Exploring the structure and content of datasets using Python

#Summary for Each data item: The below are the data items present in the Atlantic data set:
Basin, name, year, cyclone_of_the_year, date, time, status_of_system, latitude, longitude, max_sustained_wind and central_pressure.

CODE:

```
import pandas as pd
import os
atlantic_data=pd.read_csv("C:/Users/chandana/Documents/Atlantic.csv")
atlantic_data
atlantic_data.dtypes
```

- Summary for Year

```
atlantic_data.year.describe()
```

```
count    49105.000000
mean      1949.711944
std         44.618521
min       1851.000000
25%       1911.000000
50%       1956.000000
75%       1989.000000
max       2015.000000
Name: year, dtype: float64
```

- Summary for cyclone_of_the_year

```
In [5]: print(atlantic_data['cyclone_of_the_year'].describe())
```

```
count    49105.000000
mean         7.439487
std          5.226704
min          1.000000
25%          3.000000
50%          6.000000
75%         10.000000
max         31.000000
Name: cyclone_of_the_year, dtype: float64
```

- Summary for date

```
In [6]: print(atlantic_data['date'].describe())
```

```
count    4.910500e+04
mean     1.949802e+07
std      4.461850e+05
min      1.851062e+07
25%      1.911110e+07
50%      1.956093e+07
75%      1.989081e+07
max      2.015111e+07
Name: date, dtype: float64
```

- Summary for time

```
In [7]: print(atlantic_data['time'].describe())
```

```
count    49105.000000
mean       910.125975
std        671.043363
min          0.000000
25%        600.000000
50%       1200.000000
75%       1800.000000
max       2330.000000
Name: time, dtype: float64
```

- Summary for status_of_system

```
In [8]: print(atlantic_data['status_of_system'].describe())
```

```
count     49105
unique        9
top          TS
freq      17804
Name: status_of_system, dtype: object
```

- Summary for latitude

```
In [9]: print(atlantic_data['latitude'].describe())
```

```
count     49105
unique      597
top       28.0N
freq        299
Name: latitude, dtype: object
```

- Summary for Longitude

```
In [10]: print(atlantic_data['longitude'].describe())
```

```
count     49105
unique     1036
top       65.0W
freq        181
Name: longitude, dtype: object
```

- Summary for Max wind Sustained

```
In [11]: print(atlantic_data['max_sustained_wind'].describe())
```

```
count    49105.000000
mean        52.005091
std         27.681902
min        -99.000000
25%         35.000000
50%         45.000000
75%         70.000000
max        165.000000
Name: max_sustained_wind, dtype: float64
```

- Summary for Central Pressure

```
In [12]: print(atlantic_data['central_pressure'].describe())
```

```
count    18436.000000
mean       992.244250
std         19.113748
min        882.000000
25%        984.000000
50%        999.000000
75%       1006.000000
max       1024.000000
Name: central_pressure, dtype: float64
```

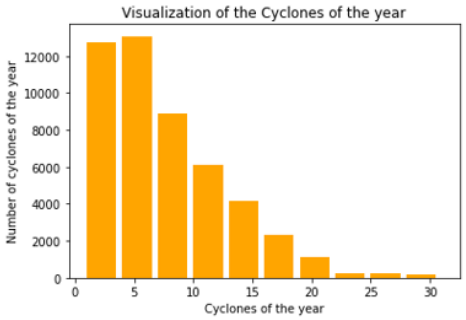
#Visualizations for the data items

import matplotlib.pyplot as plt

- Visualization for Cyclones of the Year

```
In [35]: import matplotlib.pyplot as plt

plt.hist(atlantic_data['cyclone_of_the_year'],width=2.5,color='orange')
plt.xlabel('Cyclones of the year',color='Black')
plt.ylabel('Number of cyclones of the year',color='Black')
plt.title('Visualization of the Cyclones of the year')
```

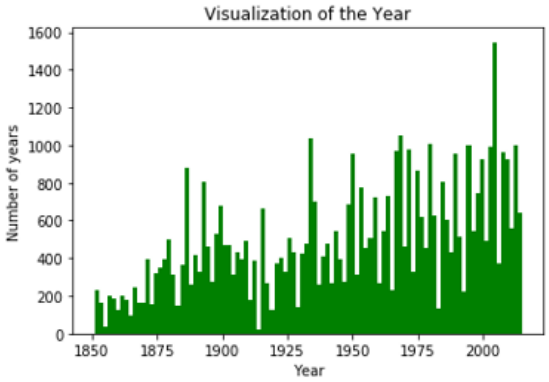
```
Out[35]: Text(0.5, 1.0, 'Visualization of the Cyclones of the year')
```



- Visualization for Year

```
In [15]: plt.hist(atlantic_data['year'],color='Green',bins=100)
         plt.xlabel('Year',color='Black')
         plt.ylabel('Number of years',color='Black')
         plt.title('Visualization of the Year')
```

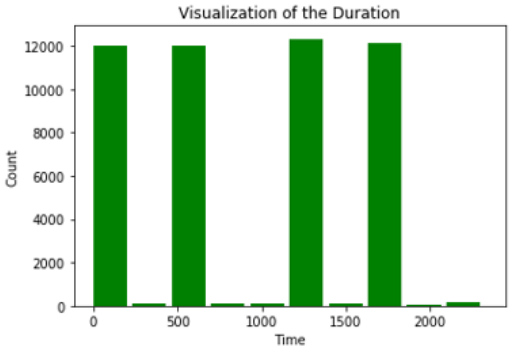
```
Out[15]: Text(0.5, 1.0, 'Visualization of the Year')
```



- Visualization for Time

```
plt.hist(atlantic_data['time'],color='Green',width=200)
plt.xlabel('Time',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the Duration')
```

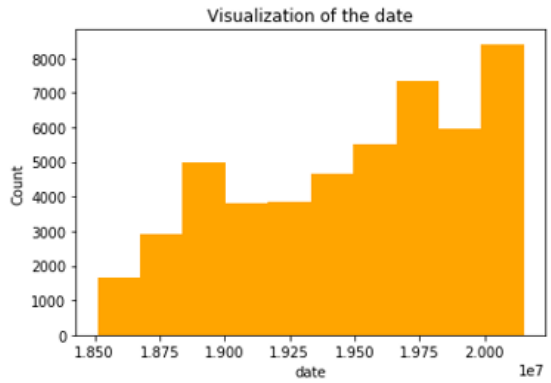
```
Text(0.5, 1.0, 'Visualization of the Duration')
```



- Visualization for Date

```
plt.hist(atlantic_data['date'],color='Orange')
plt.xlabel('date',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the date')
```

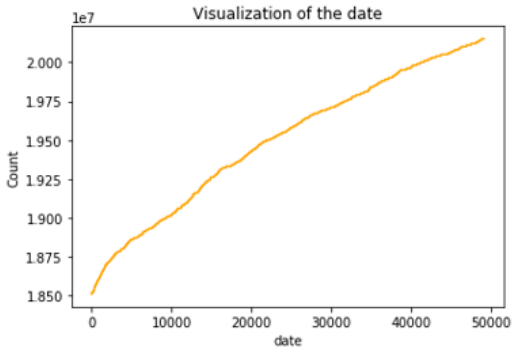
```
Text(0.5, 1.0, 'Visualization of the date')
```



- Additional Visualization for Date

```
plt.plot(atlantic_data['date'],color='Orange')
plt.xlabel('date',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the date')
```

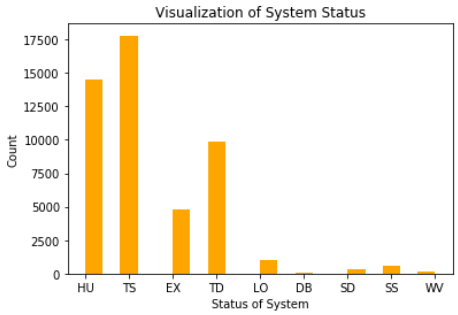
```
Text(0.5, 1.0, 'Visualization of the date')
```



- Visuazlization for Status of System

```
In [7]:
plt.hist(atlantic_data['status_of_system'],color='orange',bins=20)
plt.title('Visualization of System Status')
plt.xlabel('Status of System')
plt.ylabel('Count')
```

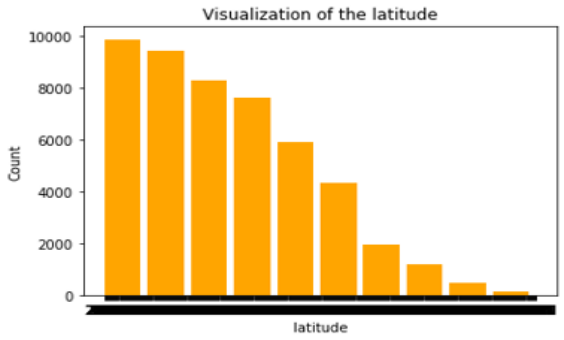
```
Out[7]: Text(0, 0.5, 'Count')
```



- Visualization for Latitude

```
plt.hist(atlantic_data['latitude'],color='Orange',width=50)
plt.xlabel('latitude',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the latitude')
```

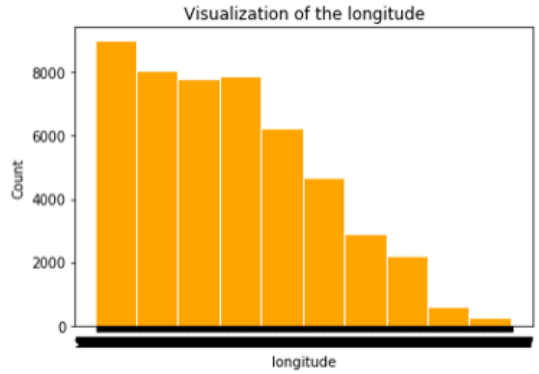
```
Text(0.5, 1.0, 'Visualization of the latitude')
```



- Visualization for Longitude

```
plt.hist(atlantic_data['longitude'],color='Orange',width=100)
plt.xlabel('longitude',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the longitude')
```

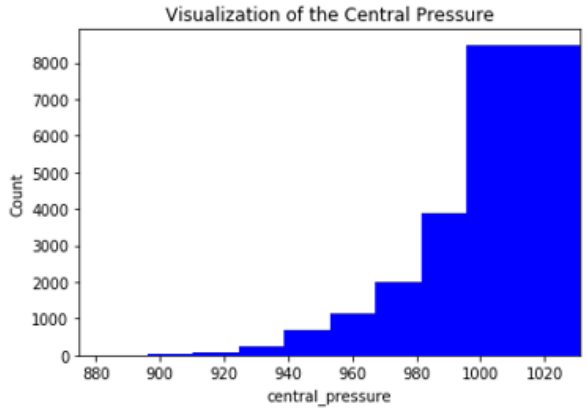
```
Text(0.5, 1.0, 'Visualization of the longitude')
```



- Visualization for Central Pressure

```
plt.hist(atlantic_data['central_pressure'],color='Blue',width=150)
plt.xlabel('central_pressure',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the Central Pressure')
```

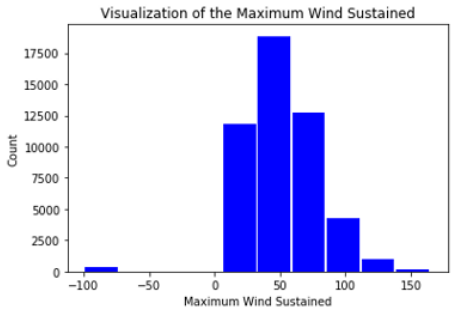
```
Text(0.5, 1.0, 'Visualization of the Central Pressure')
```



- Visualization for Max Wind Sustained

```
In [57]:
plt.hist(atlantic_data['max_sustained_wind'],color='Blue',width=25)
plt.xlabel('Maximum Wind Sustained',color='Black')
plt.ylabel('Count',color='Black')
plt.title('Visualization of the Maximum Wind Sustained')
```

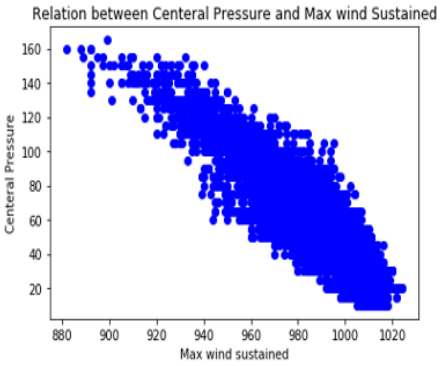
```
Out[57]: Text(0.5, 1.0, 'Visualization of the Maximum Wind Sustained')
```



#Relationship between Max Wind Sustained and Central Pressure

```
In [14]: import matplotlib.pyplot as plt
plt.scatter(atlantic_data['central_pressure'],atlantic_data['max_sustained_wind'],color='Blue')
plt.title('Relation between Centeral Pressure and Max wind Sustained')
plt.xlabel('Max wind sustained')
plt.ylabel('Centeral Pressure')
```

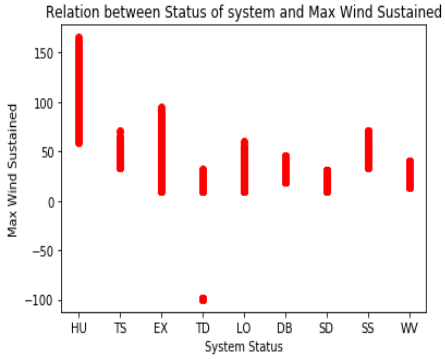
```
Out[14]: Text(0, 0.5, 'Centeral Pressure')
```



#Relationship Between Max wind Sustained and Status of the System

```
In [11]: plt.scatter(atlantic_data['status_of_system'],atlantic_data['max_sustained_wind'],color='Red')
         plt.xlabel('System Status')
         plt.ylabel('Max Wind Sustained')
         plt.title('Relation between Status of system and Max Wind Sustained')
```

```
Out[11]: Text(0.5, 1.0, 'Relation between Status of system and Max Wind Sustained')
```



#Dealing with Missing Values

#To check the presence of null values and represented using True
print(atlantic_data.isnull())

#To give the summary of missing values
atlantic_data.isnull().sum()
atlantic_data.fillna(atlantic_data.mean(), inplace=True)
atlantic_data.isnull().sum()

There are missing values which are present in the atlantic data set are normalized using the mean values.

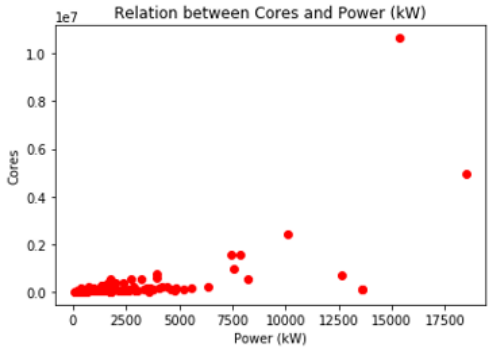
```
In [10]: #atlantic_data2= atlantic_data.replace(np.nan,'NA')
         atlantic_data.fillna(atlantic_data.mean(), inplace=True)

         atlantic_data.isnull().sum()
```

```
Out[10]: basin                0
         name                 0
         year                 0
         cyclone_of_the_year  0
         date                 0
         time                 0
         status_of_system     0
         latitude             0
         longitude            0
         max_sustained_wind   0
         central_pressure     0
         dtype: int64
```

Relationship between Cores and Power

```
In [85]: plt.scatter(list_comp['Power (kW)'],list_comp['Cores'],color='Red')
         plt.xlabel('Power (kW)')
         plt.ylabel('Cores')
         plt.title('Relation between Cores and Power (kW)')
```

```
Out[85]: Text(0.5, 1.0, 'Relation between Cores and Power (kW)')
```



Hence the Relation between Cores and RPeak is strong with 0.712 as the correlation coefficient. Also the Relation between Cores and Power is strong with 0.633 as the correlation Coefficient.

```
In [156]: list_comp.corr()
```

Out[156]:

	Rank	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
Rank	1.000000	-0.208757	-0.306730	-0.311959	-0.230166
Cores	-0.208757	1.000000	0.706636	0.712682	0.633211
Rmax (TFlop/s)	-0.306730	0.706636	1.000000	0.992196	0.567248
Rpeak (TFlop/s)	-0.311959	0.712682	0.992196	1.000000	0.576107
Power (kW)	-0.230166	0.633211	0.567248	0.576107	1.000000