Comcast_telecom_complaints_Analysis-Correct last question

July 19, 2020

1 Analysis of Comcast telecom complaints

1.1 This notebook answer below questions:

1.1.1 Q1. Import data into Python environment.

1.1.2 Q2. Provide the trend chart for the number of complaints at monthly and daily granularity levels.

1.1.3 Q3. Provide a table with the frequency of complaint types.

1.1.4 Q4. Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

1.1.5 Q5. 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.

1.1.6 Q6. Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from above question. Provide insights on:

1.1.7 Q7. Which state has the maximum complaints

1.1.8 Q8. Which state has the highest percentage of unresolved complaints

1.1.9 Q9. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

import necessary libraries

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

1.1.10 Q1. Import data into Python environment.

[2]:
```
df_comcast_complaints = pd.read_csv("Comcast_telecom_complaints_data_old.csv")
```

view first 5 rows

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

[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

[4]:
```
df_comcast_complaints.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
```

[5]:
```
#check again if any column has nulls
df_comcast_complaints.isnull().sum()
```

[5]:
```
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
```

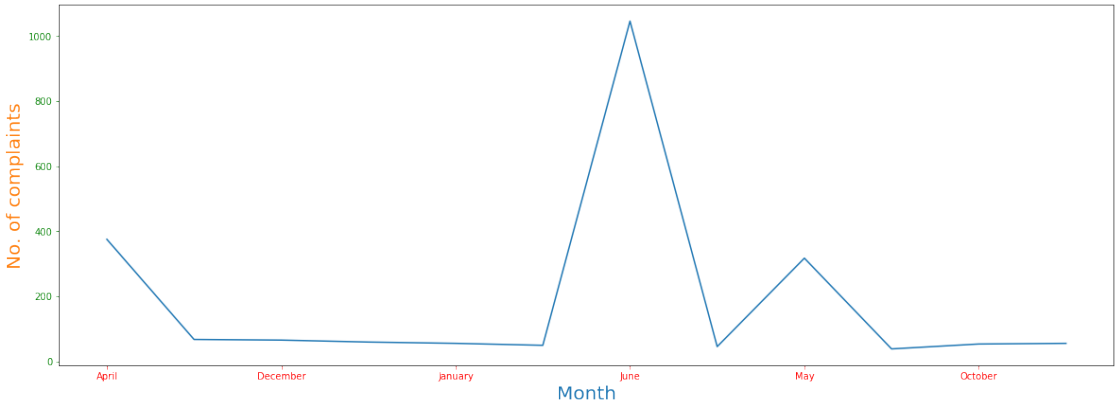
Convert Date and Time column to new DateTime Column

[6]:
```
df_comcast_complaints["DateTime"] = df_comcast_complaints.Date + " " +␣
 ↪df_comcast_complaints.Time
df_comcast_complaints["DateTime"] = pd.to_datetime(df_comcast_complaints.
 ↪DateTime, format="%d-%m-%y %I:%M:%S %p")
df_comcast_complaints["Date_month_year"] = pd.
 ↪to_datetime(df_comcast_complaints["Date_month_year"])
```

1.1.11 Q2. (a) Provide the trend chart for the number of complaints at monthly granularity levels.

---

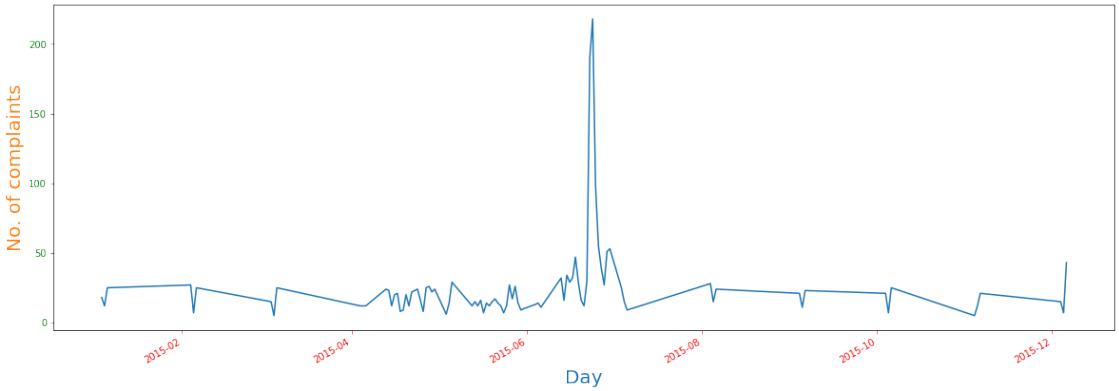
[7]:
```
df_comcast_complaints["month"] = pd.
 ↪to_datetime(df_comcast_complaints["DateTime"]).apply(lambda x: x.
 ↪strftime('%B'))
ax = df_comcast_complaints.groupby("month").count()["Customer Complaint"].
 ↪plot(kind = "line", figsize = (20, 7))
ax.set_xlabel("Month",  fontsize=20, color="C0")
ax.set_ylabel("No. of complaints", fontsize=20, color="C1")
ax.tick_params(axis='x', colors='red')
ax.tick_params(axis='y', colors='green')
```



1.1.12 Q2(b). Provide the trend chart for the number of complaints at daily granularity levels.

---

[8]:
```
ax = df_comcast_complaints.groupby("Date_month_year").count()["Customer␣
↪Complaint"].plot(kind = "line", figsize = (20, 7))
ax.set_xlabel("Day",  fontsize=20, color="C0")
ax.set_ylabel("No. of complaints", fontsize=20, color="C1")
ax.tick_params(axis='x', colors='red')
ax.tick_params(axis='y', colors='green')
```



Inspect Customer Complaint data

[9]:
```
pd.set_option('display.max_rows', 20)
```

```
df_comcast_complaints.groupby("Customer Complaint")["Customer Complaint"].
 ↪size().sort_values(ascending=False)
```

[9]:
```
Customer Complaint
Comcast                                                                 83
Comcast Internet                                                        18
Comcast Data Cap                                                        17
comcast                                                                 13
Comcast Data Caps                                                       11
                                                                        ..
Lack of availability                                                     1
Lack of communication and poor customer service                          1
Lack of consistent service                                               1
Lack of internet speed                                                   1
(Comcast is not my complaint!) Cyber Tele-marketing is my complaint!     1
Name: Customer Complaint, Length: 1841, dtype: int64
```

Let's find out most frequent complaints

[10]:
```
# Clean up stop words and punctuations
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string

stop = set(stopwords.words('english'))
stop.add("comcast") # let's exclude the company name
exclude = set(string.punctuation + "0123456789")
lemma = WordNetLemmatizer()

def cleanup_text(sentence):
    stop_free = " ".join([i for i in sentence.lower().split() if i not in stop])
    punc_free = "".join([ch for ch in stop_free if ch not in exclude])
    normalised = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalised
```

[11]:
```
sentence_total = df_comcast_complaints["Customer Complaint"].tolist()
all_sentence_2_words_list_cleaned = [cleanup_text(word).split() for word in␣
 ↪sentence_total]
all_words_cleaned = [item for sublist in all_sentence_2_words_list_cleaned for␣
 ↪item in sublist]
df_all_words = pd.DataFrame(all_words_cleaned, index =␣
 ↪range(len(all_words_cleaned)), columns=["Complaint Type"])
df_all_words.groupby("Complaint Type").size().sort_values(ascending=False)
```

[11]:
```
Complaint Type
internet    517
service     496
```

```
billing     283
data        219
speed       187
            …
nightly       1
night         1
nh            1
next          1
loses         1
Length: 1344, dtype: int64
```

1.1.13 From above we can answer below:

1.1.14 Q4. Which complaint types are maximum i.e., around internet, network issues, or across any other domains

1.1.15 A. internet is the highest complained service of Comcast, next is service and the billing, data and speed.

1.1.16 Q5. 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.

---

[12]:
```
#View the Status types and frequency
print(df_comcast_complaints.groupby('Status')["Status"].size())


#Create a function to add new column with new Status
def rename_Status(status):
    if status in ('Open','Pending'):
        return 'Open'
    elif status in ('Closed', 'Solved'):
        return 'Closed'
#Apply the function to Status Column to create newcolumn with new status
df_comcast_complaints["OpenClosed"] = df_comcast_complaints["Status"].
 ↪apply(rename_Status)

#View the new status type
df_comcast_complaints.groupby("OpenClosed").size()
```

```
Status
Closed     734
```

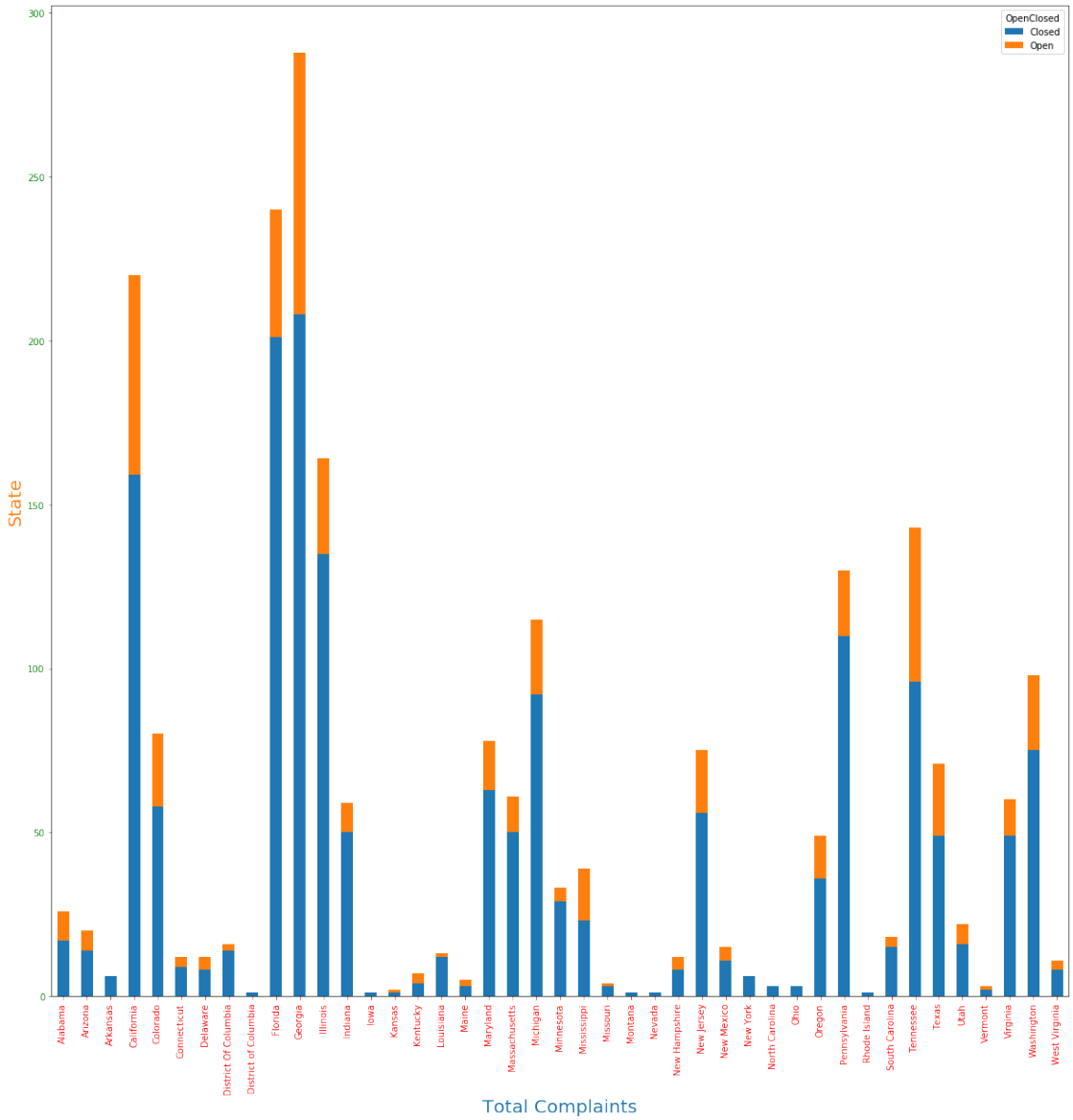
```
Open       363
Pending    154
Solved     973
Name: Status, dtype: int64
```

[12]:
```
OpenClosed
Closed    1707
Open       517
dtype: int64
```

Q6. Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from above question (Q5).

---

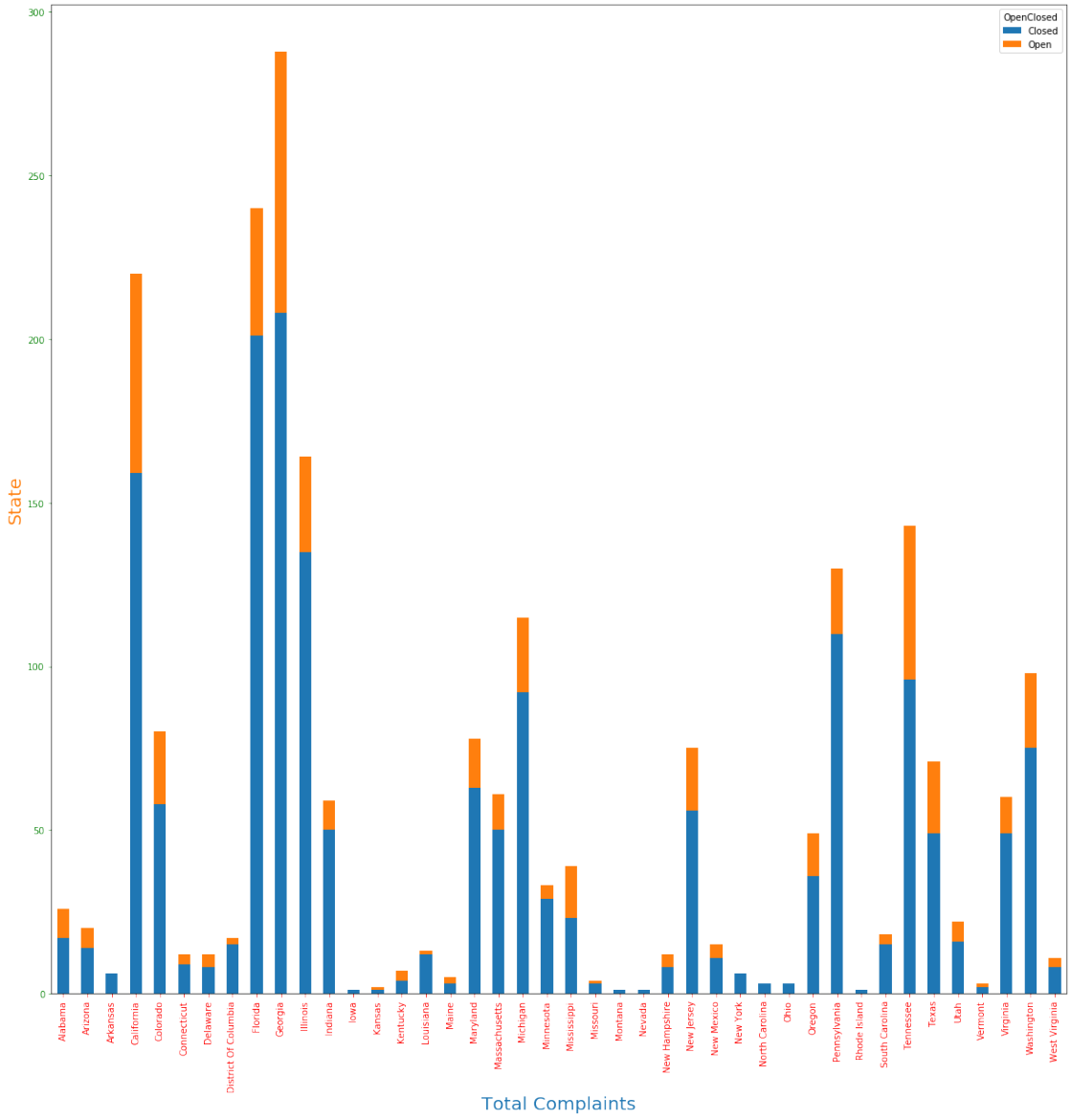
[13]:
```
#View the Total complaints by each State using stacked bar chart
ax = df_comcast_complaints.groupby(["State", "OpenClosed"]).size().unstack().
 ↪plot(kind = "bar",figsize=(20,20), stacked = True)
ax.set_xlabel("Total Complaints",  fontsize=20, color="C0")
ax.set_ylabel("State", fontsize=20, color="C1")
ax.tick_params(axis='x', colors='red')
ax.tick_params(axis='y', colors='green')
```



[14]:
```
#Adjust district of columbia as you can observe its repeated twice
df_comcast_complaints["State"].replace({"District of Columbia": "District Of␣
↪Columbia"}, inplace=True)
```

[15]:
```
#View the Total complaints by each State using stacked bar chart after␣
↪adjusting "District of columbia"
ax = df_comcast_complaints.groupby(["State", "OpenClosed"]).size().unstack().
↪plot(kind = "bar",figsize=(20,20), stacked = True)
ax.set_xlabel("Total Complaints",  fontsize=20, color="C0")
ax.set_ylabel("State", fontsize=20, color="C1")
ax.tick_params(axis='x', colors='red')
```

```
ax.tick_params(axis='y', colors='green')
```



[16]:
```
#View the Total complaints by each State
pd.set_option('display.max_rows', 50)
df_comcast_complaints.groupby("State").size().sort_values(ascending=False).
 ↪to_frame('Total Complaints')
```

[16]:

State	Total Complaints
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	17
New Mexico	15
Louisiana	13
New Hampshire	12
Connecticut	12
Delaware	12
West Virginia	11
Kentucky	7
New York	6
Arkansas	6
Maine	5
Missouri	4
Ohio	3
Vermont	3
North Carolina	3
Kansas	2
Rhode Island	1
Montana	1
Iowa	1
Nevada	1

From above we can answer below

Q7: Which state has the maximum complaints

A: Georgia state has the highest compaints

Q8. Which state has the highest percentage of unresolved complaints

---

[17]:
```
df_comcast_complaints.groupby(["State","OpenClosed"]).size().
↪sort_values(ascending=False).unstack().fillna(0).apply(lambda r: (r/r.
↪sum())*100, axis=1).sort_values("Open",ascending=False)
```

[17]:

OpenClosed	Closed	Open
State		
Kansas	50.000000	50.000000
Kentucky	57.142857	42.857143
Mississippi	58.974359	41.025641
Maine	60.000000	40.000000
Alabama	65.384615	34.615385
Vermont	66.666667	33.333333
New Hampshire	66.666667	33.333333
Delaware	66.666667	33.333333
Tennessee	67.132867	32.867133
Texas	69.014085	30.985915
Arizona	70.000000	30.000000
Georgia	72.222222	27.777778
California	72.272727	27.727273
Colorado	72.500000	27.500000
West Virginia	72.727273	27.272727
Utah	72.727273	27.272727
New Mexico	73.333333	26.666667
Oregon	73.469388	26.530612
New Jersey	74.666667	25.333333
Missouri	75.000000	25.000000
Connecticut	75.000000	25.000000
Washington	76.530612	23.469388
Michigan	80.000000	20.000000
Maryland	80.769231	19.230769
Virginia	81.666667	18.333333
Massachusetts	81.967213	18.032787
Illinois	82.317073	17.682927
South Carolina	83.333333	16.666667
Florida	83.750000	16.250000
Pennsylvania	84.615385	15.384615
Indiana	84.745763	15.254237
Minnesota	87.878788	12.121212
District Of Columbia	88.235294	11.764706
Louisiana	92.307692	7.692308

```
Iowa                 100.000000   0.000000
Rhode Island         100.000000   0.000000
Ohio                 100.000000   0.000000
North Carolina       100.000000   0.000000
New York             100.000000   0.000000
Arkansas             100.000000   0.000000
Nevada               100.000000   0.000000
Montana              100.000000   0.000000
```

[18]:
```
pd.set_option('display.max_rows', 5)
```

Q8/A: As we can see above "Kansas" has the highest number of unresolved complaints with 50 %

Q9. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

---

[19]:
```
#let's look at all "Received via" column values
df_comcast_complaints.groupby("Received Via").size().to_frame('Total')
```

[19]:
```
                    Total
Received Via
Customer Care Call   1119
Internet             1105
```

[26]:
```
#Lets filter the dataset again with received via Internet and Customer Care Call
df_internet_customercare_complaints_Received_Via = df_comcast_complaints[np.
 ↪logical_or(df_comcast_complaints["Received Via"] == "Internet",␣
 ↪df_comcast_complaints["Received Via"] == "Customer Care Call")]
df_internet_customercare_complaints_Received_Via_Internet =␣
 ↪df_comcast_complaints[df_comcast_complaints["Received Via"] == "Internet"]
df_internet_customercare_complaints_Received_Via_Customer_Care_Call =␣
 ↪df_comcast_complaints[df_comcast_complaints["Received Via"] == "Customer␣
 ↪Care Call"] #
```

[27]:
```
df_internet_customercare_complaints_Received_Via_Internet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1105 entries, 1 to 2221
Data columns (total 14 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Ticket #                    1105 non-null   object
```

```
 1   Customer Complaint           1105 non-null   object
 2   Date                         1105 non-null   object
 3   Date_month_year              1105 non-null   datetime64[ns]
 4   Time                         1105 non-null   object
 5   Received Via                 1105 non-null   object
 6   City                         1105 non-null   object
 7   State                        1105 non-null   object
 8   Zip code                     1105 non-null   int64
 9   Status                       1105 non-null   object
 10  Filing on Behalf of Someone  1105 non-null   object
 11  DateTime                     1105 non-null   datetime64[ns]
 12  month                        1105 non-null   object
 13  OpenClosed                   1105 non-null   object
dtypes: datetime64[ns](2), int64(1), object(11)
memory usage: 129.5+ KB
```

[28]:
```
df_internet_customercare_complaints_Received_Via_Customer_Care_Call.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1119 entries, 0 to 2223
Data columns (total 14 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Ticket #                     1119 non-null   object
 1   Customer Complaint           1119 non-null   object
 2   Date                         1119 non-null   object
 3   Date_month_year              1119 non-null   datetime64[ns]
 4   Time                         1119 non-null   object
 5   Received Via                 1119 non-null   object
 6   City                         1119 non-null   object
 7   State                        1119 non-null   object
 8   Zip code                     1119 non-null   int64
 9   Status                       1119 non-null   object
 10  Filing on Behalf of Someone  1119 non-null   object
 11  DateTime                     1119 non-null   datetime64[ns]
 12  month                        1119 non-null   object
 13  OpenClosed                   1119 non-null   object
dtypes: datetime64[ns](2), int64(1), object(11)
memory usage: 131.1+ KB
```

[42]:
```
df_internet_customercare_complaints_Received_Via.groupby(["Received␣
↪Via","OpenClosed"])["OpenClosed"].size().unstack()
```

[42]:

OpenClosed	Closed	Open
Received Via		
Customer Care Call	864	255
Internet	843	262

[43]:
```
df_internet_customercare_complaints_Received_Via.groupby(["Received␣
↪Via","OpenClosed"])["OpenClosed"].size().unstack().apply(lambda x: 100*x/x.
↪sum(), axis =1)
```

[43]:

OpenClosed	Closed	Open
Received Via		
Customer Care Call	77.211796	22.788204
Internet	76.289593	23.710407

From above we can answer below:

Q9/A: Total percentage of complaints resolved till date, which were received through the Internet - 76.289593% and customer care calls - 77.211796 % :

2 © - July 2020 : Lanka, Nirmal Kumar