**Import data into Python environment.**

**import** pandas **as** pd

df**=**pd.read\_csv(r"C:\Users\Libin\Desktop\Comcast\_telecom\_complaints\_data.csv")

df.head()

|  | **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 |

Identify the data types in the dataset

df.dtypes

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

Rename the column and verifying it

df**=**df.rename(columns**=**{'Ticket #':'Ticket Number'})

df.head()

|  | **Ticket Number** | **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 |

**Change the data type to datetime from Object for Date Column**

df['Date']**=**df['Date'].astype('datetime64[ns]')

df.dtypes

Ticket Number object

Customer Complaint object

Date datetime64[ns]

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

**Creating a new Dataframe with wanted columns for better analysis**

wanted\_columns**=**df[['Ticket Number','Date','Received Via','State','Status']]

​

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

Provide a table with the frequency of complaint types.

To get better insight in data, Using matplotlib to plot the data

wanted\_columns.head()

**import** matplotlib.pyplot **as** plt

**%**matplotlib inline

df1 **=**wanted\_columns.groupby(wanted\_columns['Date']).size().reset\_index(name**=**'Count')

df1**=**df1.set\_index('Date')

df1['Year'] **=** df1.index.year

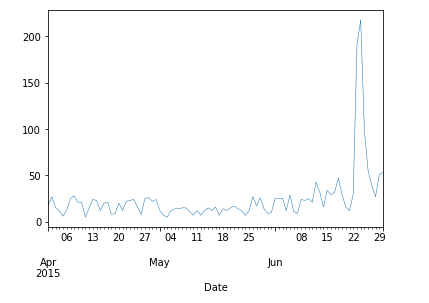
df1['Month'] **=** df1.index.month

df1['Weekday Name'] **=** df1.index.weekday\_name

df1['day']**=**df1.index.day

df1['Count'].plot(linewidth**=**0.5);

​



dif\_mnth**=**df1.groupby(df1['Month']).size().reset\_index(name**=**'Count')

​

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

​

Labels**=**['April','May','June']

size**=**dif\_mnth['Count']

plt.pie(size,labels**=**Labels,autopct**=**'%1.1f%%')

([<matplotlib.patches.Wedge at 0x1d2f36f6f48>,

<matplotlib.patches.Wedge at 0x1d2f3702748>,

<matplotlib.patches.Wedge at 0x1d2f3702fc8>],

[Text(0.56092583464912, 0.9462358099454851, 'April'),

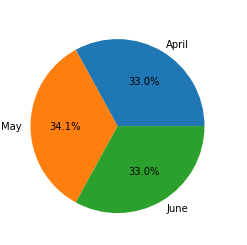
Text(-1.0999999999999954, -1.0298943258065002e-07, 'May'),

Text(0.5609260118350907, -0.9462357049101349, 'June')],

[Text(0.3059595461722473, 0.5161286236066281, '33.0%'),

Text(-0.5999999999999974, -5.6176054134900006e-08, '34.1%'),

Text(0.3059596428191404, -0.5161285663146189, '33.0%')])

​ 

*#- 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.*

​

wanted\_columns['Cstatus'] **=** ['Open' **if** x **==**'Pending' **or** x**==**'Open' **else** 'Closed' **for** x **in** wanted\_columns['Status']]

wanted\_columns.tail()

|  | **Ticket Number** | **Date** | **Received Via** | **State** | **Status** | **Cstatus** |
| --- | --- | --- | --- | --- | --- | --- |
| 2219 | 213550 | 2015-04-02 | Customer Care Call | Florida | Closed | Closed |
| 2220 | 318775 | 2015-06-02 | Customer Care Call | Michigan | Solved | Closed |
| 2221 | 331188 | 2015-06-09 | Internet | Michigan | Solved | Closed |
| 2222 | 360489 | 2015-06-23 | Customer Care Call | Michigan | Solved | Closed |
| 2223 | 363614 | 2015-06-24 | Customer Care Call | Michigan | Open | Open |

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

​

dfx **=** wanted\_columns.groupby(['State', 'Cstatus'])['State'].count().reset\_index(name**=**'Value')

dfx.head()

|  | **State** | **Cstatus** | **Value** |
| --- | --- | --- | --- |
| 0 | Alabama | Closed | 17 |
| 1 | Alabama | Open | 9 |
| 2 | Arizona | Closed | 14 |
| 3 | Arizona | Open | 6 |
| 4 | Arkansas | Closed | 6 |

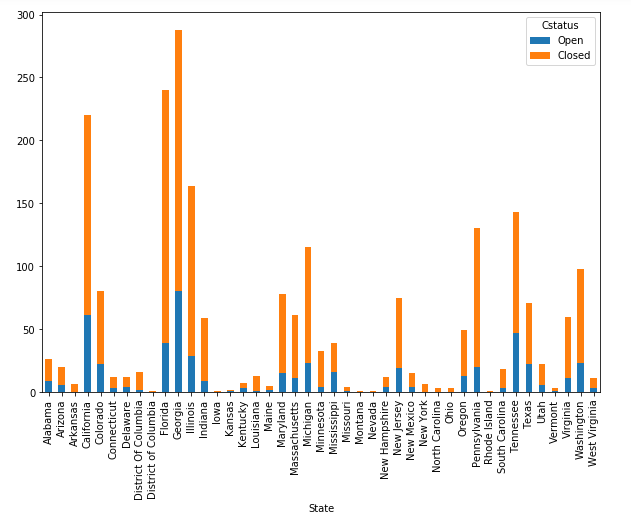
pivot\_df **=** dfx.pivot(index**=**'State', columns**=**'Cstatus', values**=**'Value')

​

pivot\_df.loc[:,['Open', 'Closed']].plot.bar(stacked**=True**, figsize**=**(10,7))

​

<matplotlib.axes.\_subplots.AxesSubplot at 0x1d2f3710f48>



**Which state has the maximum complaints**

open**=**dfx[dfx['Cstatus']**==**'Open']

closed**=**dfx[dfx['Cstatus']**==**'Closed']

open**=**open.rename(columns**=**{'Value':'Count of Open'})

closed**=**closed.rename(columns**=**{'Value':'Count of Closed'})

total**=**pd.merge(open,closed,on**=**'State')

total['total\_count']**=**total['Count of Open']**+**total['Count of Closed']

toatl**=**total.sort\_values(by**=**'total\_count',ascending**=False**)

np.max(total['total\_count'])

288

total.head()

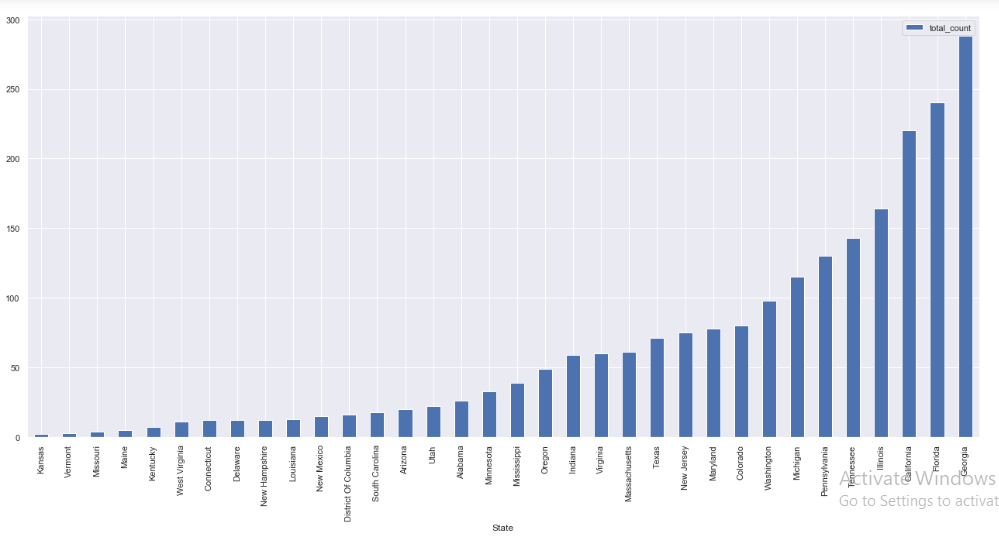
|  | **State** | **Cstatus\_x** | **Count of Open** | **Cstatus\_y** | **Count of Closed** | **total\_count** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | Alabama | Open | 9 | Closed | 17 | 26 |
| 1 | Arizona | Open | 6 | Closed | 14 | 20 |
| 2 | California | Open | 61 | Closed | 159 | 220 |
| 3 | Colorado | Open | 22 | Closed | 58 | 80 |
| 4 | Connecticut | Open | 3 | Closed | 9 | 12 |

total**=**total.drop('Cstatus\_y',axis**=**1)

total**=**total.sort\_values(by**=**'total\_count')

total.plot(kind**=**'bar',x**=**'State',y**=**'total\_count')

<matplotlib.axes.\_subplots.AxesSubplot at 0x1d2f6ee0b88>



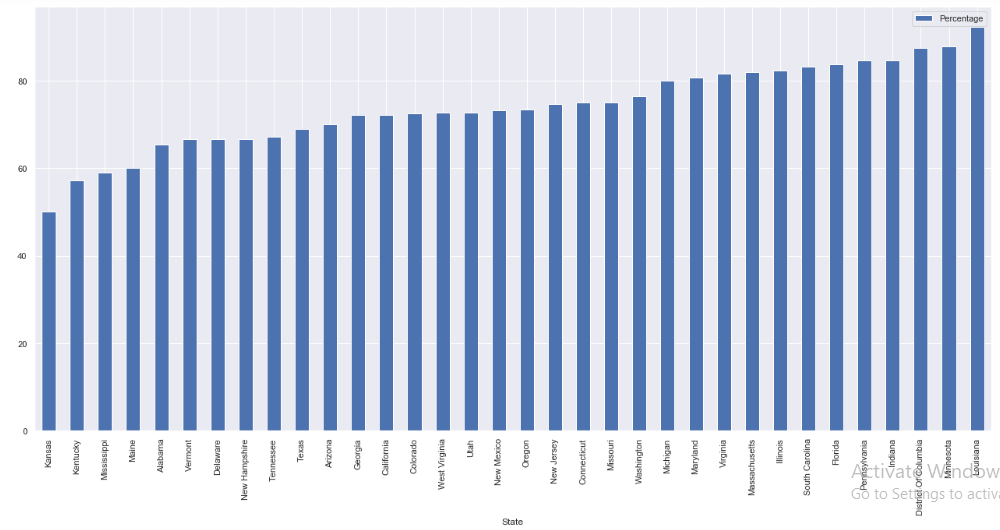
**From the graph we can understand Georgia has maximum complaints**

**Which state has the highest percentage of unresolved complaints**

total['Percentage']**=**(total['Count of Closed']**/**total['total\_count'])**\***100

total.sort\_values(by**=**'Percentage').plot(kind**=**'bar',x**=**'State',y**=**'Percentage')

<matplotlib.axes.\_subplots.AxesSubplot at 0x1d2f6610188>



**From the graph we can understand Louisiana has** **highest percentage of unresolved complaints**

wanted\_columns**=**wanted\_columns.drop('Status',axis**=**1)

wanted\_columns**=**wanted\_columns.drop('Date',axis**=**1)

wanted\_columns.head()

|  | **Ticket Number** | **Received Via** | **State** | **Cstatus** |
| --- | --- | --- | --- | --- |
| 0 | 250635 | Customer Care Call | Maryland | Closed |
| 1 | 223441 | Internet | Georgia | Closed |
| 2 | 242732 | Internet | Georgia | Closed |
| 3 | 277946 | Internet | Georgia | Open |
| 4 | 307175 | Internet | Georgia | Closed |

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

ticket\_unre**=** wanted\_columns.groupby(['Received Via', 'Cstatus'])['State'].count().reset\_index(name**=**'Value')

ticket\_unre

|  | **Received Via** | **Cstatus** | **Value** |
| --- | --- | --- | --- |
| 0 | Customer Care Call | Closed | 864 |
| 1 | Customer Care Call | Open | 255 |
| 2 | Internet | Closed | 843 |
| 3 | Internet | Open | 262 |

ticket\_unre\_closed**=**ticket\_unre[ticket\_unre['Cstatus']**==**'Closed']

ticket\_unre\_open**=**ticket\_unre[ticket\_unre['Cstatus']**==**'Open']

Un\_resolved**=**pd.merge(ticket\_unre\_closed,ticket\_unre\_open,on**=**'Received Via')

Un\_resolved.head()

|  | **Received Via** | **Cstatus\_x** | **Value\_x** | **Cstatus\_y** | **Value\_y** |
| --- | --- | --- | --- | --- | --- |
| 0 | Customer Care Call | Closed | 864 | Open | 255 |
| 1 | Internet | Closed | 843 | Open | 262 |

Un\_resolved**=**Un\_resolved.rename(columns**=**{'Value\_x':'Closed\_Count','Value\_y':'Open\_count'})

Un\_resolved**=**Un\_resolved.drop('Cstatus\_x',axis**=**1)

Un\_resolved**=**Un\_resolved.drop('Cstatus\_y',axis**=**1)

Un\_resolved.head()

|  | **Received Via** | **Closed\_Count** | **Open\_count** |
| --- | --- | --- | --- |
| 0 | Customer Care Call | 864 | 255 |
| 1 | Internet | 843 | 262 |

Un\_resolved['Total\_count']**=**Un\_resolved['Closed\_Count']**+**Un\_resolved['Open\_count']

Un\_resolved.head()

|  | **Received Via** | **Closed\_Count** | **Open\_count** | **Total\_count** |
| --- | --- | --- | --- | --- |
| 0 | Customer Care Call | 864 | 255 | 1119 |
| 1 | Internet | 843 | 262 | 1105 |

Un\_resolved['Percentage\_unresolved']**=**(Un\_resolved['Open\_count']**/**Un\_resolved['Total\_count'])**\***100

Un\_resolved.head()

:

|  | **Received Via** | **Closed\_Count** | **Open\_count** | **Total\_count** | **Percentage\_unresolved** |
| --- | --- | --- | --- | --- | --- |
| 0 | Customer Care Call | 864 | 255 | 1119 | 22.788204 |
| 1 | Internet | 843 | 262 | 1105 | 23.710407 |

plt.pie(Un\_resolved['Percentage\_unresolved'],labels**=**Un\_resolved['Received Via'],autopct**=**'%1.1f%%')

Out[295]:

([<matplotlib.patches.Wedge at 0x1d2f6629308>,

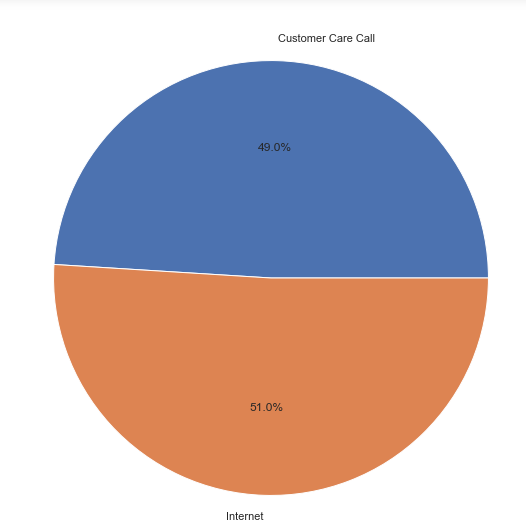
<matplotlib.patches.Wedge at 0x1d2f6629f48>],

[Text(0.034263263924554005, 1.0994662472059962, 'Customer Care Call'),

Text(-0.034263160985094576, -1.0994662504139496, 'Internet')],

[Text(0.018689053049756727, 0.5997088621123614, '49.0%'),

Text(-0.018688996900960674, -0.5997088638621543, '51.0%')])



**import** seaborn **as** sns

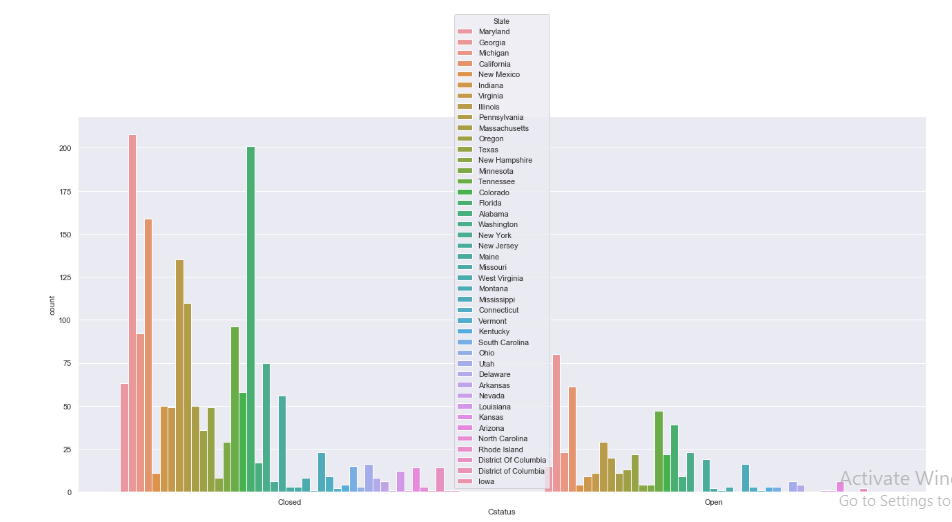
wanted\_columns

|  | **Ticket Number** | **Received Via** | **State** | **Cstatus** |
| --- | --- | --- | --- | --- |
| 0 | 250635 | Customer Care Call | Maryland | Closed |
| 1 | 223441 | Internet | Georgia | Closed |
| 2 | 242732 | Internet | Georgia | Closed |
| 3 | 277946 | Internet | Georgia | Open |
| 4 | 307175 | Internet | Georgia | Closed |
| ... | ... | ... | ... | ... |
| 2219 | 213550 | Customer Care Call | Florida | Closed |
| 2220 | 318775 | Customer Care Call | Michigan | Closed |
| 2221 | 331188 | Internet | Michigan | Closed |
| 2222 | 360489 | Customer Care Call | Michigan | Closed |
| 2223 | 363614 | Customer Care Call | Michigan | Open |

2224 rows × 4 columns

sns.countplot(x**=**'Cstatus',hue**=**"State",data**=**wanted\_columns)

<matplotlib.axes.\_subplots.AxesSubplot at 0x1d2f6515a88>



sns.countplot(y**=**'State',hue**=**"Cstatus",data**=**wanted\_columns)

sns.set(rc**=**{'figure.figsize':(22,10)})

sns.countplot(x**=**'Cstatus',hue**=**"Received Via",data**=**wanted\_columns)

<matplotlib.axes.\_subplots.AxesSubplot at 0x1d2f6ff1e88>

