Comcast Telecom Consumer Complaints Project

In [23]:
```
# importing required libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [24]:
```
# loading the data
comcast_df = pd.read_csv('comcast.csv')
# viewing top 5 rows of dataset
comcast_df.head()
```

Out[24]:

	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

Checking for null values

In [25]:
```
comcast_df[comcast_df.isnull()].count()
# Zero null values found in dataset
```

Out[25]:
```
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 [26]:
```
comcast_df.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 [27]:
```
comcast_df.shape
```

Out[27]:
```
(2224, 11)
```

Parsing dates

In [28]:
```
comcast_df['Date_month_year'] = pd.to_datetime(comcast_df['Date_month_year'])
```

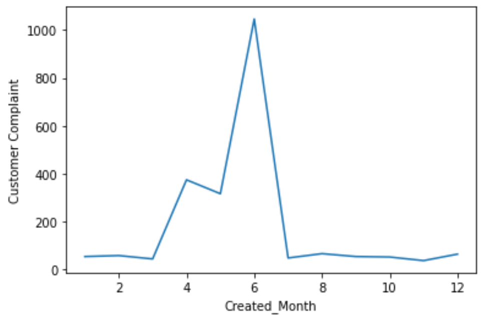
Seperating days and months for monthly and yearly granularity level charts

```
In [29]: comcast_df['Created_Year'] =  comcast_df['Date_month_year'].dt.year
         comcast_df['Created_Month'] = comcast_df['Date_month_year'].dt.month
         comcast_df['Created_Day'] = comcast_df['Date_month_year'].dt.day
         comcast_df['Created_Day_Name'] = comcast_df['Date_month_year'].dt.dayofweek
```

```
In [30]: dmap = {0:'Mon',1:'Tue',2:'Wed',3:'Thur',4:'Fri',5:'Sat',6:'Sun'}
         comcast_df['Created_Day_Name']=comcast_df['Created_Day_Name'].map(dmap)
```

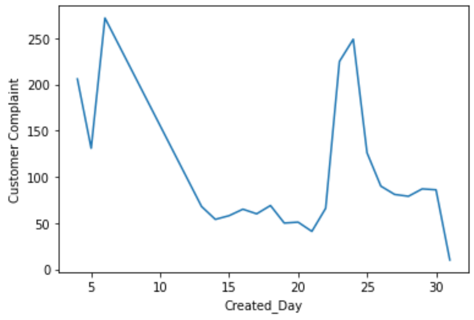
Number of Complaints Monthly

```
In [31]: bymonth = comcast_df.groupby('Created_Month').count().reset_index()
         sns.lineplot(x='Created_Month',y='Customer Complaint',data=bymonth).axes

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7fda3be635d0>
```

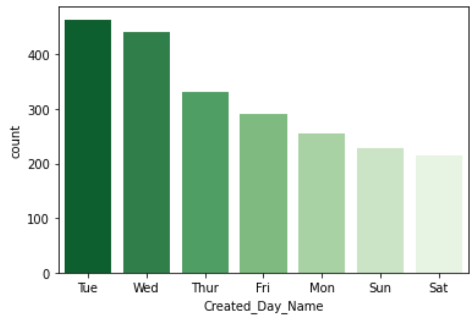


Number of Complaints Daily

```
In [32]: byday = comcast_df.groupby('Created_Day').count().reset_index()
         sns.lineplot(x='Created_Day',y='Customer Complaint',data=byday).axes

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7fda3bdfa910>
```



```
In [33]: sns.countplot(x='Created_Day_Name', data = comcast_df, order=comcast_df['Created_Day_Name'].value_counts().index, palette="Greens_r")

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7fda3ac764d0>
```



Insight: Most complaints are registered on Tuesday

Provide a table with the frequency of complaint types.

In [34]:
```
comcast_df['Customer Complaint'].value_counts()
# top 5 complaint types
```

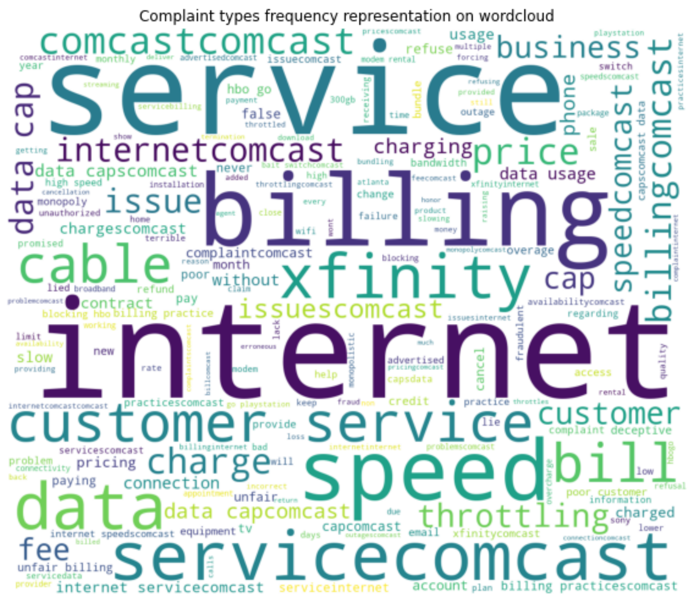
Out[34]:
```
Comcast                                            83
Comcast Internet                                   18
Comcast Data Cap                                   17
comcast                                            13
Comcast Billing                                    11
                                                   ..
Comcast refused to install internet                 1
internet services                                   1
Comcast Internet usage caps                         1
Comcast.                                            1
comcast lowering internet speeds on constant basis  1
Name: Customer Complaint, Length: 1841, dtype: int64
```

Which type of complaints are more

In [35]:
```
pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /usr/local/lib/python3.7/dist-packages (1.5.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from wordcloud) (7.1.2)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.7/dist-packages (from wordcloud) (1.19.5)
```

In [36]:
```
from wordcloud import WordCloud, STOPWORDS
complaint_types = comcast_df['Customer Complaint'].dropna().tolist()
complaint_types =''.join(complaint_types).lower()

# adding unnecessary words from customer complaint to stop words
list_stops = ('Comcast','Now','Company','Day','Someone','Thing','Also','Got','Way','Call','Called','One','Said','Tell')

for word in list_stops:
    STOPWORDS.add(word)

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

plt.figure( figsize=(10,15) )
plt.imshow(wordcloud)
plt.title('Complaint types frequency representation on wordcloud')
plt.axis('off')
plt.show()
```



From wordcloud we can observe that internet related complaint types are more, followed by service and billing complaints

Complaint Types table

In [37]:
```python
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

In [38]:
```python
stop_words = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()
```

In [39]:
```python
# defining clean function to remove stopwords, punctuations and applying lemmatizer to each word
def clean(doc):
  stop_free = " ".join([i for i in doc.lower().split() if i not in stop_words])
  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
```

In [40]:
```python
# loading customer complaint data and cleaning data using above defined function
complaint_doc = comcast_df['Customer Complaint'].tolist()
cleaned_doc = [clean(doc).split() for doc in complaint_doc]
```

In [41]:
```python
from gensim.corpora import Dictionary
```

In [42]:
```python
dct = Dictionary(cleaned_doc)
dct_term_matrix = [dct.doc2bow(doc) for doc in cleaned_doc]
```

In [43]:
```python
from gensim.models import LdaModel
```

In [44]:
```python
num_topics = 9
ldamodel = LdaModel(dct_term_matrix,num_topics=num_topics,id2word=dct,passes=10)
```

In [45]:
```python
topics = ldamodel.show_topics()
for topic in topics:
  print(topic)
```

```
(0, '0.120*"complaint" + 0.116*"comcast" + 0.092*"service" + 0.035*"paying" + 0.034*"connection" + 0.018*"terrible" + 0.016*"unreliable" +
0.015*"slowing" + 0.014*"failure" + 0.014*"access"')
(1, '0.288*"comcast" + 0.118*"internet" + 0.069*"service" + 0.034*"charge" + 0.028*"throttling" + 0.020*"xfinity" + 0.017*"problem" + 0.013
*"pricing" + 0.013*"fraudulent" + 0.011*"business"')
(2, '0.096*"comcast" + 0.027*"monopoly" + 0.027*"switch" + 0.022*"false" + 0.022*"home" + 0.021*"advertising" + 0.018*"contract" + 0.018*"e
mail" + 0.018*"bait" + 0.015*"availability"')
(3, '0.044*"bill" + 0.042*"comcast" + 0.036*"comcastxfinity" + 0.029*"charged" + 0.029*"promised" + 0.028*"high" + 0.026*"back" + 0.024*"ac
count" + 0.021*"installation" + 0.017*"without"')
(4, '0.213*"speed" + 0.190*"internet" + 0.061*"slow" + 0.024*"lack" + 0.021*"comcast" + 0.020*"bandwidth" + 0.019*"help" + 0.015*"rate" +
0.012*"provider" + 0.012*"throttle"')
(5, '0.176*"service" + 0.094*"comcast" + 0.057*"internet" + 0.048*"customer" + 0.036*"cable" + 0.028*"charge" + 0.017*"bill" + 0.014*"overa
ge" + 0.013*"cramming" + 0.013*"fee"')
(6, '0.151*"comcast" + 0.144*"billing" + 0.133*"data" + 0.108*"cap" + 0.044*"issue" + 0.038*"practice" + 0.034*"unfair" + 0.025*"usage" +
0.012*"xfinity" + 0.012*"monopolistic"')
(7, '0.077*"internet" + 0.070*"service" + 0.042*"day" + 0.037*"poor" + 0.037*"deceptive" + 0.032*"pay" + 0.030*"sale" + 0.029*"outage" + 0.
028*"connectivity" + 0.018*"several"')
(8, '0.071*"comcast" + 0.048*"price" + 0.042*"internet" + 0.031*"bill" + 0.028*"without" + 0.027*"month" + 0.023*"2" + 0.022*"show" + 0.018
*"get" + 0.018*"intermittent"')
```

In [46]:
```python
# Arranging data into tables
word_dict = {}
for i in range(num_topics):
  words = ldamodel.show_topic(i,topn=20)
  word_dict['Topic ' + "{}".format(i)] = [i[0] for i in words]
```

```
In [47]: pd.DataFrame(word_dict)
```

Out[47]:

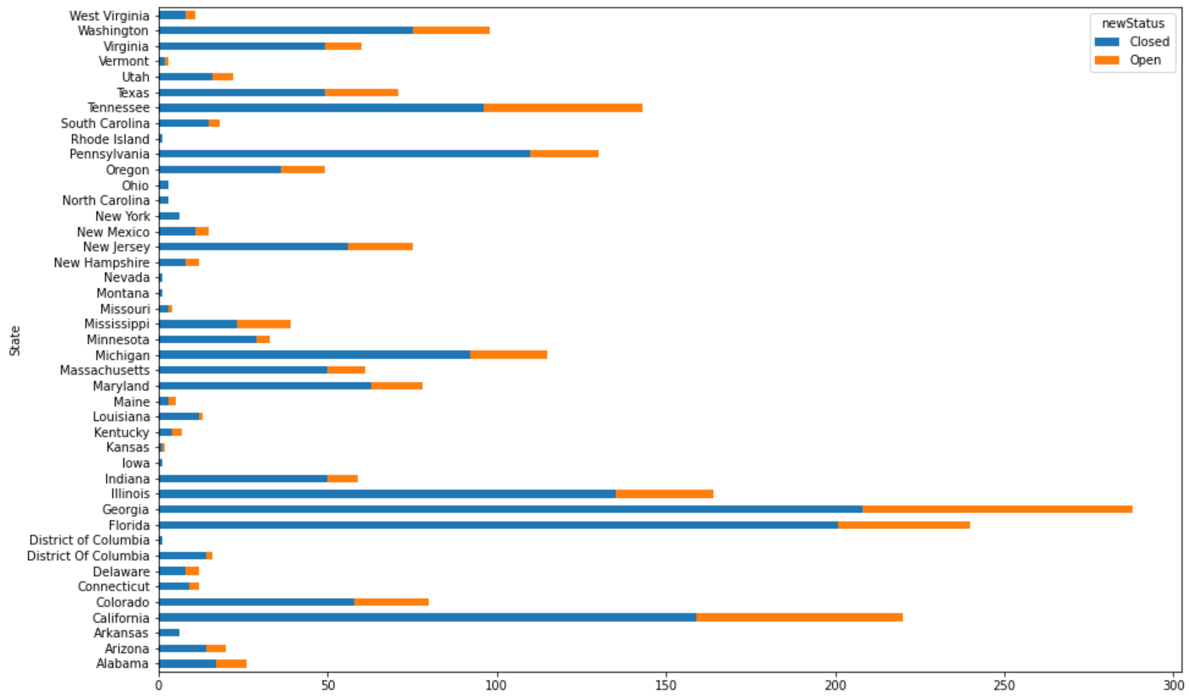
	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
0	complaint	comcast	comcast	bill	speed	service	comcast	internet	comcast
1	comcast	internet	monopoly	comcast	internet	comcast	billing	service	price
2	service	service	switch	comcastxfinity	slow	internet	data	day	internet
3	paying	charge	false	charged	lack	customer	cap	poor	bill
4	connection	throttling	home	promised	comcast	cable	issue	deceptive	without
5	terrible	xfinity	advertising	high	bandwidth	charge	practice	pay	month
6	unreliable	problem	contract	back	help	bill	unfair	sale	2
7	slowing	pricing	email	account	rate	overage	usage	outage	show
8	failure	fraudulent	bait	installation	provider	cramming	xfinity	connectivity	get
9	access	business	availability	without	throttle	fee	monopolistic	several	intermittent
10	shitty	charging	xfinity	option	extremely	monthly	limit	issue	broadband
11	provide	refund	scam	fee	isp	poor	pricing	time	service
12	lied	issue	system	phone	download	increased	12	payment	said
13	provided	equipment	service	throttled	lying	3	service	practice	people
14	internet	mb	security	loss	xfinitycomcast	bad	modem	10	overcharge
15	still	low	improper	one	way	horrible	refusal	disconnection	information
16	getting	fee	communication	hbo	please	year	plan	xfinity	credit
17	every	quality	misleading	ps4	contract	call	incorrect	incorrect	notice
18	cost	regarding	fee	go	upload	unauthorized	atlanta	trade	false
19	install	awful	device	terrible	wacko	contract	returned	fix	appointment

Creating a new categorical variable with value as Open and Closed.

```
In [48]: comcast_df['newStatus'] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" for Status in comcast_df['Status']]
         comcast_df['newStatus'].unique()

Out[48]: array(['Closed', 'Open'], dtype=object)
```

Stacked bar chart representing status of complaints in states

```
In [49]: complaints_type = comcast_df.groupby(["State","newStatus"]).size().unstack().fillna(0)
         complaints_type.sort_values('Closed',axis = 0,ascending=False)
         complaints_type.plot(kind='barh',stacked=True,figsize=(15,10))

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7fda2c3ae210>
```



Statewise Status of Complaints

In [50]:
```
complaints_by_state = comcast_df.groupby(['State']).size().sort_values(ascending=False).to_frame().rename({0:'No of complaints'},axis=1)
complaints_by_state[:5]
```

Out[50]:

State	No of complaints
Georgia	288
Florida	240
California	220
Illinois	164
Tennessee	143

Insight: From above table we can conclude **georgia** has maximum number of complaints

State with highest percentage of unresolved complaints

In [51]:
```
complaints_type['Unres_complaints_percent'] = complaints_type['Open']/complaints_type['Open'].sum()*100
```

In [52]:
```
complaints_type['Unres_complaints_percent'].sort_values(ascending=False)[:1]
```

Out[52]:
```
State
Georgia    15.473888
Name: Unres_complaints_percent, dtype: float64
```

Georgia is the state with highest percentage of unresolved complaints

State with highest percentage of resolved complaints

In [53]:
```
complaints_type['Res_complaints_percent'] = complaints_type['Closed']/complaints_type['Closed'].sum()*100
complaints_type['Res_complaints_percent'].sort_values(ascending=False)[:1]
```

Out[53]:
```
State
Georgia    12.18512
Name: Res_complaints_percent, dtype: float64
```

Georgia is the state with highest percentage of resolved complaints