In [1]:
```
import os
os.getcwd()
```

Out[1]:
```
'C:\\Users\\218882'
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
# Import data into python environment.

com_df=pd.read_csv("C:\\Users\\218882\\Comcast_telecom_complaints_data.csv")
```

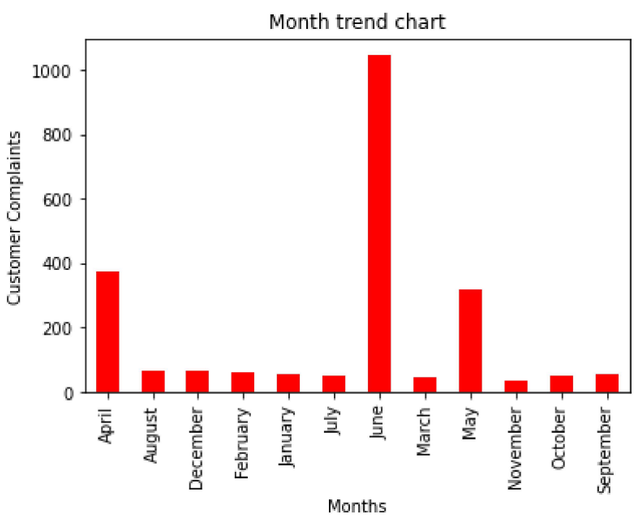
In [4]:
```
com_df.head()
```

Out[4]:

	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

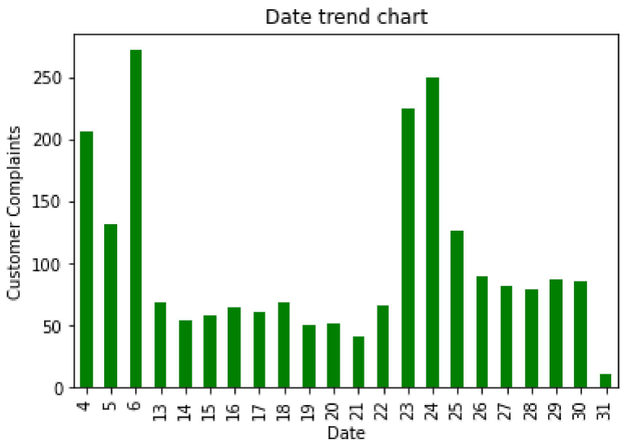
In [6]:
```
# Provide the trend chart for the number of complaints at monthly and daily granularity levels.
# Graph for month wise

com_df["Month"]=pd.to_datetime(com_df["Date_month_year"]).dt.month_name()
com_df.groupby(["Month"])["Customer Complaint"].count().plot.bar(color="r")
plt.title("Month trend chart")
plt.xlabel("Months")
plt.ylabel("Customer Complaints")
plt.show()
```



In [8]:
```
# Graph for date wise

com_df["Date"]=pd.to_datetime(com_df["Date_month_year"]).dt.day
com_df.groupby(["Date"])["Customer Complaint"].count().plot.bar(color="g")
plt.title("Date trend chart")
plt.xlabel("Date")
plt.ylabel("Customer Complaints")
plt.show()
```



In [9]:
```
# Provide a table with the frequency of complaint types.

com_df["Customer Complaint"].value_counts().to_frame().reset_index()
```

Out[9]:

	index	Customer Complaint
0	Comcast	83
1	Comcast Internet	18
2	Comcast Data Cap	17
3	comcast	13
4	Comcast Billing	11
...	...	...
1836	Improper Billing and non resolution of issues	1
1837	Deceptive trade	1
1838	intermittent internet	1
1839	Internet Speed on Wireless Connection	1
1840	Comcast, Ypsilanti MI Internet Speed	1

1841 rows × 2 columns

In [10]:
```
# Which complaint types are maximum i.e., arround internet, network issues, or across any other domains.

com_df["Customer Complaint"].value_counts().head()
```

Out[10]:
```
Comcast             83
Comcast Internet    18
Comcast Data Cap    17
comcast             13
Comcast Billing     11
Name: Customer Complaint, dtype: int64
```

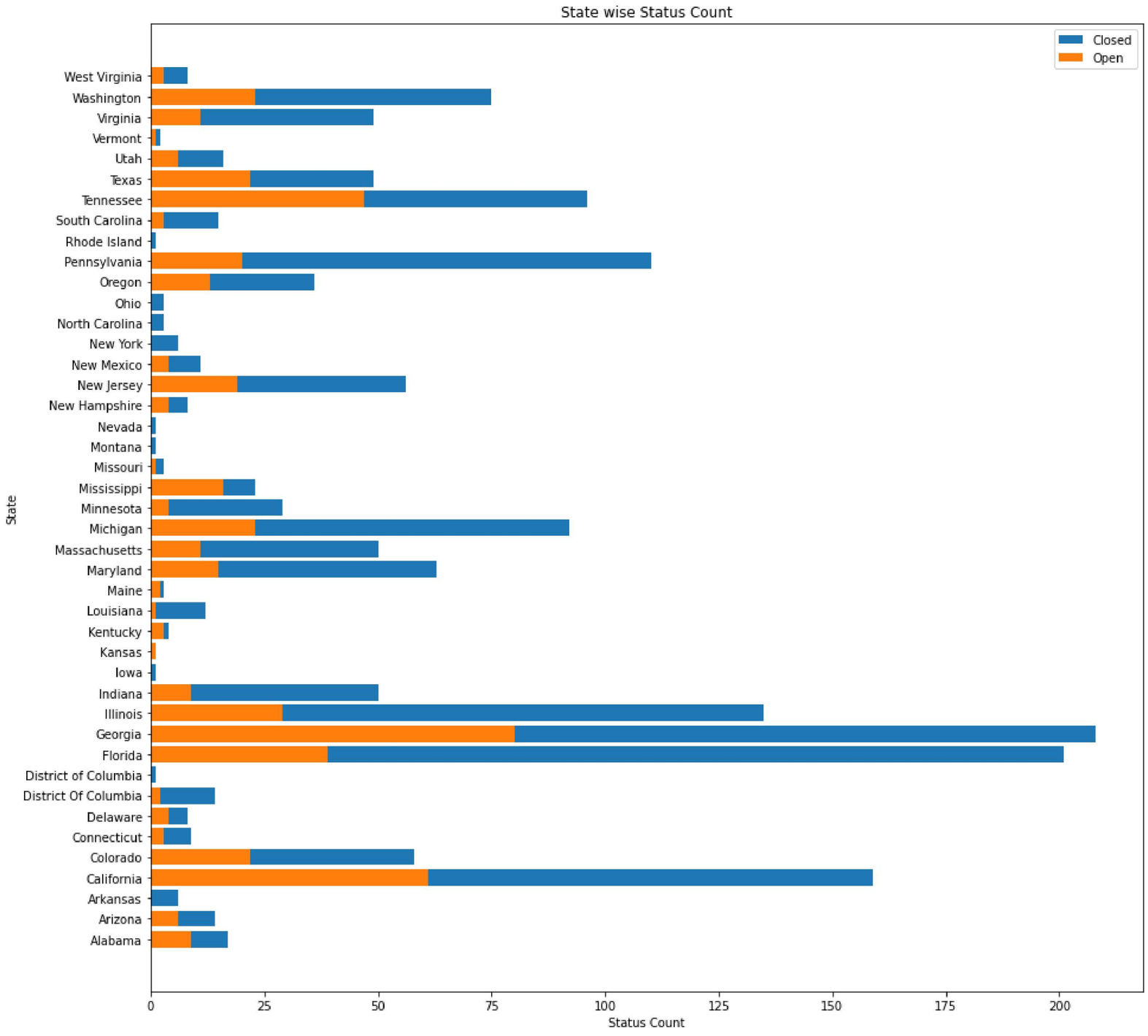
In [20]:
```
# Create a new categorical variable with value as open and closed.
#open and pending it to be categorized as open and closed and solved is to be categorized as closed.

com_df["Status"]=com_df["Status"].apply(lambda x: "Open" if ((x=="Open") | (x=="Pending")) else "Closed")
```

In [24]:
```
# Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3.


open=com_df[com_df["Status"]=="Open"].groupby(["State"])["Status"].count().to_frame().reset_index()
close=com_df[com_df["Status"]=="Closed"].groupby(["State"])["Status"].count().to_frame().reset_index()
```

In [26]:
```
fig=plt.figure(figsize=(15,15))
plt.barh(close.State, close.Status)
plt.barh(open.State, open.Status)
plt.title("State wise Status Count")
plt.xlabel("Status Count")
plt.ylabel("State")
plt.legend(["Closed", "Open"])
plt.show()
```



In [28]:
```
# Which state has maximum complaints.

com_df.groupby("State")["Customer Complaint"].agg("count").sort_values(ascending=False).head(1)
```

Out[28]:
```
State
Georgia    288
Name: Customer Complaint, dtype: int64
```

In [36]:
```
# Which state has the highest percentage of unresolved complaints.

State_Unsolved=com_df.loc[com_df["Status"]=="Open",["State"]].value_counts()
State_Unsolved.head(1)/State_Unsolved.sum()*100
```

Out[36]:
```
State
Georgia    15.473888
dtype: float64
```

In [37]:
```
# Provide the percentage of complaints resolved till date, which were received through the internet and customer care calls.

com_df[com_df["Status"]=="Closed"].groupby("Status")["Received Via"].value_counts(normalize=True)*100
```

Out[37]:
```
Status  Received Via
Closed  Customer Care Call    50.615114
        Internet              49.384886
Name: Received Via, dtype: float64
```

In [ ]:
```

```