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

In [2]:
```
### Import data into Python environment
comcast = pd.read_csv("D:/SimpliLearn-DataScience/2) Post Graduate Program in Data Science/5. P
```

In [3]:
```
comcast.head()
```

Out[3]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing Behalf Someo
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing Behalf Someo
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	
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	

In [4]:
```
comcast[comcast.isnull()].count()
```

Out[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
```

In [5]:
```
comcast.describe(include='all')
```

Out[5]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	F B S
count	2224	2224	2224	2224	2224	2224	2224	2224	2224.000000	2224	
unique	2224	1841	91	91	2190	2	928	43	NaN	4	
top	336981	Comcast	24-06-15	24-Jun-15	1:26:23 PM	Customer Care Call	Atlanta	Georgia	NaN	Solved	
freq	1	83	218	218	2	1119	63	288	NaN	973	
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	47994.393435	NaN	
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28885.279427	NaN	
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1075.000000	NaN	
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	30056.500000	NaN	
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	37211.000000	NaN	
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	77058.750000	NaN	
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	99223.000000	NaN	

In [6]:
```
# EDA and Cleanup the data set
comcast.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2224 entries, 0 to 2223
Data columns (total 11 columns):
```

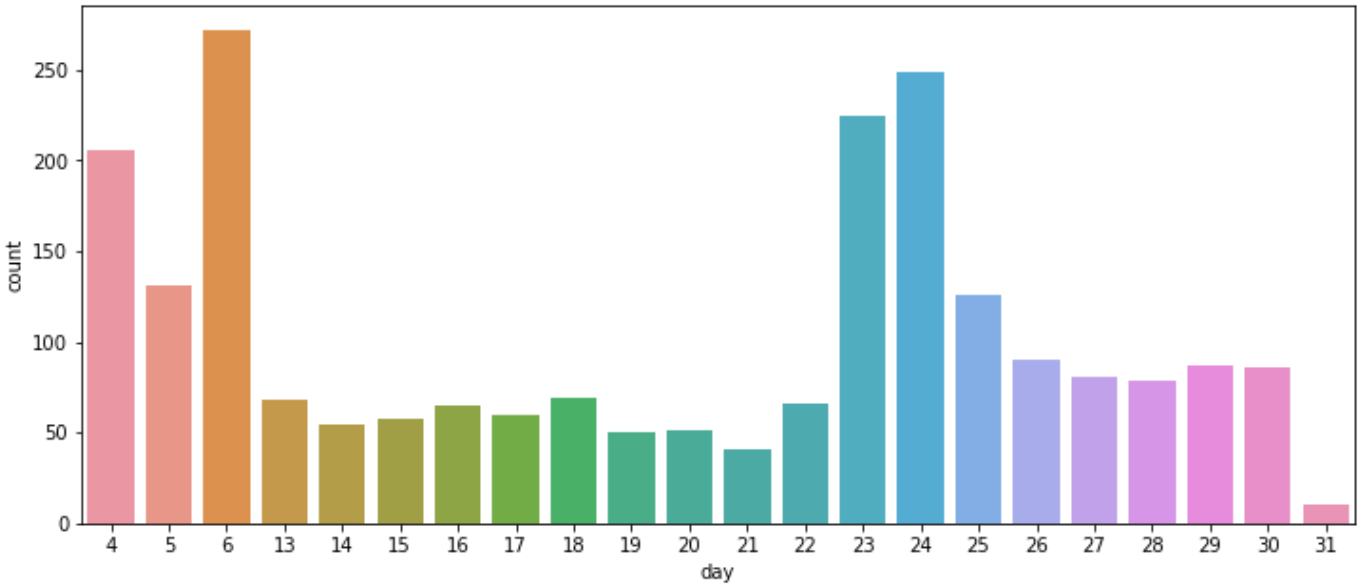
```
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Ticket #                   2224 non-null   object
 1   Customer Complaint         2224 non-null   object
 2   Date                       2224 non-null   object
 3   Date_month_year            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
```

In [7]:

```
import datetime
comcast.Date = pd.to_datetime(comcast.Date, format='%d-%m-%y')
```

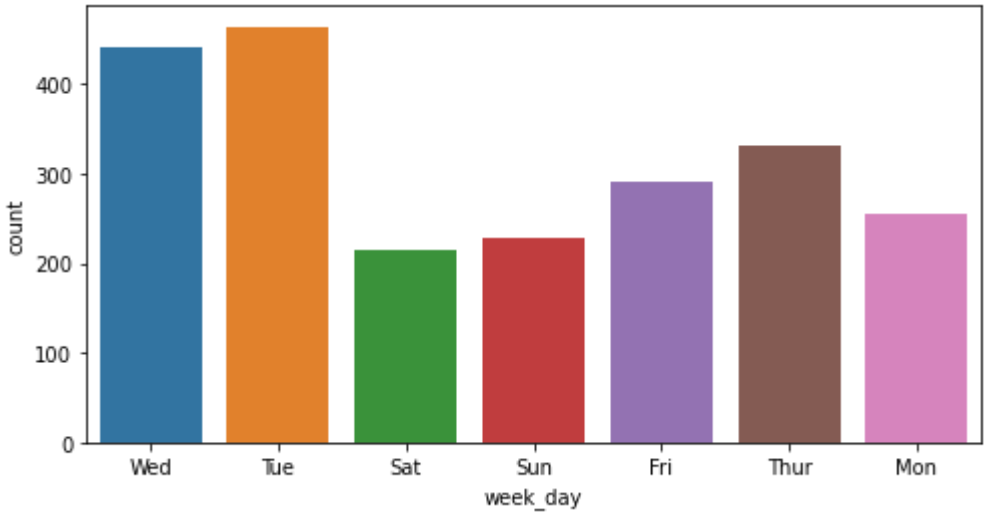
In [8]:

```
### Provide the trend chart for the number of complaints at monthly and daily granularity level
comcast['day'] = comcast['Date'].dt.day
plt.figure(figsize=(12,5))
sns.countplot(x="day", data=comcast);    # trend at daily granularity levels
# Maximum complaints are recieved on 6th day of month.
```



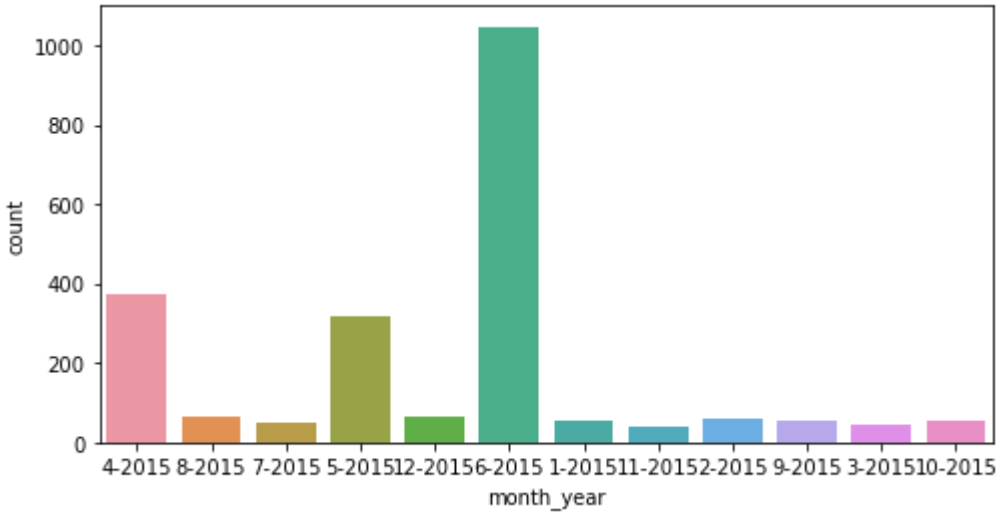
In [9]:

```
month = comcast['Date'].dt.month.astype(str)
year = comcast['Date'].dt.year.astype(str)
comcast["month_year"]= month.str.cat(year, sep ="-")
week_day = comcast['Date'].dt.dayofweek
dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thur',4:'Fri',5:'Sat',6:'Sun'}
comcast['week_day']=week_day.map(dmap)
plt.figure(figsize=(8,4))
sns.countplot(x="week_day", data=comcast);     # trend at week day granularity levels
# Maximum complaints are recieved on Tuesday and Wednesday
```



In [10]:

```
plt.figure(figsize=(8,4))
sns.countplot(x="month_year", data=comcast);      # trend at monthly granularity levels
# Maximum complaints are in June-2015 month
```



In [11]:

```
### Provide a table with the frequency of complaint types.
comcast['Customer Complaint'].str.lower().value_counts()
```

Out[11]:

```
comcast                                                 102
comcast data cap                                         30
comcast internet                                         29
comcast data caps                                        21
comcast billing                                          18
                                                       ...
comcast internet in emeryville throttles speed            1
fraudulent charges and poor or no service                 1
data allowance                                            1
outrageous internet bill adjustments from comcast         1
data caps for cable isp;s                                 1
Name: Customer Complaint, Length: 1740, dtype: int64
```

In [12]:

```
# import nltk
# nltk.download('wordnet')
# %pip install wordcloud
```

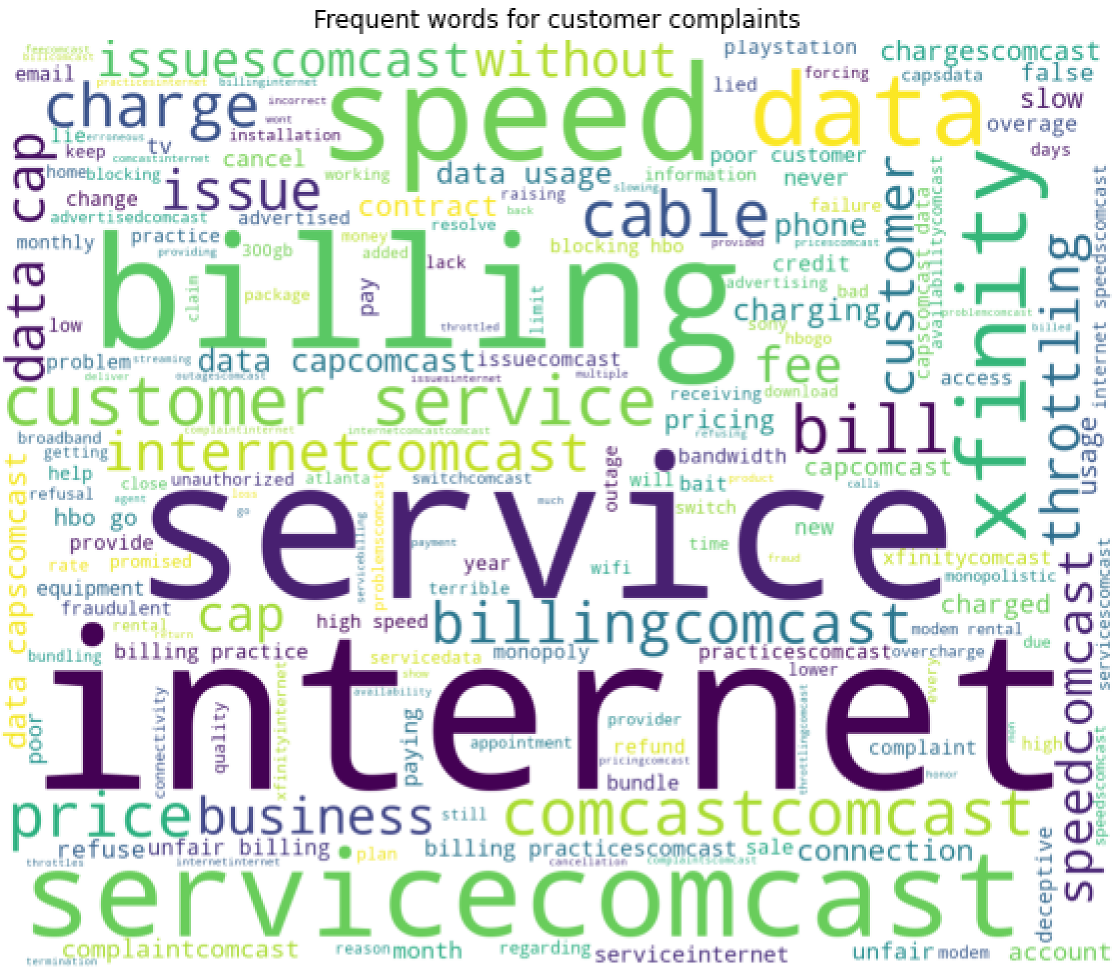
In [13]:

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

list_stops = ('Comcast','Now','Company','Day','Someone','Thing','Also','Got','Way','Call','Call
```

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

In [14]:

```
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()
# Maximum complaints are related to internet, followed by service and billing.
```



In [15]:

```
### Create a new categorical variable with value as Open and Closed. Open & Pending is to be cc
comcast['Status_Update'] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" for Sta
comcast['Status_Update'].unique()
```

Out[15]: array(['Closed', 'Open'], dtype=object)

In [16]:

```
"""
Provide state wise status of complaints in a stacked bar chart. Use the categorized variable fr
 - Which state has the maximum complaints
 - Which state has the highest percentage of unresolved complaints
"""
comcast['State'] = comcast['State'].str.title()
state_compl = comcast.groupby(['State','Status_Update']).size().unstack().fillna(0)
state_compl
```

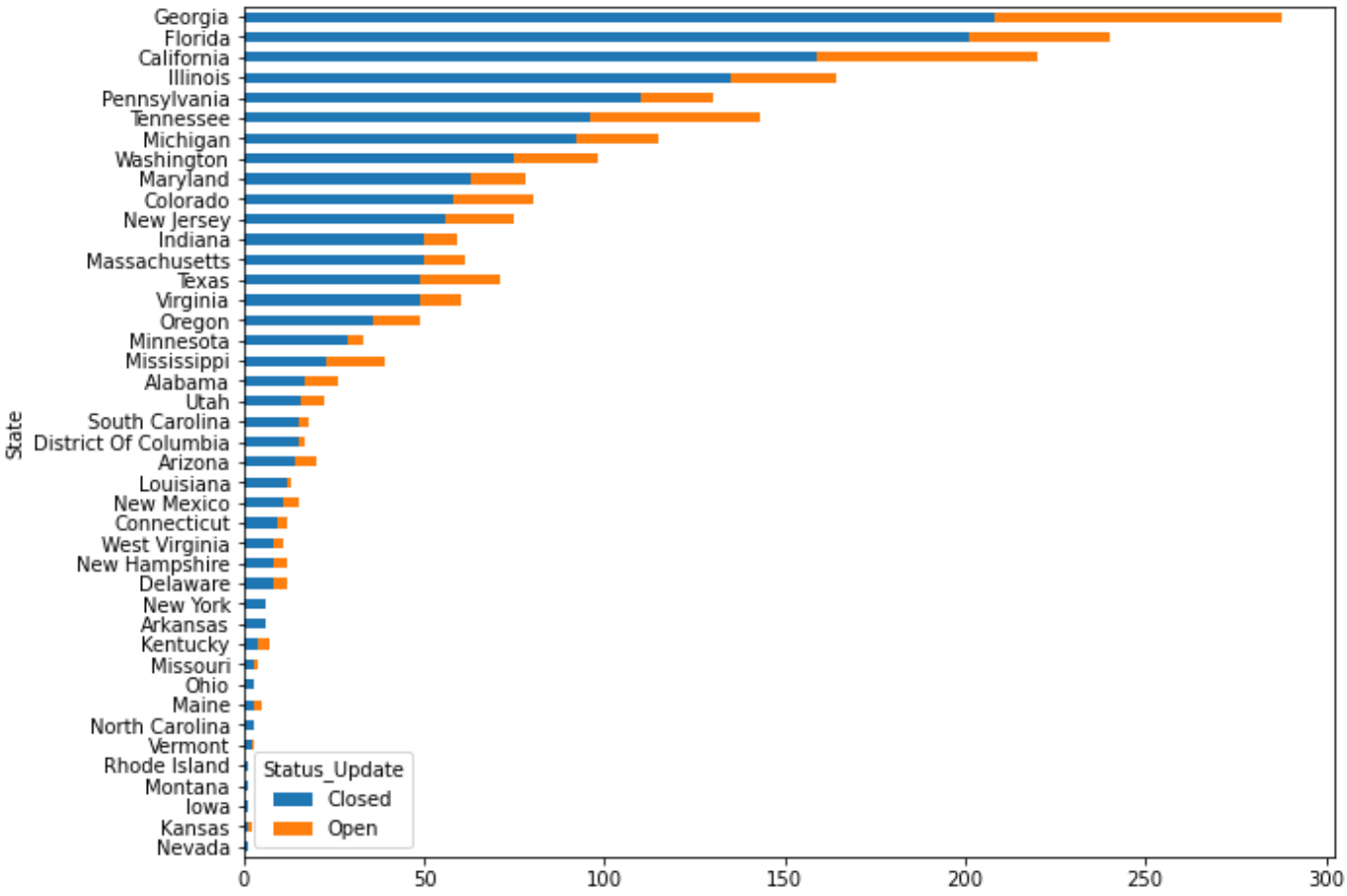
Out[16]:

Status_Update	Closed	Open
State		

Status_Update	Closed	Open
State		
Alabama	17.0	9.0
Arizona	14.0	6.0
Arkansas	6.0	0.0
California	159.0	61.0
Colorado	58.0	22.0
Connecticut	9.0	3.0
Delaware	8.0	4.0
District Of Columbia	15.0	2.0
Florida	201.0	39.0
Georgia	208.0	80.0
Illinois	135.0	29.0
Indiana	50.0	9.0
Iowa	1.0	0.0
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	0.0
Nevada	1.0	0.0
New Hampshire	8.0	4.0
New Jersey	56.0	19.0
New Mexico	11.0	4.0
New York	6.0	0.0
North Carolina	3.0	0.0
Ohio	3.0	0.0
Oregon	36.0	13.0
Pennsylvania	110.0	20.0
Rhode Island	1.0	0.0
South Carolina	15.0	3.0
Tennessee	96.0	47.0
Texas	49.0	22.0
Utah	16.0	6.0

Status_Update	Closed	Open
State		
Vermont	2.0	1.0
Virginia	49.0	11.0
Washington	75.0	23.0
West Virginia	8.0	3.0

In [17]:
```
state_compl.sort_values('Closed',axis = 0,ascending=True).plot(kind="barh", figsize=(10,8), sta
```



In [18]:
```
state_compl['Total'] = state_compl['Open'] + state_compl['Closed']
state_compl.sort_values("Total", axis = 0, ascending = False)[:1]
# Georgia has highest complaints - 288
```

Out[18]:

Status_Update	Closed	Open	Total
State			
Georgia	208.0	80.0	288.0

In [19]:
```
state_compl['Perc_Unres'] = state_compl['Open']/ state_compl['Open'].sum()*100
state_compl.sort_values("Perc_Unres", axis = 0, ascending = False)[:1]
# Georgia state has highest Unresolved complaints compared to other states which is around 15.4
```

Out[19]:

Status_Update	Closed	Open	Total	Perc_Unres
State				
Georgia	208.0	80.0	288.0	15.473888

In [20]:
```
# Provide the percentage of complaints resolved till date, which were received through the Inte
```

```
compl_res = comcast.groupby(['Received Via','Status_Update']).size().unstack().fillna(0)
compl_res['resolved'] = compl_res['Closed']/compl_res['Closed'].sum()*100
compl_res['resolved']
```

Out[20]:

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

In [ ]:

```
```