**Project No. 4: Comcast Telecom Consumer Complaints.**

Step1: Import the required libraries import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

Step2: Read the data from

data = pd.read\_csv("/content/sample\_data/Comcast\_telecom\_complaints\_data.csv")

Step3: Print the first five rows in the dataframe

(data.head(5))

Step4: Provide the trend chart for the number of complaints at daily granularity levels.

data['Date\_Index'] = pd.to\_datetime(data['Date\_month\_year'])

data = data.set\_index(data['Date\_Index'])

data['Date\_Index'].value\_counts().head(5)

data['Date\_Index'].value\_counts().plot()

Step5: Provide the trend chart for the number of complaints at monthly data.groupby(pd.Grouper(freq='M')).size()

data.groupby(pd.Grouper(freq='M')).size().plot()

Step6: Provide a table with the frequency of complaint types.

data['Customer Complaint'].value\_counts(dropna=False)[:9]

data['Customer Complaint'].value\_counts(dropna=False)[:9].plot.bar()

Step7: Which complaint types are maximum across any other domains.

data.Status.unique()

data['Current\_Status'] = data['Status'].apply(lambda present\_status: "Open" if present\_status=="Open" or present\_status=="Pending" else "Closed")

state\_wise\_complaints = data.groupby(['State','Current\_Status']).size().unstack().fillna(0.0)

# Plotting a stacked bar chart horizontally

state\_wise\_complaints.plot(kind = 'barh', figsize=(30,40), stacked='true')

Step8: State with maximum complaint

data.groupby(['State']).size().sort\_values(ascending=False).to\_frame().rename({0: "Total number of complaints"}, axis=1).head(1)

state\_wise\_complaints.sort\_values('Closed',axis = 0,ascending=False)

Step9: Which state has the highest percentage of unresolved complaints.

state\_wise\_complaints['Resolved\_complaint\_percentage'] = state\_wise\_complaints['Closed']/state\_wise\_complaints['Closed'].sum()\*100

state\_wise\_complaints['Unresolved\_complaint\_percentage'] = state\_wise\_complaints['Open']/state\_wise\_complaints['Open'].sum()\*100

state\_wise\_complaints.sort\_values('Unresolved\_complaint\_percentage',axis = 0,ascending=False).head(1)

df\_unresolved = data.groupby(['State',"Current\_Status"]).size().unstack().sort\_values(by='Open', ascending =False)

df\_unresolved['Unresolved'] = df\_unresolved['Open']/df\_unresolved['Open'].sum()\*100

df\_unresolved

Step10:Provide the percentage of complaints resolved through the Internet and  customer care calls.

df = data.groupby(['Received Via',"Current\_Status"]).size().unstack()

df['Resolved'] = df['Closed']/df['Closed'].sum()\*100

df

df.plot(kind="bar", figsize=(8,8))

# Screenshot of the Output

## Q1. Import data into Python environment.

Step1: Import the required libraries import pandas as pd

import numpy as np

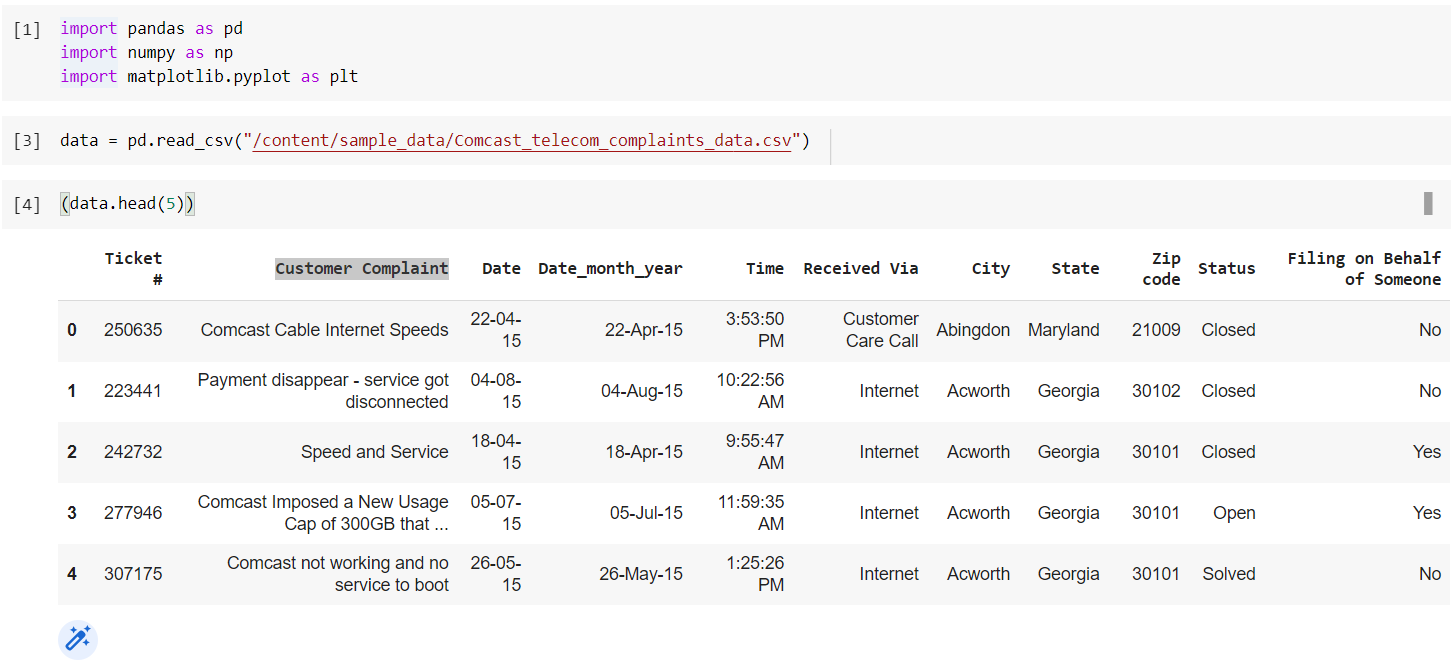
import matplotlib.pyplot as plt

Step2: Read the data from

data = pd.read\_csv("/content/sample\_data/Comcast\_telecom\_complaints\_data.csv")

Step3: Print the first five rows in the dataframe

(data.head(5))



## Q2. Provide the trend chart for the number of complaints at monthly and daily granularity levels.

Step4: Provide the trend chart for the number of complaints at daily granularity levels.

data['Date\_Index'] = pd.to\_datetime(data['Date\_month\_year'])

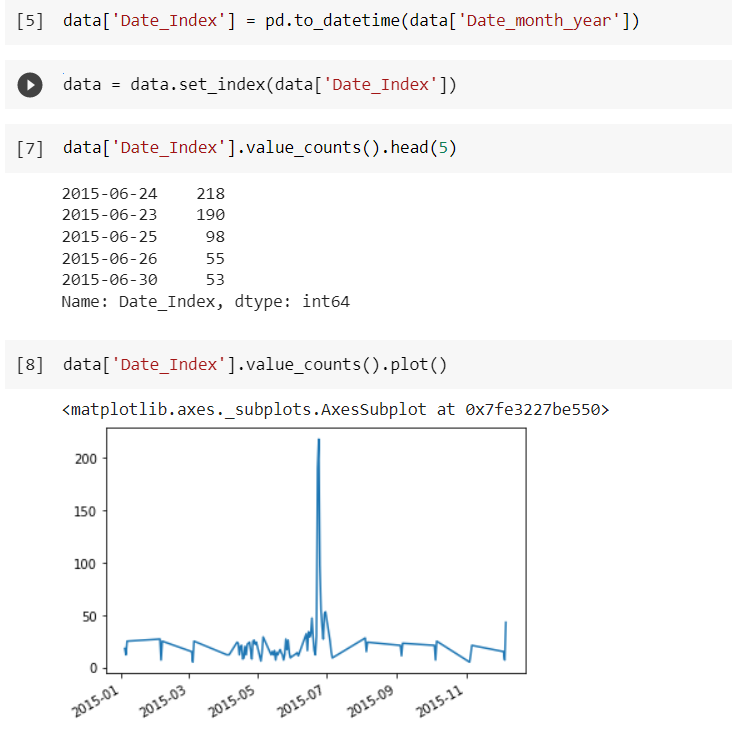
data = data.set\_index(data['Date\_Index'])

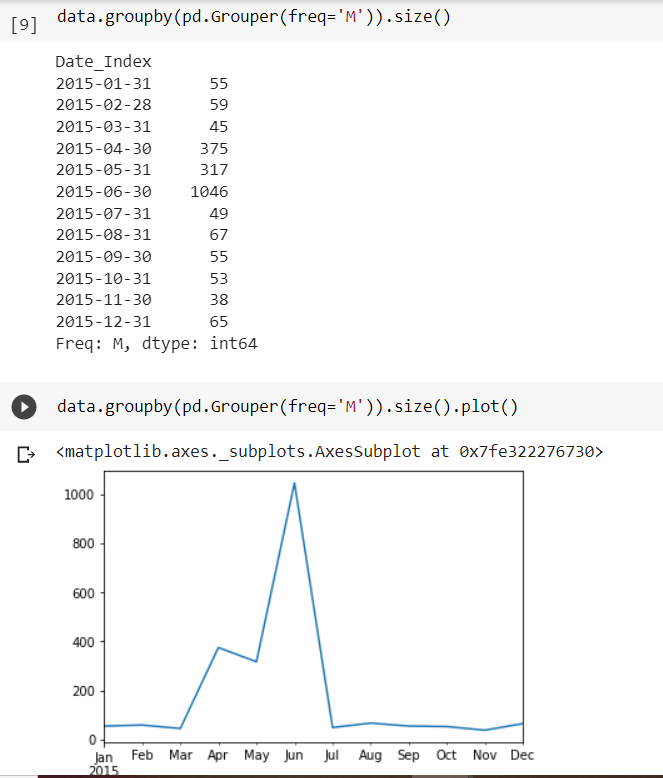
data['Date\_Index'].value\_counts().head(5)

data['Date\_Index'].value\_counts().plot()

Step5: Provide the trend chart for the number of complaints at monthly data.groupby(pd.Grouper(freq='M')).size()

data.groupby(pd.Grouper(freq='M')).size().plot()





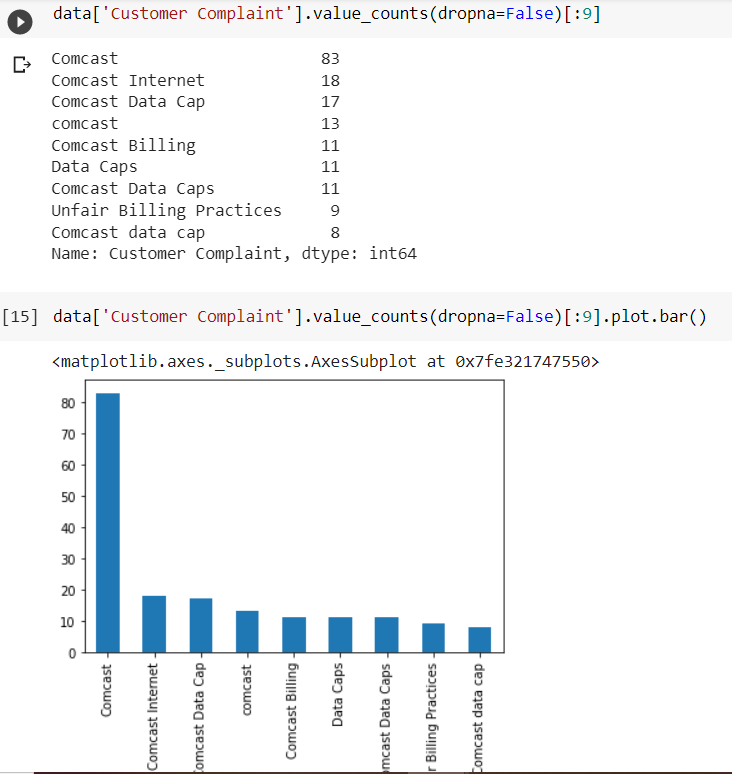
Insights: from the above trend chart, we can clearly see that complaints for the month of June 2015 are maximum.

## Q3. Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

## Step6: Provide a table with the frequency of complaint types.

data['Customer Complaint'].value\_counts(dropna=False)[:9]

data['Customer Complaint'].value\_counts(dropna=False)[:9].plot.bar()



## Q4. - Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3:

Step7: Which complaint types are maximum across any other domains.

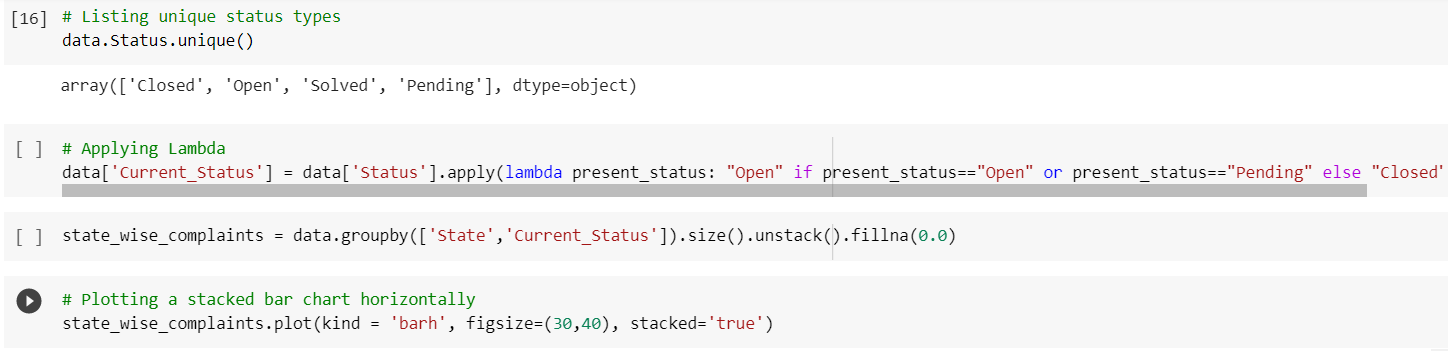
data.Status.unique()

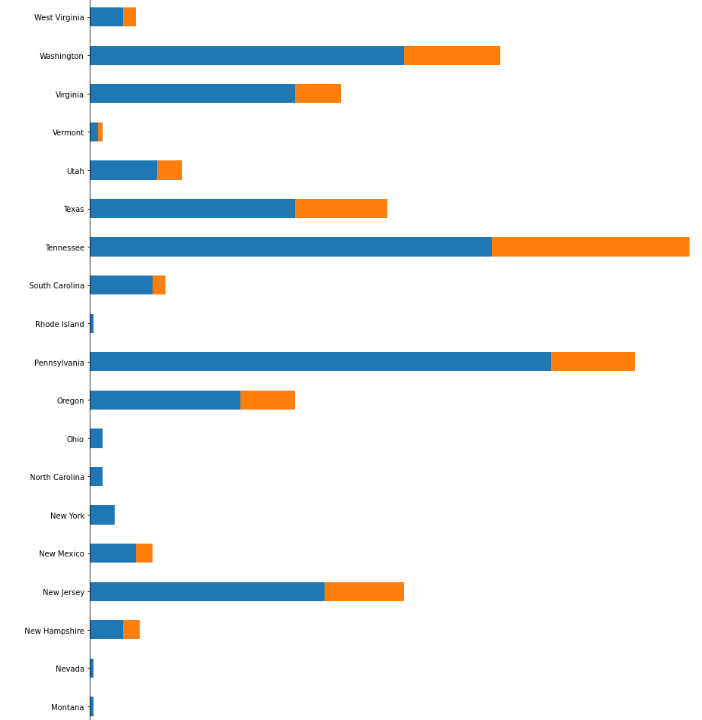
data['Current\_Status'] = data['Status'].apply(lambda present\_status: "Open" if present\_status=="Open" or present\_status=="Pending" else "Closed")

state\_wise\_complaints = data.groupby(['State','Current\_Status']).size().unstack().fillna(0.0)

# Plotting a stacked bar chart horizontally

state\_wise\_complaints.plot(kind = 'barh', figsize=(30,40), stacked='true')



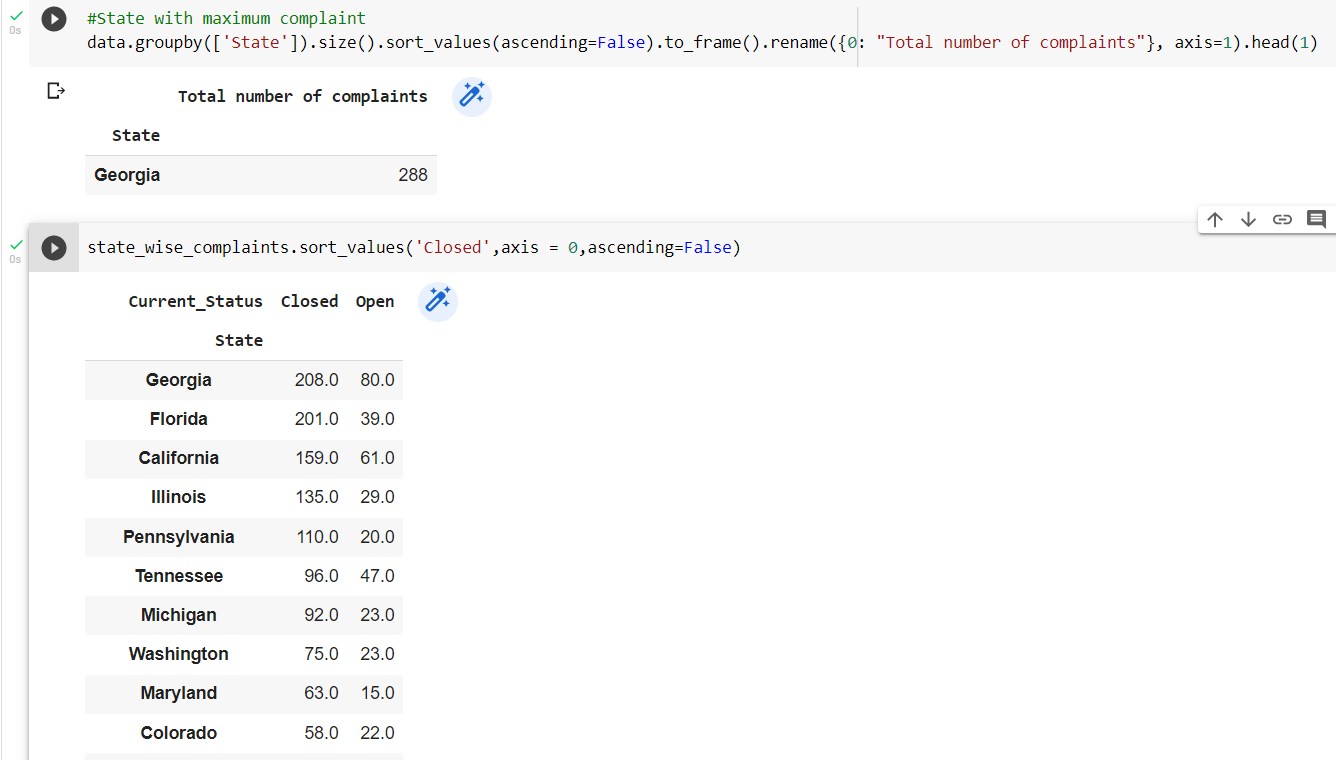


## Q5. Which state has the maximum complaints

Step8: State with maximum complaint

data.groupby(['State']).size().sort\_values(ascending=False).to\_frame().rename({0: "Total number of complaints"}, axis=1).head(1)

state\_wise\_complaints.sort\_values('Closed',axis = 0,ascending=False)



Insights: From the above table we can see Georgia has the maximum complaints

## Q6. Which state has the highest percentage of unresolved complaints.

Step9: state that has the highest percentage of unresolved complaints.

state\_wise\_complaints['Resolved\_complaint\_percentage'] = state\_wise\_complaints['Closed']/state\_wise\_complaints['Closed'].sum()\*100

state\_wise\_complaints['Unresolved\_complaint\_percentage'] = state\_wise\_complaints['Open']/state\_wise\_complaints['Open'].sum()\*100

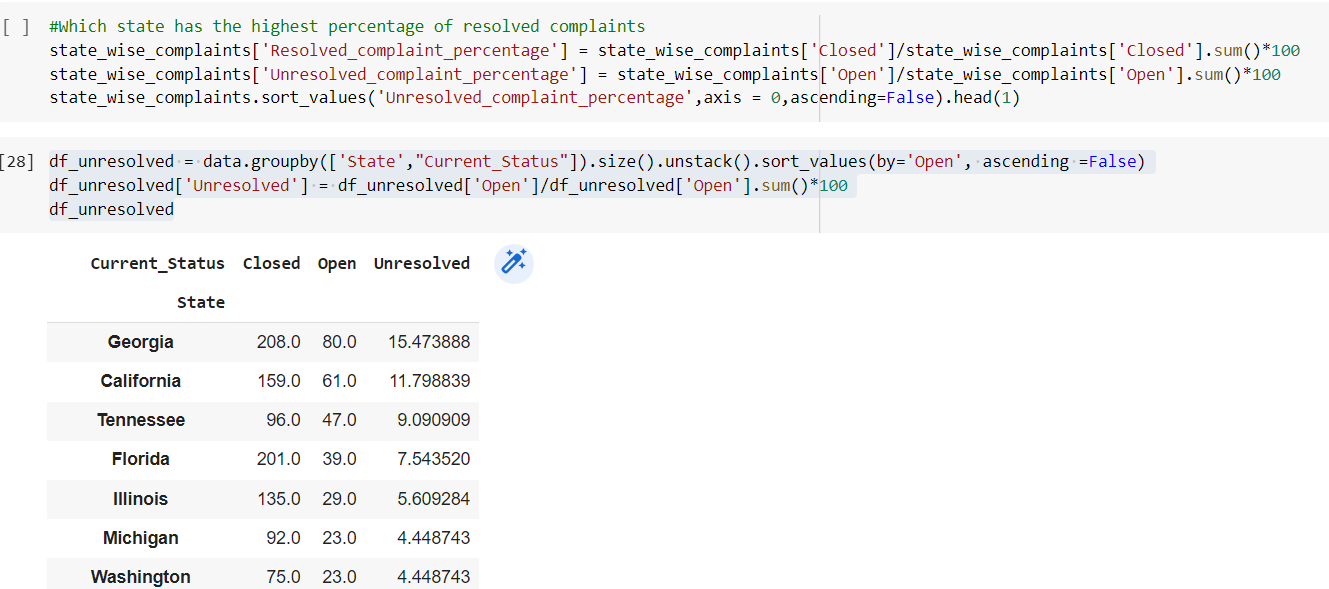
state\_wise\_complaints.sort\_values('Unresolved\_complaint\_percentage',axis = 0,ascending=False).head(1)

df\_unresolved = data.groupby(['State',"Current\_Status"]).size().unstack().sort\_values(by='Open', ascending =False)

df\_unresolved['Unresolved'] = df\_unresolved['Open']/df\_unresolved['Open'].sum()\*100

df\_unresolved

Insights: from the below table, we can clearly see that the Georgia has the highest unresolved complaints.



## Q7. Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

Step10:Provide the percentage of complaints resolved through the Internet and  customer care calls.

df = data.groupby(['Received Via',"Current\_Status"]).size().unstack()

df['Resolved'] = df['Closed']/df['Closed'].sum()\*100

df

df.plot(kind="bar", figsize=(8,8))

