**DATA EXPLORING**

**Introduction:**

Data Cleaning is the process of correcting and deleting inaccurate records from a data base. It also consists of identifying and replacing incomplete, inaccurate, irrelevant data and records. The importance of Data Cleaning is to fix the poor quality data.

**Data Migration:**

It is the process of extracting data from one location and transferring to another.

**Benefits of Data Cleaning:**

* Streamlined business practices
* Increased productivity
* Faster sales cycle
* Better decisions

**Types of data issues:**

Duplicate data – two or more identical data

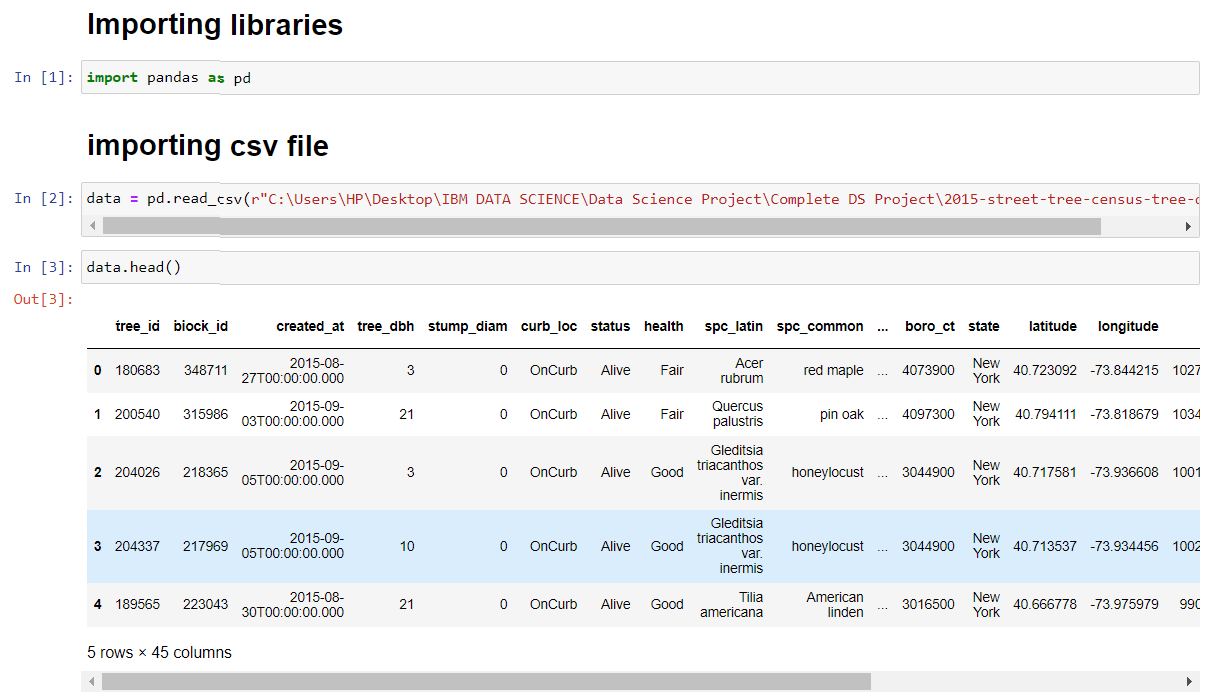
Conflicting data - When there are same records with different attributes, it means data is conflicting. For example, a company with different versions of addresses may cause delivery issues.

Incomplete data - The data that has missing attributes.

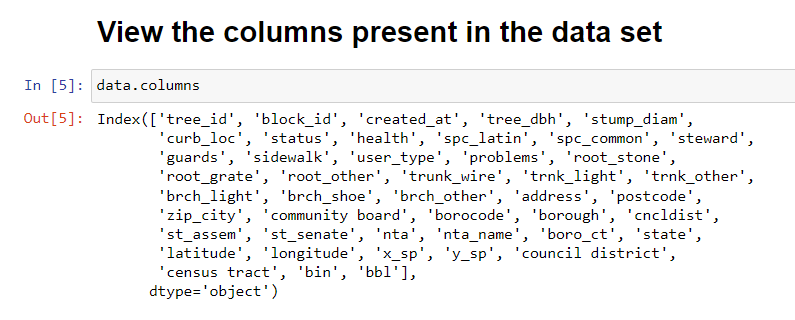
Invalid data - Data attributes are not conforming to standardization. For example, 9 digits phone number records rather than 10 digits.

**Step by step procedure on exploring a data set:**

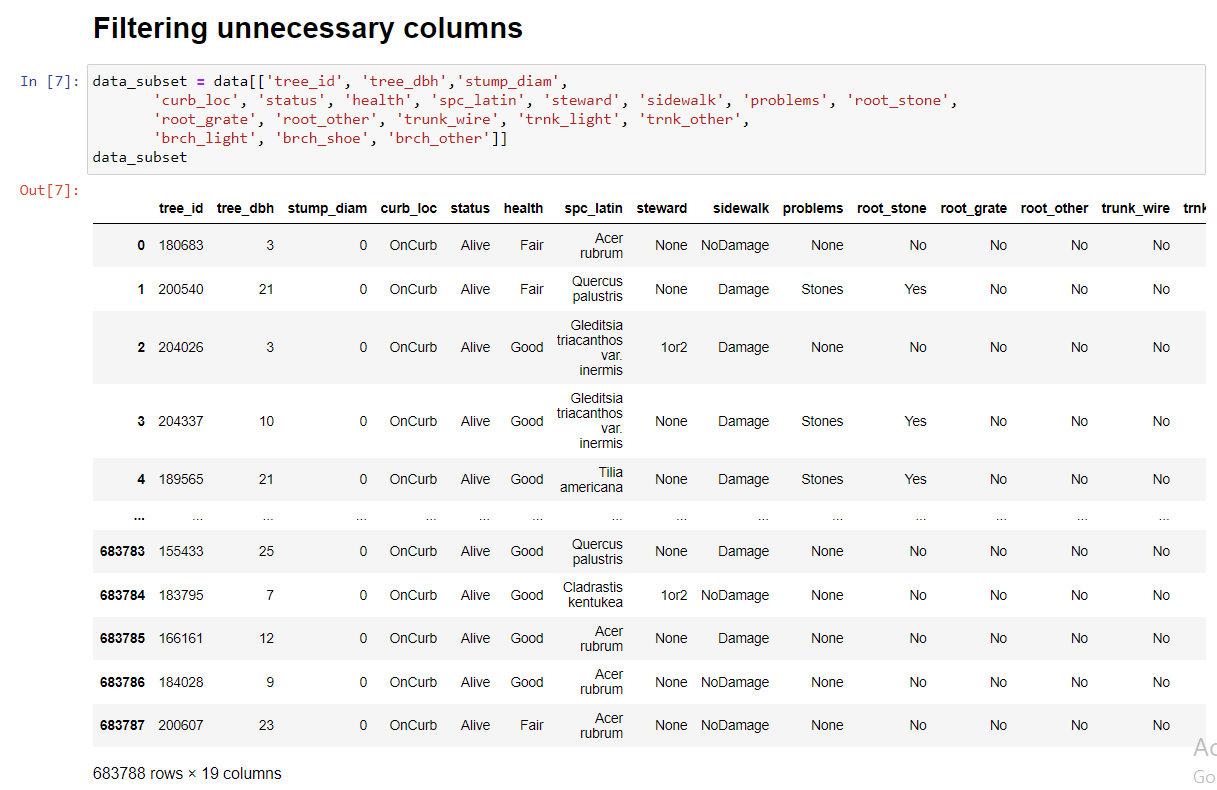
**Step 1:** Import necessary libraries and view the csv file where the data has to be explored. **Pandas** is the main library used to work with csv files.



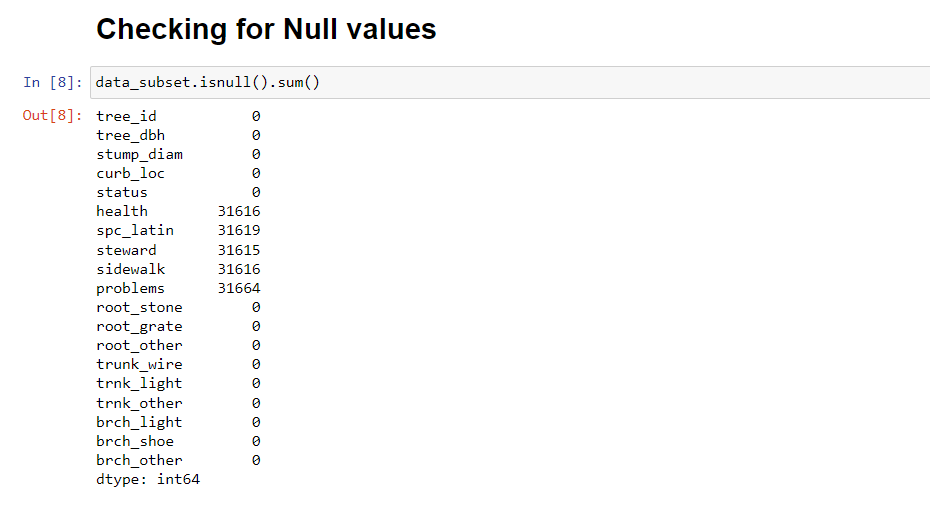
**Step 2:** First and foremost step in data exploring process is to get to know about the columns present in the data base.



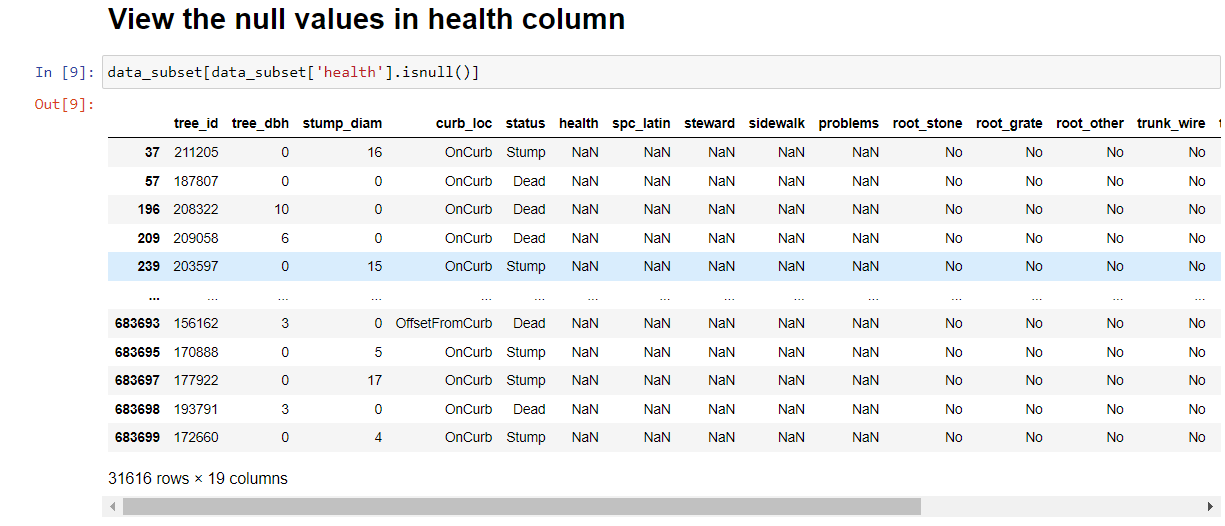
**Step 3:** After known about the columns in the data set, filter the columns that are not necessary for data analysis



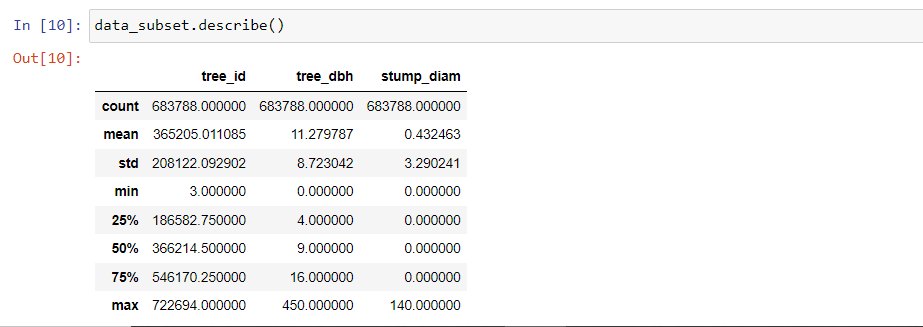
**Step 4:** After filtering the columns it could be easier to deal with the limited number of columns. Now it is necessary to check for the presence of null values in the data set. For this process a simple command is used **isnull().** The command **isnull().sum()** gives the total number of null values present in the data set. The above filtered data set is named as data\_subset.



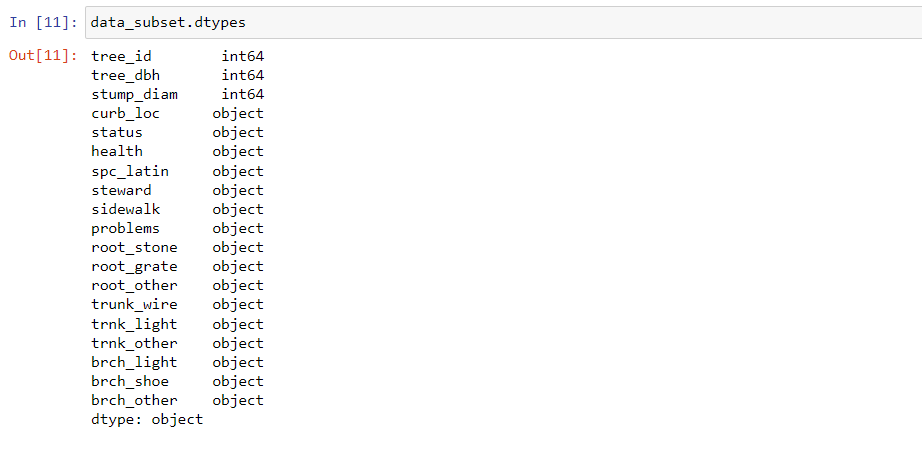
**Step 5:** From the above step we get to know how many values are missing from each column of the data set. Now view the null values present in the particular ‘health’ column using the following command **data\_subset [data\_subset ['health'].isnull()]** and the output of the dataset is as follows,



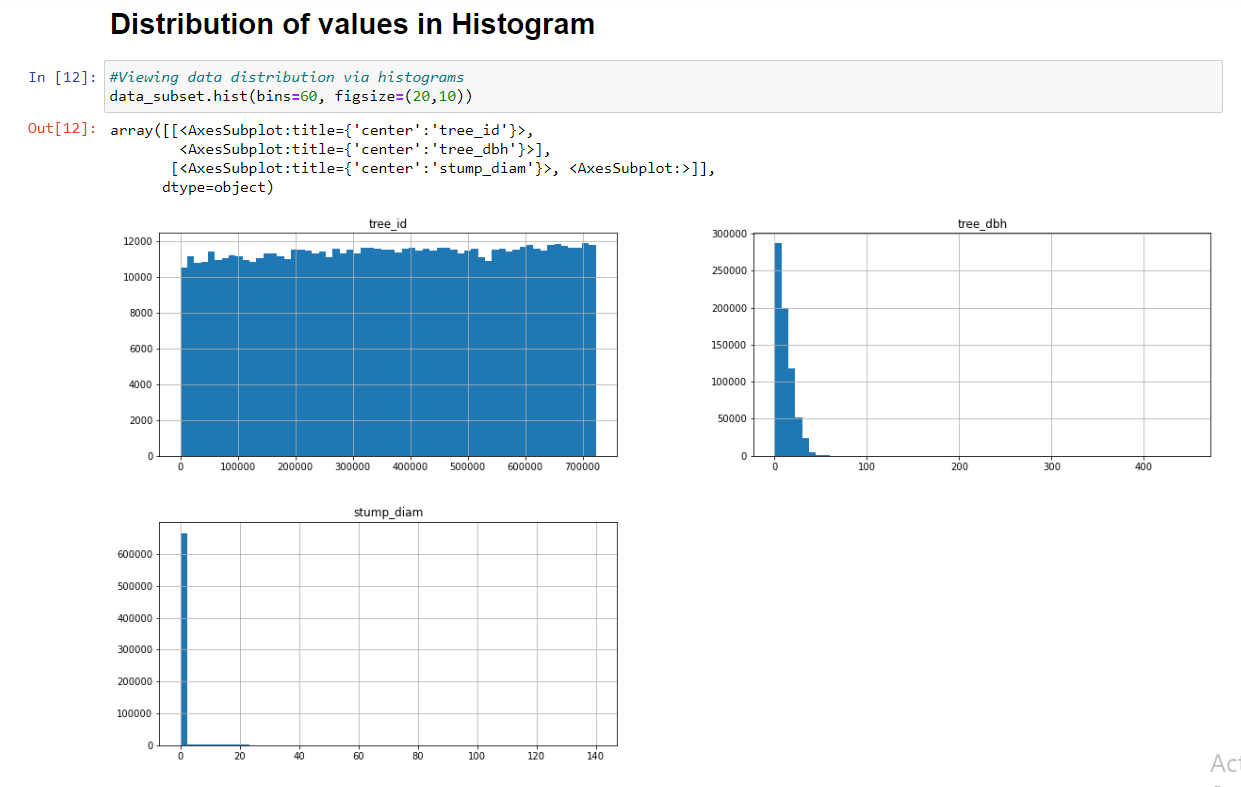
**Step 6:** Now view the count, average, standard value, minimum and maximum values for the data present in the subset using the function **describe()**

****

**Step 7:** Also view the data type of the values present in the filtered data set using the following command **data\_subset.dtypes**

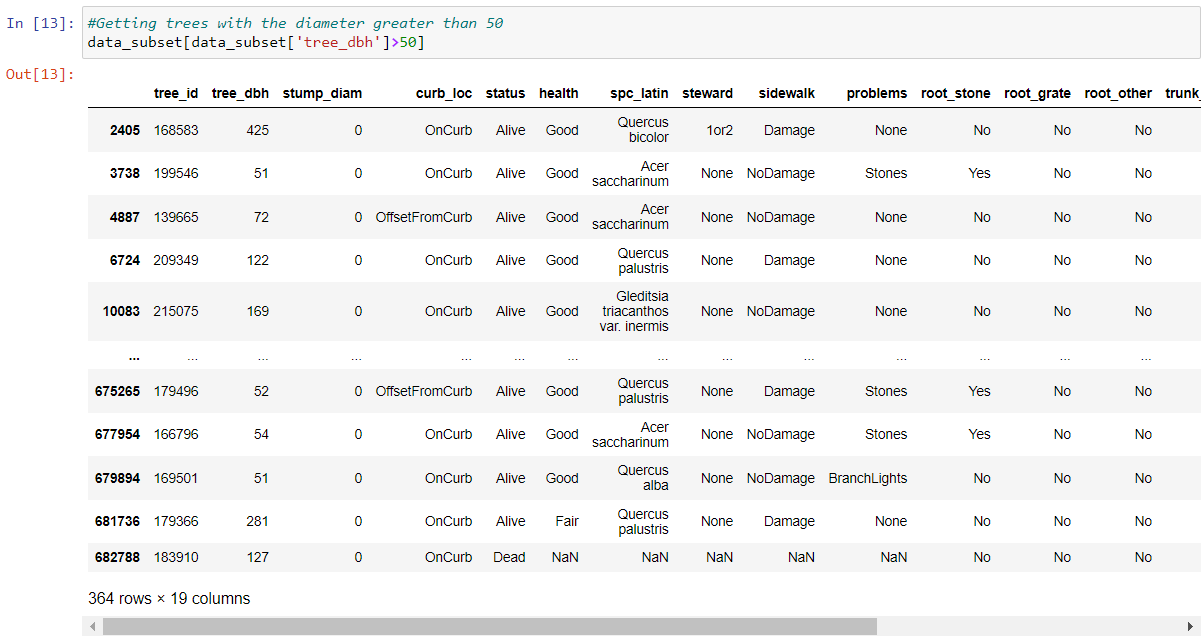


**Step 8:** Lets’ view how the data are distributed in the dataset through histograms. The command for creating a histogram chart is **data\_subset.hist(bins=60, figsize=(20,10))**

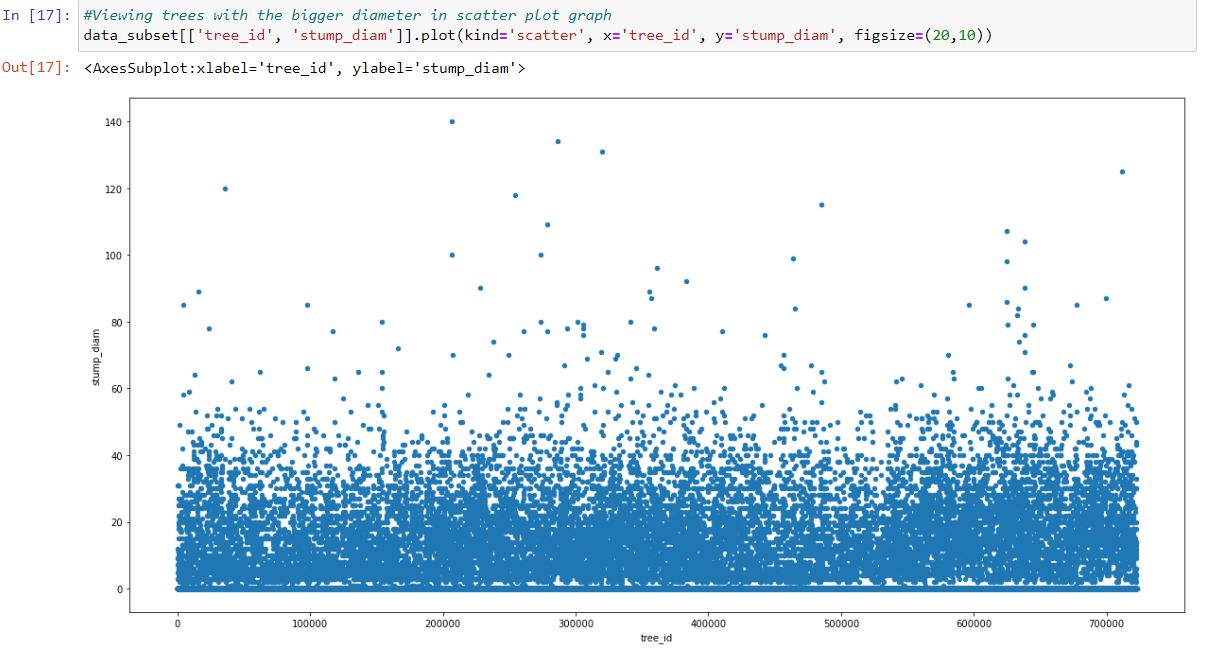


The above graph shows a pictorial representation of data distribution in the data set.

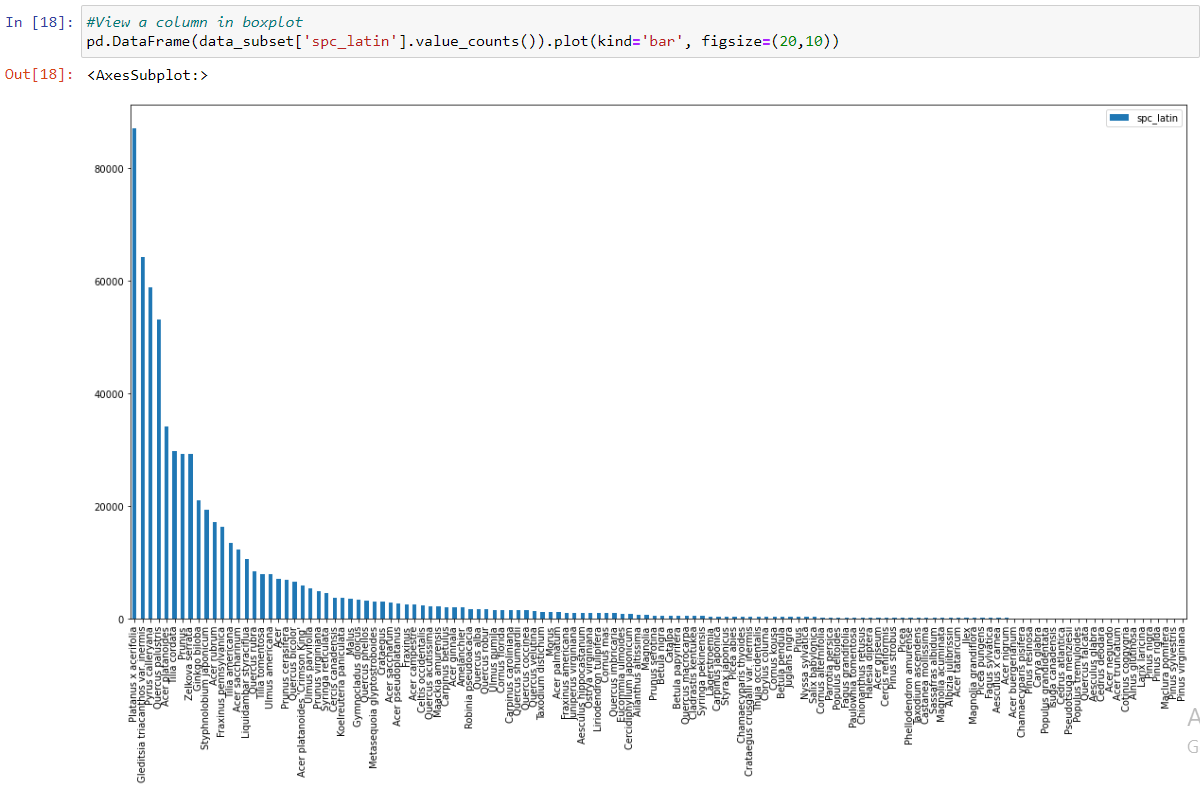
**Step 9:** We can also view the particular value of columns by filtering the value using an expression. The following expression filters the row values which has the diameter value of the tree greater than 50.



**Step 10:** Now we can view trees with the bigger diameter in scatter plot graphical presentation, which has made easier to view the data values. Here we have declared x-axis as tree\_id and y-axis as stump\_diam and also kind of the graph is declared as scatter hence we can view the values in scatterplot.

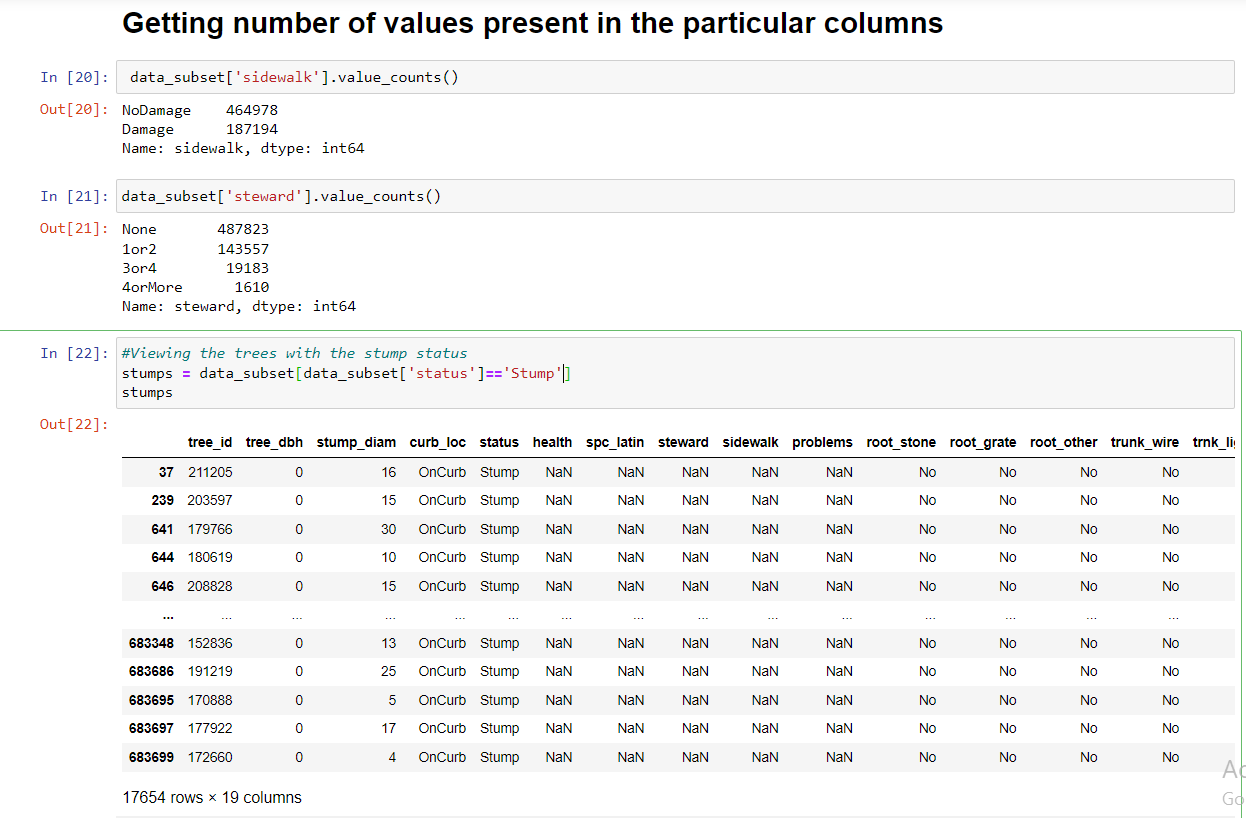


**Step 11:** Now we can view the number of value using the value\_counts() function and also a graphical representation in a single command of a specific column ‘spc\_latin’ and the command is as follows **pd.DataFrame(data\_subset['spc\_latin'].value\_counts()).plot(kind='bar', figsize=(20,10)).** The following screenshot gives the result of the graph.



The above graph shows how the data gets distributed in the data set individually.

**Step 12:** From this step we can get the number of values present in the particular columns.

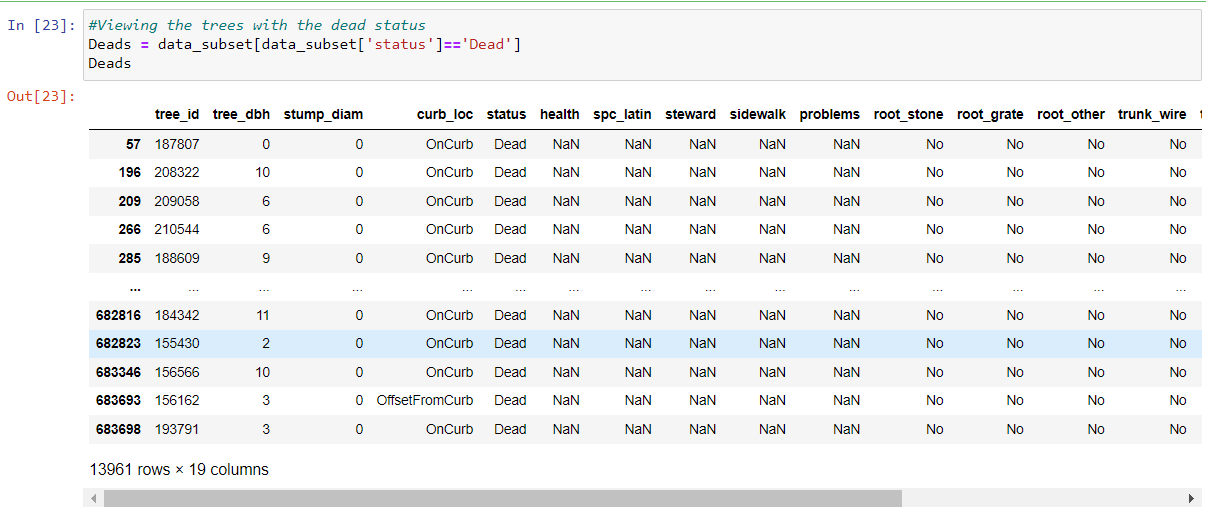


**Step 13:** We can also view the details of the trees that are dead by declaring the ‘Dead’ value to the ‘status’ column using the following command

Deads = data\_subset[data\_subset['status']=='Dead']

Deads

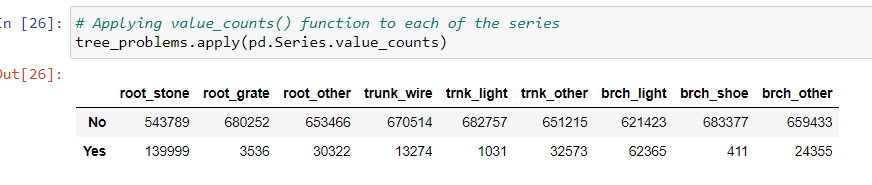
And the output is as follows,



# Step 14: From the above exploration steps we've get to know that the details of the dead tree are not updated for the columns like health, spc\_latin, steward, sidewalk, problems. Now explore remaining columns of the data set.

# 

# Step 15: Now apply value\_counts() function to each of the series and extract the values from the data set and the output is as follows.



**Conclusion:**

This Data Exploring process helps in reducing the size of the data by removing the irrelevant data, missing values, duplicates, replacing null values and finally we get to know the insight of the data for further exploration process.