Comcast-telecom_Project-Python

August 30, 2023

0.1 Tiisetso Khasebe

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

Data Exploratory

[2]:
```
data=pd.read_csv("Comcast_telecom_complaints_data.csv")
```

[3]:
```
data.head()
```

[3]:
```
  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
```

[4]:
```
#the null values
data.isnull().sum()
```

[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
```

[5]:
```
#data description
data.describe(include='all')
```

[5]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time
count	2224	2224	2224	2224	2224
unique	2224	1841	91	91	2190
top	250635	Comcast	24-06-15	24-Jun-15	12:41:14 PM
freq	1	83	218	218	2
mean	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN

	Received Via	City	State	Zip code	Status
count	2224	2224	2224	2224.000000	2224
unique	2	928	43	NaN	4
top	Customer Care Call	Atlanta	Georgia	NaN	Solved
freq	1119	63	288	NaN	973
mean	NaN	NaN	NaN	47994.393435	NaN
std	NaN	NaN	NaN	28885.279427	NaN
min	NaN	NaN	NaN	1075.000000	NaN
25%	NaN	NaN	NaN	30056.500000	NaN
50%	NaN	NaN	NaN	37211.000000	NaN
75%	NaN	NaN	NaN	77058.750000	NaN
max	NaN	NaN	NaN	99223.000000	NaN

	Filing on Behalf of Someone
count	2224
unique	2
top	No
freq	2021

