

Partha_Comcast_Final _Project

February 12, 2021

0.0.1 Comcast Telecom Consumer Complaints:

Q1. Import and Analysis:

Q1. A. Import data into Python environment.

Q1. B. Provide the trend chart for the number of complaints at monthly and daily granularity 1

Q1. C. Provide a table with the frequency of complaint types.

Q2. Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

Q3. 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.

Q4. Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:

Q4. A. Which state has the maximum complaints .

Q4. B. Which state has the highest percentage of unresolved complaints.

Q5. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

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

0.0.2 Q1. A.

0.0.3 Import data into Python environment.

```
[2]: # Here index_col=0 is removing the 0 indexed column.
comcast_df=pd.read_csv('Comcast_telecom_complaints_data.csv',index_col=0)
```

```
[3]: # Display the first three rows of the dataset.
comcast_df.head(3)
```

```
[3]:
```

| | Customer Complaint | Date \ |
|----------|--|----------|
| Ticket # | | |
| 250635 | Comcast Cable Internet Speeds | 22-04-15 |
| 223441 | Payment disappear - service got disconnected | 04-08-15 |
| 242732 | Speed and Service | 18-04-15 |

| | Date_month_year | Time | Received Via | City | State \ |
|----------|-----------------|-------------|--------------------|----------|----------|
| Ticket # | | | | | |
| 250635 | 22-Apr-15 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland |
| 223441 | 04-Aug-15 | 10:22:56 AM | Internet | Acworth | Georgia |
| 242732 | 18-Apr-15 | 9:55:47 AM | Internet | Acworth | Georgia |

| | Zip code | Status | Filing on Behalf of Someone |
|----------|----------|--------|-----------------------------|
| Ticket # | | | |
| 250635 | 21009 | Closed | No |
| 223441 | 30102 | Closed | No |
| 242732 | 30101 | Closed | Yes |

```
[ ]:
```

0.1 EDA(Explanatory Data Analysis) and Cleanup the Dataset

```
[4]: # Check the null values present in the dataset(Column wise):
comcast_df.isnull().sum()
```

```
[4]: 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]: # or
# To check if there is any null or not in the whole dataset:
comcast_df.isnull().values.any()
```

```
[5]: False
```

So There is no Null values in the whole dataset.

Also to count the scan for null value checkin in the dataset is : `comcast_df.isnull().count()`

1

```
[6]: # The describe() function computes a summary of statistics pertaining to the
      ↪ DataFrame columns.
      # (include='all') parameter is assuring he inclusion of all column in this
      ↪ calculation of the function.
      # 'all' : All columns of the input will be included in the output. OR we can
      ↪ say that Describing a column from a DataFrame by accessing it as an
      ↪ attribute:
      comcast_df.describe(include='all')
```

```
[6]:
```

| | Customer Complaint | Date | Date_month_year | Time | \ |
|--------|--------------------|----------|-----------------|------------|---|
| count | 2224 | 2224 | 2224 | 2224 | |
| unique | 1841 | 91 | 91 | 2190 | |
| top | Comcast | 24-06-15 | 24-Jun-15 | 5:28:32 PM | |
| freq | 83 | 218 | 218 | 2 | |
| mean | NaN | NaN | NaN | NaN | |
| std | NaN | NaN | NaN | NaN | |
| min | NaN | NaN | NaN | NaN | |
| 25% | NaN | NaN | NaN | NaN | |
| 50% | NaN | NaN | NaN | NaN | |
| 75% | NaN | NaN | NaN | NaN | |
| max | 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 |

```
[7]: comcast_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2224 entries, 250635 to 363614
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Customer Complaint                    2224 non-null   object
1   Date                                  2224 non-null   object
2   Date_month_year                       2224 non-null   object
3   Time                                  2224 non-null   object
4   Received Via                          2224 non-null   object
5   City                                  2224 non-null   object
6   State                                 2224 non-null   object
7   Zip code                             2224 non-null   int64
8   Status                               2224 non-null   object
9   Filing on Behalf of Someone           2224 non-null   object
dtypes: int64(1), object(9)
memory usage: 191.1+ KB
```

```
[8]: comcast_df.head(1)
```

```
[8]:
```

| | Customer Complaint | Date | Date_month_year | Time | \ |
|----------|-------------------------------|----------|-----------------|------------|---|
| Ticket # | | | | | |
| 250635 | Comcast Cable Internet Speeds | 22-04-15 | 22-Apr-15 | 3:53:50 PM | |

| | Received Via | City | State | Zip code | Status | \ |
|----------|--------------------|----------|----------|----------|--------|---|
| Ticket # | | | | | | |
| 250635 | Customer Care Call | Abingdon | Maryland | 21009 | Closed | |

| | Filing on Behalf of Someone |
|----------|-----------------------------|
| Ticket # | |
| 250635 | No |

```
[ ]:
```

