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

[18]:
```
data= pd.read_csv(r"C:\Users\Dell\OneDrive\Documents\ET practical\comcost.csv")
```

[19]:
```
data.head(5)
```

[19]:

	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	15:53:50	Customer Care Call	Abingdon	Maryland	21009	Closed	No
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56	Internet	Acworth	Georgia	30102	Closed	No
2	242732	Speed and Service	18-04-15	18-Apr-15	09:55:47	Internet	Acworth	Georgia	30101	Closed	Yes

[20]:
```
data.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: 191.3+ KB
```

[21]:
```
data.columns
```

[21]:
```
data.columns
```

[21]:
```
Index(['Ticket #', 'Customer Complaint', 'Date', 'Date_month_year', 'Time',
       'Received Via', 'City', 'State', 'Zip code', 'Status',
       'Filing on Behalf of Someone'],
      dtype='object')
```

[22]:
```
data['Date'] = pd.to_datetime(data['Date'])
data['Month'] = data['Date'].dt.to_period('M')  # # monthly granularity
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_6596\2069294920.py:1: UserWarning: Could not infer format, so e
vidually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a fo
  data['Date'] = pd.to_datetime(data['Date'])
```

[26]:
```
monthly_trend = data.groupby('Month').size()
```

[29]:
```
plt.figure(figsize=(4,2))
monthly_trend.plot(kind='bar', color='skyblue')
plt.title("Monthly complain")
plt.xlabel("Month")
plt.ylabel("Num of Complaints")
plt.show()
```

```
plt.show()
```


```
#Ye chart batata hai kis month me sabse zyada complaints ayi hai .
#Usually, trend line dikha rha  hai increasing ya decreasing pattern ka
```

[30]:
```
complaint_freq = data['Customer Complaint'].value_counts()
```

[31]:
```
complaint_freq
```

[31]:
```
Customer Complaint
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: count, Length: 1841, dtype: int64
```

```
# in above code Most complaints are related to Internet and Network issues,
# indicating major service quality problems.
```

[32]:
```
#Create a new categorical variable with value as Open and Closed.
```

[32]:
```
#Create a new categorical variable with value as Open and Closed.
data['Status'] = data['Status'].str.lower()

def status_group(x):
    if x in ['open','pending']:
        return 'Open'
    elif x in ['closed','solved']:
        return 'Closed'
    else:
        return 'Other'
```

[33]:
```
data['Status_Category'] = data['Status'].apply(status_group)
print(data['Status_Category'].value_counts())
```

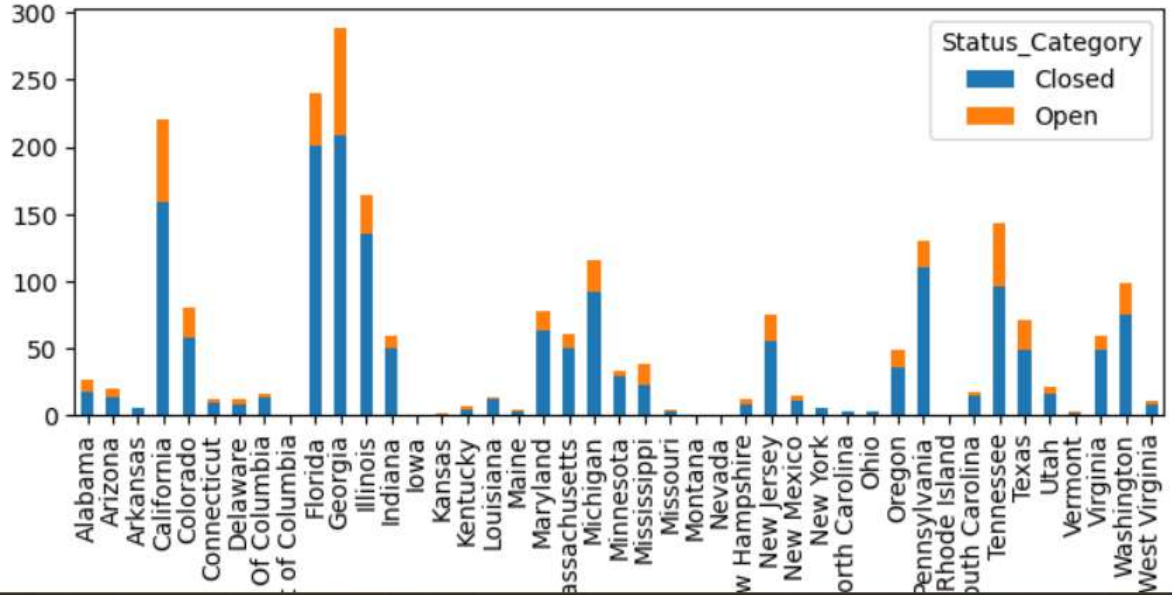
```
Status_Category
Closed    1707
Open       517
Name: count, dtype: int64
```

[42]:
```
state_status = pd.crosstab(data['State'], data['Status_Category'])

# Plot stacked bar chart
```

```
state_status.plot(kind='bar', stacked=True, figsize=(8,3))
```

[42]: <Axes: xlabel='State'>



[43]:
```
#Which state has maximum complaints
print(data['State'].value_counts().head(1))
```

```
State
Georgia    288
Name: count, dtype: int64
```

[45]:
```
# percentwise complian
unresolved = state_status.apply(lambda x: x['Open'] / x.sum() * 100, axis=1)
print(unresolved.sort_values(ascending=False).head(1))
```

```
State
Kansas    50.0
dtype: float64
```

[44]:
```
#Percentage of Complaints Resolved by Mode Received
received_mode = pd.crosstab(data['Received Via'], data['Status_Category'])
received_mode['Resolved %'] = round(received_mode['Closed'] / (received_mode['Closed'] + received_mode['Open']) * 100, 2)
print(received_mode)
```

```
Status_Category     Closed  Open  Resolved %
Received Via
Customer Care Call     864   255       77.21
Internet               843   262       76.29
```

[44]:
```
#Percentage of Complaints Resolved by Mode Received
received_mode = pd.crosstab(data['Received Via'], data['Status_Category'])
received_mode['Resolved %'] = round(received_mode['Closed'] / (received_mode['Closed'] + received_mode['Open']) * 100, 2)
print(received_mode)
```

```
Status_Category     Closed  Open  Resolved %
Received Via
Customer Care Call     864   255       77.21
Internet               843   262       76.29
```

[46]:
```
#here inferance is Check which communication mode (Internet / Customer Care Call) has a higher resolution rate
```

[ ]: