Data Science with Python

Name: Varsha Deshpande

Project : Comcast Telecom Consumer Complaints

Question 1: Import data into Python environment.

- **Python Code:**
  import pandas as pd
  cust_data =
  pd.read_csv("/home/labsuser/Datasets/Comcast_telecom_complaints_data.csv")
  cust_data.head(10)

- **Screenshots with output: (**#printing 10 records output**)**

```
import pandas as pd
cust_data = pd.read_csv("/home/labsuser/Datasets/Comcast_telecom_complaints_data.csv")
cust_data.head(10)
```

	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
5	338519	ISP Charging for arbitrary data limits with ov...	06-12-15	06-Dec-15	9:59:40 PM	Internet	Acworth	Georgia	30101	Solved	No
6	361148	Throttling service and unreasonable data caps	24-06-15	24-Jun-15	10:13:55 AM	Customer Care Call	Acworth	Georgia	30101	Pending	No
7	359792	Comcast refuses to help troubleshoot and corre...	23-06-15	23-Jun-15	6:56:14 PM	Internet	Adrian	Michigan	49221	Solved	No
8	318072	Comcast extended outages	06-01-15	06-Jan-15	11:46:30 PM	Customer Care Call	Alameda	California	94502	Closed	No
9	371214	Comcast Raising Prices and Not Being Available...	28-06-15	28-Jun-15	6:46:31 PM	Customer Care Call	Alameda	California	94501	Open	Yes

- **Insights:**
  Pandas library and read_csv method was used to extract data and head was used to print out only first 10 records.

Question 2 : Provide the trend chart for the number of complaints at monthly and daily granularity levels.

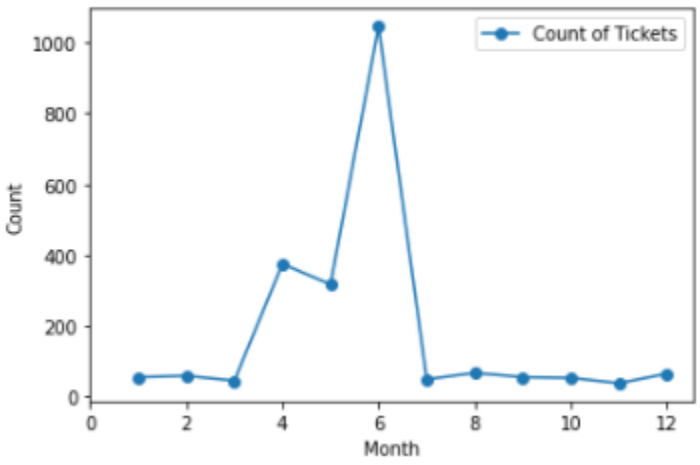
I. Monthly Trend Chart:

- **Python Code:**

```
import pandas as pd
import datetime as dt
import numpy as np
import matplotlib.pyplot as plt

cust_data =
pd.read_csv("/home/labsuser/Datasets/Comcast_telecom_complaints_data.csv")
cust_data.head(10)
cust_data['Date'] = pd.to_datetime(cust_data['Date'],format="%d-%m-%y")
cust_data['Month'] = cust_data['Date'].dt.month

#Month wise data trend chart

plt.plot(cust_data.groupby(['Month']).count()['Ticket #'],label="Count of
Tickets",marker='o')
plt.xlabel('Month')
plt.ylabel('Count')
plt.xticks(np.arange(0,14,2))
plt.legend()
```

- **Screenshots with output:**

```
<matplotlib.legend.Legend at 0x7f5580743ed0>
```



- **Insights:**
  As per the trend chart, 6th month(June) has the most number of complaints. Second to it is 4th month(April) and then the 5rd month(May).

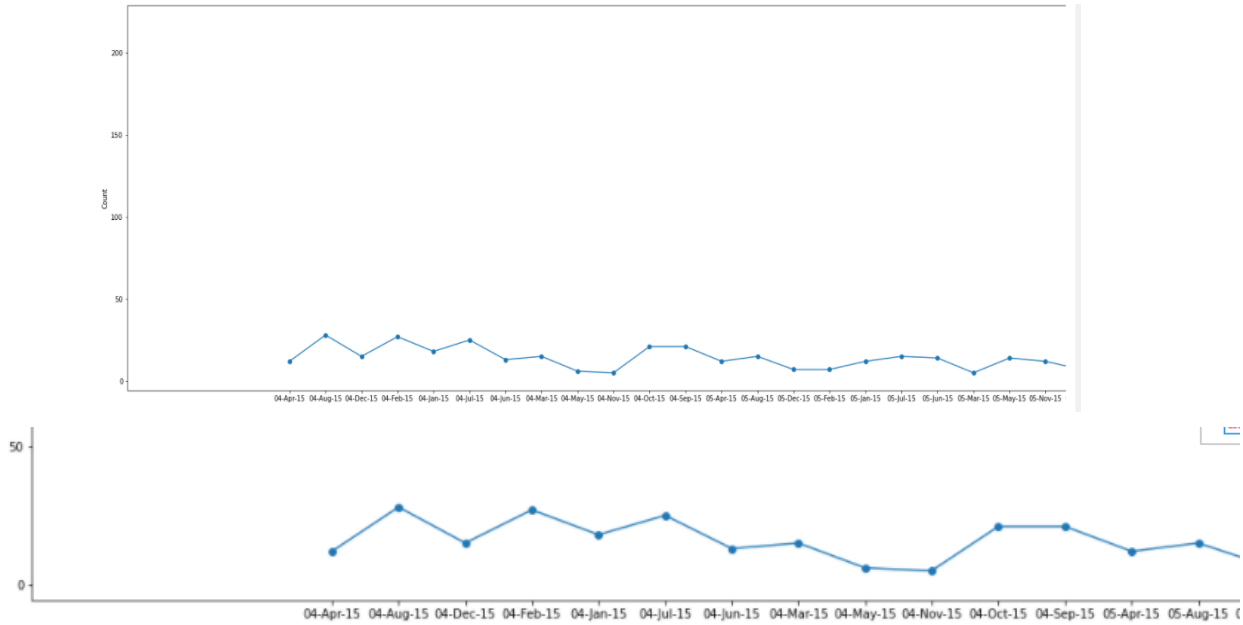
The trend shows a decline in the number of complaints at the start and towards the end of the considered years.

II. Daily Trend Chart:

- **Python Code:**

```
import pandas as pd
import datetime as dt
import numpy as np
import matplotlib.pyplot as plt

cust_data =
pd.read_csv("/home/labsuser/Datasets/Comcast_telecom_complaints_data.csv")
cust_data.head(10)
cust_data['Date'] = pd.to_datetime(cust_data['Date'],format="%d-%m-%y")
cust_data['Month'] = cust_data['Date'].dt.month

#Month wise data trend chart

plt.plot(cust_data.groupby(['Month']).count()['Ticket #'],label="Count of
Tickets",marker='o')
plt.xlabel('Month')
plt.ylabel('Count')
plt.xticks(np.arange(0,14,2))
plt.legend()

#Date wise data trend chart
plt.figure(0)
plt.figure(figsize=(100,10))
plt.plot(cust_data.groupby(['Date_month_year']).count()['Ticket #'],label="Count of
Tickets daily",marker='o')
plt.xlabel('Date')
plt.ylabel('Count')
plt.legend()
```

- **Screenshots with output: (attaching zoomed in screenshot due to poor visibility of overall data plot) (Attached saved plot in submission of screenshots section)**



- **Insights:**
  As per the daily trend chart, 23rd June 2015 (190) and 24th June 2015(218) have the highest number of complaints. The rest dates have complaints number at a consistent scale between 0-50 or 50 -100.

Question 3: Provide a table with the frequency of complaint types.
Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

- **Python Code:**
  print("Grouped_data-> \n",cust_data.groupby(['Customer Complaint']).count().sort_values('Ticket #',ascending=False)['Ticket #'])
  print("As per the counts most of the complaints are of the category - Comcast. Some of them are related to Comcast internet and even billing.")

- **Screenshots with output:**

```
Grouped_data->
 Customer Complaint
Comcast                                             83
Comcast Internet                                    18
Comcast Data Cap                                    17
comcast                                             13
Comcast Billing                                     11
                                                    ..
Comcast internet speeds extremely slow               1
Comcast internet speeds                              1
Comcast internet service that I was NEVER able to use    1
Comcast internet price high                          1
xfinity customer service                             1
Name: Ticket #, Length: 1841, dtype: int64
As per the counts most of the complaints are of the category - Comcast. Some of them are related to Comcast internet and even billi
ng.
```

- **Insights:**
  As per the analysed result, most of the complaints are of the category: Comcast (83 complaints)
  Some other categories like Comcast Internet, Comcast Data Cap are also seen to have some complaints to the lower than Comcast but similar scale of number with each other(10-20 complaints).

Question 4: 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.

- **Python Code:**
  #Creating new Status variable for Open and Closed Complaints

```
cust_data.loc[cust_data['Status'].str.contains('Open') | cust_data['Status'].str.contains('Pending'), 'Cstatus'] = 'Open'
cust_data.loc[cust_data['Status'].str.contains('Solved') | cust_data['Status'].str.contains('Closed'), 'Cstatus'] = 'Closed'
cust_data[['Cstatus','Status']].head(10)
```

- **Screenshots with output:(printing the values of this new column for first 10 records):**

:

	Cstatus	Status
0	Closed	Closed
1	Closed	Closed
2	Closed	Closed
3	Open	Open
4	Closed	Solved
5	Closed	Solved
6	Open	Pending
7	Closed	Solved
8	Closed	Closed
9	Open	Open

- **Insights:** **None**

Question 5: Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:

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

- **Python Code:**

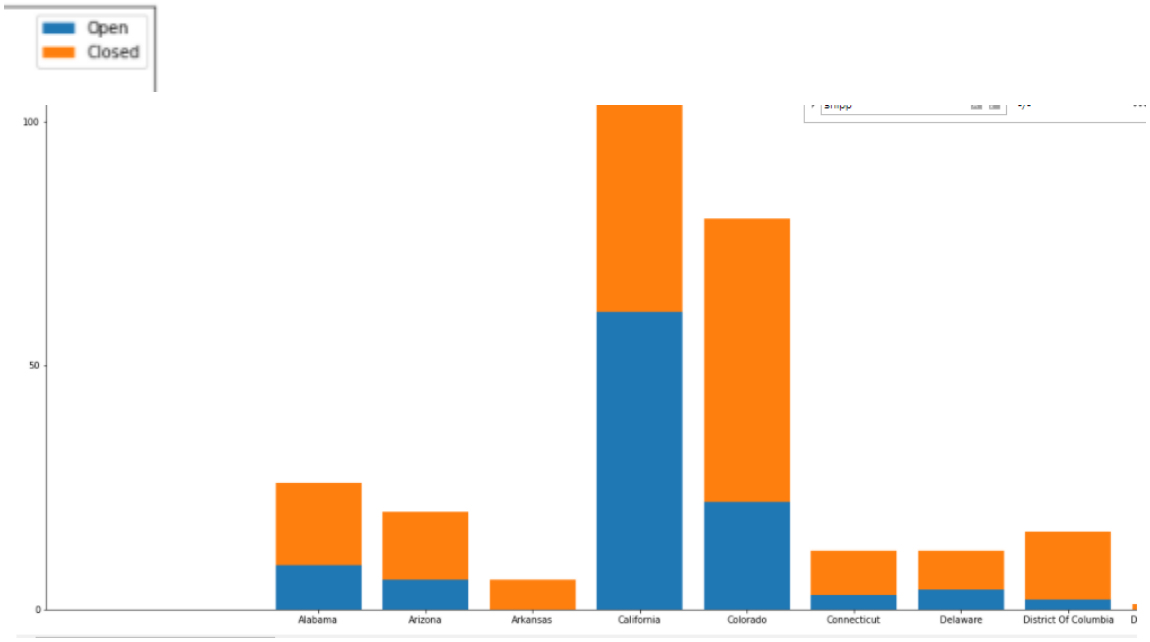
```
#Statewise Open and closed complaints bar graph. Blue -> Open and Orange -> Closed

aggregated = pd.crosstab(cust_data['State'],cust_data['Cstatus'])
aggregated_new = pd.DataFrame(aggregated)
print("Aggregated Data -> \n",aggregated_new.head(10))
plt.figure(figsize=(100,30))
plt.bar(aggregated_new.index,aggregated_new.Open,label='Open')
plt.bar(aggregated_new.index,aggregated_new.Closed,bottom=aggregated_new.Open,label='Clo
sed')
plt.legend()
aggregated_new['Total'] = aggregated_new.sum(axis=1)
aggregated_new['Unresolved Complaints %'] = (aggregated_new.Open/aggregated_new.Total) *100

print("Aggregated Data with Total and Percentage data -> \n",aggregated_new.head(10))
highest_comp = aggregated_new[aggregated_new.Total == aggregated_new.Total.max()]['Total']
print("State with max number of complaints : ", aggregated_new[aggregated_new.Total ==
aggregated_new.Total.max()].index.values,"\nNumber of complaints -> \n",highest_comp)
highest_perc = aggregated_new[aggregated_new['Unresolved Complaints %'] ==
aggregated_new['Unresolved Complaints %'].max()]['Unresolved Complaints %']
print("State with max percentage of unresolved complaints  : ",
aggregated_new[aggregated_new['Unresolved Complaints %'] == aggregated_new['Unresolved Complaints
%'].max()].index.values, "\nPercentage -> \n",highest_perc )
```

- **Screenshots with output: (Printing first 10 for visibility)**

```
Aggregated Data ->
 Cstatus               Closed  Open
State
Alabama                   17     9
Arizona                   14     6
Arkansas                   6     0
California               159    61
Colorado                  58    22
Connecticut                9     3
Delaware                   8     4
District Of Columbia      14     2
District of Columbia       1     0
Florida                  201    39
```

```
Aggregated Data with Total and Percentage data ->
 Cstatus               Closed  Open  Total  Unresolved Complaints %
State
Alabama                   17     9     26                34.615385
Arizona                   14     6     20                30.000000
Arkansas                   6     0      6                 0.000000
California               159    61    220                27.727273
Colorado                  58    22     80                27.500000
Connecticut                9     3     12                25.000000
Delaware                   8     4     12                33.333333
District Of Columbia      14     2     16                12.500000
District of Columbia       1     0      1                 0.000000
Florida                  201    39    240                16.250000
State with max number of complaints :  ['Georgia']
Number of complaints ->
 State
Georgia    288
Name: Total, dtype: int64
State with max percentage of unresolved complaints  :  ['Kansas']
Percentage ->
 State
Kansas    50.0
Name: Unresolved Complaints %, dtype: float64
```

Zoomed in version, due to visibility. Attaching the saved image to submission screenshots as a jpg.



- **Insights:**

  As per the graph analysis

  - **the state “Georgia” has the most number of complaints (288)**
  - **the state “Kansas” has the highest percentage of unresolved complaints. (50%)**

Question 6: Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls

- **Python Code:**

```
aggregated_recvd_via = pd.crosstab(cust_data['Received Via'],cust_data['Cstatus'])
aggregated_recvd = pd.DataFrame(aggregated_recvd_via)
aggregated_recvd['Total'] = aggregated_recvd_via.sum(axis=1)
aggregated_recvd['Resolved Complaints %'] =
(aggregated_recvd['Closed']/aggregated_recvd['Total']) *100
print("Aggregated Data with total & Percentage columns ->\n",aggregated_recvd)

print("Percentage of Closed complaints via - Customer Care Call is - " +
str(int(aggregated_recvd.loc['Customer Care Call']['Resolved Complaints %'])) + "%")
print("Percentage of Closed complaints via - Internet is - " +
str(int(aggregated_recvd.loc['Internet']['Resolved Complaints %'])) + "%")
```

- **Screenshots with output :**

```
Aggregated Data with total & Percentage columns ->
 Cstatus             Closed  Open  Total  Resolved Complaints %
Received Via
Customer Care Call     864   255   1119              77.211796
Internet               843   262   1105              76.289593
Percentage of Closed complaints via - Customer Care Call is - 77%
Percentage of Closed complaints via - Internet is - 76%
```

- **Insights:**

  As per the analysis results the percentage of resolved complaints received via
  - **Customer Care Call - 77.2%**
  - **Internet - 76.2%**