1.0.1 Q1.B.

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

```
[9]: # Convert the 'Date_month_year' field from string object to date type object.
comcast_df['Date_month_year']=pd.to_datetime(comcast_df['Date_month_year'])
comcast_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2224 entries, 250635 to 363614
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Customer Complaint                    2224 non-null   object
1   Date                                  2224 non-null   object
2   Date_month_year                      2224 non-null   datetime64[ns]
3   Time                                  2224 non-null   object
4   Received Via                         2224 non-null   object
5   City                                  2224 non-null   object
6   State                                2224 non-null   object
7   Zip code                             2224 non-null   int64
8   Status                               2224 non-null   object
9   Filing on Behalf of Someone          2224 non-null   object
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 191.1+ KB
```

```
[10]: # Extract the month, day from the the given field to create seperate new fields:
comcast_df['Create_Month'] = comcast_df['Date_month_year'].apply(lambda x: x.
    ↪month)
comcast_df['Create_Day'] = comcast_df['Date_month_year'].apply(lambda x: x.day)
comcast_df.head(2)
```

```
[10]:
```

| | Customer Complaint | Date \ |
|----------|--|----------|
| Ticket # | | |
| 250635 | Comcast Cable Internet Speeds | 22-04-15 |
| 223441 | Payment disappear - service got disconnected | 04-08-15 |

| | Date_month_year | Time | Received Via | City | State \ |
|----------|-----------------|-------------|--------------------|----------|----------|
| Ticket # | | | | | |
| 250635 | 2015-04-22 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland |
| 223441 | 2015-08-04 | 10:22:56 AM | Internet | Acworth | Georgia |

| | Zip code | Status | Filing on Behalf of Someone | Create_Month \ |
|----------|----------|--------|-----------------------------|----------------|
| Ticket # | | | | |
| 250635 | 21009 | Closed | No | 4 |

| | | | | |
|--------|-------|--------|----|---|
| 223441 | 30102 | Closed | No | 8 |
|--------|-------|--------|----|---|

| | Create_Day |
|--|------------|
|--|------------|

| Ticket # | |
|----------|----|
| 250635 | 22 |
| 223441 | 4 |

```
[11]: # To get the date wise day of the weeeek for the field named_
      ↪ 'Create_Days_of_Week'
comcast_df['Create_Days_of_Week']=comcast_df['Date_month_year'].dt.day_name()

# Change the 'Create_Month' field from number to name:
comcast_df['Create_Month']= comcast_df['Date_month_year'].dt.month_name()

comcast_df.head(3)
```

```
[11]:
```

| | Customer Complaint | Date \ |
|----------|--|----------|
| Ticket # | | |
| 250635 | Comcast Cable Internet Speeds | 22-04-15 |
| 223441 | Payment disappear - service got disconnected | 04-08-15 |
| 242732 | Speed and Service | 18-04-15 |

| | Date_month_year | Time | Received Via | City | State \ |
|----------|-----------------|-------------|--------------------|----------|----------|
| Ticket # | | | | | |
| 250635 | 2015-04-22 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland |
| 223441 | 2015-08-04 | 10:22:56 AM | Internet | Acworth | Georgia |
| 242732 | 2015-04-18 | 9:55:47 AM | Internet | Acworth | Georgia |

| | Zip code | Status | Filing on Behalf of Someone | Create_Month \ |
|----------|----------|--------|-----------------------------|----------------|
| Ticket # | | | | |
| 250635 | 21009 | Closed | No | April |
| 223441 | 30102 | Closed | No | August |
| 242732 | 30101 | Closed | Yes | April |

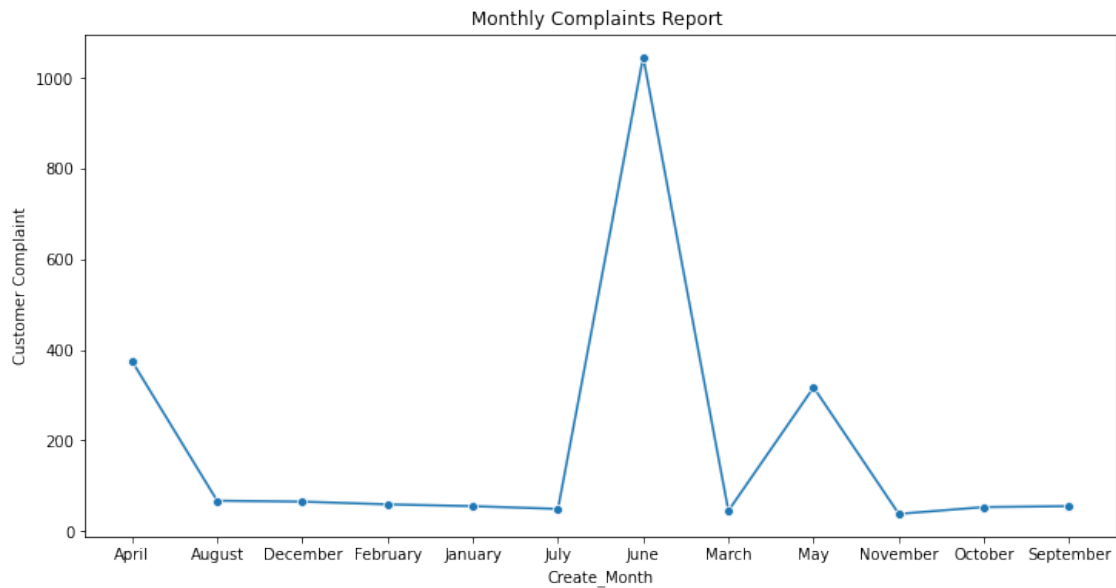
| | Create_Day | Create_Days_of_Week |
|----------|------------|---------------------|
| Ticket # | | |
| 250635 | 22 | Wednesday |
| 223441 | 4 | Tuesday |
| 242732 | 18 | Saturday |

```
[12]: # Find the number of complaints Monthly and Plot:
```

```
[13]: # Apply plotting using Seaborn:
plt.figure(figsize=(12,6))
Group_by_Month=comcast_df.groupby('Create_Month').count().reset_index()
```

```
Comp_M_Plot = sns.lineplot(x='Create_Month', y= 'Customer Complaint', data =_
↳Group_by_Month, marker='o')
Comp_M_Plot.set_title("Monthly Complaints Report ")
```

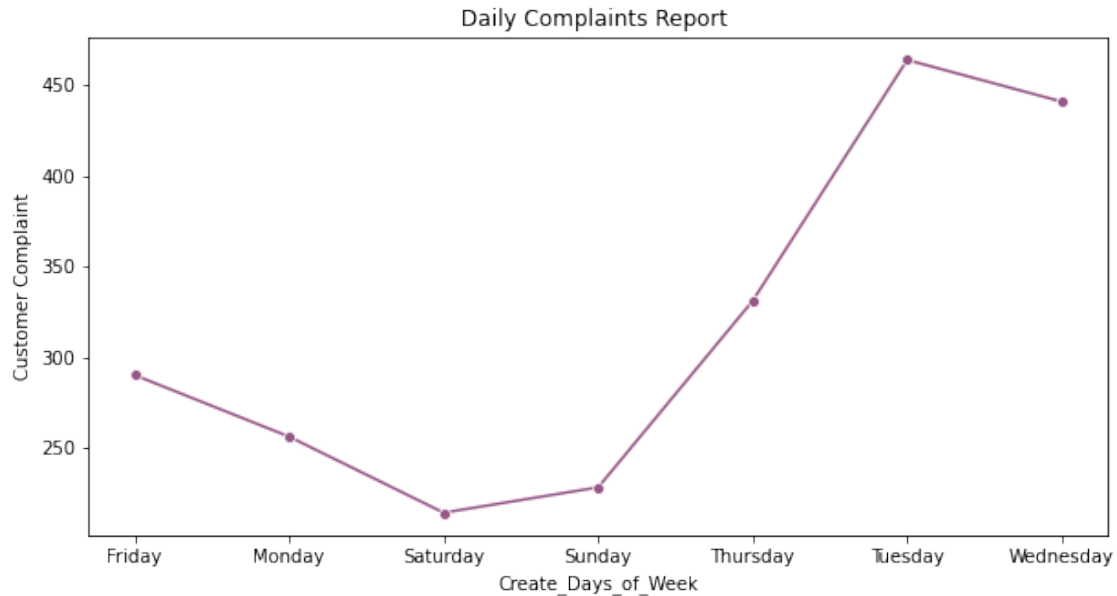
[13]: Text(0.5, 1.0, 'Monthly Complaints Report ')



```
[14]: plt.figure(figsize=(10,5))
Group_by_Day=comcast_df.groupby('Create_Days_of_Week').count().reset_index()
#Group_by_Day['Create_Day']=Group_by_Day['Date_month_year'].dt.day_name()
Comp_D_Plot = sns.lineplot(x='Create_Days_of_Week', y= 'Customer Complaint',_
↳data = Group_by_Day, marker='o', color='#965786')

Comp_D_Plot.set_title("Daily Complaints Report ")
```

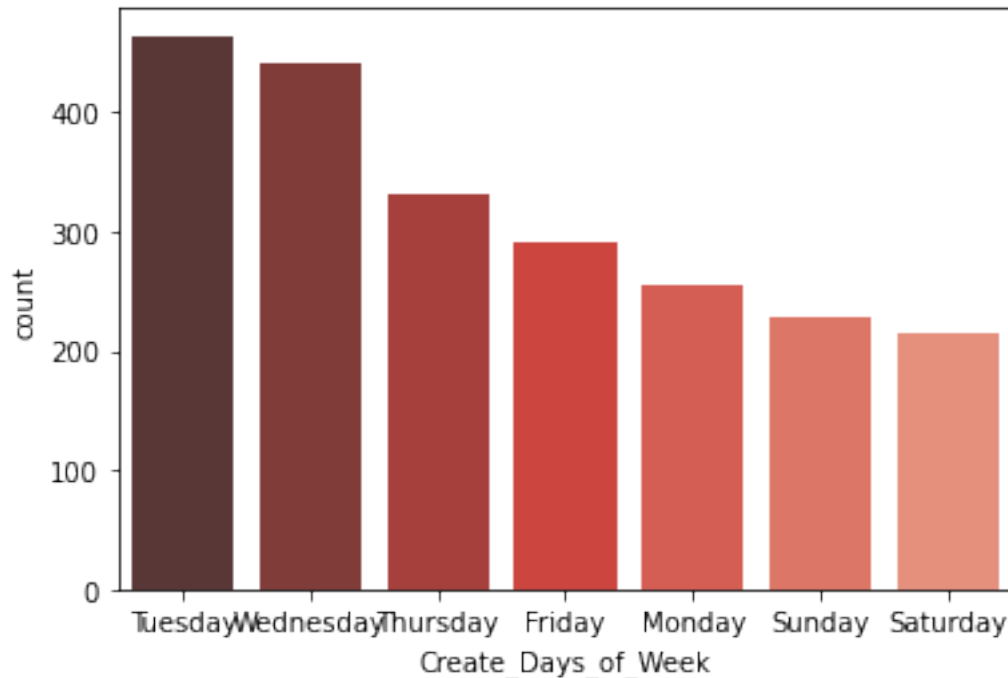
[14]: Text(0.5, 1.0, 'Daily Complaints Report ')



1.1 Conclusion:----- Tuesday and Wednesday is having maximum complaints report.

```
[15]: # OR We can use another way to represent the daily complaints report:
sns.countplot(x='Create_Days_of_Week', data = comcast_df,
              order=comcast_df['Create_Days_of_Week'].value_counts().index,
              palette = "Reds_d")
```

```
[15]: <AxesSubplot:xlabel='Create_Days_of_Week', ylabel='count'>
```

1.2 Above is the Daily Complaints Status.

1.2.1 Q1.C.

1.2.2 Provide a table with the frequency of complaint types.

```
[16]: comcast_df['Customer Complaint'] = comcast_df['Customer Complaint'].str.title()
      Comp_freq_Table = comcast_df['Customer Complaint'].value_counts()
      Comp_freq_Table
```

```
[16]: Comcast
      102
      Comcast Data Cap
      30
      Comcast Internet
      29
      Comcast Data Caps
      21
      Comcast Billing
      18
      ...
      Internet System Reliability
      1
```

Data Usage

1

Comcast: No Service For One Month

1

Video Throttling

1

Long Term Billing Issue With Rude Customer Service Caused Depression And Trauma

1

Name: Customer Complaint, Length: 1740, dtype: int64

```
[17]: import nltk
      %pip install wordcloud
```

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: wordcloud in /usr/local/lib/python3.7/site-packages (1.6.0)

Requirement already satisfied: pillow in /usr/local/lib/python3.7/site-packages (from wordcloud) (7.1.1)

Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/site-packages (from wordcloud) (3.3.0)

Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.7/site-packages (from wordcloud) (1.18.2)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /usr/local/lib/python3.7/site-packages (from matplotlib->wordcloud) (2.4.6)

Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/site-packages (from matplotlib->wordcloud) (2.8.1)

Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/site-packages (from matplotlib->wordcloud) (0.10.0)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/site-packages (from matplotlib->wordcloud) (1.2.0)

Requirement already satisfied: six in /usr/local/lib/python3.7/site-packages (from cycler>=0.10->matplotlib->wordcloud) (1.14.0)

WARNING: You are using pip version 20.3.3; however, version 21.0.1 is available.

You should consider upgrading via the '/usr/local/bin/python3.7 -m pip install --upgrade pip' command.

Note: you may need to restart the kernel to use updated packages.

1.2.3 Q2.

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

```
[18]: from wordcloud import WordCloud, STOPWORDS
common_complaints = comcast_df['Customer Complaint'].dropna().tolist()
common_complaints = ''.join(common_complaints).lower()

list_to_stops = _
↳('Comcast', 'Now', 'Company', 'Day', 'Someone', 'Thing', 'Also', 'Got', 'Way', 'Call', 'Called', 'One'

for word in list_to_stops:
    STOPWORDS.add(word)

wordcloud = WordCloud(stopwords=STOPWORDS,
                       background_color='white',
                       width=1200,
                       height=1000).generate(common_complaints)

plt.figure( figsize=(10,12) )
plt.imshow(wordcloud)
plt.title('Frequent words for customer complaints')
plt.axis('off')
plt.show()
```



```

punc_free = "".join([ch for ch in stop_free if ch not in exclude])
normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
return normalized

doc_complete = comcast_df['Customer Complaint'].tolist()
doc_clean = [clean(doc).split() for doc in doc_complete]

import gensim
from gensim import corpora

dictionary = corpora.Dictionary(doc_clean)
dictionary

doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
doc_term_matrix

from gensim.models import LdaModel

num_topic = 9
ldamodel = LdaModel(doc_term_matrix,num_topics=num_topic,id2word = _
↳dictionary,passes=10)

topics = ldamodel.show_topics()
for topic in topics:
    print(topic)
    print()

word_dict = {}
for i in range(num_topic):
    words = ldamodel.show_topic(i,topn = 20)
    word_dict['Topic '+"{}".format(i)]=[i[0] for i in words]

pd.DataFrame(word_dict)

import pyLDAvis.gensim

Lda_display = pyLDAvis.gensim.
↳prepare(ldamodel,doc_term_matrix,dictionary,sort_topics=False)
pyLDAvis.display(Lda_display)

```

[nltk_data] Downloading package wordnet to /home/labsuser/nltk_data...

```
[nltk_data]   Unzipping corpora/wordnet.zip.

(0, '0.172*"internet" + 0.167*"speed" + 0.046*"slow" + 0.029*"comcast" +
0.021*"false" + 0.020*"promised" + 0.014*"intermittent" + 0.013*"paying" +
0.013*"month" + 0.013*"charged"')

(1, '0.136*"comcast" + 0.128*"complaint" + 0.043*"fee" + 0.035*"internet" +
0.030*"service" + 0.028*"connection" + 0.025*"fraudulent" + 0.022*"home" +
0.011*"comcastxfinity" + 0.011*"provider"')

(2, '0.087*"comcast" + 0.064*"contract" + 0.045*"price" + 0.035*"cable" +
0.023*"xfinitycomcast" + 0.022*"monopoly" + 0.022*"year" + 0.020*"back" +
0.016*"equipment" + 0.016*"overcharge"')

(3, '0.229*"service" + 0.066*"customer" + 0.049*"internet" + 0.045*"poor" +
0.022*"pay" + 0.017*"misleading" + 0.016*"comcast" + 0.016*"unreliable" +
0.015*"without" + 0.014*"option"')

(4, '0.117*"comcast" + 0.116*"charge" + 0.025*"overage" + 0.023*"cramming" +
0.023*"account" + 0.022*"internet" + 0.021*"comcastxfinity" + 0.020*"issue" +
0.017*"several" + 0.016*"connectivity"')

(5, '0.078*"comcast" + 0.069*"internet" + 0.066*"service" + 0.039*"problem" +
0.030*"outage" + 0.019*"customer" + 0.018*"bad" + 0.017*"horrible" +
0.014*"claim" + 0.013*"day"')

(6, '0.112*"comcast" + 0.057*"service" + 0.029*"without" + 0.026*"internet" +
0.025*"bill" + 0.021*"help" + 0.019*"failure" + 0.018*"high" + 0.016*"charging"
+ 0.015*"payment"')

(7, '0.177*"billing" + 0.145*"comcast" + 0.067*"issue" + 0.057*"practice" +
0.045*"unfair" + 0.032*"service" + 0.029*"pricing" + 0.026*"xfinity" +
0.020*"deceptive" + 0.020*"monopolistic"')

(8, '0.214*"comcast" + 0.106*"data" + 0.086*"cap" + 0.078*"internet" +
0.064*"service" + 0.024*"throttling" + 0.020*"usage" + 0.020*"bill" +
0.011*"xfinity" + 0.010*"day"')
```

[19]: <IPython.core.display.HTML object>

```
[20]: comcast_df.head(3)
```

```
[20]:
```

| | Customer Complaint | Date \ |
|----------|--|----------|
| Ticket # | | |
| 250635 | Comcast Cable Internet Speeds | 22-04-15 |
| 223441 | Payment Disappear - Service Got Disconnected | 04-08-15 |
| 242732 | Speed And Service | 18-04-15 |

| | Date_month_year | Time | Received Via | City | State \ |
|----------|-----------------|-------------|--------------------|----------|----------|
| Ticket # | | | | | |
| 250635 | 2015-04-22 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland |
| 223441 | 2015-08-04 | 10:22:56 AM | Internet | Acworth | Georgia |
| 242732 | 2015-04-18 | 9:55:47 AM | Internet | Acworth | Georgia |

| | Zip code | Status | Filing on Behalf of Someone | Create_Month \ |
|----------|----------|--------|-----------------------------|----------------|
| Ticket # | | | | |
| 250635 | 21009 | Closed | No | April |
| 223441 | 30102 | Closed | No | August |
| 242732 | 30101 | Closed | Yes | April |

| | Create_Day | Create_Days_of_Week |
|----------|------------|---------------------|
| Ticket # | | |
| 250635 | 22 | Wednesday |
| 223441 | 4 | Tuesday |
| 242732 | 18 | Saturday |

1.3.1 Q3.

1.3.2 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.

```
[21]: comcast_df['New_Comp_Status'] = ["Open" if Status=="Open" or Status=="Pending"
    ↪else "Closed" for Status in comcast_df["Status"]]
```

```
comcast_df['New_Comp_Status'].unique()
```

```
[21]: array(['Closed', 'Open'], dtype=object)
```

```
[22]: comcast_df.head(3)
```

```
[22]:
```

| | Customer Complaint | Date \ |
|----------|--|----------|
| Ticket # | | |
| 250635 | Comcast Cable Internet Speeds | 22-04-15 |
| 223441 | Payment Disappear - Service Got Disconnected | 04-08-15 |
| 242732 | Speed And Service | 18-04-15 |

| | Date_month_year | Time | Received Via | City | State \ |
|----------|-----------------|-------------|--------------------|----------|----------|
| Ticket # | | | | | |
| 250635 | 2015-04-22 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland |
| 223441 | 2015-08-04 | 10:22:56 AM | Internet | Acworth | Georgia |
| 242732 | 2015-04-18 | 9:55:47 AM | Internet | Acworth | Georgia |

| Ticket # | Zip code | Status | Filing on Behalf of Someone | Create_Month | \ |
|----------|----------|--------|-----------------------------|--------------|---|
| 250635 | 21009 | Closed | No | April | |
| 223441 | 30102 | Closed | No | August | |
| 242732 | 30101 | Closed | Yes | April | |

| Ticket # | Create_Day | Create_Days_of_Week | New_Comp_Status |
|----------|------------|---------------------|-----------------|
| 250635 | 22 | Wednesday | Closed |
| 223441 | 4 | Tuesday | Closed |
| 242732 | 18 | Saturday | Closed |

1.4 Conclusion: ----- New Categorical variable created with the values 'Closed' and 'Open'.

1.4.1 Q4.

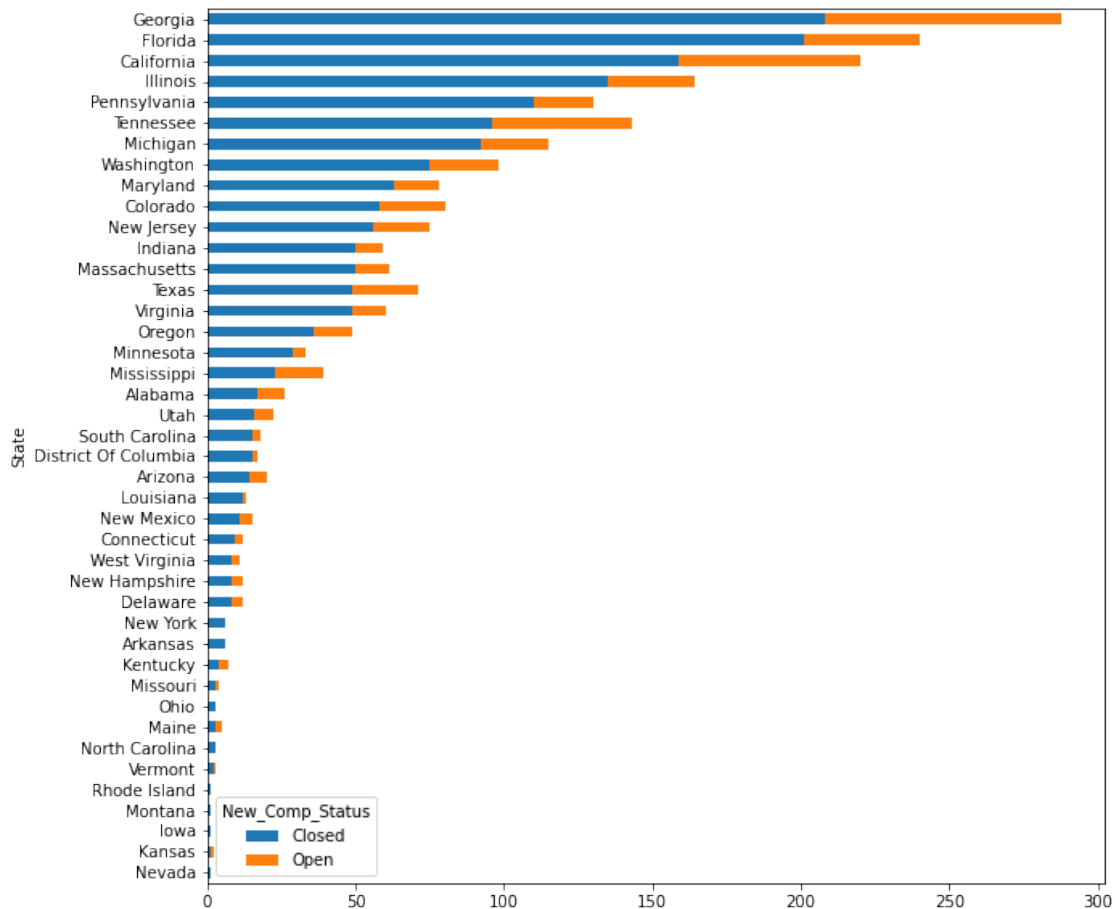
1.4.2 Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3.

```
[23]: comcast_df['State'] = comcast_df['State'].str.title()
st_comp = comcast_df.groupby(['State', 'New_Comp_Status']).size().unstack().
    ↪ fillna(0)

st_comp

st_comp.sort_values('Closed', axis = 0, ascending=True).plot(kind="barh",
    ↪ figsize=(10,10), stacked=True)
```

