In [1]:
```
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# read using csv

df  = pd.read_csv('Comcast_telecom_complaints_data.csv')
```

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

Out[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

In [4]:
```
df.dtypes
```

Out[4]:
```
Ticket #                    object
Customer Complaint          object
Date                        object
Date_month_year             object
Time                        object
Received Via                object
City                        object
State                       object
Zip code                     int64
Status                      object
Filing on Behalf of Someone    object
dtype: object
```

In [5]:
```
# changing the columns into lower case

df.columns = df.columns.str.lower()
```

In [6]:
```
# checking for NaN values

df.isnull().sum()
```

Out[6]:
```
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 [7]:
```
# No. of unique values in df
df.nunique()
```

Out[7]:
```
ticket #                    2224
customer complaint          1841
date                          91
date_month_year               91
time                        2190
received via                   2
city                         928
state                         43
zip code                    1543
status                         4
filing on behalf of someone    2
dtype: int64
```

In [8]:
```
# renaming the columns name

df.rename(columns={'ticket #':'ticket'},inplace = True)
```

Changing column date_month_year from object to datetime dtype

In [9]:
```
# converting the variable 'date_time_year' to date_time format

dmy = pd.to_datetime(df['date_month_year'])
```

In [10]:
```
# adding the variable to the main dataset

df['dmy']  = dmy
```

In [11]:
```
# extracting the month from the 'dmy ' variable

df['month'] = df['dmy'].dt.month
```

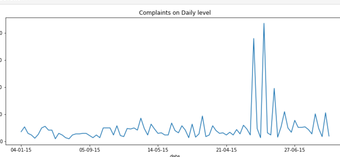
In [12]:
```
# renaming the column - customer complaint with '_'

df.rename(columns={'customer complaint':'customer_complaint'},inplace=True)
```

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

In [13]:
```
df.groupby('month').size().plot()
plt.title('Monthly complaints')
plt.show()
```

In [14]:
```
plt.figure(figsize=(13,5))
x = df.groupby('date').size()
x.plot()
plt.title('Complaints on Daily level')
plt.show()
```



Provide a table with the frequency of complaint types.

In [15]:
```
frequency = df.customer_complaint.value_counts().to_frame().reset_index().rename(columns={'index':'complaint
frequency
```

Out[15]:

	complaint type	complaint_count
0	Comcast	83
1	Comcast Internet	18
2	Comcast Data Cap	17
3	comcast	13
4	Data Caps	11
...	...	...
1836	Internet speeds not as advertised. Bandwith no...	1
1837	Comcast over billing	1
1838	Throttling bandwidth.	1
1839	Comcast Internet prices & speeds	1
1840	Comcast knowingly over billed	1

1841 rows × 2 columns

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

In [16]:
```
frequency.sort_values(by='complaint_count',ascending=False).[:5]
```

Out[16]:

	complaint type	complaint_count
0	Comcast	83
1	Comcast Internet	18
2	Comcast Data Cap	17
3	comcast	13
4	Data Caps	11

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.

In [17]:
```
check = ['Open' if status=='Open' or status=='Pending' else 'Closed' for status in df['status']]
```

In [18]:
```
df['new_status'] = check
```

In [19]:
```
df.head()
```

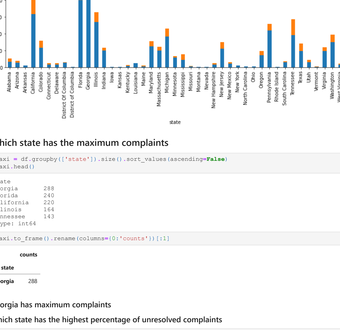
Out[19]:

	ticket	customer_complaint	date	date_month_year	time	received via	city	state	zip code	status	filing on behalf of someone	dmy	month	n
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No	2015-04-22	4	
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No	2015-08-04	8	
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes	2015-04-18	4	
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	2015-07-05	7	
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	2015-05-26	5	

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

In [20]:
```
# plt.figure(figsize=(13,4))

df.groupby(['state','new_status']).size().unstack().plot(kind = 'bar',stacked=True,figsize=(13,4))
plt.show()
```



Which state has the maximum complaints

In [21]:
```
maxi = df.groupby(['state']).size().sort_values(ascending=False)
maxi.head()
```

Out[21]:
```
state
Georgia       288
Florida       240
California    220
Illinois      164
Tennessee     143
dtype: int64
```

In [22]:
```
maxi.to_frame().rename(columns={0:'counts'})[:1]
```

Out[22]:

	counts
state	
Georgia	288

Georgia has maximum complaints

Which state has the highest percentage of unresolved complaints

In [23]:
```
opens = df.groupby(['new_status','state']).size().unstack()
opens = opens.T
opens.fillna(0,inplace=True)
```

In [24]:
```
tot_len = (opens['Open'])/(opens['Closed']+opens['Open'])*100
```

In [25]:
```
tot_len = tot_len.sort_values(ascending=False).to_frame().head()
```

In [26]:
```
tot_len = tot_len.reset_index().rename(columns= {0:'percentage'})
```

In [27]:
```
tot_len[:1]
```

Out[27]:

	state	percentage
0	Kansas	50.0

Kansas has the highest percentage of Unresolved complaint

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

In [28]:
```
final_task = df.groupby(['received via','new_status']).size().unstack()
final_task
```

Out[28]:

new_status	Closed	Open
received via		
Customer Care Call	864	255
Internet	843	262

In [29]:
```
((final_task['Closed']) / (final_task['Closed']+final_task['Open']) *100).to_frame().rename(columns={0:'Solv
```

Out[29]:

	Solved percentage
received via	
Customer Care Call	77.211796
Internet	76.289593