```
mean                          NaN
std                           NaN
min                           NaN
25%                           NaN
50%                           NaN
75%                           NaN
max                           NaN
```

[6]:
```
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
```

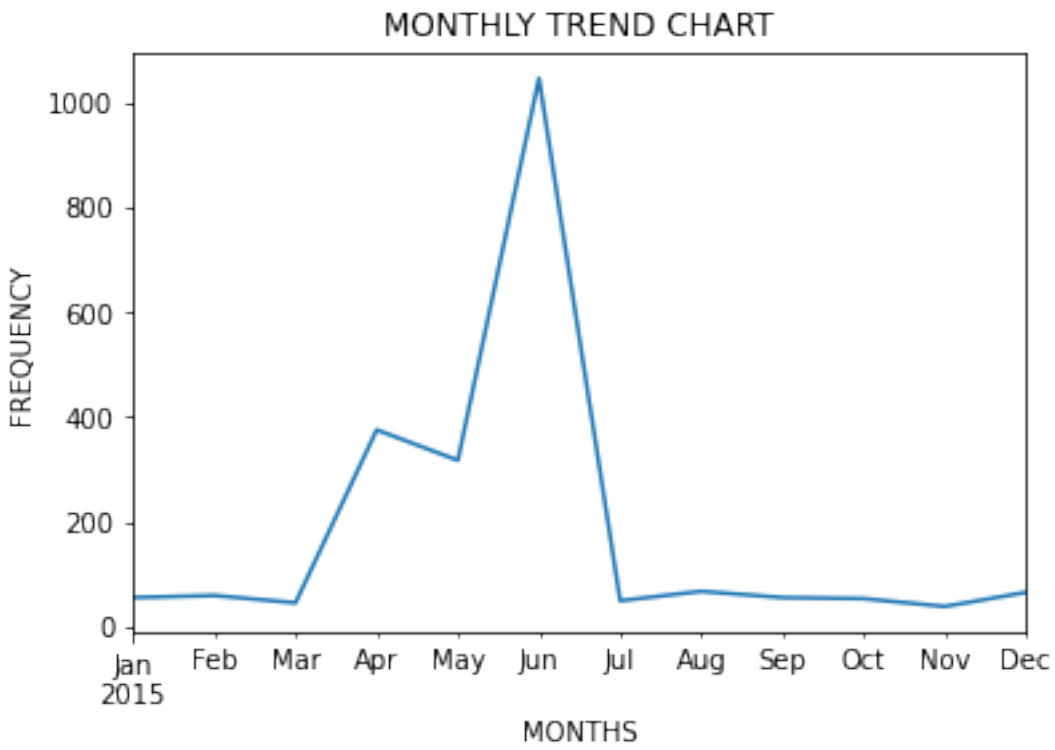
[7]:
```
#shape of the dataset
data.shape
```

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

The trend chart for the number of complaints

[8]:
```
data['Date_month_year']=data['Date_month_year'].apply(pd.to_datetime)
data=data.set_index('Date_month_year')
months=data.groupby(pd.Grouper(freq='M')).size().plot()
plt.xlabel("MONTHS")
plt.ylabel("FREQUENCY")
plt.title("MONTHLY TREND CHART")
```

[8]:
```
Text(0.5, 1.0, 'MONTHLY TREND CHART')
```



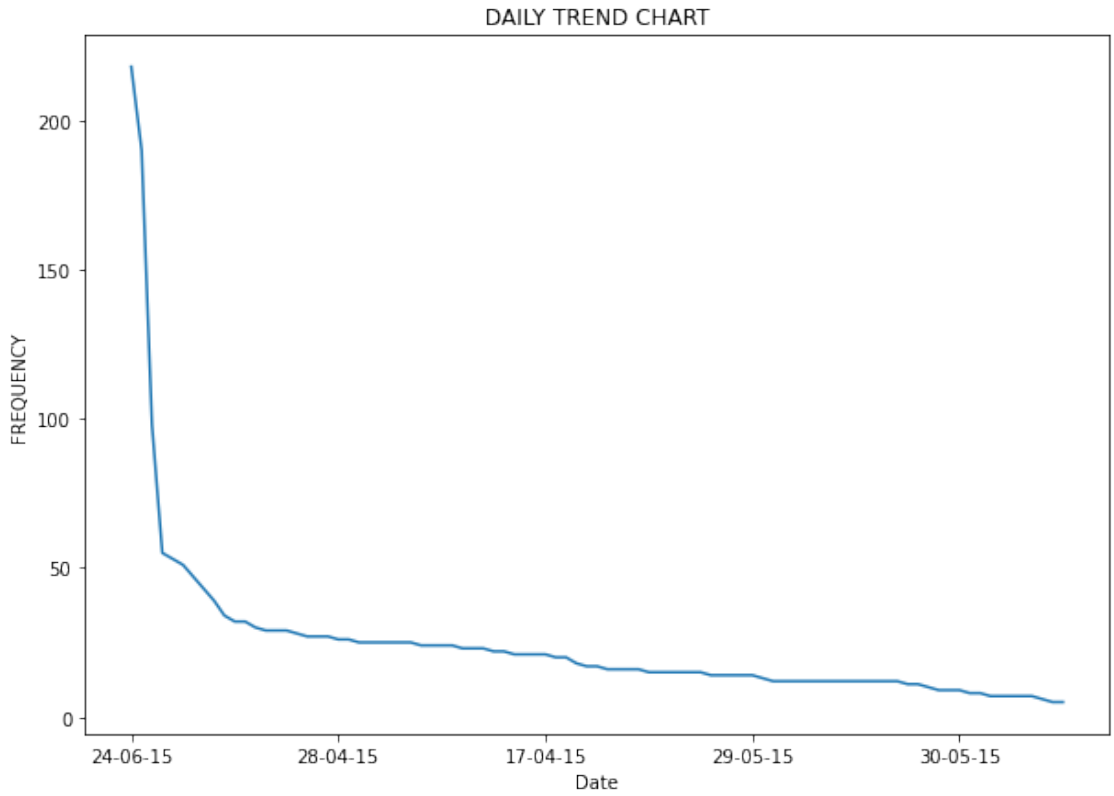
[9]:
```
data['Date'].value_counts()[:10]
```

[9]:
```
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
```

Plotting the daily chart

[10]:
```
data=data.sort_values(by='Date')
plt.figure(figsize=(10,7))
data['Date'].value_counts().plot()
plt.xlabel("Date")
plt.ylabel("FREQUENCY")
plt.title("DAILY TREND CHART")
```

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



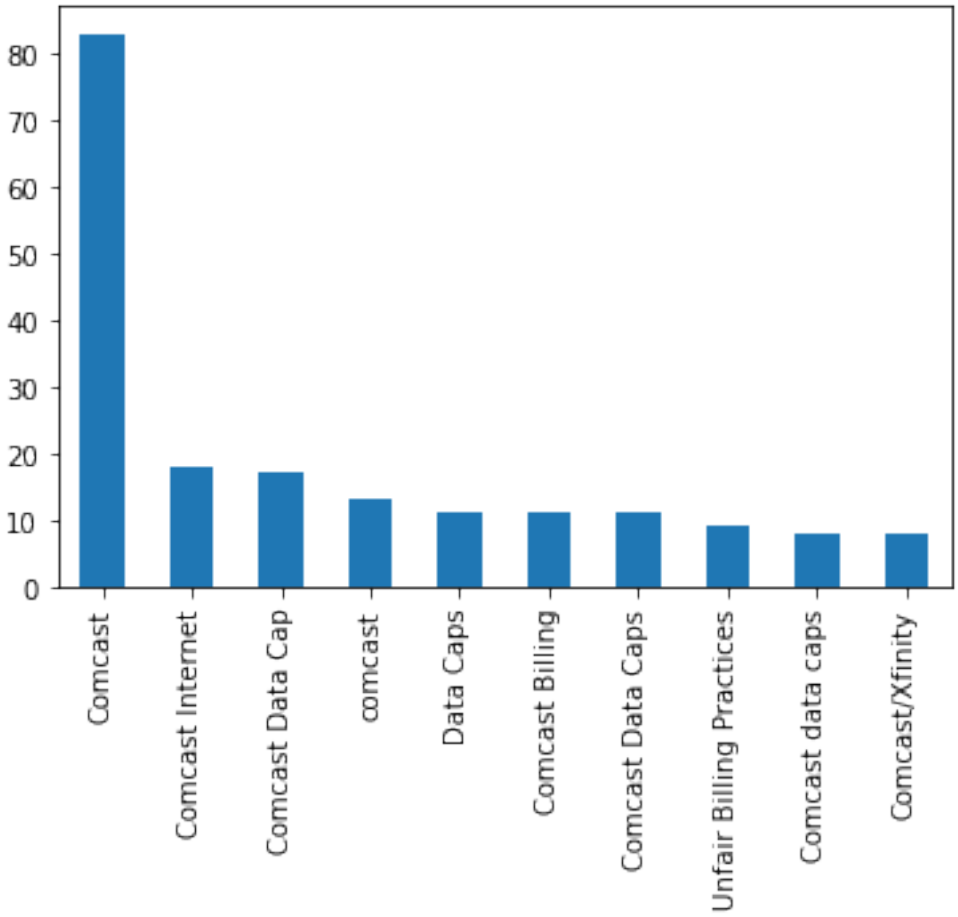
A Table with frequency of complaint types

[11]:
```
data['Customer Complaint'].value_counts()[:10]
```

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

[12]:
```
data['Customer Complaint'].value_counts()[:10].plot.bar()
```

[12]:
```
<AxesSubplot: >
```



Which complaint types are maximum

[13]:
```
internet_issues1=data[data['Customer Complaint'].str.contains('network')].
  ↪count()
internet_issues2=data[data['Customer Complaint'].str.contains('speed')].count()
internet_issues3=data[data['Customer Complaint'].str.contains('data')].count()
internet_issues4=data[data['Customer Complaint'].str.contains('internet')].
  ↪count()
billing_issues1=data[data['Customer Complaint'].str.contains('billing')].count()
billing_issues2=data[data['Customer Complaint'].str.contains('charges')].count()
service_issues1=data[data['Customer Complaint'].str.contains('service')].count()
service_issues2=data[data['Customer Complaint'].str.contains('customer')].
  ↪count()
total_internet_issues=internet_issues1+internet_issues2+internet_issues3+internet_issues4
total_billing_issues=billing_issues1+billing_issues2
