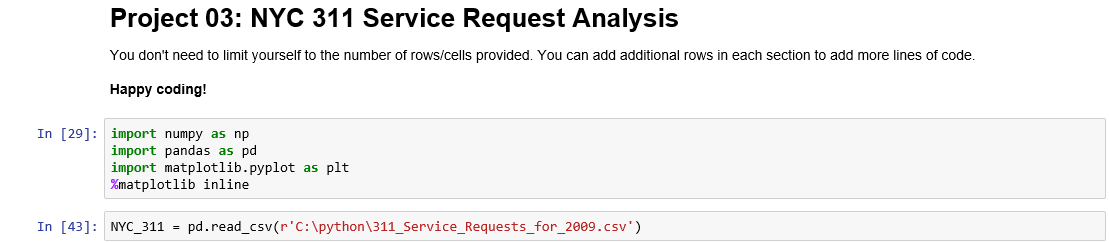
**Python - Real Time Project**

**New York City 311 calls**

Perform a service request data analysis of New York City 311 calls. You will focus on the data wrangling techniques to understand the pattern in the data and also visualize the major complaint types.

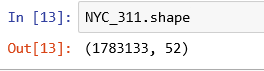
**ANSWERS:**

* **Import a 311 NYC service request**



The above command shows how we can import a dataset into the jupyter notebook. We have imported the 311 NYC service request dataset here, using the command ‘*pd.read.csv(file path)*’.

* **Basic data exploratory analysis** 
  + **Explore data**

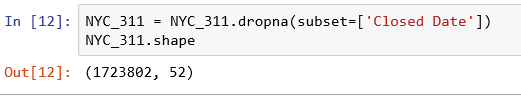


The above command shows the dimensions of the array of the dataset, which is denoted by *‘.shape*’. Here we can see that the dataset has **1783133** rows/observations and **52** columns/variables. Below is a table showing the variables involved in the given dataset:

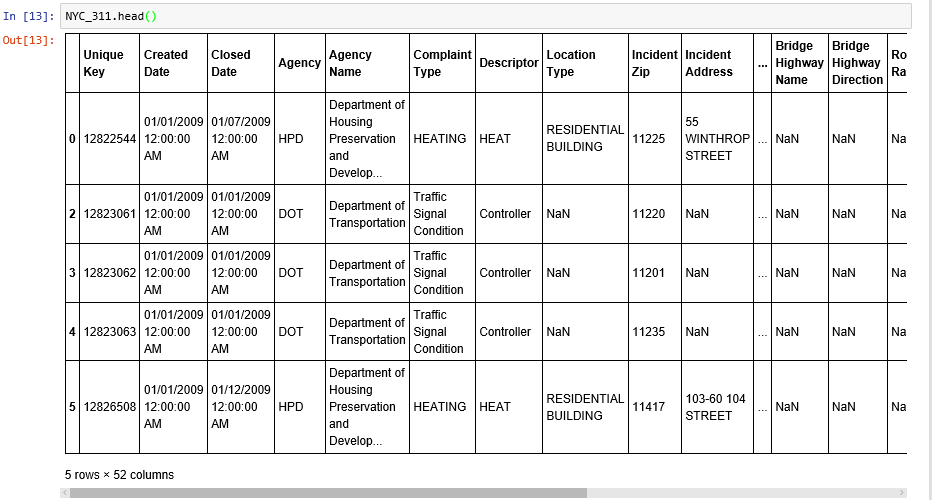
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute names** | | | |
| Unique Key  Created Date  Closed Date  Agency  Agency Name  Complaint Type  Descriptor  Location Type  Incident Zip  Incident Address  Street Name  Cross Street 1  Cross Street 2  Intersection Street 1 | Intersection Street 2  Address Type  City  Landmark  Facility Type  Status  Due Date  Resolution Action Updated Date  Community Board  Borough  X Coordinate (State Plane)  Y Coordinate (State Plane) | Park Facility Name  Park Borough  School Name  School Number  School Region  School Code  School Phone Number  School Address  School City  School State  School Zip  School Not Found  School or Citywide Complaint | Vehicle Type  Taxi Company Borough  Taxi Pick Up Location  Bridge Highway Name  Bridge Highway Direction  Road Ramp  Bridge Highway Segment  Garage Lot Name  Ferry Direction  Ferry Terminal Name  Latitude  Longitude  Location |

* + **Find patterns**

Seeing the dataset, we can observe that there are few observations for which close date is empty, meaning that those cases are still open/pending. If we narrow down the dataset to show only the cases which are open we get the following:

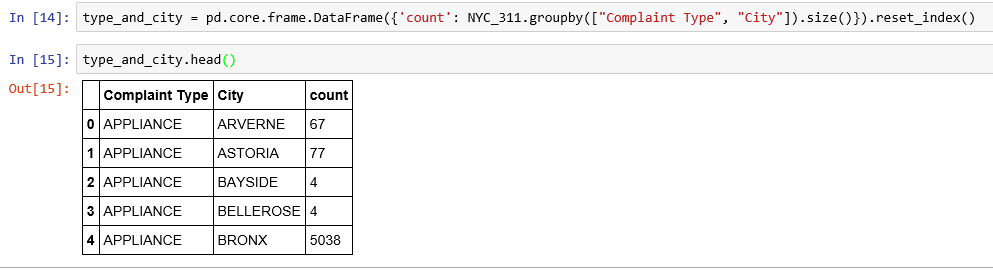


What we did above is that we narrowed down the dataset to show only those cases which are either open/pending. The function we used here was ‘*dropna(subset=[variable])*’. After doing this, we input the dimensions of the dataset again, and the dataset is now narrowed down to **1723802** rows/observations and the same **52** columns/variables.



The above command and results shows the **first 5 rows** of the filtered dataset. This is done by using the ‘*head()*’ function.

* + **Display the complaint type and city together**

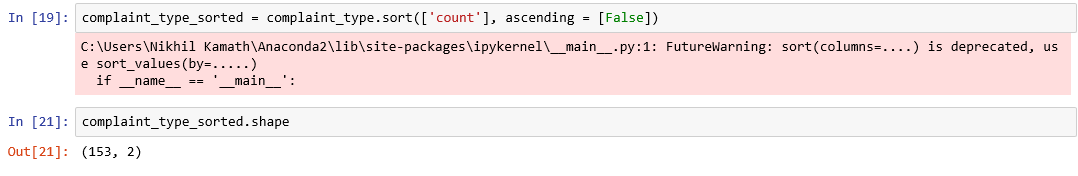


The above result shows records where the complaint type and city have been put together. Here, the groupby function has been used to group the set by complaint type and city, and the count function gives the number of times the same complaint type and city have been reported.

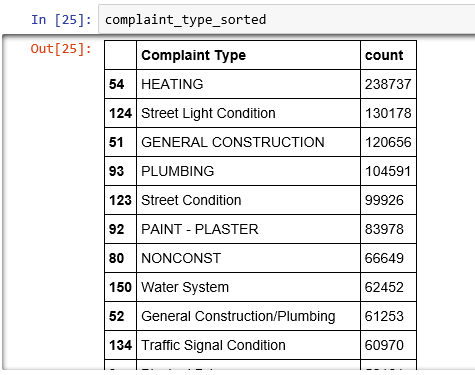
* **Find major complaint types**
  + **Find the top 10 complaint types**



Firstly, we need to filter the dataset to sort the cases as per the complaint type. The source code and the result is shown above.



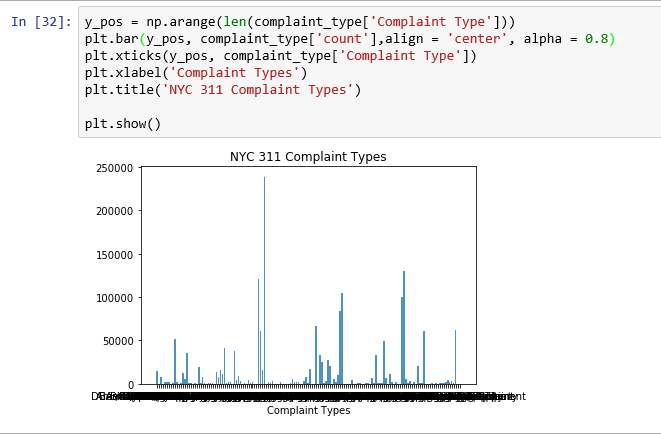
Next, we need to sort the filtered data in descending order as per the complaint type count. Here, the ‘*sort*’ function is used, and the ascending order is put as ‘False’ as we want the data to be sorted in descending order. After sorting, we can see though the shape function that the data now has **153** observations and **2** variables.



The final outcome of the data of top 10 complaint types sorted as per descending order is tabulated above. The visualized pie chart of the top 10 complaint types is also available which is shown below.



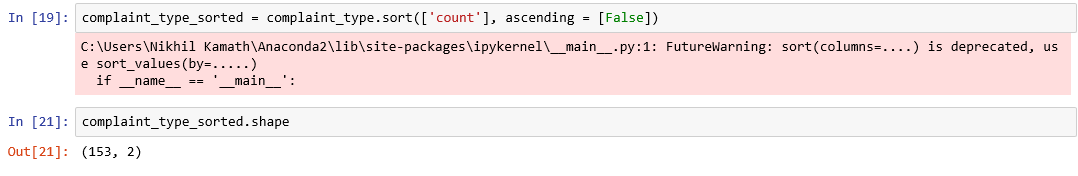
* + **Plot a bar graph of count vs. complaint types**



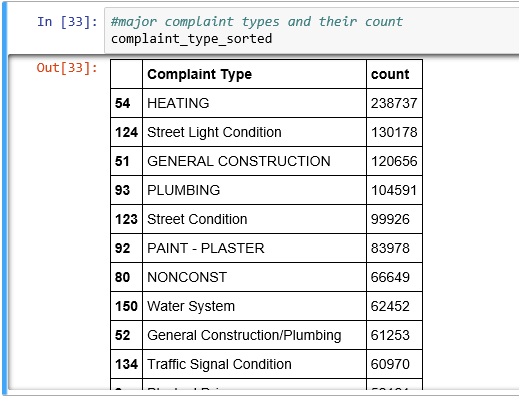
* **Visualize the complaint types**
  + **Display the major complaint types and their count**

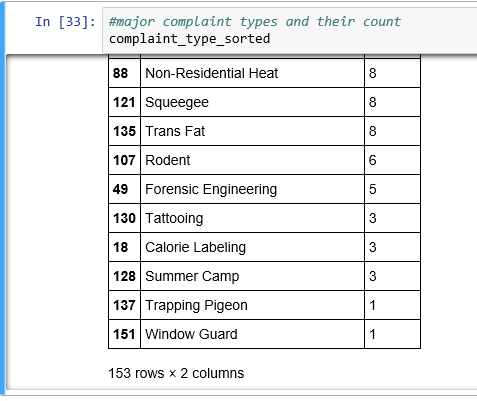


Firstly, we need to filter the dataset to sort the cases as per the complaint type. The source code and the result is shown above.



Next, we need to sort the filtered data in descending order as per the complaint type count. Here, the ‘*sort*’ function is used, and the ascending order is put as ‘False’ as we want the data to be sorted in descending order. After sorting, we can see though the shape function that the data now has **153** observations and **2** variables.





The above table shows the major complaint types and the count of the complaint types. They are also visualized below.

