# Importing Necessary Libraries

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| In [144...  In [145...  In [146... | import pandas as pd import numpy as np import matplotlib. pyplot as PI t  PI t. rcParams[ 'figure.figsize' ] - [15, 8] import seaborn as sns sns . set\_ style ( "whitegrid " )  data-pd . read\_csv( " Comcast telecom complaints data . csv" )  \_ \_ \_  data. head() |  |  |  |  |  |  |
| Out [146] : | Ticket | Received |  |  | Zip |  | Filing on |
|  | Customer Complaint Date Date\_month\_year Time | Via | City | State | code | Status | Behalf of  Someone |

 Comcast Cable Internet 22-  Customer

o 250635 22-Apr-15Abingdon Maryland 21009 Closed No

Speeds 04-15  Care Call

 Payment disappear - 04-

1. 223441 04-Aug-15Internet Acworth Georgia 30102 Closed No

service got disconnected 08-15 AM

18-

1. 242732 Speed and Service 18-Apr-15Internet Acworth Georgia 30101 Closed Yes

04-15 AM

Comcast Imposed a New

05-

1. 277946 Usage Cap of 300GB that 05-Jul-15Internet Acworth Georgia 30101 Open Yes

07-15 AM

 Comcast not working 26-

1. 307175 26-May-15Internet Acworth Georgia 30101 Solved No and no service to boot 05-15

In [149... data . dtypes

Ticket # object Out [149] :

Customer Complaint object

Date object

Date\_month\_year object

Time object

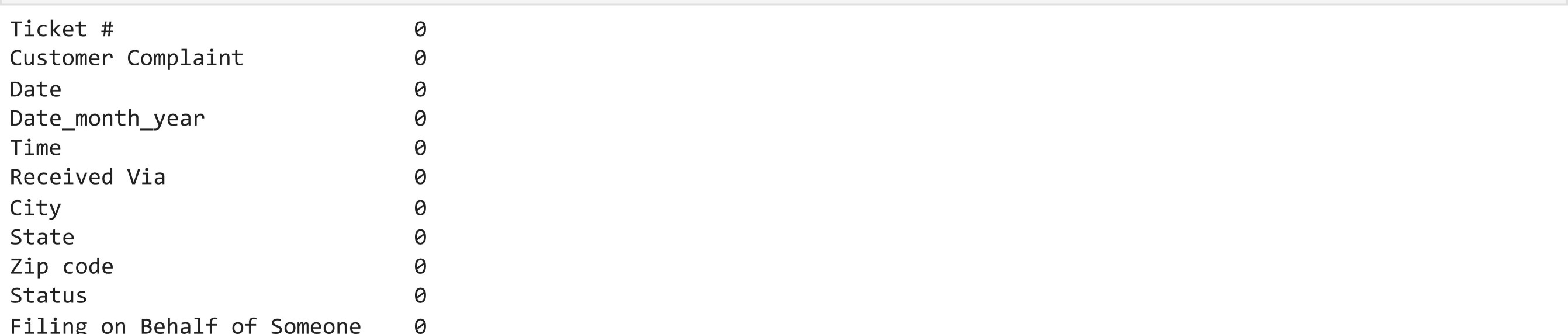
Received Via object

City object State object Zip code int64

Status object Filing on Behalf of Someone object dtype: object

In [ 148... data.isna() .sum()

Out [148] :



Filing

Behalf

of

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| In [ 150...    In [ 152... | dtype: int64  data [ "           " ] . to\_datetime (data [ "            " ] )  C:                py: l: UserWarning:  Could not infer format, so each element will be parsed individually, falling back to dateutil\* . To ensure parsing is c onsistent and as-expected, please specify a format.  data . dtypes  Extracting month and day from Date column to get trend chart  data [ " month" ] -data [ "Date\_month\_year" ] . dt . month data [ "day" ] -data [ "Date\_month\_year" ] . dt . day | | | | | | | | | | | | | |
| In [ 153...  Out[153] : | | | data. head()  Ticket Customer |  |  |  | Received |  |  | Zip |  | Filing on |  |  |
|  | | | Complaint | Date | Date\_month\_year | Time | Via | City | State | code | Status | Behalf of  Someone | month | day |

22-

Comcast Cable  Customer o 250635 04- 201 5-04-22Abingdon Maryland 21009 Closed No 4 22

Internet Speeds  Care Call

15

Payment 04disappear -

1. 223441 08- 2015-08-04 Internet Acworth Georgia 30102 Closed No 8 4

service got AM

15 disconnected

18-

Speed and 9:55:47

1. 242732 04- 2015-04-18 Internet Acworth Georgia 30101 Closed Yes 4 18

Service AM

15

Comcast Imposed 05-

1. 277946 a New Usage Cap 07- 2015-07-05 Internet Acworth Georgia 30101 Open Yes 7 5 AM of 300GB that .15

Comcast not 26-

|  |  |  |  |
| --- | --- | --- | --- |
| 4 307175 working and no 05-  service to boot 15 | 2015-05-26 Internet Acworth Georgia 30101  PM | Solved | No 5 26 |
| month\_group=data . groupby( "month " , | as index-False) ["Customer Complaint"] .count() |  |  |

In [ 160...

|  |  |  |
| --- | --- | --- |
| In | [161... | month\_group |

Out[161] : month Customer Complaint

55

59 45

375 317

1046 49

67

55

* 1. 53

38

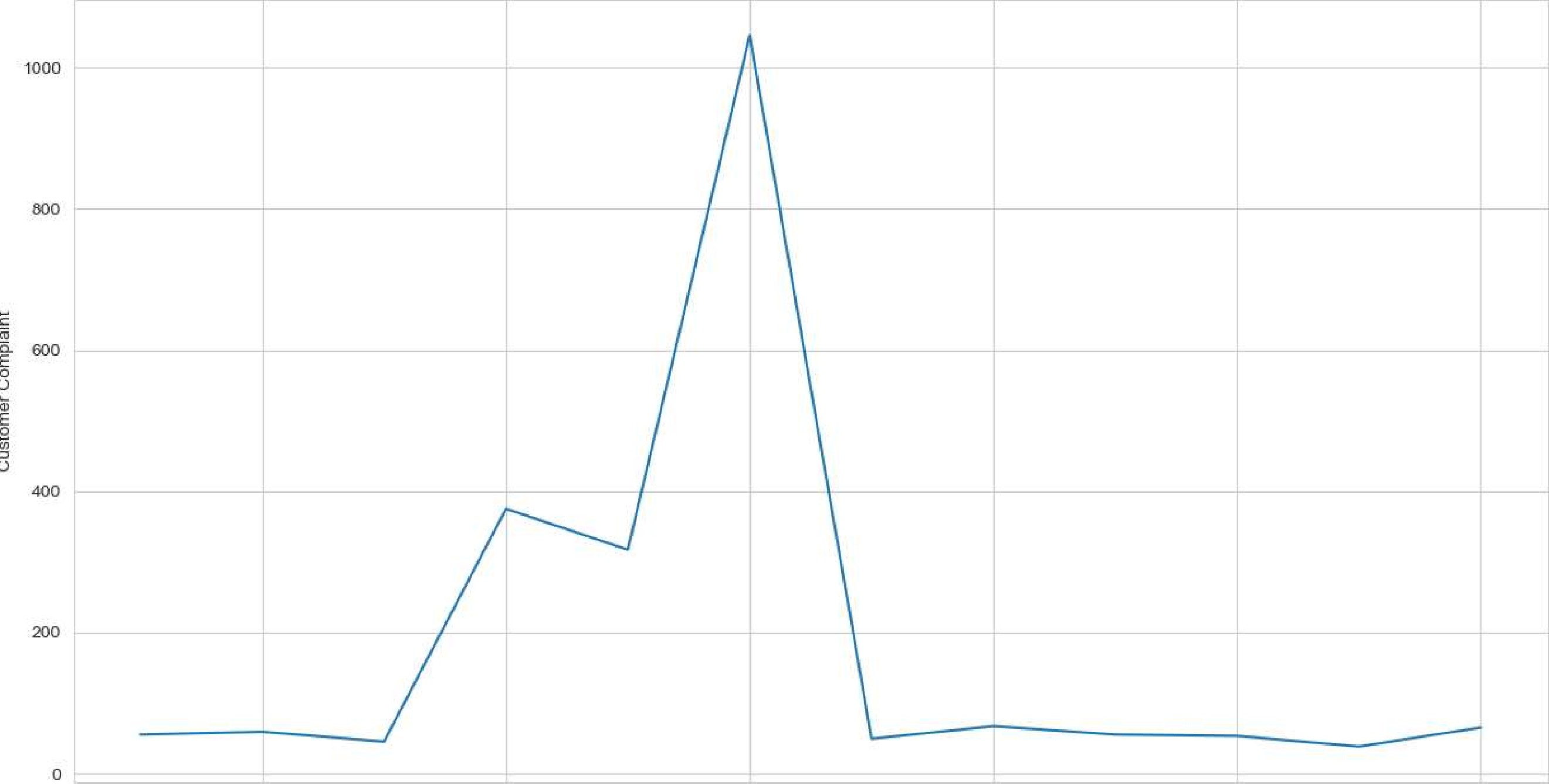
* 1. 12 65

# TREND CHART BASED ON MONTHLY COMPLAINTS

In [ 162... sns.lineplot(data=month\_group, x: "month" ,  Complaint" )

<AxesSubp10t: xlabel='month' , ylabe1='Customer Complaint' >

Out [162] :



2 4 6 8 10 12 nTnth

164... daily\_group=data . groupby( ”day” , as index-False) ["Customer Complaint"] . count()

[ 165... daily\_group

Out[165] : day Customer Complaint

206

131

272

13 68

4 14 54

15 58

1. 16 65
2. 17 60
3. 18 69
4. 19 50
5. 20
6. 21 41
   1. 66
   2. 225

14 24 249

25 126

1. 26 90
2. 27 81
3. 28 79

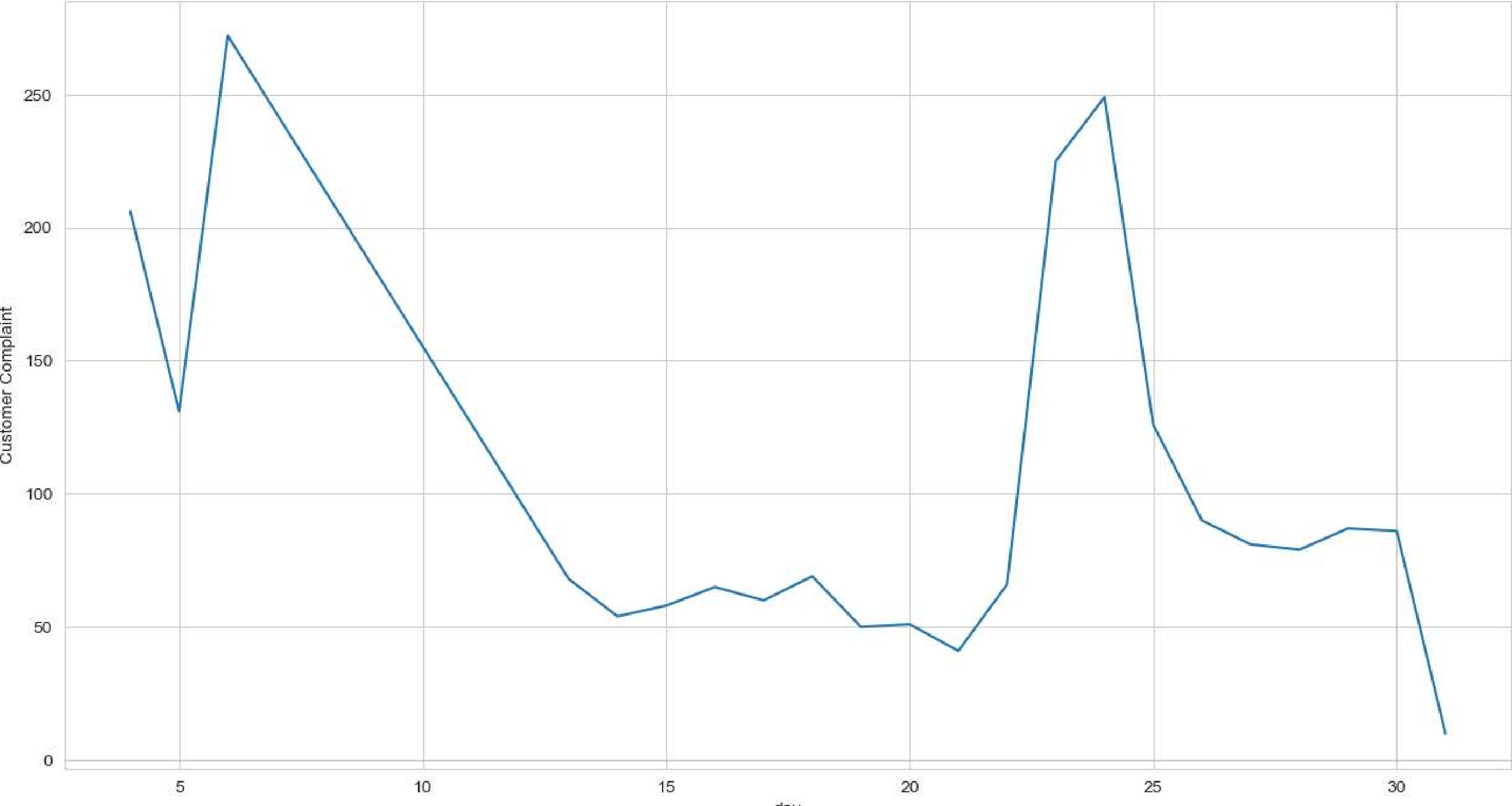
29 87

20 30 86

31 10

# TREND CHART BASED ON DAILY COMPLAINTS

|  |  |
| --- | --- |
|  | day  table with the frequency of complaint types |
| In [ 167... | complaints\_frequency-datal "Customer Complaint"] . value\_counts() |
| In [ 168... | complaints\_frequency=complaints\_frequency.rename\_axis( 'Customer Complaints ' ) . reset\_index(name= 'Number of Complaints' ) |
| In [ 169... | complaints\_frequency |



|  |  |
| --- | --- |
| In [ 166... | sns.lineplot(data=daily\_group, x: "day", y="Customer Complaint") |
| Out [166] : | <AxesSubp10t: x label='day' , ylabe1='Customer Complaint' > |

Out[169] : Customer Complaints Number of Complaints

Comcast 83

1. Comcast Internet 18
2. Comcast Data Cap 17
3. comcast 13
4. Comcast Billing 1 1
5. Improper Billing and non resolution of issues 1
6. Deceptive trade 1
7. intermittent internet 1
8. Internet Speed on Wireless Connection 1
9. Comcast, Ypsilanti Ml Internet Speed 1
10. rows x 2 columns

In [ 170... 

data. head()

|  |  |  |
| --- | --- | --- |
| In [ 172... | datal "Complaint types"] . |  |
| Out [172] : | Complaint types  Customer Care Call 1119  Internet 1105  Name: count, dtype: int64 |  |
| In [ 173... | data [ "Status " ] . value\_ counts ( ) |  |
| Out[173] •. | Status  Solved 973  Closed 734 Open 363 pending 154  Name: count, dtype: int64 |  |

In [ 174...

I n 58 ] :

In [59] :



In [ 176...

Out [176] •.

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.

I (data["Status"]=="Pending"),

I (data ["Status"]=="S01ved")] values=[ "Open " , "Closed" ] datal "New Status"]=np.select(conditions, values)

"Open Open "

#data. Loc [ (datal "Status " "Open ") (datal "Status " "Pending ") "Open " "Open "

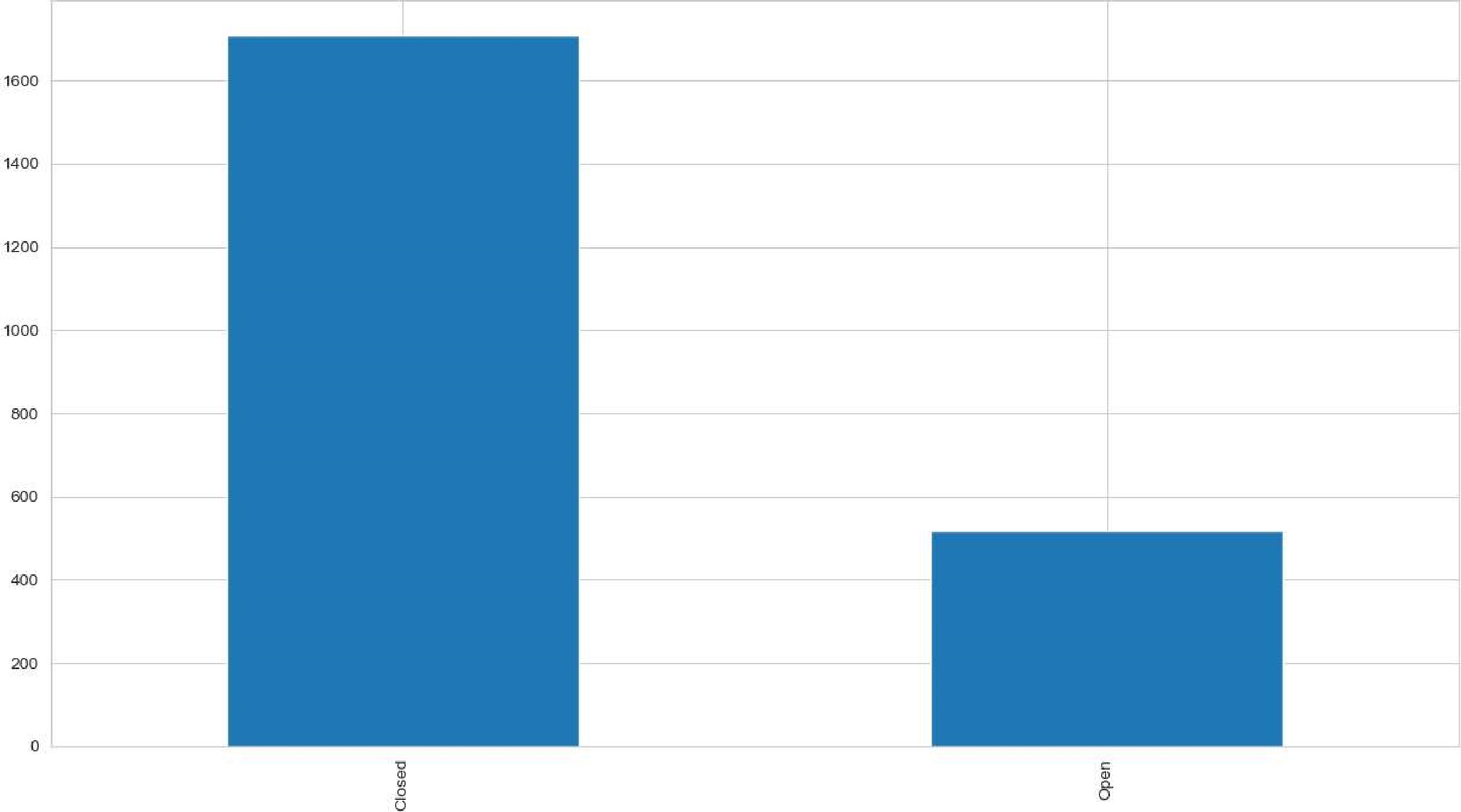
## "Not Closed"

#data. Loc [ (datal "Status " "Closed ") / (data[ "Status" "Sol ved"),  "Closed"

data. head()

data [ "New\_Status " ] . value\_counts ( ) . plot (kind: " bar" )

<AxesSubp10t: xlabel= 'New Status



New Status

# state wise status of complaints in a stacked bar chart

## In [ 184... #state\_group=data. groupby( [ "State ", "New\_ Status " ] , as\_index=FaLse) . count ( )

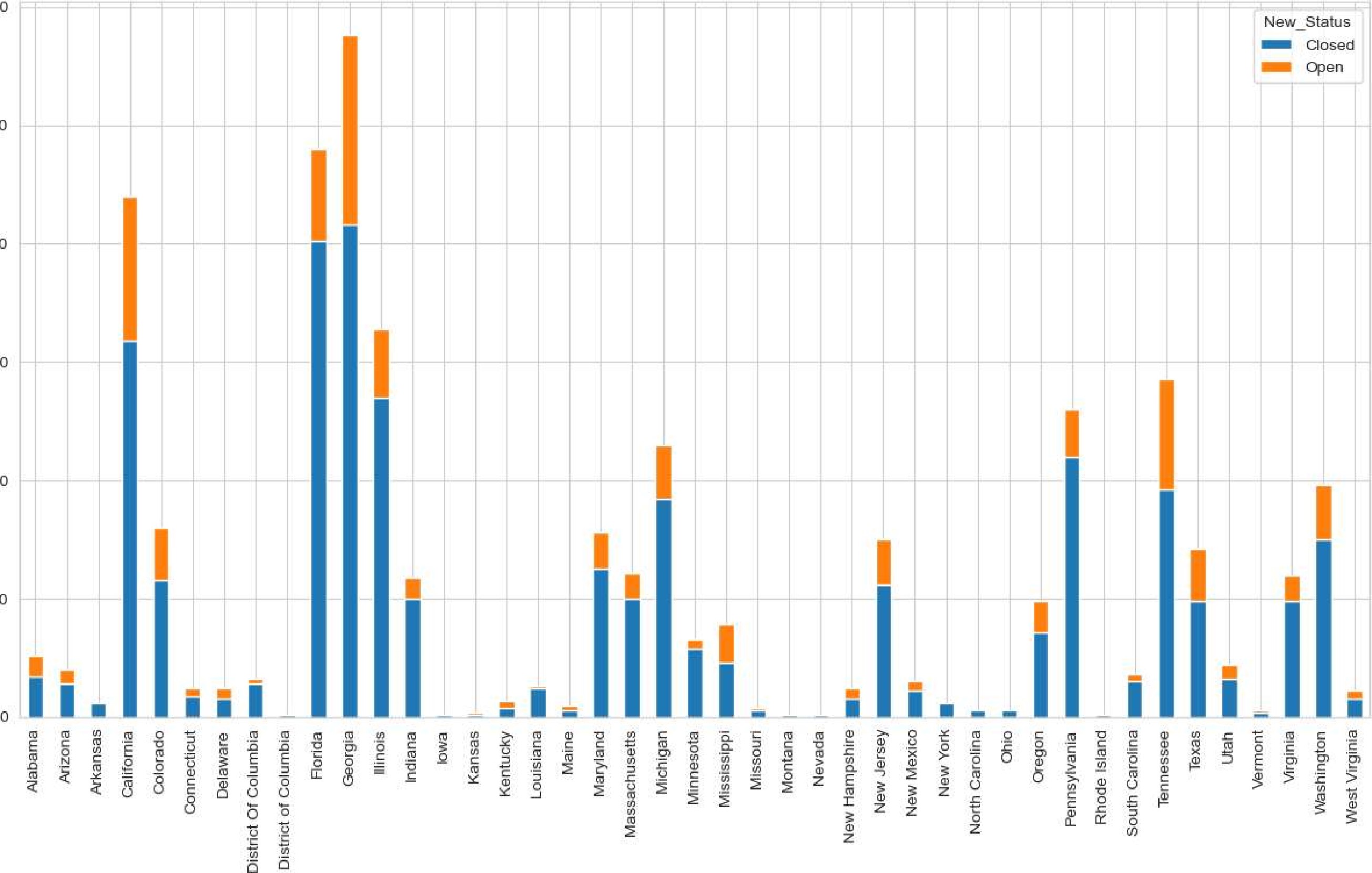
In [ 188... state\_group-data . groupby( ["State", "New\_Status"]) • size() . reset\_index() . pivot (columns= New Status' , index= ' State' ,

In [ 190... state\_group. plot (kind= bar l , stacked-True)

<AxesSubp10t : xlabel= State I >

Out[190] :

localhost:8888/lab/tree/Comcast telecom\_complaints\_data\_Analysis.ipynb



300

Comcast

telecom\_complaints\_data\_Analysis

State

250

200

150

100

In [ 194... state\_group. sort \_values ( by: " Closed " , ascending-False)

Out [194] : New Status Closed Open

State

|  |  |  |
| --- | --- | --- |
| Georgia | 208.0 | 80.0 |
| Florida | 201.0 | 39.0 |
| California | 159.0 | 61.0 |
| Illinois | 135.0 | 29.0 |
| Pennsylvania | 1 10.0 | 20.0 |
| Tennessee | 96.0 | 47.0 |
| Michigan | 92.0 | 23.0 |
| Washington | 75.0 | 23.0 |
| Maryland | 63.0 | 15.0 |
| Colorado | 58.0 | 22.0 |
| New Jersey | 56.0 | 19.0 |
| Indiana | 50.0 | 9.0 |
| Massachusetts | 50.0 | 1 1.0 |
| Texas | 49.0 | 22.0 |
| Virginia | 49.0 | 1 1.0 |
| Oregon | 36.0 | 13.0 |
| Minnesota | 29.0 | 4.0 |
| Mississippi | 23.0 | 16.0 |
| Alabama | 17.0 | 9.0 |
| Utah | 16.0 | 6.0 |
| South Carolina | 15.0 | 3.0 |
| Arizona | 14.0 | 6.0 |
| District Of Columbia | 14.0 | 2.0 |
| Louisiana | 12.0 | 1.0 |

New Status Closed Open

State

|  |  |  |
| --- | --- | --- |
| New Mexico | 1 1.0 | 4.0 |
| Connecticut | 9.0 | 3.0 |
| West Virginia | 8.0 | 3.0 |
| New Hampshire | 8.0 | 4.0 |
| Delaware | 8.0 | 4.0 |
| Arkansas | 6.0 | NaN |
| New York | 6.0 | NaN |
| Kentucky | 4.0 | 3.0 |
| Maine | 3.0 | 2.0 |
| North Carolina | 3.0 | NaN |
| Missouri | 3.0 | 1.0 |
| Ohio | 3.0 | NaN |
| Vermont | 2.0 | 1.0 |
| District of Columbia | 1.0 | NaN |
| Rhode Island | 1.0 | NaN |
| Iowa | 1.0 | NaN |
| Kansas | 1.0 | 1.0 |
| Nevada | 1.0 | NaN |
| Montana | 1.0 | NaN |

# From the stacked bar and above we can say that Georgia state has the maximum complaints

#state\_group=data. groupby( [ "State ", "New Status . count ()

## #state\_group. reset\_ index( ) . sort\_ values (by=[ "City ", "New Status "l, ascending=FaLse)

|  |  |
| --- | --- |
| In [ 203...  In [ 204...  Out [204] : | counts=data["Comp1aint types"] . value\_counts() percentage-data [ "Complaint types" ] . value\_counts (normalize-True) .mu1(100) . round(2)       perc\_of\_complaints=pd.concat( [counts, percentage], axis-I, keys-I "counts", "percentage"] )  Customer Care Call complaints are of more than 50%  perc\_of\_complaints  counts percentage  Complaint types |

 Customer Care Call 1 1 19 50.31%

Internet 1 105 49.69% state has the highest percentage of unresolved complaints

In [ 205... state\_group. sort\_values ( by: "Open " , ascending-False)

out [205] : New Status Closed Open

State

|  |  |  |
| --- | --- | --- |
| Georgia | 208.0 | 80.0 |
| California | 159.0 | 61.0 |
| Tennessee | 96.0 | 47.0 |
| Florida | 201.0 | 39.0 |
| Illinois | 135.0 | 29.0 |
| Michigan | 92.0 | 23.0 |
| Washington | 75.0 | 23.0 |
| Colorado | 58.0 | 22.0 |
| Texas | 49.0 | 22.0 |
| Pennsylvania | 1 10.0 | 20.0 |
| New Jersey | 56.0 | 19.0 |
| Mississippi | 23.0 | 16.0 |
| Maryland | 63.0 | 15.0 |
| Oregon | 36.0 | 13.0 |
| Massachusetts | 50.0 | 1 1.0 |
| Virginia | 49.0 | 1 1.0 |
| Alabama | 17.0 | 9.0 |
| Indiana | 50.0 | 9.0 |
| Arizona | 14.0 | 6.0 |
| Utah | 16.0 | 6.0 |
| Delaware | 8.0 | 4.0 |
| New Hampshire | 8.0 | 4.0 |
| New Mexico | 1 1.0 | 4.0 |
| Minnesota | 29.0 | 4.0 |

New Status Closed Open

State

|  |  |  |
| --- | --- | --- |
| South Carolina | 1 5.0 | 3.0 |
| Connecticut | 9.0 | 3.0 |
| West Virginia | 8.0 | 3.0 |
| Kentucky | 4.0 | 3.0 |
| District Of Columbia | 14.0 | 2.0 |
| Maine | 3.0 | 2.0 |
| Louisiana | 12.0 | 1.0 |
| Vermont | 2.0 | 1.0 |
| Missouri | 3.0 | 1.0 |
| Kansas | 1.0 | 1.0 |
| Arkansas | 6.0 | NaN |
| District of Columbia | 1.0 | NaN |
| Iowa | 1.0 | NaN |
| Montana | 1.0 | NaN |
| Nevada | 1.0 | NaN |
| New York | 6.0 | NaN |
| North Carolina | 3.0 | NaN |
| Ohio | 3.0 | NaN |
| Rhode Island | 1.0 | NaN |

From above table georgia has highest compalints in open state i.e, unresolved complaints