total_service_issues=service_issues1+service_issues2
other_issues=2224-(total_billing_issues+total_internet_issues+total_service_issues)
other_issues
```

[13]:
```
Ticket #                      1321
Customer Complaint            1321
Date                          1321
Time                          1321
Received Via                  1321
City                          1321
State                         1321
Zip code                      1321
Status                        1321
Filing on Behalf of Someone   1321
dtype: int64
```

[14]:
```
total_billing_issues
```

[14]:
```
Ticket #                      169
Customer Complaint            169
Date                          169
Time                          169
Received Via                  169
City                          169
State                         169
Zip code                      169
Status                        169
Filing on Behalf of Someone   169
dtype: int64
```

[15]:
```
total_internet_issues
```

[15]:
```
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
```

[16]:
```
total_service_issues
```

[16]:
```
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
```

Create a new categorical variable with value as Open and Closed

[17]:
```
data['newstatus']=['Open' if status=="Open" or status=='Pending' else 'Closed' ␣
 ↪for status in data['Status']]
data.sample(10)
```

[17]:

Date_month_year	Ticket #	Customer Complaint	Date	Time	Received Via	City	State	Zip code	Status
2015-04-27	258748	Comcast	27-04-15	12:28:34 PM	Customer Care Call	Placida	Florida	33946	Closed
2015-12-06	337555	Comcast Internet Complaint	06-12-15	4:13:39 PM	Customer Care Call	New Bedford	Massachusetts	2741	Open
2015-06-26	367310	Suspected Throttling	26-06-15	1:00:33 AM	Internet	Bloomington	Illinois	61704	Open
2015-06-23	360356	Comcast Data Caps	23-06-15	10:27:40 PM	Customer Care Call	Covington	Georgia	30016	Pending
2015-06-29	372741	Non working service	29-06-15	3:40:47 PM	Customer Care Call	Rahway	New Jersey	7065	Closed
2015-06-17	347688	Comcast Data usage caps	17-06-15	11:40:30 PM	Internet	Powell	Tennessee	37849	Open
2015-05-06	324795	Rates	06-05-15	9:59:39 AM	Internet				
2015-04-23	251802	Erroneous charges on Comcast bill	23-04-15	10:12:28 AM	Customer Care Call				
2015-06-13	338856	Comcast Monopoly and Abuse	13-06-15	11:06:05 AM	Internet				
2015-06-24	362687	Poor customer service, terrible Internet speed…	24-06-15	4:27:43 PM	Customer Care Call				

```
2015-05-06              Bala Cynwyd   Pennsylvania     19004   Closed
2015-04-23       Palm Beach Gardens        Florida     33418   Closed
2015-06-13              Somerville  Massachusetts      2143   Solved
2015-06-24              Hudsonville       Michigan     49426     Open
```

Date_month_year	Filing on Behalf of Someone newstatus	
2015-04-27	No	Closed
2015-12-06	Yes	Open
2015-06-26	Yes	Open
2015-06-23	No	Open
2015-06-29	No	Closed
2015-06-17	Yes	Open
2015-05-06	No	Closed
2015-04-23	No	Closed
2015-06-13	No	Closed
2015-06-24	No	Open

[18]:
```
#Provide state wise status of complaints
state_complain=data.groupby(['State','newstatus']).size().unstack()
state_complain
```

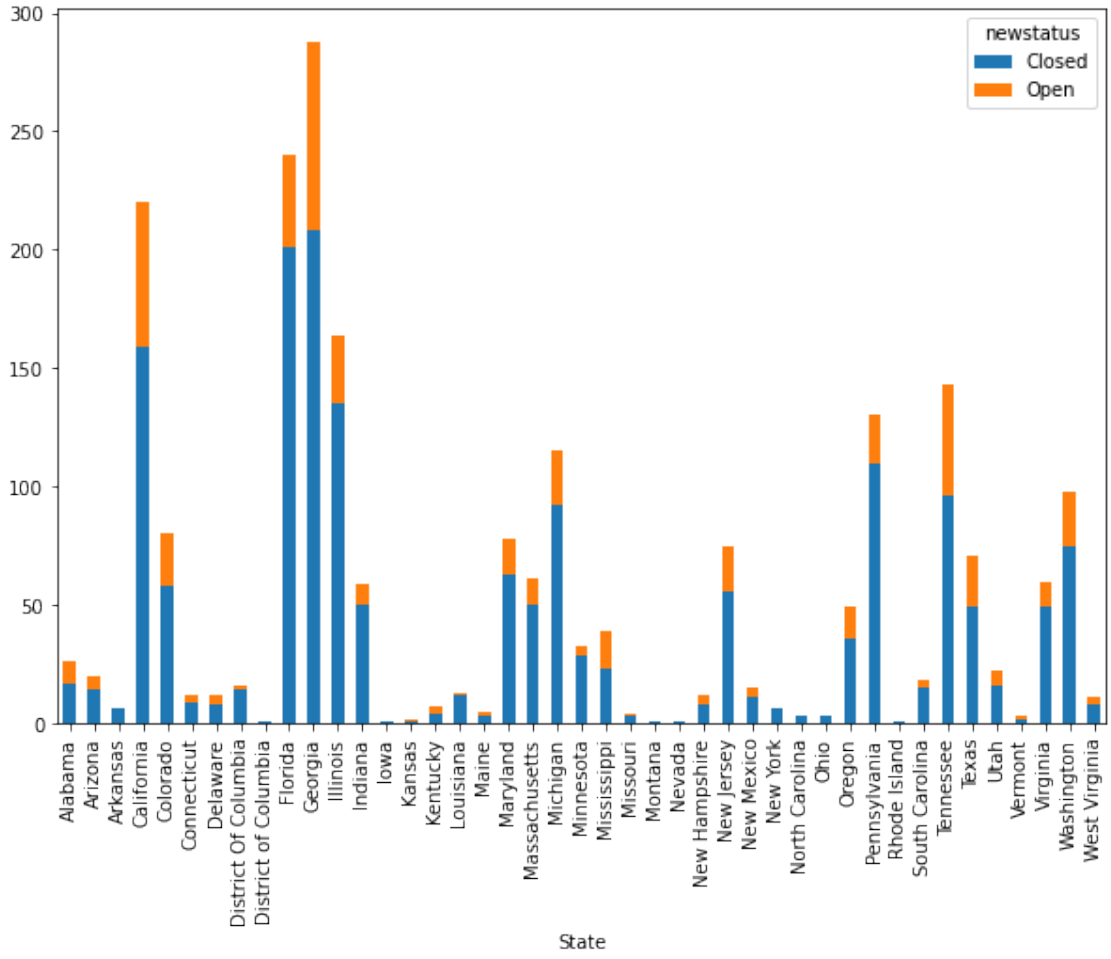
[18]:

State	Closed	Open
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
```

[19]:
```
state_complain.plot.bar(figsize=(10,7),stacked=True)
```

[19]:
```
<AxesSubplot: xlabel='State'>
```



Which state has the maximum complaints

[20]:
```
data.groupby(['State']).size().sort_values(ascending=False)[:1]
```

[20]:
```
State
Georgia    288
dtype: int64
```

Which state has the highest percentage of unresolved complaints

[21]:
```
data.newstatus.value_counts()
```

[21]:
```
Closed    1707
Open       517
Name: newstatus, dtype: int64
```

[22]:
```
unresolved_data=data.groupby(['State','newstatus']).size().unstack().fillna(0).
 ↪sort_values(by="Open",ascending=False)
unresolved_data['unresolved_cmp_prct']=unresolved_data['Open']/
 ↪unresolved_data['Open'].sum()*100
unresolved_data
```

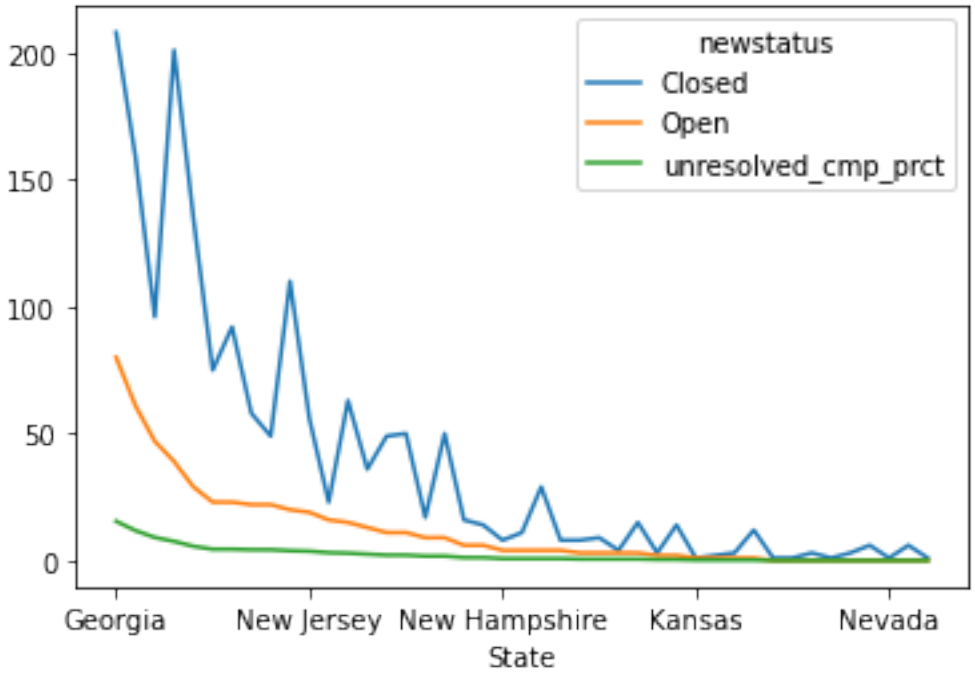
[22]:

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
```

[23]:
```
unresolved_data.plot()
```

[23]:
```
<AxesSubplot: xlabel='State'>
```



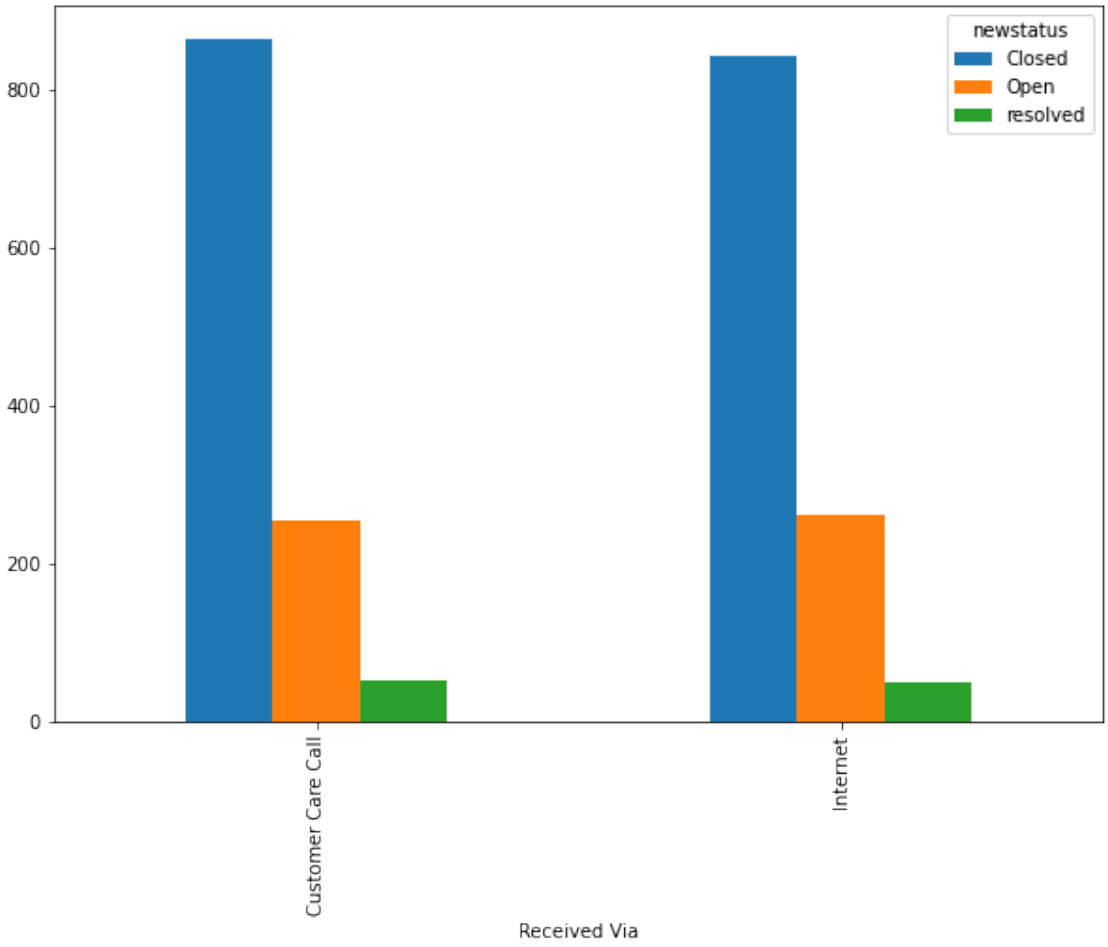
Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls

[24]:
```
resolved_data=data.groupby(['Received Via','newstatus']).size().unstack().
  ↪fillna(0)
resolved_data['resolved']=resolved_data['Closed']/resolved_data['Closed'].
  ↪sum()*100
resolved_data['resolved']
```

[24]:
```
Received Via
Customer Care Call    50.615114
Internet              49.384886
Name: resolved, dtype: float64
```

[25]:
```
resolved_data.plot(kind='bar',figsize=(10,7))
```

[25]: <AxesSubplot: xlabel='Received Via'>



[ ]: