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

In [2]:
```
df = pd.read_csv('Comcast_telecom_complaints_data.csv')
df.head()
```

Out[2]:

	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 [3]:
```
df.shape
```

Out[3]:
```
(2224, 11)
```

In [4]:
```
df.isnull().sum()
```

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]:
```
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 [6]:
```
df['Date_month_year'] = pd.to_datetime(df['Date_month_year'])
```

In [7]:
```
df.dtypes
```

Out[7]:
```
Ticket #                             object
Customer Complaint                   object
Date                                 object
Date_month_year              datetime64[ns]
Time                                 object
Received Via                         object
City                                 object
State                                object
Zip code                              int64
Status                               object
Filing on Behalf of Someone          object
dtype: object
```

In [8]:
```
# Create day,month and day name from the dataframe
df['create_day'] = df['Date_month_year'].dt.day
df['create_month'] = df['Date_month_year'].dt.month
df['create_day of week'] = df['Date_month_year'].dt.day_name()
```

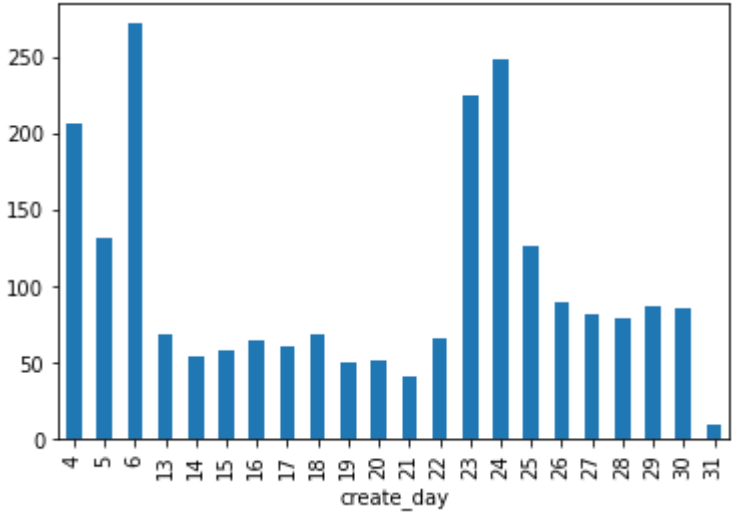
In [9]:
```
df.head()
```

Out[9]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	create_day	create_month	create_day of week
0	250635	Comcast Cable Internet Speeds	22-04-15	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No	22	4	Wednesday
1	223441	Payment disappear - service got disconnected	04-08-15	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No	4	8	Tuesday
2	242732	Speed and Service	18-04-15	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes	18	4	Saturday
3	277946	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	2015-07-05	11:59:35 AM	Internet	Acworth	Georgia	30101	Open	Yes	5	7	Sunday
4	307175	Comcast not working and no service to boot	26-05-15	2015-05-26	1:25:26 PM	Internet	Acworth	Georgia	30101	Solved	No	26	5	Tuesday

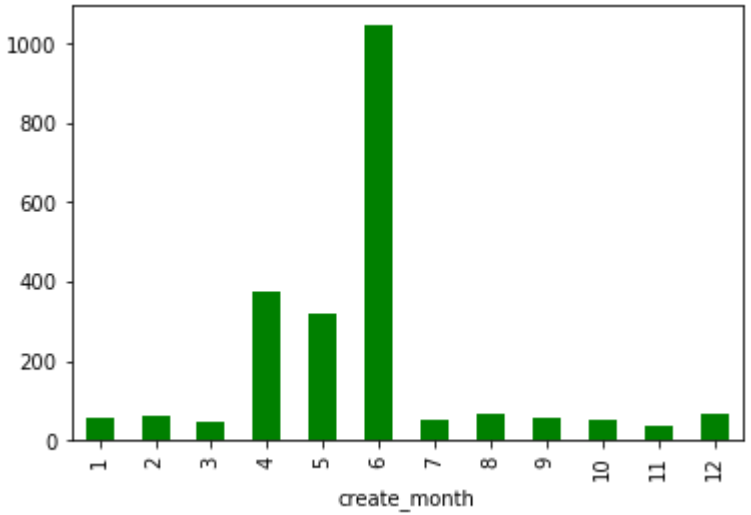
In [10]:
```
# date wise visuals
df.groupby(['create_day'])['Customer Complaint'].count().plot(kind='bar')
```

Out[10]:
```
<AxesSubplot:xlabel='create_day'>
```



In [11]:
```
# month wise visuals
df.groupby(['create_month'])['Customer Complaint'].count().plot(kind='bar',color='g')
```

Out[11]:
```
<AxesSubplot:xlabel='create_month'>
```



In [12]:
```
# Frequency count for unique complaint type
df['Customer Complaint'].value_counts()
```

Out[12]:
```
Comcast                                          83
Comcast Internet                                 18
Comcast Data Cap                                 17
comcast                                          13
Comcast Billing                                  11
                                                 ..
Improper Billing and non resolution of issues     1
Deceptive trade                                   1
intermittent internet                             1
Internet Speed on Wireless Connection             1
Comcast, Ypsilanti MI Internet Speed              1
Name: Customer Complaint, Length: 1841, dtype: int64
```

In [13]:
```
df['Customer Complaint'].unique
```

Out[13]:
```
<bound method Series.unique of 0                                Comcast Cable Internet Speeds
1            Payment disappear - service got disconnected
2                                         Speed and Service
3       Comcast Imposed a New Usage Cap of 300GB that ...
4              Comcast not working and no service to boot
                              ...
2219                                  Service Availability
2220           Comcast Monthly Billing for Returned Modem
2221                                complaint about comcast
2222               Extremely unsatisfied Comcast customer
2223                 Comcast, Ypsilanti MI Internet Speed
Name: Customer Complaint, Length: 2224, dtype: object>
```

In [14]:
```
# Creating table for the frequency count for each unique complaint type
df['Customer Complaint'].value_counts().to_frame().reset_index()
```

Out[14]:

	index	Customer Complaint
0	Comcast	83
1	Comcast Internet	18
2	Comcast Data Cap	17
3	comcast	13
4	Comcast Billing	11
...	...	...
1836	Improper Billing and non resolution of issues	1
1837	Deceptive trade	1
1838	intermittent internet	1
1839	Internet Speed on Wireless Connection	1
1840	Comcast, Ypsilanti MI Internet Speed	1

1841 rows × 2 columns

In [15]:
```
df['Status'].value_counts()
```

Out[15]:
```
Solved     973
Closed     734
Open       363
Pending    154
Name: Status, dtype: int64
```

In [16]:
```
# Convert all the values in the lower case so that we won't miss any values to stored in dataframe.
df['Status'] = df['Status'].str.lower()
```

In [17]:
```
df['Status'].head()
```

Out[17]:
```
0    closed
1    closed
2    closed
3      open
4    solved
Name: Status, dtype: object
```

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 [18]:
```
df['new_status'] = df['Status'].apply(lambda x:'Open' if((x=='open')|(x=='pending')) else'Closed')
```

In [19]:
```
df['new_status'].value_counts()
```

Out[19]:
```
Closed    1707
Open       517
Name: new_status, dtype: int64
```

State wise status of complaints in a stacked bar chart. using categorized variable from Q3.

In [20]:
```
state_open = df[df['new_status']=='Open'].groupby(['State'])['new_status'].count().to_frame().reset_index()
state_close = df[df['new_status']=='Closed'].groupby(['State'])['new_status'].count().to_frame().reset_index()
```

In [21]:
```
state_open
```

Out[21]:

	State	new_status
0	Alabama	9
1	Arizona	6
2	California	61
3	Colorado	22
4	Connecticut	3
5	Delaware	4
6	District Of Columbia	2
7	Florida	39
8	Georgia	80
9	Illinois	29
10	Indiana	9
11	Kansas	1
12	Kentucky	3
13	Louisiana	1
14	Maine	2
15	Maryland	15
16	Massachusetts	11
17	Michigan	23
18	Minnesota	4
19	Mississippi	16
20	Missouri	1
21	New Hampshire	4
22	New Jersey	19
23	New Mexico	4
24	Oregon	13
25	Pennsylvania	20
26	South Carolina	3

	State	new_status
27	Tennessee	47
28	Texas	22
29	Utah	6
30	Vermont	1
31	Virginia	11
32	Washington	23
33	West Virginia	3

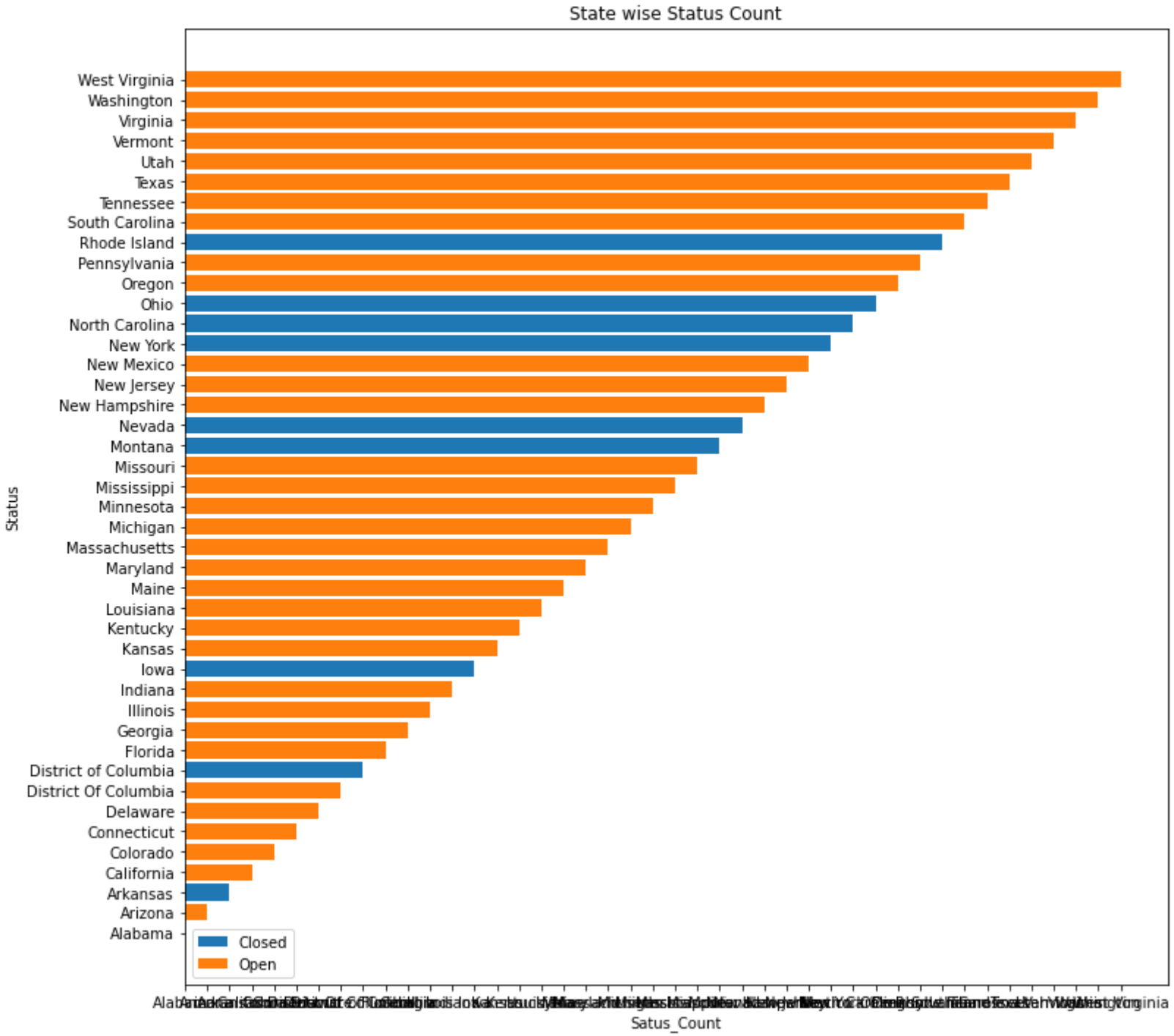
In [22]:
```
state_close
```

Out[22]:

	State	new_status
0	Alabama	17
1	Arizona	14
2	Arkansas	6
3	California	159
4	Colorado	58
5	Connecticut	9
6	Delaware	8
7	District Of Columbia	14
8	District of Columbia	1
9	Florida	201
10	Georgia	208
11	Illinois	135
12	Indiana	50
13	Iowa	1
14	Kansas	1
15	Kentucky	4
16	Louisiana	12
17	Maine	3
18	Maryland	63
19	Massachusetts	50
20	Michigan	92
21	Minnesota	29
22	Mississippi	23
23	Missouri	3
24	Montana	1
25	Nevada	1
26	New Hampshire	8

	State	new_status
27	New Jersey	56
28	New Mexico	11
29	New York	6
30	North Carolina	3
31	Ohio	3
32	Oregon	36
33	Pennsylvania	110
34	Rhode Island	1
35	South Carolina	15
36	Tennessee	96
37	Texas	49
38	Utah	16
39	Vermont	2
40	Virginia	49
41	Washington	75
42	West Virginia	8

In [23]:
```
fig = plt.figure(figsize=(12,12))
plt.barh(state_close.State,state_close.State)
plt.barh(state_open.State,state_open.State)
plt.ylabel('Status')
plt.xlabel('Satus_Count')
plt.legend(['Closed','Open'])
plt.title("State wise Status Count")
plt.show()
```



state has the maximum complaints

In [24]:
```
df.groupby(['State']).size().sort_values(ascending=False).to_frame().reset_index().rename({0:' Statewise_complaint'},axis=1)
```

Out[24]:

	State	Statewise_complaint
0	Georgia	288
1	Florida	240
2	California	220
3	Illinois	164
4	Tennessee	143
5	Pennsylvania	130
6	Michigan	115
7	Washington	98
8	Colorado	80
9	Maryland	78
10	New Jersey	75
11	Texas	71
12	Massachusetts	61
13	Virginia	60
14	Indiana	59
15	Oregon	49
16	Mississippi	39
17	Minnesota	33
18	Alabama	26
19	Utah	22
20	Arizona	20
21	South Carolina	18
22	District Of Columbia	16
23	New Mexico	15
24	Louisiana	13
25	Connecticut	12
26	New Hampshire	12

	State	Statewise_complaint
27	Delaware	12
28	West Virginia	11
29	Kentucky	7
30	Arkansas	6
31	New York	6
32	Maine	5
33	Missouri	4
34	North Carolina	3
35	Vermont	3
36	Ohio	3
37	Kansas	2
38	District of Columbia	1
39	Rhode Island	1
40	Iowa	1
41	Nevada	1
42	Montana	1

state having the highest percentage of unresolved complaints

In [25]:
```
open_complaint = df.loc[df['new_status']=='Open',['State']].value_counts()
open_complaint
```

Out[25]:
```
State
Georgia                 80
California              61
Tennessee               47
Florida                 39
Illinois                29
Michigan                23
Washington              23
Colorado                22
Texas                   22
Pennsylvania            20
New Jersey              19
Mississippi             16
Maryland                15
Oregon                  13
Massachusetts           11
Virginia                11
Alabama                  9
Indiana                  9
Arizona                  6
Utah                     6
Delaware                 4
New Hampshire            4
New Mexico               4
Minnesota                4
South Carolina           3
Connecticut              3
West Virginia            3
Kentucky                 3
District Of Columbia     2
Maine                    2
Louisiana                1
Vermont                  1
Missouri                 1
Kansas                   1
dtype: int64
```

In the above output it is clear that Georgia state has highest unresolved complaint.

In [26]:
```
# Finding the percentage of the georgia state
complain_percent = open_complaint.head(1)/open_complaint.sum()*100
complain_percent
```

Out[26]:
```
State
Georgia    15.473888
dtype: float64
```

In [27]:
```
 # minimize the floating values using round function
round(complain_percent,2)
```

Out[27]:
```
State
Georgia    15.47
dtype: float64
```

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

In [28]:
```
solved_complain = df[df['new_status']=="Closed"].groupby('new_status')['Received Via'].value_counts()
```

In [29]:
```
solved_complain/solved_complain.sum()*100
```

Out[29]:
```
new_status  Received Via
Closed      Customer Care Call    50.615114
            Internet              49.384886
Name: Received Via, dtype: float64
```

In [30]:
```
# finding percentage using normalise
solved_percent = df[df['new_status']=="Closed"].groupby('new_status')['Received Via'].value_counts(normalize=True)*100
# With normalize set to True, returns the relative frequency by dividing all values by the sum of values.
solved_percent
```

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

In [31]:
```
df['Received Via'].value_counts()
```

Out[31]:
```
Customer Care Call    1119
Internet              1105
Name: Received Via, dtype: int64
```

In [32]:
```
1119/(1119+1105)*100
```

Out[32]:
```
50.31474820143885
```

In [33]:
```
1105/(1119+1105)*100
```

Out[33]:
```
49.685251798561154
```

In [ ]: