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.

The analysis results to be provided with insights wherever applicable.

In [53]:
```
#Importing required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Import data into Python environment.

In [54]:
```
#Importing CCTC(Comcast Telecom Consumer Complaints) Dataset into python environment
path=r'D:\Python Simplilearn Material\Comcast Telecom Consumer Complaints'
CTCC_Data=pd.read_csv(path+r'\Comcast_telecom_complaints_data.csv')
```

In [55]:
```
#Displaying the starting 5 rows of CTCC Dataset
CTCC_Data.head()
```

Out[55]:

	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

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone
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 [56]:
```
print('Dimensions of the Dataset',CTCC_Data.shape)
print('Size of the Dataset',CTCC_Data.size)
```

```
Dimensions of the Dataset (2224, 11)
Size of the Dataset 24464
```

In [57]:
```
#Getting the information of the dataset
CTCC_Data.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
```

From the above information we can see that there are 2224 rows and 11 columns in the dataset and all the columns has non-null values

In [58]:
```
#Cross checking for number of missing values to treat them in case if present
print('Number of missing values:\n',CTCC_Data.isnull().sum())
```

```
Number of missing values:
 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
```

From the above result we can conculde that there are no missing values present in the Comcast Telecom Consumer Complaints(CTCC) Dataset,hence we can proceed to perform the analysis tasks

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

In [59]:
```
# Creating a new columns 'Date_New' (by adding Date_month_year with Time) and 'Day of the Month'
CTCC_Data['Date_New'] = CTCC_Data['Date_month_year'] + ' ' + CTCC_Data['Time']

#Converting 'Date','Date_month_year' and 'Date_New' to Datetime Format
CTCC_Data['Date_New'] = pd.to_datetime(CTCC_Data['Date_New'])
CTCC_Data['Date_month_year'] = pd.to_datetime(CTCC_Data['Date_month_year'])
CTCC_Data['Day of the Month'] = pd.to_datetime(CTCC_Data['Date'])

#Displaying the datset after modification
CTCC_Data.head()
```

Out[59]:

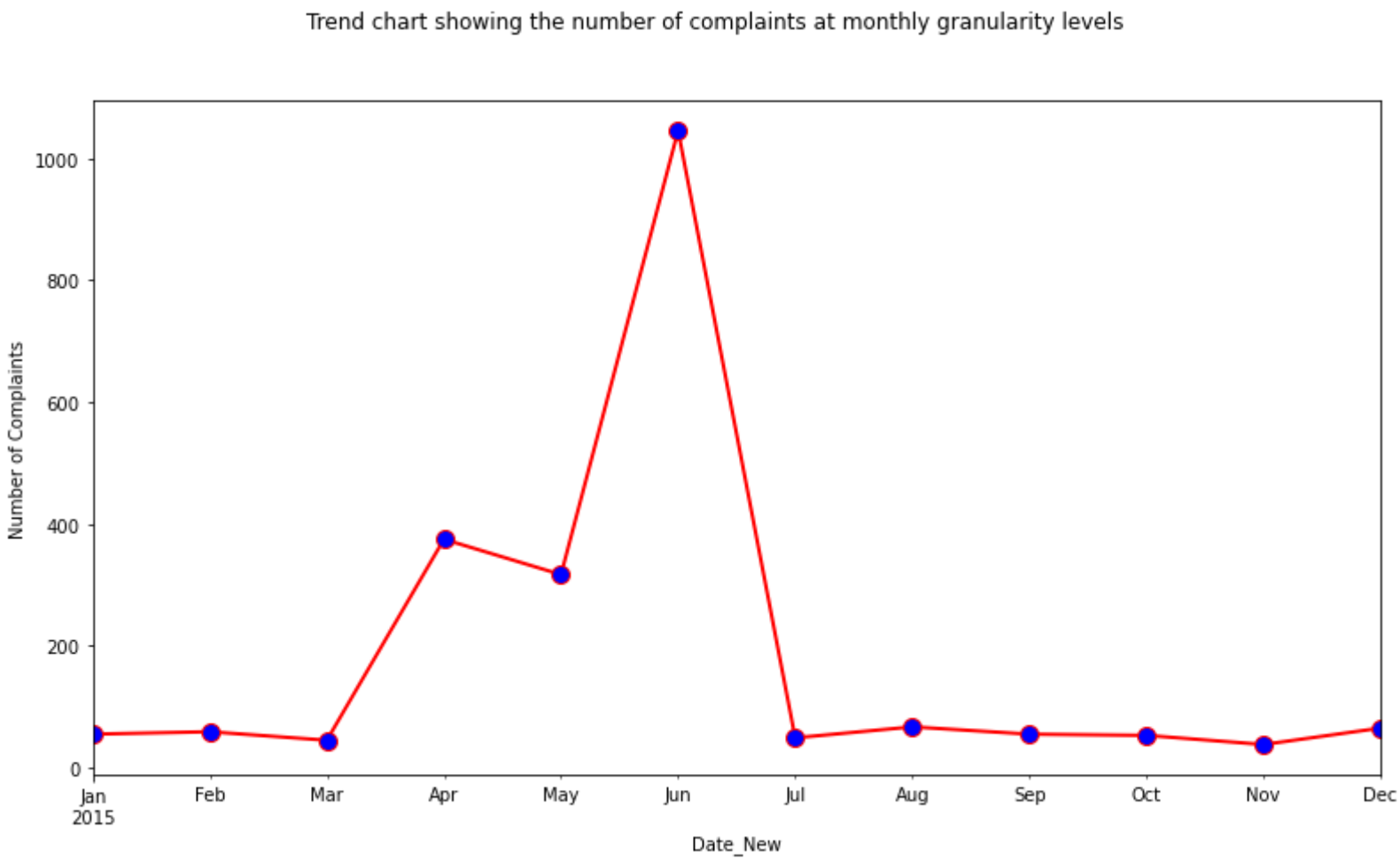
	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	Date_New	Day of the Month
0	250635	Comcast Cable Internet Speeds	22-04-15	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No	2015-04-22 15:53:50	2015-04-22
1	223441	Payment disappear - service got disconnected	04-08-15	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No	2015-08-04 10:22:56	2015-04-08
2	242732	Speed and Service	18-04-15	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes	2015-04-18 09:55:47	2015-04-18
3	277946	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	2015-07-05	11:59:35 AM	Internet	Acworth	Georgia	30101	Open	Yes	2015-07-05 11:59:35	2015-05-07

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	Date_New	Day of the Month
4	307175	Comcast not working and no service to boot	26-05-15	2015-05-26	1:25:26 PM	Internet	Acworth	Georgia	30101	Solved	No	2015-05-26 13:25:26	2015-05-26

From the above result we can see the new columns Date_New and 'Day of the Month' is added to the dataset and also the columns are modified to the required format successfully!

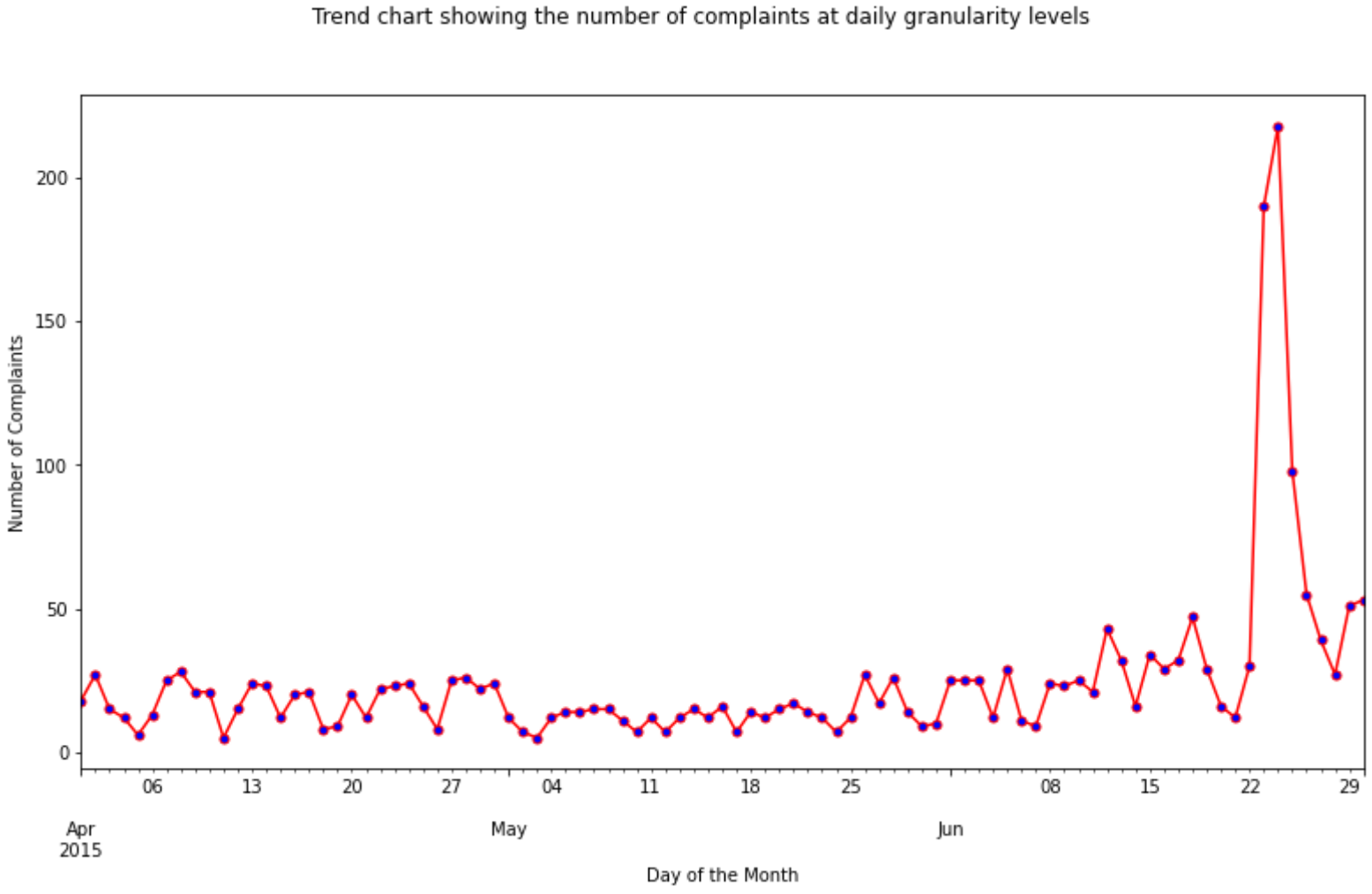
In [63]:
```
# Trend chart for the number of complaints at monthly granularity levels
CTCC_Data_Monthly = CTCC_Data.set_index(CTCC_Data['Date_New'])
plt.figure(figsize=(13,7))# to increase the plot Size
CTCC_Data_Monthly.groupby(pd.Grouper(freq='M')).size().plot(c="r",lw=2,marker='o',mfc='b',ms=10)
plt.ylabel('Number of Complaints')
plt.suptitle('Trend chart showing the number of complaints at monthly granularity levels')
```

Out[63]:
```
Text(0.5, 0.98, 'Trend chart showing the number of complaints at monthly granularity levels')
```



In [65]:
```
# Trend chart for the number of complaints at daily granularity levels.
CTCC_Data_Daily = CTCC_Data.set_index(CTCC_Data['Day of the Month'])
plt.figure(figsize=(13,7))# to increase the plot Size
CTCC_Data_Daily.groupby(pd.Grouper(freq='D')).size().plot(c="r",lw=1.5,marker='o',mfc='b',ms=5)
plt.ylabel('Number of Complaints')
plt.suptitle('Trend chart showing the number of complaints at daily granularity levels')
```

Out[65]:
```
Text(0.5, 0.98, 'Trend chart showing the number of complaints at daily granularity levels')
```



2. Provide a table with the frequency of complaint types.

In [75]:
```
# For getting frequency of complaint types first we shall see all the customer complaint types present in the dataset
# and also get the count of complaints that are repeated.
CTCC_Data_Complaint_type = CTCC_Data['Customer Complaint'].value_counts()
CTCC_Data_Complaint_type
```

Out[75]:
```
Comcast                                          83
Comcast Internet                                 18
Comcast Data Cap                                 17
comcast                                          13
Comcast Billing                                  11
                                                 ..
Improper Billing and non resolution of issues     1
Deceptive trade                                   1
intermittent internet                             1
Internet Speed on Wireless Connection             1
Comcast, Ypsilanti MI Internet Speed              1
Name: Customer Complaint, Length: 1841, dtype: int64
```

The above result might not be proper because of repeated complaint types,so to make better analysis lets convert all the customer complaints to "Lower Case"

In [77]:
```
# Converting all complaint types to lower case
CTCC_Data_Complaint_type=CTCC_Data['Customer Complaint'].str.lower().value_counts()
CTCC_Data_Complaint_type
```

Out[77]:
```
comcast                                                    102
comcast data cap                                            30
comcast internet                                            29
comcast data caps                                           21
comcast billing                                             18
                                                          ...
monthly data caps                                            1
comcast/xfinity poor service, fraudulent billing and collection     1
lost emails/billing                                          1
improper billing and non resolution of issues                1
comcast, ypsilanti mi internet speed                         1
Name: Customer Complaint, Length: 1740, dtype: int64
```

After converting to "lower case" we can see a reduction of length in the "Customer Complaint" from 1841 to 1740.This shows that all the repeated complaints are counted now!

In [79]:
```
#Displaying top 15 complaints since we cannot display the entire table as the data is huge!
print('Table with the frequency of complaint types:')
CTCC_Data_Complaint_type.head(15)
```

Out[79]:
```
Table with the frequency of complaint types:
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
Name: Customer Complaint, dtype: int64
```

Which complaint types are maximum ?

In [80]:
```
import nltk
import wordcloud
```

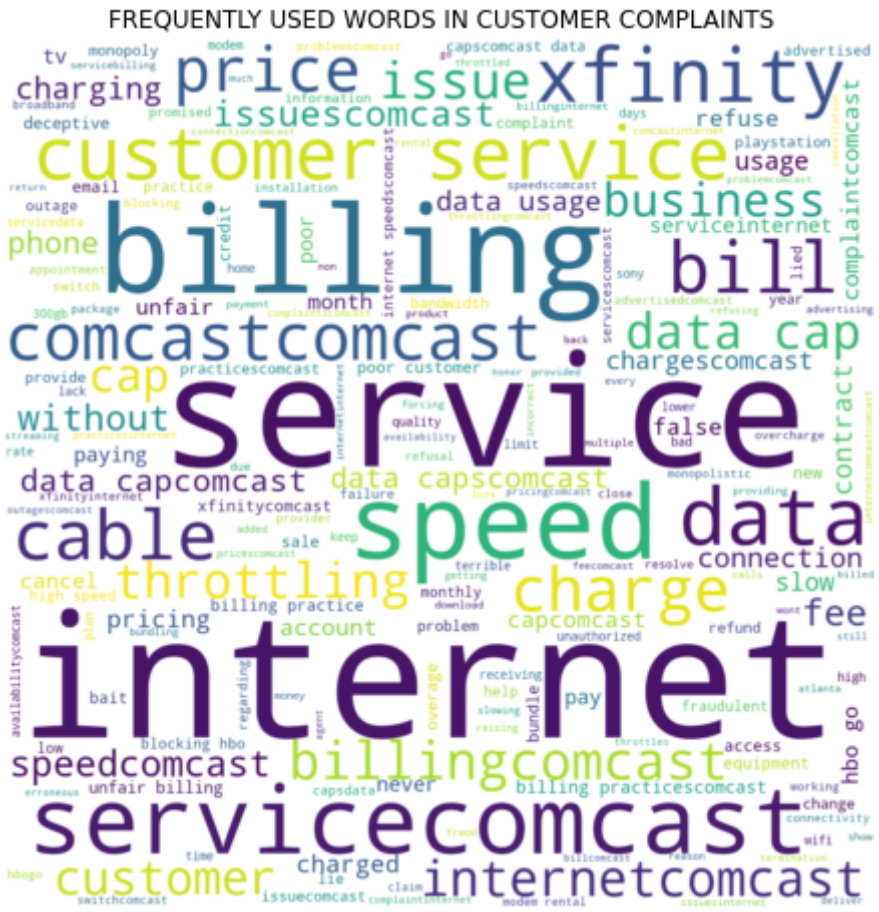
In [81]:
```
from wordcloud import WordCloud, STOPWORDS
Common_Complaint_Types = CTCC_Data['Customer Complaint'].dropna().tolist()
Common_Complaint_Types =''.join(Common_Complaint_Types).lower()

list_stops=list_stops = ('comcast','got','way','call','called','now','company','complaint'
                         'day','someone','thing','also','one','said','tell')

for word in list_stops:
    STOPWORDS.add(word)
```

In [82]:
```
wordcloud = WordCloud(stopwords=STOPWORDS,
                      background_color='white',
                      width=1000,
                      height=1000,
                      min_font_size=10).generate(Common_Complaint_Types)

plt.figure(figsize=(12,8) )
plt.imshow(wordcloud,interpolation='bilinear')
plt.title('FREQUENTLY USED WORDS IN CUSTOMER COMPLAINTS')
plt.axis('off')
plt.show()
```



From the above wordcloud we can clearly see that ***the complaints are maximum across internet,service,billing***

3. 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 [83]:
```
print("Unique values in the Status Variable:\n",CTCC_Data['Status'].unique())
```

```
Unique values in the Status Variable:
 ['Closed' 'Open' 'Solved' 'Pending']
```

There are ***4*** *unique values namely closed,open,solved and pending in the status column*

In [85]:
```
# Creating a new categorical variable 'New_Status' with value as Open and Closed.
# Open & Pending are considered as Open
# Closed & Solved are considered as Closed.
CTCC_Data['New_Status']= ["Open" if Status=="Open" or Status=="Pending" else
                          "Closed" for Status in CTCC_Data["Status"]]

#Displaying the 1st 5 rows
CTCC_Data.head()
```

Out[85]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	Date_New	Day of the Month	New_Status
0	250635	Comcast Cable Internet Speeds	22-04-15	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No	2015-04-22 15:53:50	2015-04-22	Closed
1	223441	Payment disappear - service got disconnected	04-08-15	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No	2015-08-04 10:22:56	2015-04-08	Closed
2	242732	Speed and Service	18-04-15	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes	2015-04-18 09:55:47	2015-04-18	Closed
3	277946	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	2015-07-05	11:59:35 AM	Internet	Acworth	Georgia	30101	Open	Yes	2015-07-05 11:59:35	2015-05-07	Open
4	307175	Comcast not working and no service to boot	26-05-15	2015-05-26	1:25:26 PM	Internet	Acworth	Georgia	30101	Solved	No	2015-05-26 13:25:26	2015-05-26	Closed

we can see a new categorical variable called ***"New_Status"*** *is created successfully!*

In [86]:
```
# lets check whether the modification as 'Open' and 'closed' is done properly in the variable "New_Status"
print("Unique values in the New_Status Variable:\n",CTCC_Data['New_Status'].unique())
```

```
Unique values in the New_Status Variable:
 ['Closed' 'Open']
```

Hence the task to create a new categorical variable 'New_Status'(only with 2 values Open and Closed) is accomplished.

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

In [87]:
```
#Performing cross tabulation on'State' and 'New_Status' to get state wise status of complaints
CTCC_Data_State_Wise_Status = pd.crosstab(CTCC_Data['State'],CTCC_Data['New_Status'])
CTCC_Data_State_Wise_Status
```

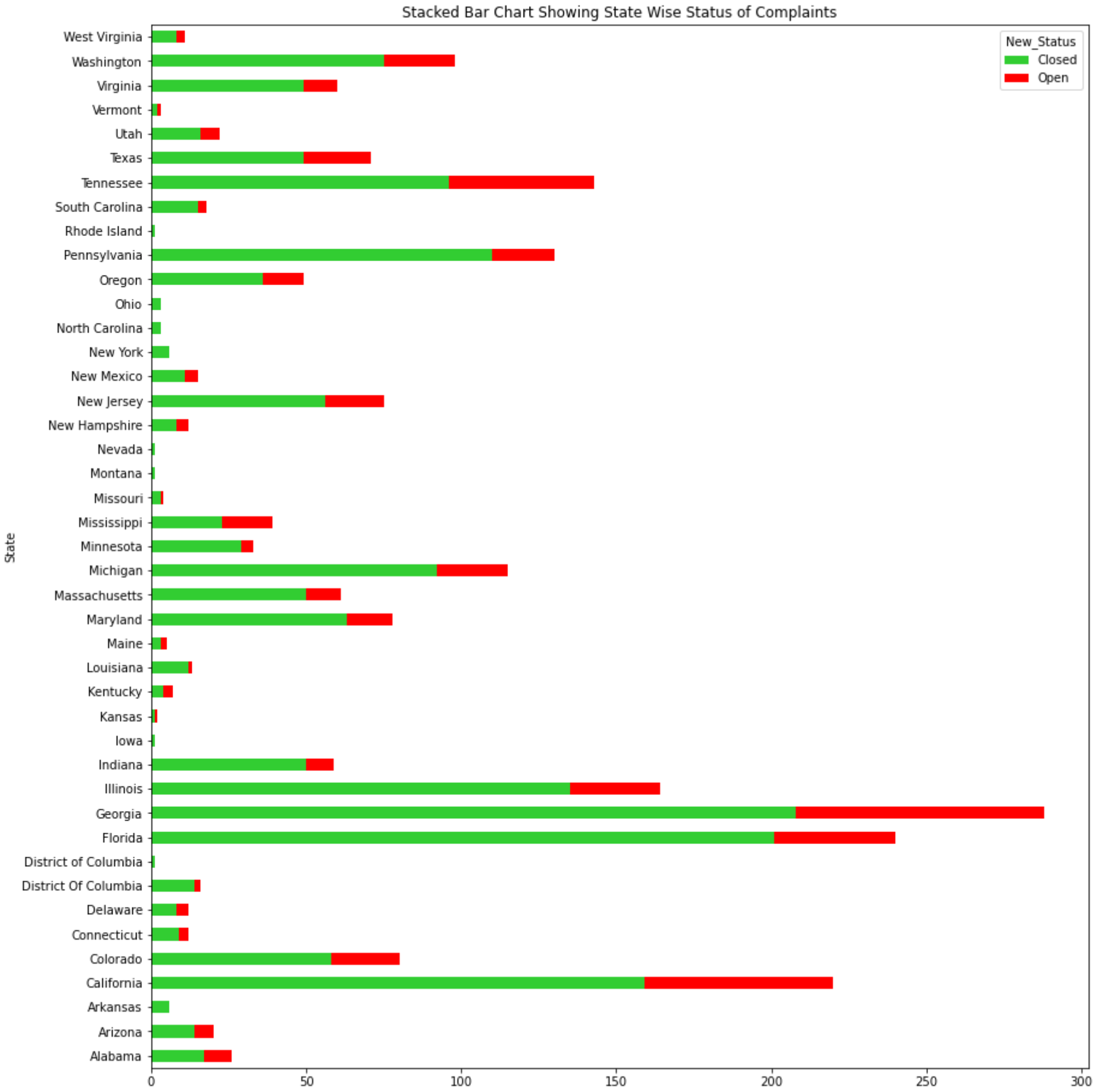
Out[87]:

New_Status	Closed	Open
State		
Alabama	17	9
Arizona	14	6
Arkansas	6	0
California	159	61
Colorado	58	22
Connecticut	9	3
Delaware	8	4
District Of Columbia	14	2
District of Columbia	1	0
Florida	201	39
Georgia	208	80
Illinois	135	29
Indiana	50	9
Iowa	1	0
Kansas	1	1
Kentucky	4	3
Louisiana	12	1
Maine	3	2
Maryland	63	15
Massachusetts	50	11
Michigan	92	23
Minnesota	29	4
Mississippi	23	16
Missouri	3	1
Montana	1	0
Nevada	1	0

New_Status State	Closed	Open
New Hampshire	8	4
New Jersey	56	19
New Mexico	11	4
New York	6	0
North Carolina	3	0
Ohio	3	0
Oregon	36	13
Pennsylvania	110	20
Rhode Island	1	0
South Carolina	15	3
Tennessee	96	47
Texas	49	22
Utah	16	6
Vermont	2	1
Virginia	49	11
Washington	75	23
West Virginia	8	3

In [88]:
```
#plotting a state wise status of complaints in a stacked bar chart
colors={'Open':'red','Closed':'limegreen'}
CTCC_Data_State_Wise_Status.plot(kind='barh',figsize=(14,16),stacked=True,color=colors,
                                 title='Stacked Bar Chart Showing State Wise Status of Complaints')
```

Out[88]:
```
<AxesSubplot:title={'center':'Stacked Bar Chart Showing State Wise Status of Complaints'}, ylabel='State'>
```



Which state has the maximum complaints?

In [89]:
```
#Sorting the data in decreasing order
CTCC_Data.groupby(["State"]).size().sort_values(ascending=False).to_frame().reset_index().rename({0: "Count"}, axis=1)[:5]
```

Out[89]:

	State	Count
0	Georgia	288
1	Florida	240
2	California	220
3	Illinois	164
4	Tennessee	143

*Seeing the stacked bar chart and the above result we can say that maximum complaints are from the State called **Georgia** (It has 288 number of complaints)*

Which state has the highest percentage of unresolved complaints?

In [90]:
```
#Calculating the percenting of unresolved complaints state wise
CTCC_Data_State_Wise_Status['Percentage of Unresolved Complaints']=(CTCC_Data_State_Wise_Status['Open']/
                                                                    CTCC_Data_State_Wise_Status['Open'].sum())*100
CTCC_Data_State_Wise_Status.head()
```

Out[90]:

New_Status	Closed	Open	Percentage of Unresolved Complaints
State			
Alabama	17	9	1.740812

New_Status	Closed	Open	Percentage of Unresolved Complaints
State			
Arizona	14	6	1.160542
Arkansas	6	0	0.000000
California	159	61	11.798839
Colorado	58	22	4.255319

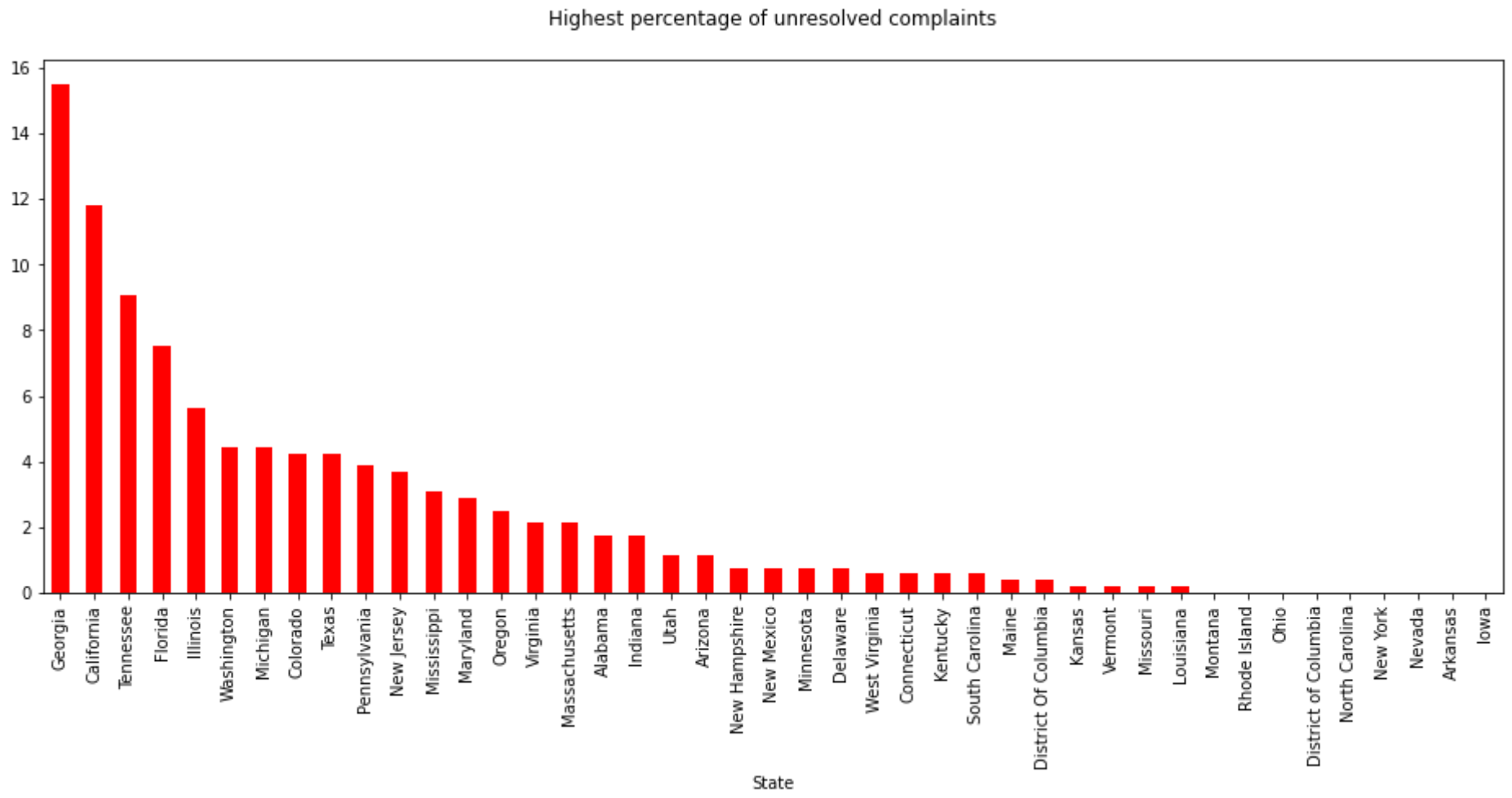
In [91]:
```
#Sorting in descending order to get the highest percentage of unresolved complaints state wise
CTCC_Data_State_Wise_Status['Percentage of Unresolved Complaints'].sort_values(ascending=False).head()
```

Out[91]:
```
State
Georgia       15.473888
California    11.798839
Tennessee      9.090909
Florida        7.543520
Illinois       5.609284
Name: Percentage of Unresolved Complaints, dtype: float64
```

*From the above result we find that **Georgia** has the highest percentage of unresolved complaints compare to other states.Lets Show this by plotting a **bar chart***

In [92]:
```
CTCC_Data_State_Wise_Status['Percentage of Unresolved Complaints'].sort_values(ascending=False).plot(kind='bar',
                                                                                                     figsize=(16,6),
                                                                                                     color="r")
plt.title('Highest percentage of unresolved complaints\n')
```

Out[92]:
```
Text(0.5, 1.0, 'Highest percentage of unresolved complaints\n')
```



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

In [23]:
```
# Checking for unique values in Received Via Column of CTCC_Data
print("The Unique Values in the Received Via Variable \n",CTCC_Data['Received Via'].unique())
```

```
The Unique Values in the Received Via Variable
 ['Customer Care Call' 'Internet']
```

*There are **2** Unique values 'Customer Care Call' and 'Internet' in the Received via variable*

In [93]:
```
#count of complaints resolved and unresolved till date
CTCC_Data.New_Status.value_counts()
```

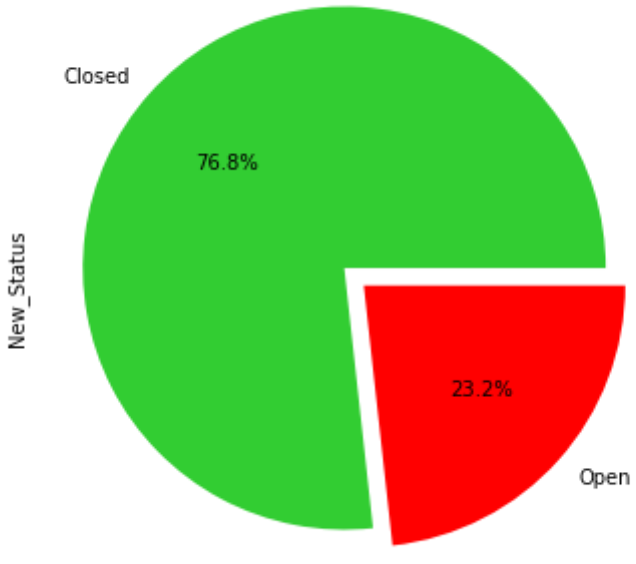
Out[93]:
```
Closed    1707
Open       517
Name: New_Status, dtype: int64
```

This shows 1707 complaints are resolved or closed

In [96]:
```
# Let show this in percentage under the pie chart using argument autopct='%1.1f%%'
mycolor=['limegreen','red']
myexplode = [0.1, 0]
```

```
plt.title('Percentage of Complaints Resolved(CLOSED) and Unresolved(OPEN) till date \n')
CTCC_Data.New_Status.value_counts().plot(kind='pie',explode = myexplode,colors=mycolor,autopct='%1.1f%%',figsize = (14,6))
```

Out[96]:
```
<AxesSubplot:title={'center':'Percentage of Complaints Resolved(CLOSED) and Unresolved(OPEN) till date \n'}, ylabel='New_Status'>
```



*From the plot we can conclude that **76.8%** of percentage of complaints resolved till date, which were received through the Internet and customer care call*

*Now out of 76.8% percentage of resolved complaints lets get the **individual percentage of complaints resolved which were received through the mode of internet and customer care call***

In [97]:
```
#Percentage of complaints resolved till date, which were received through the Internet and customer care calls
CTCC_Data_Resolved = CTCC_Data.groupby(['Received Via','New_Status']).size().unstack()
CTCC_Data_Resolved['Resolved Complaints'] = (CTCC_Data_Resolved['Closed']/CTCC_Data_Resolved['Closed'].sum())*100
CTCC_Data_Resolved['Resolved Complaints'].round(2)
```

Out[97]:
```
Received Via
Customer Care Call    50.62
Internet              49.38
Name: Resolved Complaints, dtype: float64
```

*From the above result we can see that **50.62%** of Complaints received through Customer Care call and **49.38%** of Complaints received through Internet are resolved*

In [ ]:
```

```

In [ ]:
```

```