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

1.4.3 Q.4. A.

1.4.4 Which state has the maximum complaints? [Provide Insight]

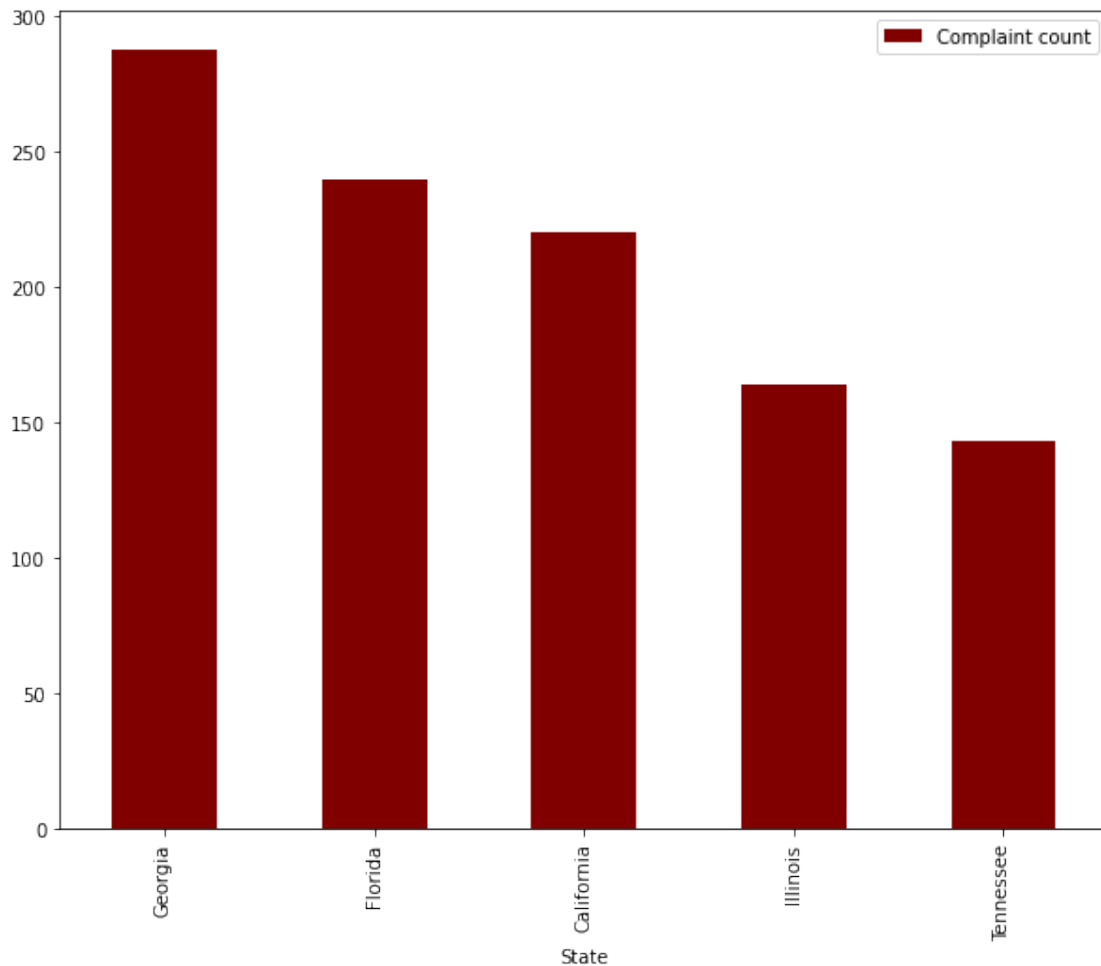
```
[24]: Comp_max=comcast_df.groupby(["State"]).size().sort_values(ascending=False).
      ↪to_frame().rename({0: "Complaint count"}, axis=1)[:5]
Comp_max
```

```
[24]:      Complaint count
State
Georgia                288
Florida                240
California              220
Illinois                164
Tennessee              143
```

1.5 Conclusion: ----- Georgia has highest complaints [Insight below]

```
[25]: Comp_max.plot(kind="bar", figsize=(10,8), stacked=True, color='maroon')
```

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



```
[26]: Comp_Status = comcast_df.groupby(['State','New_Comp_Status']).size().unstack().  
      ↪ fillna(0)  
      Comp_Status.sort_values('Closed',axis = 0,ascending=False)[:1]
```

```
[26]: New_Comp_Status  Closed  Open  
State  
Georgia           208.0  80.0
```

```
[ ]:
```

1.5.1 Q.4. B.

1.5.2 Which state has the highest percentage of unresolved complaints [Provide Insight]

```
[27]: #highest percentage of unresolved complaints
Comp_Status['Resolved_cmp_prct'] = Comp_Status['Closed']/Comp_Status['Closed'].
↳sum()*100
Comp_Status['Unresolved_cmp_prct'] = Comp_Status['Open']/Comp_Status['Open'].
↳sum()*100
Comp_Status.head(3)
```

```
[27]: New_Comp_Status  Closed  Open  Resolved_cmp_prct  Unresolved_cmp_prct
State
Alabama             17.0   9.0             0.995899             1.740812
Arizona             14.0   6.0             0.820152             1.160542
Arkansas             6.0   0.0             0.351494             0.000000
```

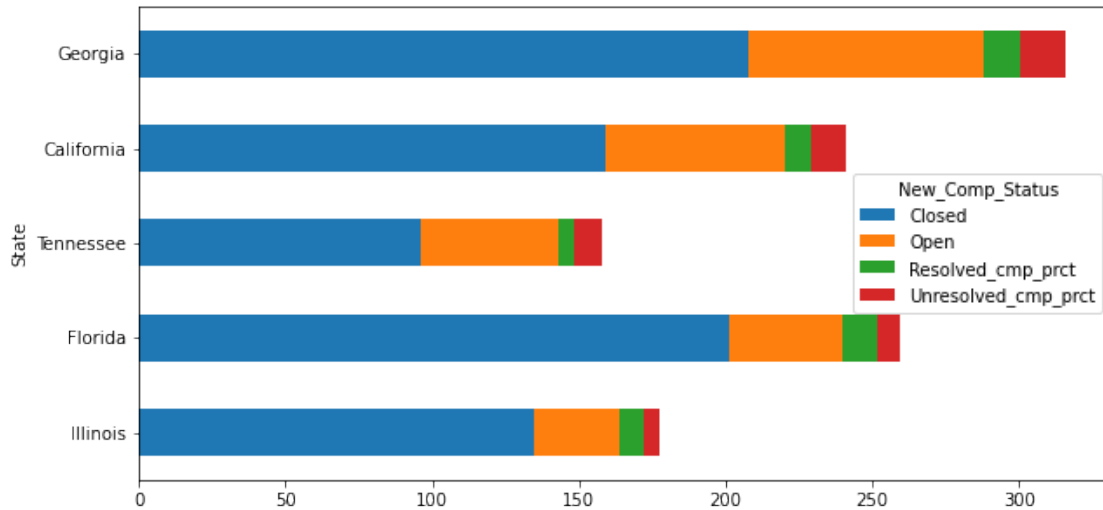
```
[28]: C_Unresolved=Comp_Status.sort_values('Unresolved_cmp_prct',axis =0,
↳0,ascending=False)[:5]
C_Unresolved
```

```
[28]: New_Comp_Status  Closed  Open  Resolved_cmp_prct  Unresolved_cmp_prct
State
Georgia             208.0  80.0             12.185120             15.473888
California           159.0  61.0              9.314587             11.798839
Tennessee            96.0  47.0              5.623902              9.090909
Florida              201.0  39.0             11.775044              7.543520
Illinois             135.0  29.0              7.908612              5.609284
```

1.6 Conclusion: ----- Georgia state has highest Unresolved complaints when compared to other states.

```
[29]: C_Unresolved.sort_values('Unresolved_cmp_prct',axis =0, ascending=True).
↳plot(kind='barh',figsize=(10,5),stacked=True)
```

```
[29]: <AxesSubplot:ylabel='State'>
```



1.6.1 Q.5.

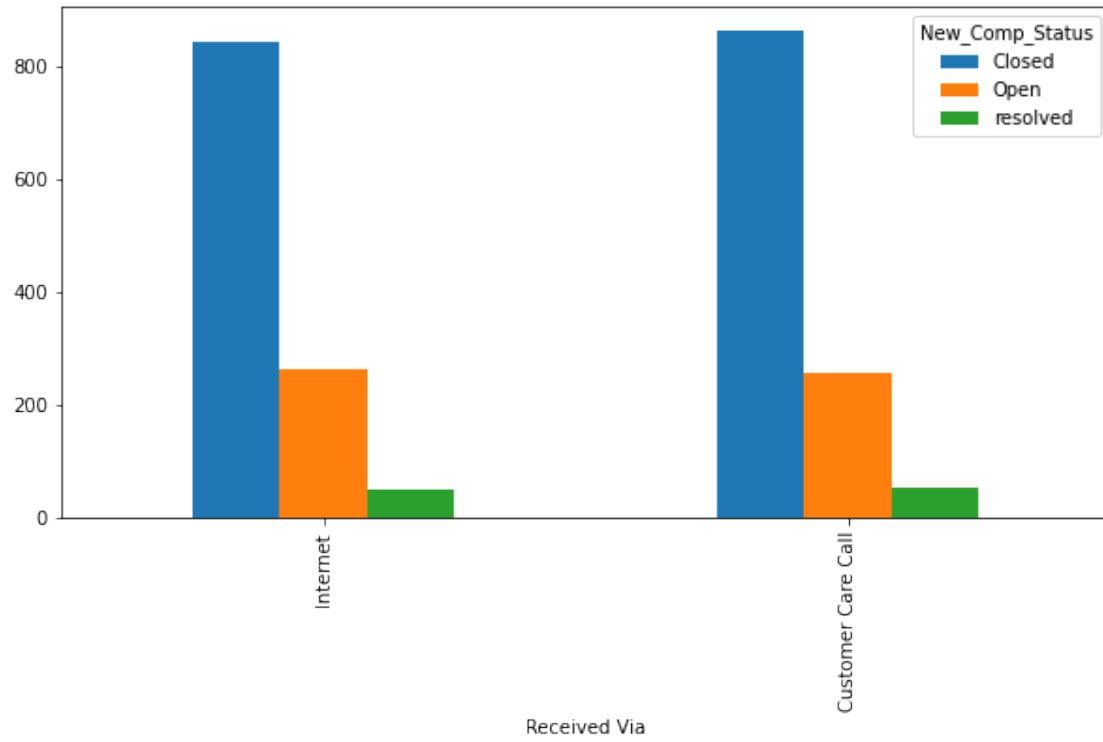
1.6.2 Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

```
[30]: Comp_Resolved = comcast_df.groupby(['Received Via', 'New_Comp_Status']).size().
      ↪unstack().fillna(0)
      Comp_Resolved['resolved'] = Comp_Resolved['Closed']/Comp_Resolved['Closed'].
      ↪sum()*100
      Comp_Resolved['resolved']
```

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

```
[31]: Comp_Resolved.sort_values('resolved', axis=0, ascending=True).
      ↪plot(kind='bar', figsize=(10,5), stacked=False)
```

```
[31]: <AxesSubplot:xlabel='Received Via'>
```



[]:

Thank You

2

[]: