In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Task 1: Import data into Python environment

In [2]:

```
comcast_tele_consumer=pd.read_csv('https://rahul-kumar.s3.ap-south-1.amazonaws.com/Data
```

In [3]:

```
# check whether data imported or not
comcast_tele_consumer.head()
```

Out[3]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Statu
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closec
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Closec
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Closec
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	Oper
4	307175	Comcast not working and no service to boot	26-05-15	26-May-15	1:25:26 PM	Internet	Acworth	Georgia	30101	Solvec

In [4]:

```
# Step 1: Check the duplicate columns or Variables with duplicate names and delete such
comcast_tele_consumer.shape
```

Out[4]:

```
(2224, 11)
```

In [5]:

```
comcast_tele_consumer.columns
```

Out[5]:

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

In [6]:

```
# Hence no duplicate names found we will go to step 2
```

In [7]:

```
# Step 2: Check for 0 columns or single value
comcast_tele_consumer.describe()
```

Out[7]:

	Zip code
count	2224.000000
mean	47994.393435
std	28885.279427
min	1075.000000
25%	30056.500000
50%	37211.000000
75%	77058.750000
max	99223.000000

In [8]:

```
# No Zero Columns or single Value Found, so we proceed to step 3
```

In [9]:

```
# Step 3: Missing value Treatment
comcast_tele_consumer.isnull().sum().sort_values(ascending=False)
```

Out[9]:

```
Ticket #                       0
Customer Complaint             0
Date                           0
Date_month_year                0
Time                           0
Received Via                   0
City                           0
State                          0
Zip code                       0
Status                         0
Filing on Behalf of Someone    0
dtype: int64
```

In [10]:

```
# No Missing value found so we proceed to step
```

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

In [11]:

```
comcast_tele_consumer.dtypes
```

Out[11]:

```
Ticket #                       object
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 [12]:

```
# Add Date Month year with Time and save it into Date_Full
comcast_tele_consumer["Date_Full"] = comcast_tele_consumer["Date_month_year"] + ' ' +
```

In [13]:

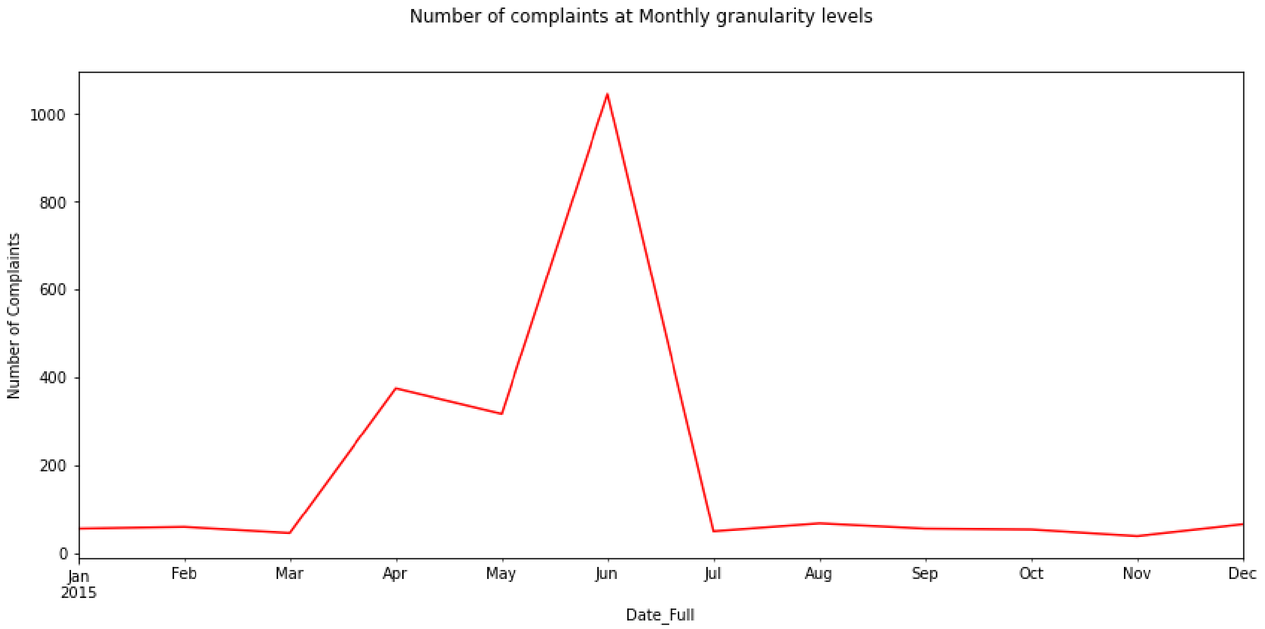
```
#Convert Date_Full and Date_month_year to Datetime Format
comcast_tele_consumer["Date_Full"] = pd.to_datetime(comcast_tele_consumer["Date_Full"])
comcast_tele_consumer["Date_month_year"] = pd.to_datetime(comcast_tele_consumer["Date_m
comcast_tele_consumer_monthly = comcast_tele_consumer.set_index(comcast_tele_consumer["
```

In [14]:

```
# Provide the trend chart for the number of complaints at monthly granularity levels.
#Increase Graph Size
plt.figure(figsize=(14,6))
plt.suptitle('Number of complaints at Monthly granularity levels')
plt.ylabel('Number of Complaints')
comcast_tele_consumer_monthly.groupby(pd.Grouper(freq="M")).size().plot(color='red')
```

Out[14]:

```
<AxesSubplot:xlabel='Date_Full', ylabel='Number of Complaints'>
```

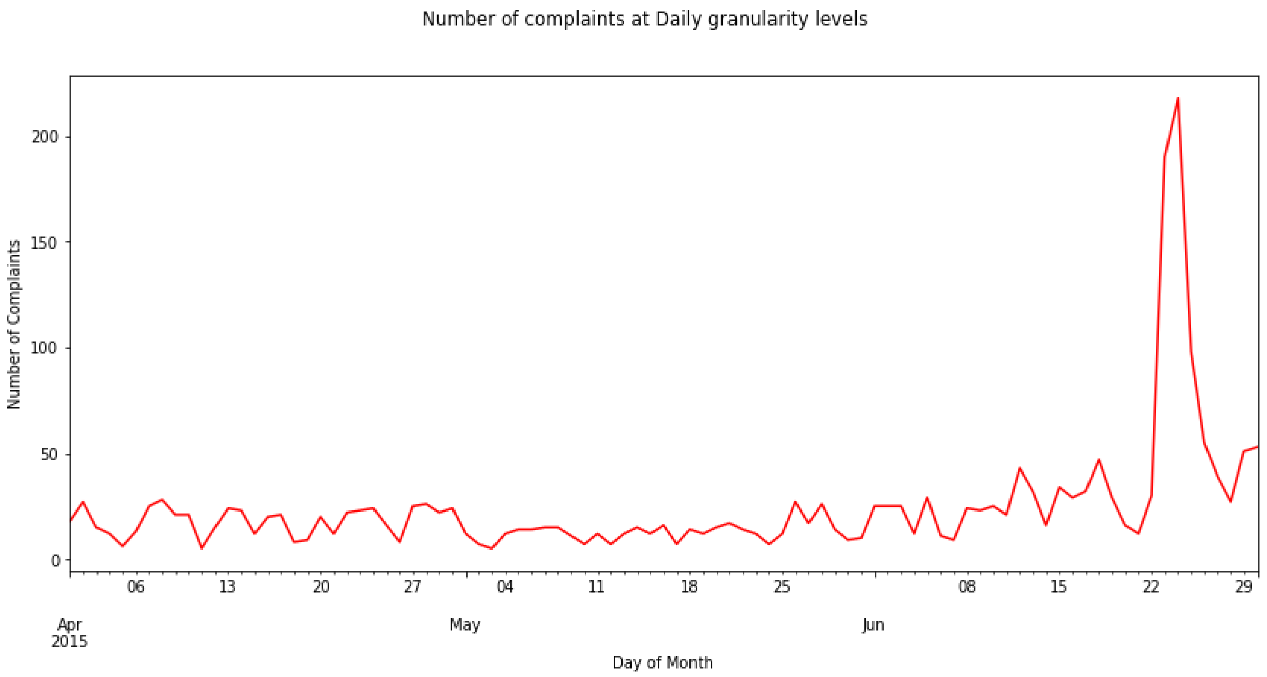


In [15]:

```
# Provide the trend chart for the number of complaints at daily granularity levels.
comcast_tele_consumer['Day of Month'] = pd.to_datetime(comcast_tele_consumer['Date'])
comcast_tele_consumer_daily = comcast_tele_consumer.set_index(comcast_tele_consumer["Da
#Increase Graph Size
plt.figure(figsize=(14,6))
plt.suptitle('Number of complaints at Daily granularity levels')
plt.ylabel('Number of Complaints')
comcast_tele_consumer_daily.groupby(pd.Grouper(freq="D")).size().plot(color='red')
```

Out[15]:

```
<AxesSubplot:xlabel='Day of Month', ylabel='Number of Complaints'>
```



Task 3: Provide a table with the frequency of complaint types.

In [16]:

```
# To get the frequency of complaint types first we have to see all complaint types and
# Incomplete data so that we can make analytics better
comcast_tele_consumer_complaint_type = comcast_tele_consumer["Customer Complaint"].valu
```

In [17]:

```
comcast_tele_consumer_complaint_type.head(10)
```

Out[17]:

```
Comcast                       83
Comcast Internet              18
Comcast Data Cap              17
comcast                       13
Data Caps                     11
Comcast Data Caps             11
Comcast Billing               11
Unfair Billing Practices       9
Data Cap                       8
Comcast internet               8
Name: Customer Complaint, dtype: int64
```

In [18]:

```
# Better to convert all data into uper case or sentence case so duplicate value will sh
comcast_tele_consumer_complaint_type=comcast_tele_consumer["Customer Complaint"].str.up
```

In [19]:

```
# Data is huge so we have showed only top 25 Complaint Types. We can show clearly in th
# COMCAST INTERNET are the Highest top 3 complaint types
comcast_tele_consumer_complaint_type.head(25)
```

Out[19]:

```
COMCAST                        102
COMCAST DATA CAP                30
COMCAST INTERNET                29
COMCAST DATA CAPS               21
COMCAST BILLING                 18
COMCAST SERVICE                 15
INTERNET SPEED                  15
UNFAIR BILLING PRACTICES        13
```

```
DATA CAPS                    13
DATA CAP                     12
COMCAST/XFINITY              11
COMCAST COMPLAINT            11
COMCAST INTERNET SERVICE     10
BILLING                       9
BILLING ISSUES                8
COMCAST CABLE                 5
COMCAST BILLING COMPLAINT     5
COMCAST ISSUES                5
COMPLAINT AGAINST COMCAST     5
SLOW INTERNET                 5
SERVICE ISSUES                5
INTERNET                      5
INTERNET SERVICE              5
COMCAST BILLING PRACTICES     5
COMCAST BILLING ISSUES        4
Name: Customer Complaint, dtype: int64
```

Task 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.

In [20]:

```
# Check how many unique values are under Status Column
comcast_tele_consumer['Status'].unique()
```

Out[20]:

```
array(['Closed', 'Open', 'Solved', 'Pending'], dtype=object)
```

In [21]:

```
# Convert as per Instruction (Task 4) into New Column without changing the main data so
# in Future
comcast_tele_consumer['New_Status']= ["Open" if Status=="Open" or Status=="Pending" els
                                      "Closed" for Status in comcast_tele_consumer["Sta
```

In [22]:

```
# Check whether all status updated or not
comcast_tele_consumer['New_Status'].unique()
```

Out[22]:

```
array(['Closed', 'Open'], dtype=object)
```

In [23]:

```
comcast_tele_consumer_status_by_state = pd.crosstab(comcast_tele_consumer.State,comcast
```

In [24]:

```
comcast_tele_consumer_status_by_state
```

Out[24]:

New_Status	Closed	Open
State		
Alabama	17	9
Arizona	14	6
Arkansas	6	0
California	159	61
Colorado	58	22

New_Status	Closed	Open
State		
Connecticut	9	3
Delaware	8	4
District Of Columbia	14	2
District of Columbia	1	0
Florida	201	39
Georgia	208	80
Illinois	135	29
Indiana	50	9
Iowa	1	0
Kansas	1	1
Kentucky	4	3
Louisiana	12	1
Maine	3	2
Maryland	63	15
Massachusetts	50	11
Michigan	92	23
Minnesota	29	4
Mississippi	23	16
Missouri	3	1
Montana	1	0
Nevada	1	0
New Hampshire	8	4
New Jersey	56	19
New Mexico	11	4
New York	6	0
North Carolina	3	0
Ohio	3	0
Oregon	36	13
Pennsylvania	110	20
Rhode Island	1	0
South Carolina	15	3
Tennessee	96	47

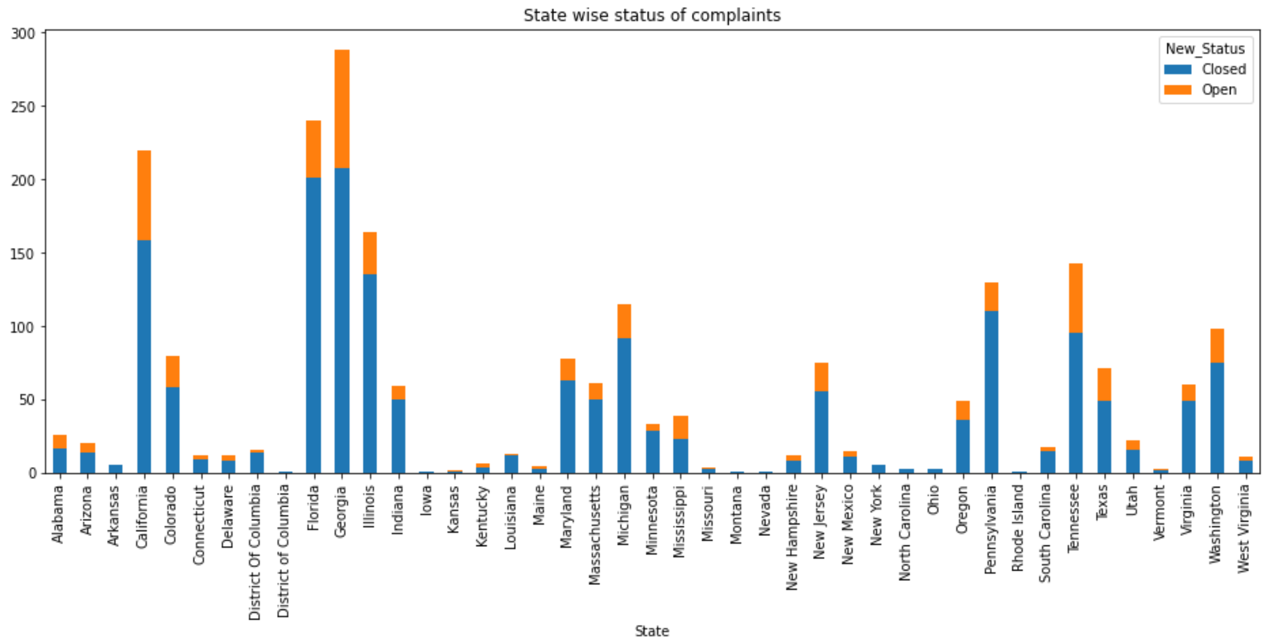
New_Status	Closed	Open
State		
Texas	49	22
Utah	16	6
Vermont	2	1
Virginia	49	11
Washington	75	23
West Virginia	8	3

In [25]:

```
pd.crosstab(comcast_tele_consumer.State,comcast_tele_consumer.New_Status).plot(kind='ba
                                                       stacked=True,
                                                       title='State wise status of compl
```

Out[25]:

```
<AxesSubplot:title={'center':'State wise status of complaints'}, xlabel='State'>
```



In [26]:

```
# Which state has the maximum complaints - Georgia has maximum number of complaints
```

Task 5: Which state has the highest percentage of unresolved complaints

In [27]:

```
comcast_tele_consumer_unresolved_complaints = comcast_tele_consumer[comcast_tele_consum
```

In [28]:

```
comcast_tele_consumer_unresolved_complaint_count = comcast_tele_consumer_unresolved_com
comcast_tele_consumer_unresolved_complaint_count
```

Out[28]:

```
Georgia                 80
California              61
Tennessee               47
Florida                 39
```

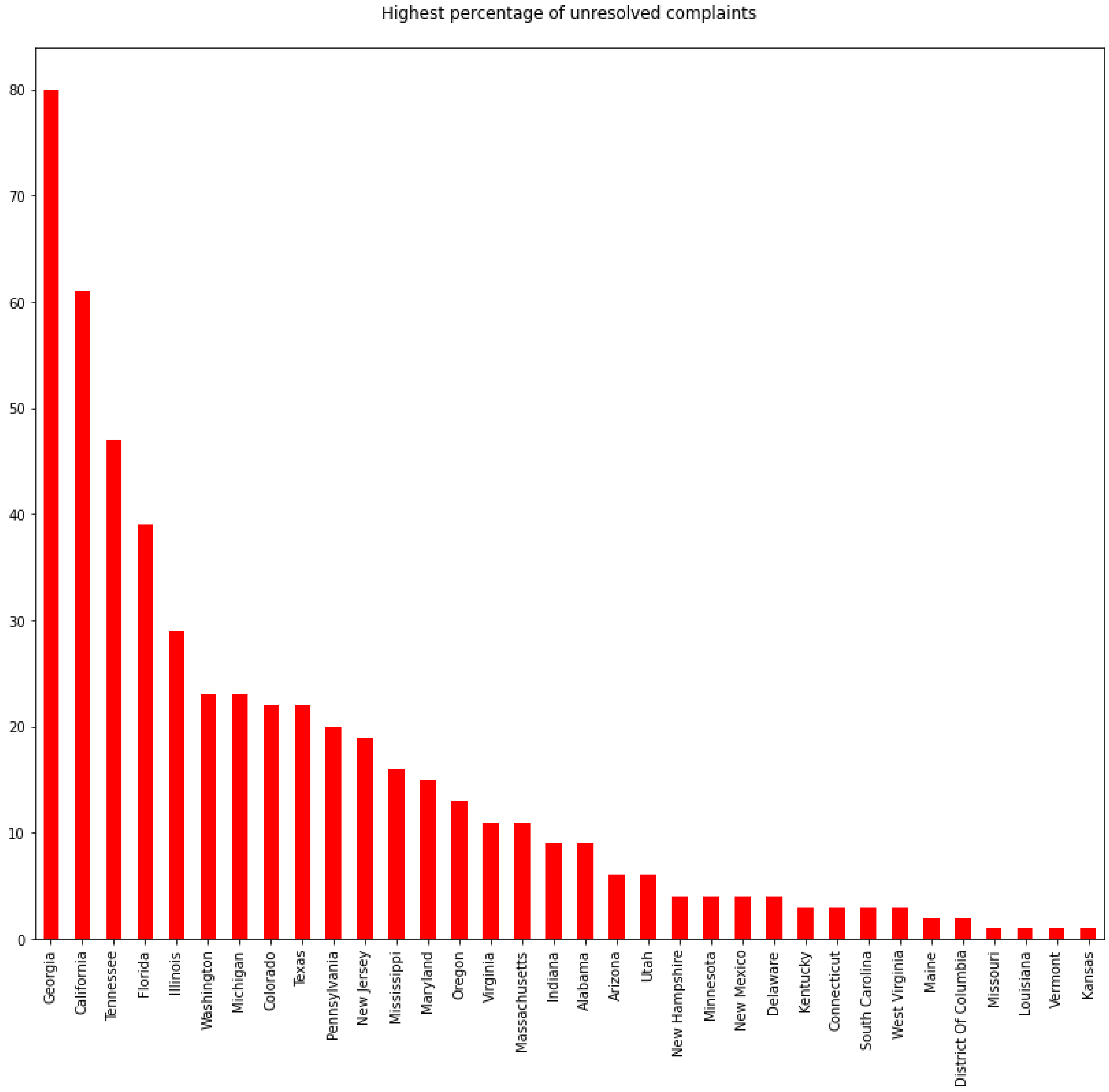
```
Illinois                29
Washington              23
Michigan                23
Colorado                22
Texas                   22
Pennsylvania            20
New Jersey              19
Mississippi             16
Maryland                15
Oregon                  13
Virginia                11
Massachusetts           11
Indiana                  9
Alabama                  9
Arizona                  6
Utah                     6
New Hampshire            4
Minnesota                4
New Mexico               4
Delaware                 4
Kentucky                 3
Connecticut              3
South Carolina           3
West Virginia            3
Maine                    2
District Of Columbia     2
Missouri                 1
Louisiana                1
Vermont                  1
Kansas                   1
Name: State, dtype: int64
```

In [29]:

```
# Georgia has the Highest Number of unresolved complaints
# Show this by Bar Graph
comcast_tele_consumer_unresolved_complaint_count.plot(kind='bar',figsize=(14,12),color=
plt.title('Highest percentage of unresolved complaints\n')
```

Out[29]:

```
Text(0.5, 1.0, 'Highest percentage of unresolved complaints\n')
```



Highest percentage of unresolved complaints

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

In [30]:

```
# Check unique values in Received Via Column
comcast_tele_consumer['Received Via'].unique()
```

Out[30]:

```
array(['Customer Care Call', 'Internet'], dtype=object)
```

In [31]:

```
# So there are only two values in that columns so no need to short we can directly proc
```

In [32]:

```
comcast_tele_consumer.New_Status.value_counts()
```

Out[32]:

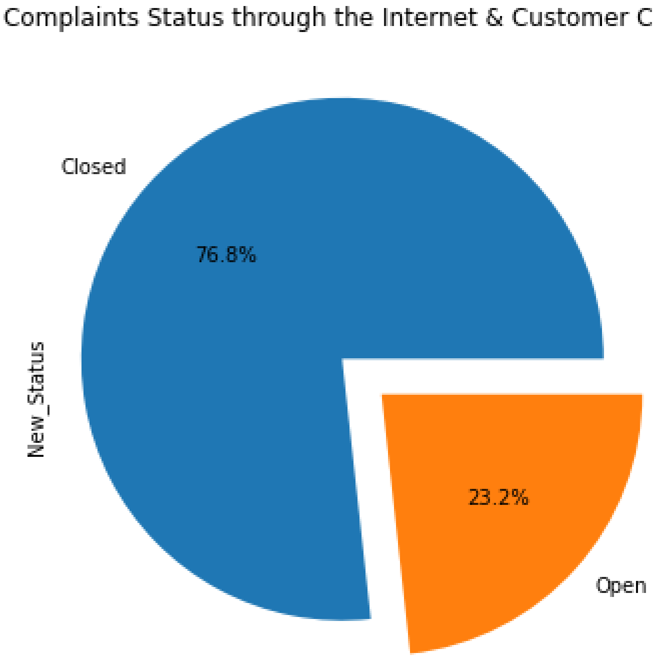
```
Closed    1707
Open       517
Name: New_Status, dtype: int64
```

In [33]:

```
# Used autopct='%1.1f%%'  to show percentage under the pie chart
myexplode = [0.2, 0]
plt.title('Complaints Status through the Internet & Customer Care Calls\n')
comcast_tele_consumer.New_Status.value_counts().plot(kind='pie',explode = myexplode,aut
                        figsize = (14,6))
```

Out[33]:

```
<AxesSubplot:title={'center':'Complaints Status through the Internet & Customer Care Cal
ls\n'}, ylabel='New_Status'>
```


Complaints Status through the Internet & Customer Care Calls

Thank You