

COMCAST TELECOM COMPLAINTS ANALYSIS

1) Importing Modules and dataset

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

In [2]: TC = pd.read_csv("H:\Exploratory DATA Analysis\TC\Comcast_telecom_complaints.csv")

In [3]: #Let's look at the first few rows of the dataframe
TC.head()

Out[3]:
```

	Ticket_no	Customer complaints	Date	Date_month_year	Time	Received via	City	State	Zip	Status
0	250635	Comcast Cable Internet Speeds	2015-04-22	22 April 2015	15:53:50	Customer Care Call	Abingdon	Maryland	21009	Closed
1	223441	Payment disappear - service got disconnected	2015-08-04	04 August 2015	10:22:56	Internet	Acworth	Georgia	30102	Closed
2	242732	Speed and Service	2015-04-18	18 April 2015	09:55:47	Internet	Acworth	Georgia	30101	Closed
3	277946	Comcast Imposed a New Usage Cap of 300GB that ...	2015-07-05	05 July 2015	11:59:35	Internet	Acworth	Georgia	30101	Open
4	307175	Comcast not working and no service to boot	2015-05-26	26 May 2015	13:25:26	Internet	Acworth	Georgia	30101	Solved

```
In [4]: #Checking for whether any column has null values
TC.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2224 entries, 0 to 2223
Data columns (total 11 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Ticket_no                            2224 non-null   object
1   Customer complaints                 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                                 2224 non-null   int64
9   Status                              2224 non-null   object
10  Filling on behalf of someone        2224 non-null   object
dtypes: int64(1), object(10)
memory usage: 191.2+ KB
```

2) Data Manipulation required for analysis

----> Categorizing all the complaints

```
In [5]: TC['Customer complaints'] = TC['Customer complaints'].astype('string')
TC['Customer complaints'] = TC['Customer complaints'].str.upper()

DC = ['DATA', 'USA', 'CAP', 'LIMIT']
i = 0
while i < 2224:
    output = any(d in TC['Customer complaints'][i] for d in DC)
    if output:
        TC['Customer complaints'][i] = 'Broadband Data Caps'
        i = i + 1

IE = ['SPEED', 'INTERNET', 'SLOW']
i = 0
while i < 2224:
    output = any(e in TC['Customer complaints'][i] for e in IE)
    if output:
        TC['Customer complaints'][i] = 'Internet Connectivity (or) speed issue'
        i = i + 1

BP = ['PAY', 'BILL', 'CHARGE', 'PRICING', 'PRICE', 'CHARGING']
i = 0
while i < 2224:
    output = any(b in TC['Customer complaints'][i] for b in BP)
    if output:
        TC['Customer complaints'][i] = 'Over prices (or) Incorrect Billing (or) Payment'
        i = i + 1

SE = ['SERVICE']
i = 0
while i < 2224:
    output = any(s in TC['Customer complaints'][i] for s in SE)
    if output:
        TC['Customer complaints'][i] = 'Poor Service'
        i = i + 1

TC.loc[~TC['Customer complaints'].isin(['Broadband Data Caps', 'Internet Connectivity (or) speed issue', 'Over prices (or) Incorrect Billing (or) Payment', 'Poor Service'])]

Out[6]:
```

Customer complaints	count
Other	619
Internet Connectivity (or) speed issue	599
Over prices (or) Incorrect Billing (or) Payment	518
Broadband Data Caps	255
Poor Service	233

Name: Customer complaints, dtype: Int64

----> Adding month column

```
In [7]: TC['Date'] = pd.to_datetime(TC['Date'], format = '%Y-%m-%d')
TC['Month'] = TC['Date'].dt.month_name()
```

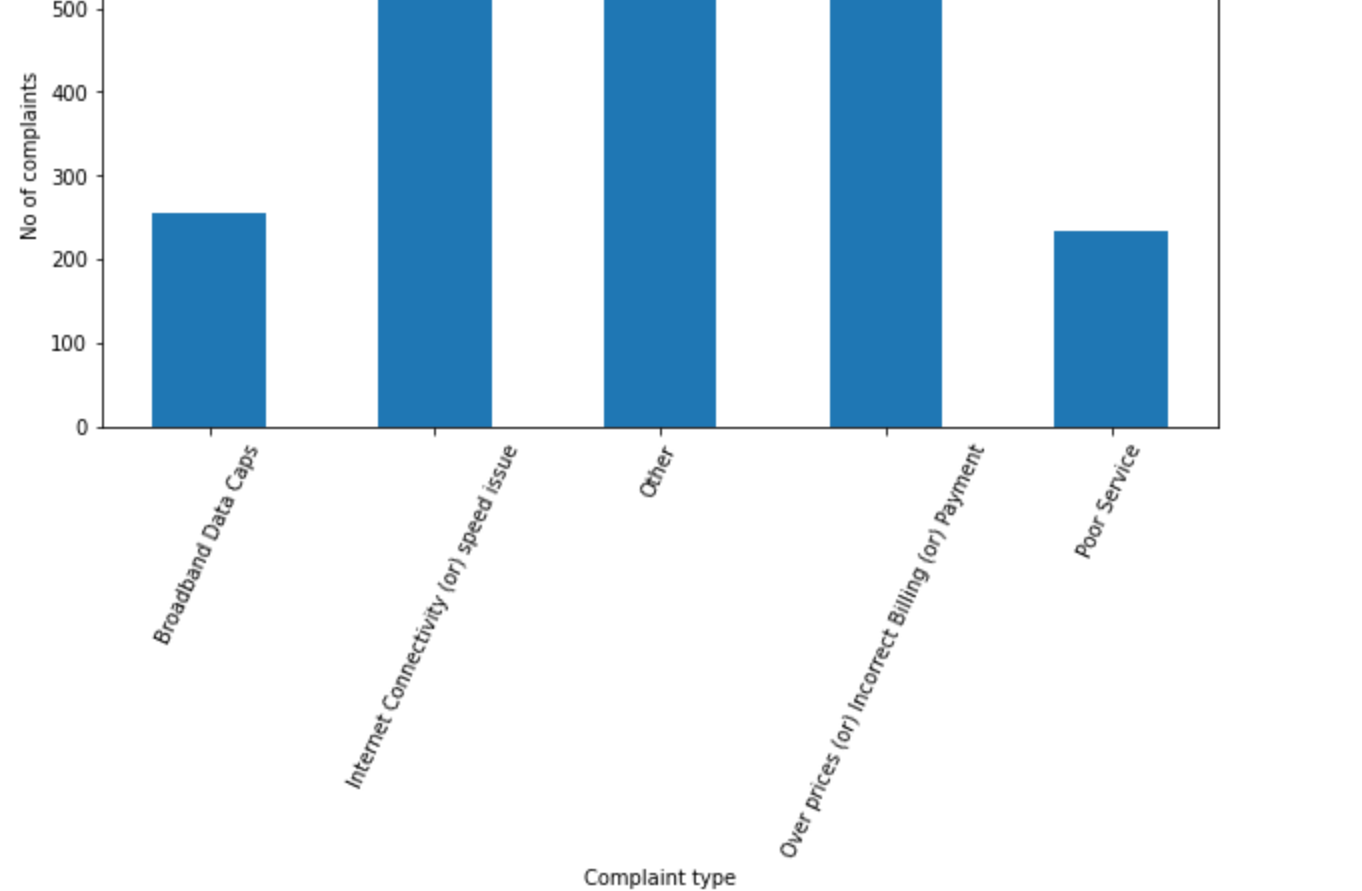
----> Changing columns containing categorical values to type category

```
In [8]: TC['Status'] = TC['Status'].astype('category')
TC['Customer complaints'] = TC['Customer complaints'].astype('category')
```

3) Plotting Data

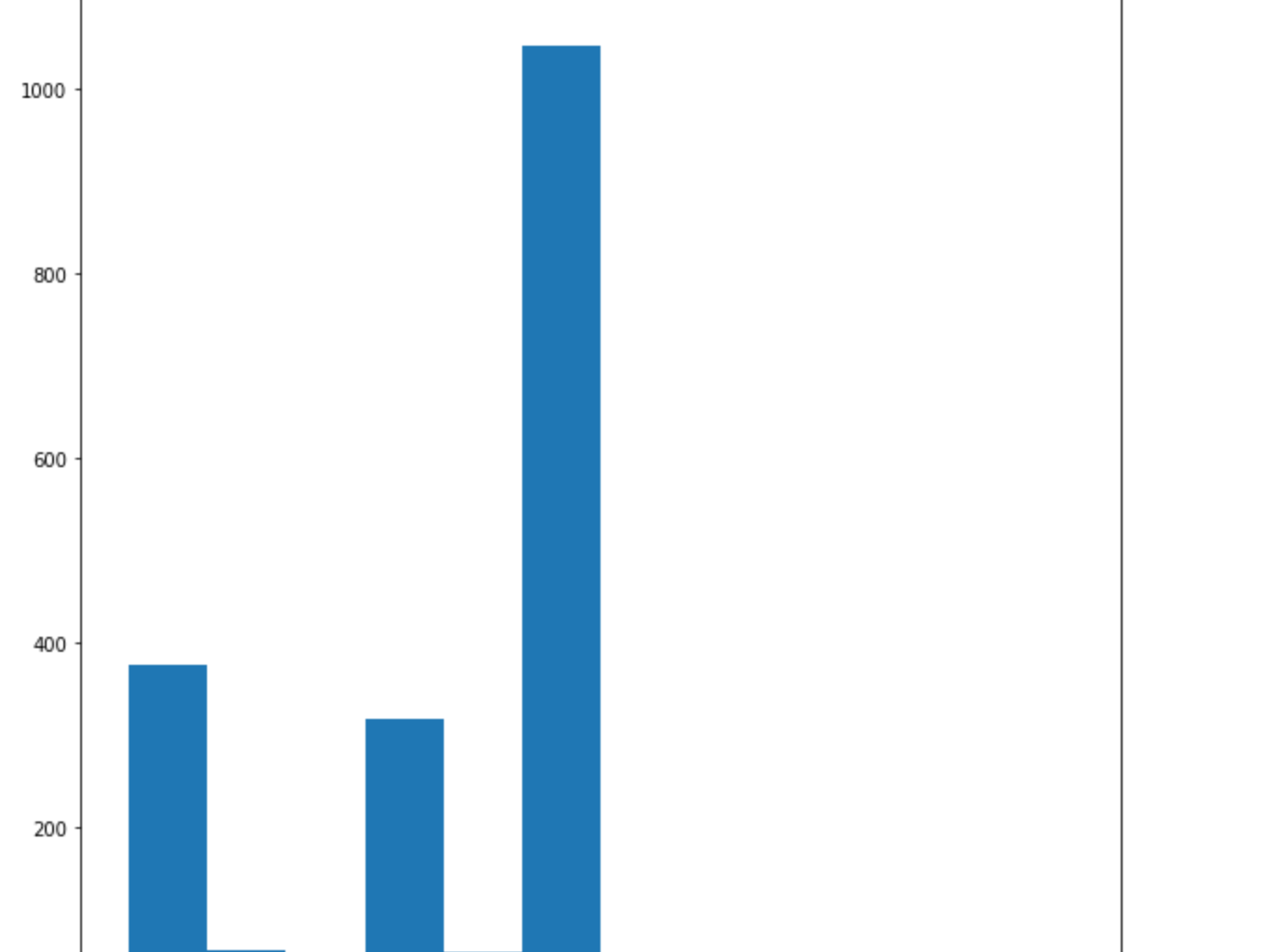
----> Total count for each type of complaint

```
In [9]: TC['count'] = 1
res = TC.groupby(['Customer complaints']).count()
fig,ax = plt.subplots(figsize = (10,5))
plt.bar(res.index, res['count'], width = 0.5)
plt.xticks(res.index, rotation = 65)
plt.xlabel("Complaint type")
plt.ylabel("No of complaints")
plt.title("Number of complaints for each type")
plt.show()
```



----> Which month has highest distribution of complaints?

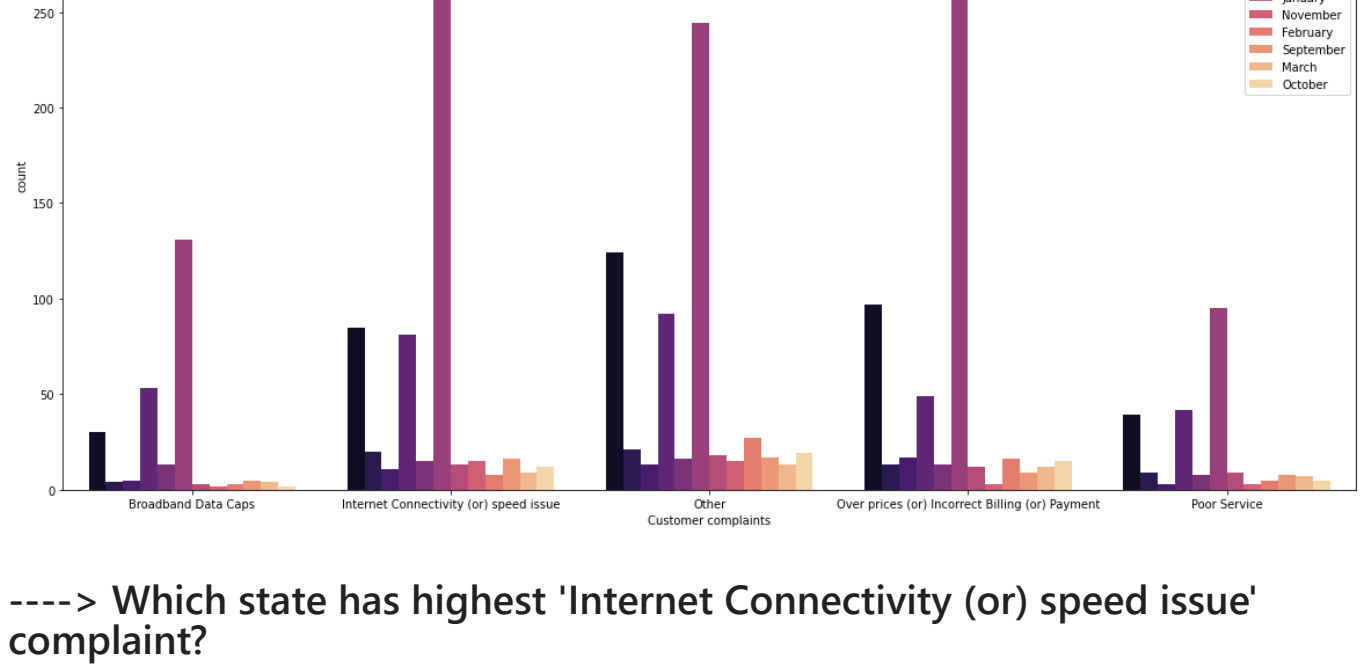
```
In [30]: fig,ax = plt.subplots(figsize = (10,10))
plt.title("Distribution of complaints across all months")
ax.hist(TC['Month'], bins = 12)
plt.show()
```



----> Number of complaints for each month for different type of complaint

```
In [12]: fig,ax = plt.subplots(figsize = (20,10))
plt.title("Total number for each complaint across all months")
sns.countplot(data = TC, x = "Customer complaints", hue = "Month", palette = 'magma')
```

```
Out[12]: <AxesSubplot:title='center':Total number for each complaint across all months', xlabel='Customer complaints', ylabel='count'>
```



----> Which state has highest 'Internet Connectivity (or) speed issue' complaint?

```
In [13]: Internet = TC.loc[TC['Customer complaints'] == 'Internet Connectivity (or) speed issue']
fig,ax = plt.subplots(figsize = (40,20))
plt.title("Number of Internet Connectivity (or) speed issue across different states")
sns.countplot(data = Internet, x = "State")
```

```
Out[13]: <AxesSubplot:title='center':Number of Internet Connectivity (or) speed issue across different states', xlabel='State', ylabel='count'>
```

