Comcast Project

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.

1. Import data into Python environment.
2. Provide the trend chart for the number of complaints at monthly and daily granularity levels.
3. Provide a table with the frequency of complaint types.
4. Which complaint types are maximum i.e., around internet, network issues, or across any other domains.
5. 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.
6. Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:
7. Which state has the maximum complaints
8. Which state has the highest percentage of unresolved complaints
9. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.
10. The analysis results to be provided with insights wherever applicable.

Solution :

1. Import data into Python environment

```
# Import Libraries
import pandas as pd
import datetime as dt
```

```
# Read Data from CSV and verify
df = pd.read_csv('Comcast_telecom_complaints_data.csv')
df.head()
df.shape
df.info
df.columns
```

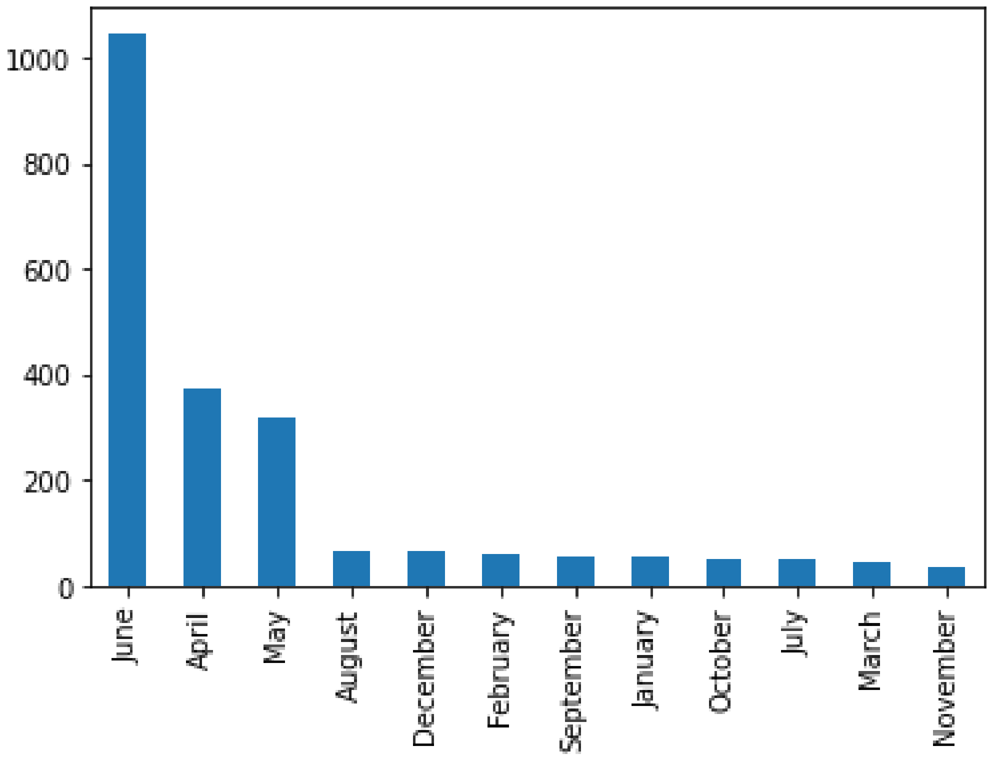
2. Provide the trend chart for the number of complaints at monthly and daily granularity levels
3. Provide a table with the frequency of complaint types.

Convert Date field to DataTime format and split the date

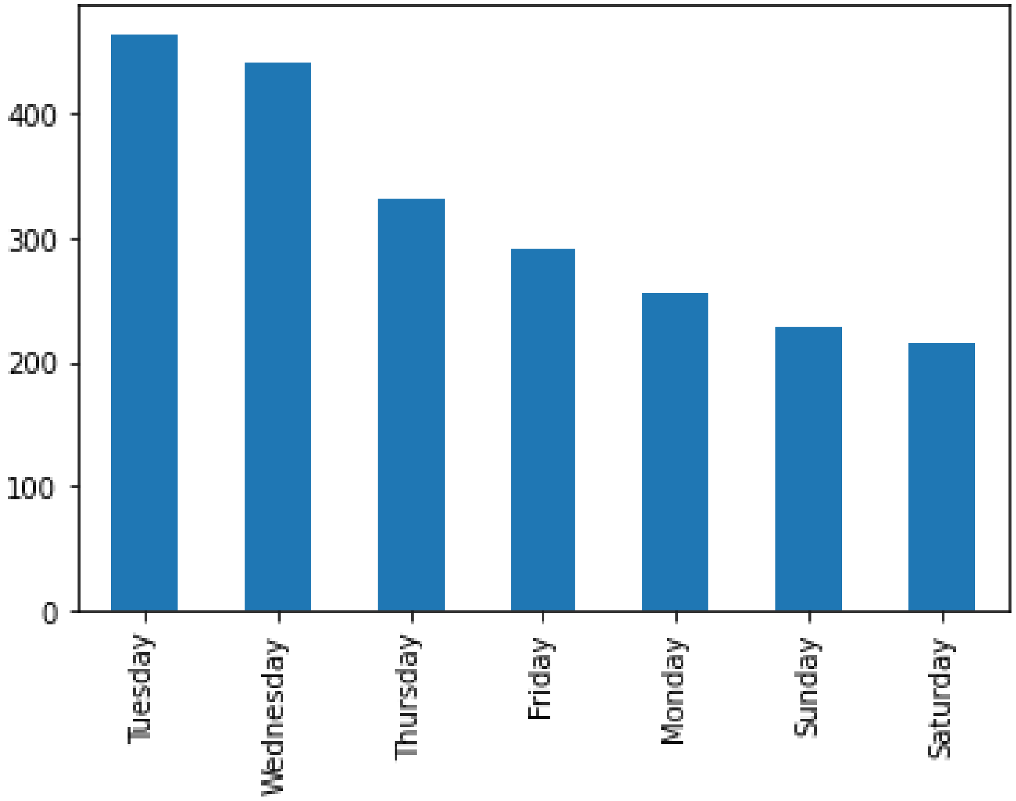
```
Convert Date field

        df['New_Date_with_time']=df['Date']+' '+df['Time']
        df['New_Date_with_time']= pd.to_datetime(df['New_Date_with_time'],format='%d-
        %m-%y %I:%M:%S %p')
df.head()
```

```
# Split the date
        df['New_Month']=df['New_Date_with_time'].dt.strftime('%B') #Extract Month from
        the New Date with time column
        df['New_Date']=df['New_Date_with_time'].dt.strftime('%d') # Extract Date from the
        New Date column
        df['New_Year']=df['New_Date_with_time'].dt.strftime('%Y') # Extract Year
        df['Day']=df['New_Date_with_time'].dt.strftime('%A')   # Extract Day
        df['Hour']=df['New_Date_with_time'].dt.strftime('%H')  # Extract Hour
        df.info()
        df.head()
```

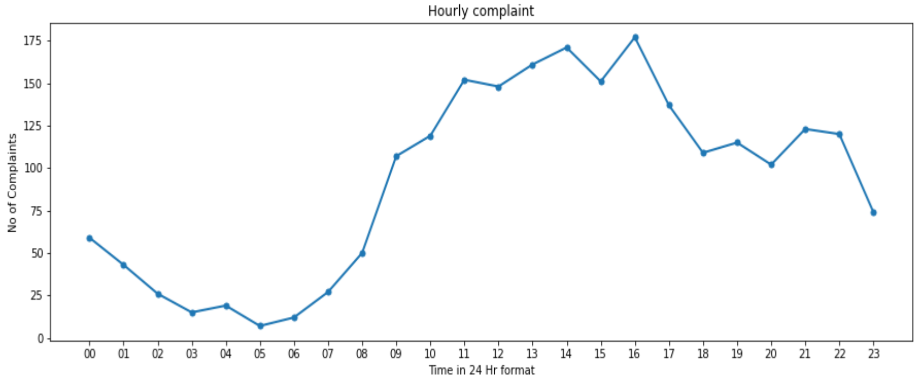
```
# Monthly Count
   df['New_Month'].value_counts()
  df['New_Month'].value_counts().plot.bar()
```



```
# Day count
    df['Day'].value_counts()
    df['Day'].value_counts().plot.bar()
```

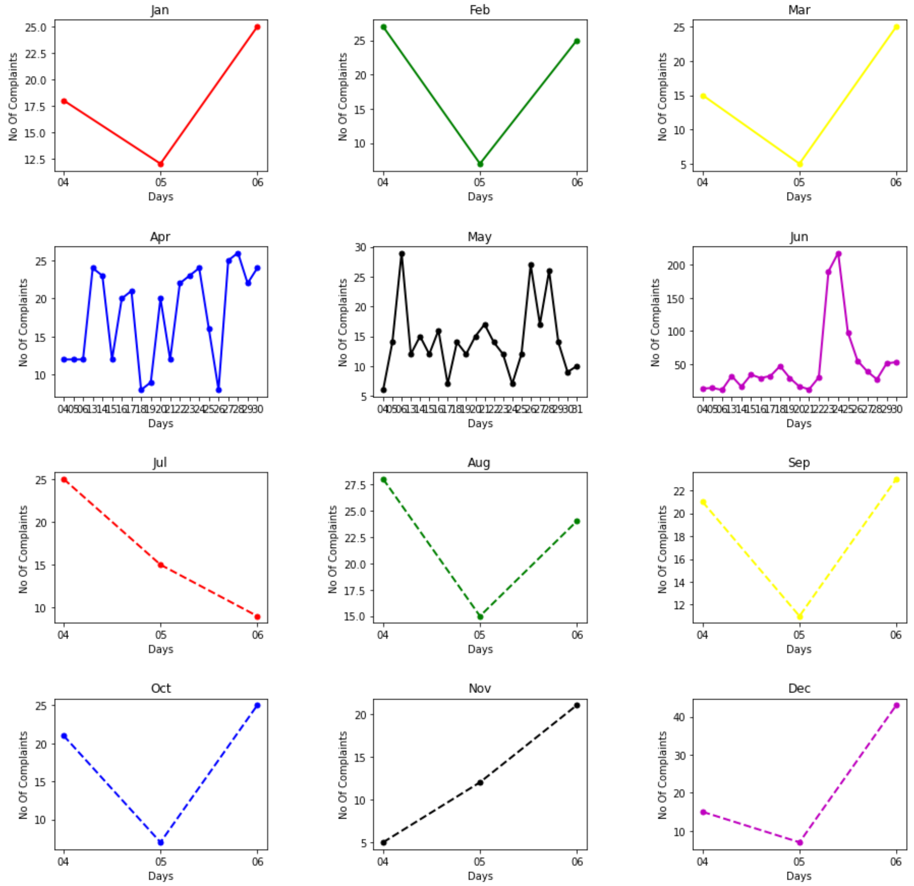


```
# hourly count
   df['Hour'].value_counts()
              import matplotlib.pyplot as plt
          from matplotlib import style

          s= df.groupby('Hour').size()
          f = plt.figure()
          f.set_figwidth(15)
          f.set_figheight(5)
          plt.plot(s , marker='o', linestyle='-',linewidth=2, markersize=5 )
          plt.ylabel('No of Complaints')
          #plt.annotate('Max',ha='center' , xy = (16,150) ,va='bottom',arrowprops={'facecolor':
          'blue'})
          plt.xlabel('Time in 24 Hr format')
          plt.title('Hourly complaint')
plt.show()
```



```
# Monthly trend on Complaints Received

          m_df=df.groupby(['New_Month','New_Date']).size()
          df_jan=df.groupby('New_Month').get_group('January')
          jan_plot = df_jan.groupby('New_Date').size()

          df_feb=df.groupby('New_Month').get_group('February')
          feb_plot = df_feb.groupby('New_Date').size()

          df_mar=df.groupby('New_Month').get_group('March')
          mar_plot = df_mar.groupby('New_Date').size()

          df_apr=df.groupby('New_Month').get_group('April')
          apr_plot = df_apr.groupby('New_Date').size()

          df_may=df.groupby('New_Month').get_group('May')
          may_plot = df_may.groupby('New_Date').size()

          df_jun=df.groupby('New_Month').get_group('June')
          jun_plot = df_jun.groupby('New_Date').size()
```

```
df_jul=df.groupby('New_Month').get_group('July')
jul_plot = df_jul.groupby('New_Date').size()

df_aug=df.groupby('New_Month').get_group('August')
aug_plot = df_aug.groupby('New_Date').size()

df_sep=df.groupby('New_Month').get_group('September')
sep_plot = df_sep.groupby('New_Date').size()

df_oct=df.groupby('New_Month').get_group('October')
oct_plot = df_oct.groupby('New_Date').size()

df_nov=df.groupby('New_Month').get_group('November')
nov_plot = df_nov.groupby('New_Date').size()

df_dec=df.groupby('New_Month').get_group('December')
dec_plot = df_dec.groupby('New_Date').size()

f = plt.figure()
f.set_figwidth(15)
f.set_figheight(15)
plt.subplots_adjust(hspace=.5 , wspace=.5)
plt.subplot(4,3,1)
plt.title('Jan')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(jan_plot , marker='o', linestyle='-',linewidth=2, markersize=5 , c='red' ,
label='Jan')

plt.subplot(4,3,2)
plt.title('Feb')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(feb_plot , marker='o', linestyle='-',linewidth=2, markersize=5, c='green' ,
label='Feb')

plt.subplot(4,3,3)
plt.title('Mar')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(mar_plot , marker='o', linestyle='-',linewidth=2, markersize=5, c='yellow' ,
label='Mar')

plt.subplot(4,3,4)
plt.title('Apr')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(apr_plot , marker='o', linestyle='-',linewidth=2, markersize=5, c='blue',
label='Apr')

plt.subplot(4,3,5)
plt.title('May')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(may_plot , marker='o', linestyle='-',linewidth=2, markersize=5, c='black',
label='May' )

plt.subplot(4,3,6)
plt.title('Jun')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(jun_plot , marker='o', linestyle='-',linewidth=2, markersize=5, c='m',
label='Jun')

plt.subplot(4,3,7)
plt.title('Jul')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(jul_plot , marker='o', linestyle='--',linewidth=2, markersize=5, c= 'red' ,
label='Jul')

plt.subplot(4,3,8)
plt.title('Aug')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(aug_plot , marker='o', linestyle='--',linewidth=2, markersize=5, c='green',
label='Aug')

plt.subplot(4,3,9)
plt.title('Sep')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(sep_plot , marker='o', linestyle='--',linewidth=2, markersize=5 ,c='yellow',
label='Sep')

plt.subplot(4,3,10)
plt.title('Oct')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(oct_plot , marker='o', linestyle='--',linewidth=2, markersize=5, c='blue',
label='Oct')

plt.subplot(4,3,11)
plt.title('Nov')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(nov_plot , marker='o', linestyle='--',linewidth=2, markersize=5 , c='black',
label='Nov')

plt.subplot(4,3,12)
plt.title('Dec')
plt.xlabel('Days')
plt.ylabel('No Of Complaints')
plt.plot(dec_plot , marker='o', linestyle='--',linewidth=2, markersize=5, c='m',
label='Dec')
```



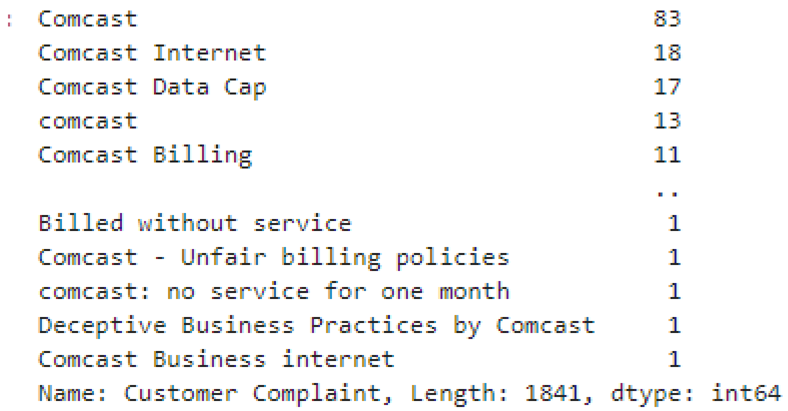
Observations:

----

a) Jun , april and may has more no of complaints , highest is June.
b) When observed on Weekday basis , Tuesday Wednesday and Thrusday has more complaints then other days. Weekend are comparatively less than weekdays
c) Complaints are at peek on mid day i.e b/w 12:PM to 4:PM . Peak at 4:Pm
d) We observe a 'V ' trend on Montly complaint basis for most of the months except June, april, and May

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

```
    df.head()
    df_com = df['Customer Complaint'].reset_index()
    df_com = df_com[['Customer Complaint']]
    df_com['Customer Complaint'].value_counts()
```



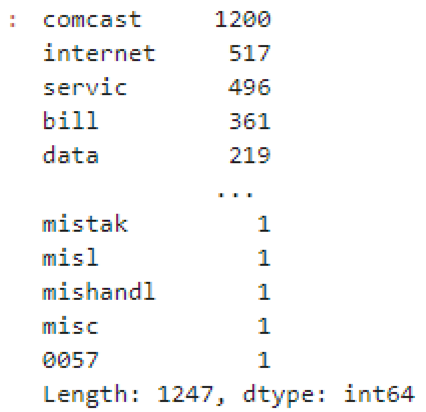
```
: Comcast                                     83
  Comcast Internet                            18
  Comcast Data Cap                            17
  comcast                                     13
  Comcast Billing                             11
                                              ..
  Billed without service                       1
  Comcast - Unfair billing policies            1
  comcast: no service for one month            1
  Deceptive Business Practices by Comcast      1
  Comcast Business internet                    1
  Name: Customer Complaint, Length: 1841, dtype: int64
```

```
# remove stop words and punctuations

            from nltk.corpus import stopwords
            from nltk.stem import PorterStemmer
import string
```

```
            def rmsw(msg):
               porter = PorterStemmer()
               non_p = [char for char in msg if char not in string.punctuation]
               non_p= ''.join(non_p)
   return[porter.stem(m.lower()) for m in non_p.split() if m not in stopwords.words('english')]
```

```
df_com['Customer Complaint'][0:5].apply(rmsw)
```

```
# Bag of Words
            from sklearn.feature_extraction.text import CountVectorizer , TfidfTransformer
            bow=CountVectorizer(analyzer=rmsw).fit(df_com['Customer Complaint'])
len(bow.vocabulary_)
            m_bow = bow.transform(df_com['Customer Complaint'])
            m_bow
            dfb = pd.DataFrame(m_bow.toarray() , columns=bow.get_feature_names())
            df_t=dfb.sum().sort_values(ascending=False)
df_t
```

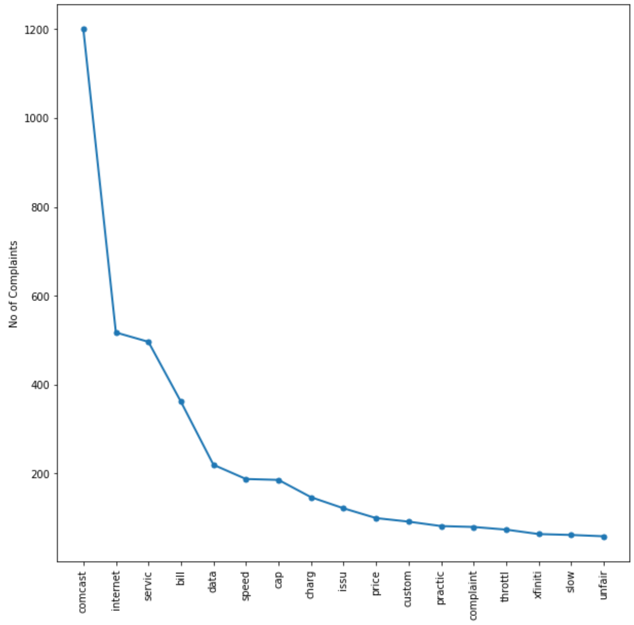


```
: comcast     1200
  internet     517
  servic       496
  bill         361
  data         219
              ...
  mistak         1
  misl           1
  mishandl       1
  misc           1
  0057           1
  Length: 1247, dtype: int64
```

```
df_t1.rename(columns = {'index' : 'Word' ,0 : 'count'} , inplace = True)
# count words that are greater than 50
        df_t2 = df_t1.loc[df_t1['count']>50]
df_t2
```

```
        Word    count
        0       comcast         1200
        1       internet        517
        2       servic  496
        3       bill    361
        4       data    219
        5       speed   187
        6       cap     185
        7       charg   146
        8       issu    121
        9       price   99
        10      custom 91
        11      practic  81
        12      complaint       79
        13      throttl  73
        14      xfiniti  63
        15      slow    61
16      unfair  58
```

```
        #df_t2['count'].plot(kind = 'barh' , figsize=(50,50))
        f = plt.figure()
        f.set_figwidth(10)
        f.set_figheight(10)
        plt.xticks(rotation = 90)
        plt.plot(df_t2['count'] , marker='o', linestyle='-',linewidth=2, markersize=5 ,  )
        plt.ylabel('No of Complaints')
        plt.show()
        #plt.xticks(rotation='vertical')
#plt.plot(df_t2['count'])
```

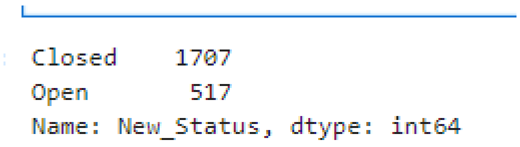


Observation
Most complaints are of Internet , service and Bill related

5. 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.
6. Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:
7. Which state has the maximum complaints
8. Which state has the highest percentage of unresolved complaints

```
df['New_Status'] =[ "Open" if Status=="Open" or Status=="Pending" else "Closed" for Status in df['Status'] ]
df.head()
```

```
df['New_Status'].value_counts()
```



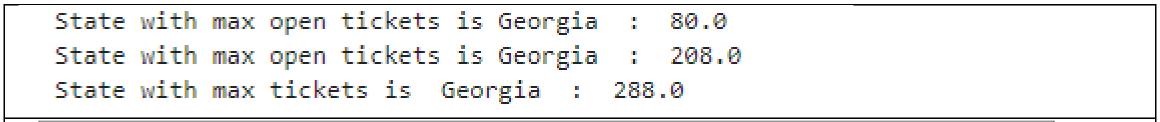
```
Closed    1707
Open       517
Name: New_Status, dtype: int64
```

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

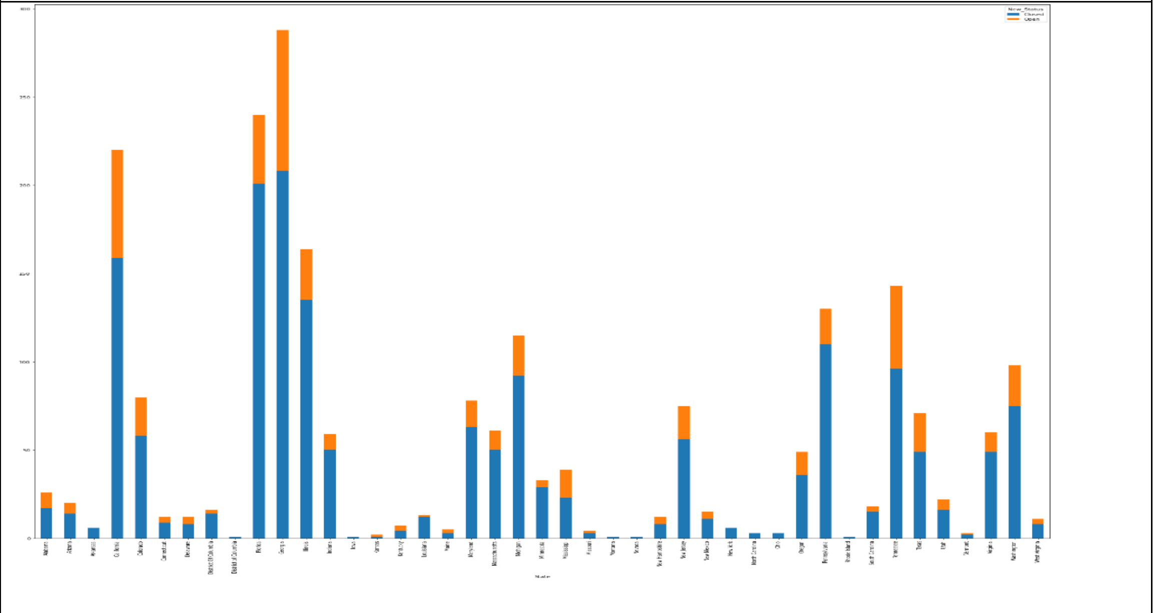
```
State
Alabama          26
Arizona         20
Arkansas          6
California      220
Colorado         80
Connecticut        12
```

```
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
```

```
        df.columns #
        St_df = df.groupby(['State','New_Status']).size().unstack().fillna(0).reset_index()
        St_df.columns
        St_df['Total'] = St_df['Open']+St_df['Closed']
        print ('State with max open tickets is', St_df.iloc[St_df['Open'].idxmax()]['State'],' :
        ',St_df['Open'].max())
        print ('State with max open tickets is', St_df.iloc[St_df['Closed'].idxmax()]['State'],' :
        ',St_df['Closed'].max())
        print ('State with max tickets is ', St_df.iloc[St_df['Total'].idxmax()]['State'],' :
        ',St_df['Total'].max())
df.groupby(['State','New_Status']).size().unstack().fillna(0).plot(kind='bar',stacked=True ,
figsize=(30,30))
```



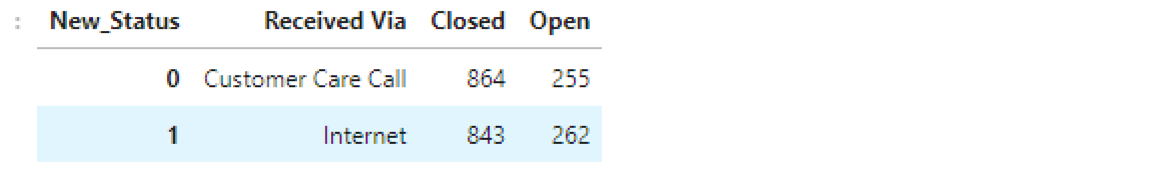
```
State with max open tickets is Georgia  :  80.0
State with max open tickets is Georgia  :  208.0
State with max tickets is  Georgia  :  288.0
```



State with max open tickets is Georgia : 80.0
State with max open tickets is Georgia : 208.0
State with max tickets is Georgia : 288.0

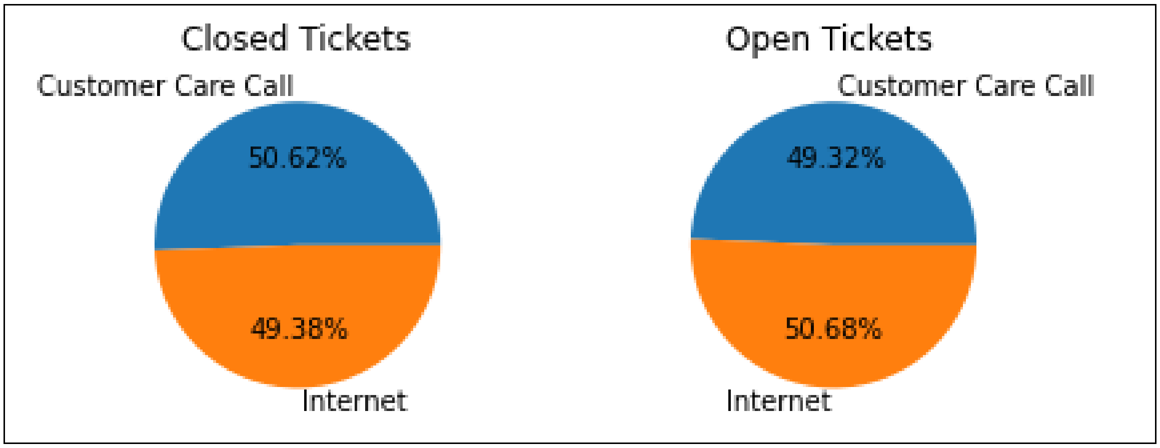
9. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

```
df_rcv = df.groupby(['Received Via','New_Status']).size().unstack().reset_index()
#count()['State']
df_rcv
```



New_Status	Received Via	Closed	Open
0	Customer Care Call	864	255
1	Internet	843	262

```
        plt.subplots_adjust(hspace=.5 , wspace=.5)
        plt.subplot(1,2,1)
        plt.title('Closed Tickets')
        plt.pie(df_rcv['Closed'],labels=df_rcv['Received Via'],autopct='%1.2f%%')
        plt.plot()
        plt.subplot(1,2,2)
        plt.title('Open Tickets ')
        plt.pie(df_rcv['Open'],labels=df_rcv['Received Via'],autopct='%1.2f%%')
plt.plot()
```



10. The analysis results to be provided with insights wherever applicable.