Meteorite Landings Analysis

Shweta Yadav
Artificial Intelligence & Machine Learning
Lambton College
North York
C0854479@mylambton.ca

Sai Srikanth Raju
Artificial Intelligence & Machine Learning
Lambton College
North York
C084551@mylambton.ca

Sandra Nicholas
Artificial Intelligence & Machine Learning
Lambton College
North York, Ontario
C0851536@mylambton.ca

Thakshak Sunkara
Artificial Intelligence & Machine Learning
Lambton College
North York, Ontario
C0846100@mylambton.ca

***Abstract*–– We will work on a project in which we will create boxplots, count plots, histograms, distplots, distribution plots and cartograms for the meteorite landings dataset. In the process, we will also clean data because the dataset for this project has some unwanted data points and visualize all the findings.**

Keywords—Boxplots, histograms, distplots, dist plots, visualisation, meteorite, cartograms and count plots.

I. INTRODUCTION

In this project, we will use NASA's meteorite landings dataset, as meteorite plays a very significant role in discovering the solar system's origin. Because some of the chondrites' meteorites contain chondrules, a chondrule is a spherical mineral grain. They were formed at the time of the birth of the solar system. Studying chondrules will help us understand the formation of the solar system. Hence, meteorites are essential to scientists.

A meteoroid is a solid rock-like object that gets separated from its parent body in outer space mainly due collision of asteroids which are humongous in size. It travels through the solar system and comes close to Earth. It's also referred to as a near-Earth object. Meteor A meteor, otherwise known as a shooting star or fireball, is a meteoroid that fires up and radiates energy when it enters Earth's atmosphere. Meteorite A meteorite is a meteor that falls on the earth's surface or ground.

II. DATA DESCRIPTION

The Meteoritical Society collects data on meteorites that have fallen to Earth from outer space. The dataset contains the variables for over 45,000 meteorites that have struck our planet. We are planning first to clean the dataset as some of the geo-locations of the meteorites are not correct by performing data cleaning. After that, we will apply visualizations such as count plot, boxplot, histogram, distplots, annotated count plot and histogram and try to plot the different sites of the meteorites by using Cartograms from the folium library.

A. Attributes:

1. name: the name of the place where a meteorite was found or observed.

2. id: a unique identifier for a meteorite.

3. nametype: one of the following:

• valid: a typical meteorite.

4. recclass: the class of the meteorite; one of a large number of classes based on physical, chemical, and other characteristics.

5. mass: the mass of the meteorite, in grams

6. fall: whether the meteorite was seen falling, or was discovered after its impact; one of the following:

• Fell: the meteorite's fall was observed.

• Found: the meteorite's fall was not observed.

7. year: the year the meteorite fell, or the year it was found (depending on the value of fell).

8. reclat: the latitude of the meteorite's landing.

9. reclong: the longitude of the meteorite's landing.

10. GeoLocation: a parentheses-enclose, comma-separated tuple that combines reclat and reclon

B) *Importing the Necessary Libraries:*

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
met_df = pd.read_csv('dataset.csv')
met_df.head()
```

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333	(50.775000, 6.083330)
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333	(56.183330, 10.233330)
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000	(54.216670, -113.000000)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000	(16.883330, -99.900000)
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000	(-33.166670, -64.950000)

Figure 1: Showing the head of the data frame

From the above snippet, we are importing the Numpy, Pandas & Matplotlib.Pyplot modules. Created a Pandas Dataframe and displayed the first 5 rows of the data frame using head().

```
met_df.tail()
```

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLo
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850	(29.0 17.0
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667	(13.7 8.9
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.66667	(49.2 17.6
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460	(49.7 41.5
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333	(33.9 -115.6

Figure 2: Showing the tail of the data frame

From the above snippet, we are calling the last 5 rows of the dataset using the tail().

C. Finding the number of rows and columns in a dataset:

```
met_df.shape
```

```
(45716, 10)
```

Figure 3: Showing the shape of the data frame

Getting the descriptive statistics for the year using describe().

```
count    45428.000000
mean      1991.772189
std         27.181247
min        301.000000
25%       1987.000000
50%       1998.000000
75%       2003.000000
max       2501.000000
Name: year, dtype: float64
```

Figure 4: Statistical analysis of the dataset

By using describe() we are getting all the statistical analysis of the dataset. We get the count which was around 45000, the mean is 1991 and the standard deviation is around 27 along with min and max in the dataset.

*D. Exploratory Data Analysis***:**

From the below figure(5), we can see, that the boxplot for the year column in the met_df DataFrame is tightly squeezed. It shows that most of the year values are very close to each other. Specifically, most of the year values are roughly around the year 2000.

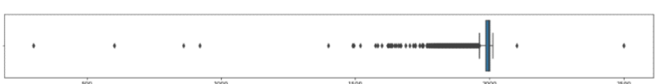


Figure 5: Boxplot in column met_df

From the plot, we can infer that most of the mass values are extremely squeezed around zero because the box has practically become a vertical line. Create a descriptive statistics summary for the mass column to further observe the variation in the mass values.

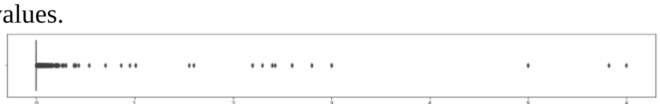


Figure 6:Boxplot on the mass column

*E. Slicing***:**

1. The column names such as recclass, reclat, reclong begin with the rec (denotes 'recommended') keyword. They are the recommended values of the classes, latitudes and longitude variables for the meteorites according to the Meteoritical Society.
2. We must remove any year less than 860 or greater than 2016 from the dataset as mentioned that data is collected between these years only.
3. Remove the 0N/0E latitude and longitude entries which are not between.

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333	(50.775000, 6.083330)
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333	(56.183330, 10.233330)
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000	(54.216670, -113.000000)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000	(16.883330, -99.900000)
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000	(-33.166670, -64.950000)
...	...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850	(29.037000, 17.018500)
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667	(13.783330, 8.966670)
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.66667	(49.250000, 17.666670)
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460	(49.789170, 41.504600)
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333	(33.983330, -115.683330)

45424 rows × 10 columns

Figure 7: Removing unwanted latitude entries

Because of the condition (met_df['year'] >= 860) & (met_df['year'] <= 2016), the met_df DataFrame will return only those rows in which the year values are greater than or equal to 860 and less than or equal to 2016 .

```
count    45424.000000
mean      1991.826413
std         25.047805
min        860.000000
25%       1987.000000
50%       1998.000000
75%       2003.000000
max       2013.000000
Name: year, dtype: float64
```

Figure 8: Statistical analysis after removing unwanted coordinates

As we can see, the minimum value of the year is 860 and the maximum value is 2013.

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333	(50.775000, 6.083330)
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333	(56.183330, 10.233330)
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000	(54.216670, -113.000000)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000	(16.883330, -99.900000)
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000	(-33.166670, -64.950000)
...	...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850	(29.037000, 17.018500)
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667	(13.783330, 8.966670)
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.66667	(49.250000, 17.666670)
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460	(49.789170, 41.504600)
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333	(33.983330, -115.683330)

38221 rows × 10 columns

Figure 9: After removing unnecessary entries

The longitude values go from −180◦to 180◦ where positive values represent the eastward direction and the negative values represent the westward direction. The 0◦longitude is called the prime meridian.

F. Removing The Rows Containing 0 N, 0 E Values:

There are also some rows which contain the 0 reclat and 0 reclong values. It is also the point of intersection of the prime meridian and the Equator. We need to remove such rows because this pair of coordinates represent the portion of the Atlantic Ocean near the west coast of Africa (refer to the image below) from where it is difficult to recover the fallen meteorites.

```
correct_long_df[(correct_long_df['reclat'] == 0) & (correct_long_df['reclong']== 0)]
```

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLocation
596	Mason Gully	53653	Valid	H5	24.54	Fell	2010.0	0.0	0.0	(0.000000, 0.000000)
1648	Allan Hills 09004	52119	Valid	Howardite	221.70	Found	2009.0	0.0	0.0	(0.000000, 0.000000)
1649	Allan Hills 09005	55797	Valid	L5	122.30	Found	2009.0	0.0	0.0	(0.000000, 0.000000)
1650	Allan Hills 09006	55798	Valid	H5	104.30	Found	2009.0	0.0	0.0	(0.000000, 0.000000)
1651	Allan Hills 09008	55799	Valid	H5	31.30	Found	2009.0	0.0	0.0	(0.000000, 0.000000)
...	...	...	...	...	...	...	...	...	...	...
45655	Yamato 984144	40764	Valid	H6	37.44	Found	1998.0	0.0	0.0	(0.000000, 0.000000)
45656	Yamato 984145	40765	Valid	L6	54.80	Found	1998.0	0.0	0.0	(0.000000, 0.000000)
45657	Yamato 984146	40766	Valid	H3	19.32	Found	1998.0	0.0	0.0	(0.000000, 0.000000)
45658	Yamato 984147	40767	Valid	LL6	118.90	Found	1998.0	0.0	0.0	(0.000000, 0.000000)
45659	Yamato 984148	40768	Valid	L5	4.59	Found	1998.0	0.0	0.0	(0.000000, 0.000000)

6185 rows × 10 columns

```
correct_lat_long_df = correct_long_df[~((correct_long_df['reclat'] == 0 ) & (correct_long_df['reclong'] == 0))]
correct_lat_long_df
```

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLocation
0	Aachen	1	Valid	L5	21.0	Fell	1880.0	50.77500	6.08333	(50.775000, 6.083330)
1	Aarhus	2	Valid	H6	720.0	Fell	1951.0	56.18333	10.23333	(56.183330, 10.233330)
2	Abee	6	Valid	EH4	107000.0	Fell	1952.0	54.21667	-113.00000	(54.216670, -113.000000)
3	Acapulco	10	Valid	Acapulcoite	1914.0	Fell	1976.0	16.88333	-99.90000	(16.883330, -99.900000)
4	Achiras	370	Valid	L6	780.0	Fell	1902.0	-33.16667	-64.95000	(-33.166670, -64.950000)
...	...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.0	Found	1990.0	29.03700	17.01850	(29.037000, 17.018500)
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.0	Found	1999.0	13.78333	8.96667	(13.783330, 8.966670)
45713	Zlin	30410	Valid	H4	3.3	Found	1939.0	49.25000	17.66667	(49.250000, 17.666670)
45714	Zubkovsky	31357	Valid	L6	2167.0	Found	2003.0	49.78917	41.50460	(49.789170, 41.504600)
45715	Zulu Queen	30414	Valid	L3.7	200.0	Found	1976.0	33.98333	-115.68333	(33.983330, -115.683330)

32036 rows × 10 columns

Figure 10: Removing unnecessary entries

```
round(correct_lat_long_df.shape[0] * 100 / met_df.shape[0], 2)
```

```
70.08
```

Figure 11: Total retained data after cleaning

So far we have retained the approximately 70% of the values which is still quite a big dataset

G. Missing Values:

We have removed all the unwanted values or rows. Now it's time to check whether we have any missing values or not. The missing values are generally reported as NaN values. We can find the rows or columns containing the NaN values using either the isnull() or the isna() function. They both return True for the NaN values.

```
correct_lat_long_df.isnull().sum()
```

```
name            0
id              0
nametype        0
recclass        0
mass          107
fall            0
year            0
reclat          0
reclong         0
GeoLocation     0
dtype: int64
```

Figure 12: Missing Values handling

From the figure, we can observe the missing values in the 'mass' column. We will replace all the NaN values in the mass column with the median mass value because most of the mass values lie between the first and the third quartile values. The second quartile is a fair representation of the values lying in the inter-quartile range. More importantly, the quartile values remain unaffected by the unusually very high or very low values. Also the outliers do not affect the quartile values because they are computed by arranging all the values in increasing order.

The median mass value is 29.6 grams. So in the below snippet, We are trying to replace all NaN values in the mass column with 29.6.

```
missing_mass_values = correct_lat_long_df.loc[row_indices, 'mass']
missing_mass_values
```

```
12       29.6
38       29.6
76       29.6
93       29.6
172      29.6
         ...
31097    29.6
36812    29.6
38278    29.6
41460    29.6
45698    29.6
Name: mass, Length: 107, dtype: float64
```

Figure 13: Imputing the Null values

H. The Good & Withered Meteorites Separation:

```
correct_lat_long_df['nametype'].value_counts()
```

```
Valid      31967
Relict        69
Name: nametype, dtype: int64
```

Figure 14: Showing relict and valid meteorites

So, there are only 69 meteorites which have withered due to prolonged exposure to probably extreme weather conditions.

	name	id	nametype	recclass	mass	fall	year	reclat	reclong	GeoLocation
1108	Abajo	4	Valid	H5	331.00	Found	1982	26.80000	-105.41667	(26.800000, -105.416670)
1109	Abar al' Uj 001	51399	Valid	H3.8	194.34	Found	2008	22.72192	48.95937	(22.721920, 48.959370)
1110	Abbott	5	Valid	H3-6	21100.00	Found	1951	36.30000	-104.28333	(36.300000, -104.283330)
1111	Abernathy	7	Valid	L6	2914.00	Found	1941	33.85000	-101.80000	(33.850000, -101.800000)
1112	Abo	8	Valid	H	1.20	Found	1840	60.43333	22.30000	(60.433330, 22.300000)
...	...	...	...	...	...	...	...	...	...	...
45711	Zillah 002	31356	Valid	Eucrite	172.00	Found	1990	29.03700	17.01850	(29.037000, 17.018500)
45712	Zinder	30409	Valid	Pallasite, ungrouped	46.00	Found	1999	13.78333	8.96667	(13.783330, 8.966670)
45713	Zlin	30410	Valid	H4	3.30	Found	1939	49.25000	17.66667	(49.250000, 17.666670)
45714	Zubkovsky	31357	Valid	L6	2167.00	Found	2003	49.78917	41.50460	(49.789170, 41.504600)
45715	Zulu Queen	30414	Valid	L3.7	200.00	Found	1976	33.98333	-115.68333	(33.983330, -115.683330)

30871 rows × 10 columns

Figure 15: Showing all the meteorites in good condition

So, there are 30,781 meteorites which were found in good condition.

I. Cartogram for Withered Meteorites:

From folium, we used cartograms to visualise all the withered meteorites

```
import folium
map1 = folium.Map(location=[0, 0], width='90%', height='90%', tiles='Stamen Toner', zoom_start=1)
for i in found_relict_df.index:
    folium.Marker(location=[found_relict_df.loc[i, 'reclat'], found_relict_df.loc[i, 'reclong']],
    popup=found_relict_df.loc[i, 'name']).add_to(map1)
map1
```

Figure 16: Showing cartograms of withered meteorites

J. Cartograms for Good Condition Meteorites:

Similarly, we visualized good condition meteorites across the different years like from years 2004 to 2007, 2007 to 2010 and 2010 to the present.

```
map2 = folium.Map(location=[0.0, 0.0], width='90%', height='80%', tiles='Stamen Toner', zoom_start=1.5)
for i in found_valid_df[found_valid_df['year'] > 2010].index:
    folium.Marker(location=[found_valid_df.loc[i, 'reclat'], found_valid_df.loc[i, 'reclong']],
    popup=found_valid_df.loc[i, 'name']).add_to(map2)
map2
```

Figure 17: Showing good condition meteorites after 2010

```
map3 = folium.Map(location=[0, 0], width='90%', height='85%', tiles='Stamen Toner', zoom_start=1.5)
for i in found_valid_df[(found_valid_df['year'] > 2007) & (found_valid_df['year'] <= 2010)].index:
    folium.Marker(location=[found_valid_df.loc[i, 'reclat'], found_valid_df.loc[i, 'reclong']],
    popup=found_valid_df.loc[i, 'name']).add_to(map3)
map3
```

Figure 18: Showing good condition meteorites between 2007 and 2010

```
map4 = folium.Map(location=[0, 0], width='90%', height='85%', tiles='Stamen Toner', zoom_start=1.5)
for i in found_valid_df[(found_valid_df['year'] > 2004) & (found_valid_df['year'] <= 2007)].index:
    folium.Marker(location=[found_valid_df.loc[i, 'reclat'], found_valid_df.loc[i, 'reclong']],popup=found_
map4
```

Figure 19: Showing good condition meteorites between 2007-2010

The popup parameter displays the name of the location on a popup box when clicked on a marker. Along with the city name, we can also display the mass of a meteorite in the popup of the markers. For this, we have used string concatenation.

*K. Count Plot***:**

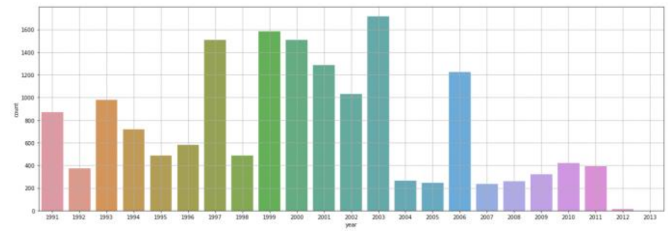


Figure 20: Count plots from year 1991-2013

So according to the above count plot, after the years 1990 (exclusive), the number of meteorites falling has been increasing and decreasing non-periodically. In other words, there appears no fixed pattern in the number of falls of meteorites. Most numbers of meteorites were observed in the year 2003. Thereafter, the number of meteorites observed dropped.

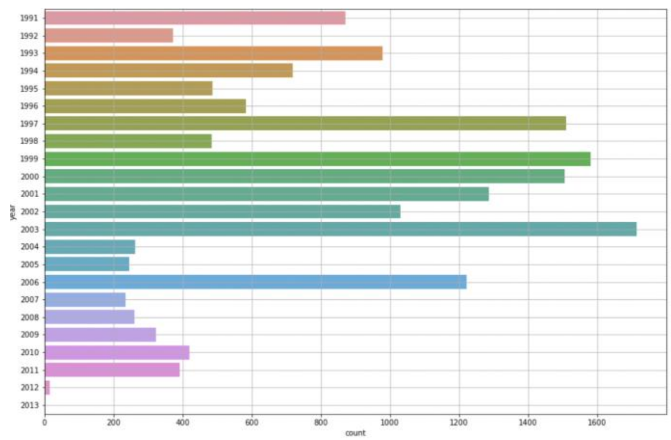


Figure 21: Countplots visualisation

We split the count bars based on a category in the count plot.

1. The number of meteorites that fell but were not found in a year.
2. The number of meteorites that fell and were found in a year.

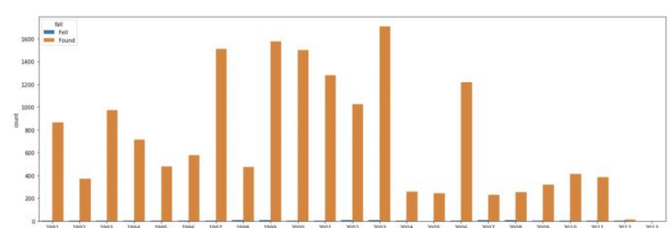


Figure 22: Countplot of found meteorites

As we can see, we have divided the count bars into the count of Fell and the count of Found bars for a year. The blue-coloured bars represent the count of the meteorites which fell but were not found in a year whereas the orange-coloured bars represent the count of the meteorites which fell but were found in a year. The Fell values are very low in comparison to the Found values. Hence, the blue-coloured bars are very short compared to the orange-coloured bars.

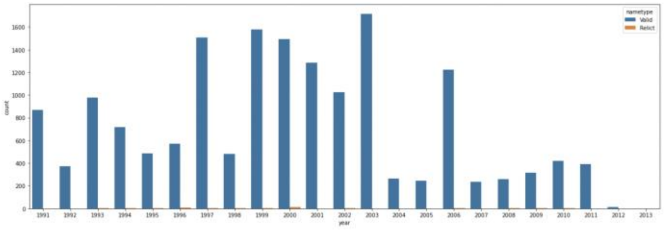


Figure 23: Countplot showing fell meteorites

As we can see, we have divided the count bars into the count of Valid and the count of Relict bars for a year. The blue-coloured bars represent the count of the meteorites that were found in a good condition in a year whereas the orange-coloured bars represent the count of the meteorites that were found in a withered condition in a year. The Relict values are very low in comparison to the Valid values. Hence, the orange-coloured bars are very short compared to the blue-coloured bars.

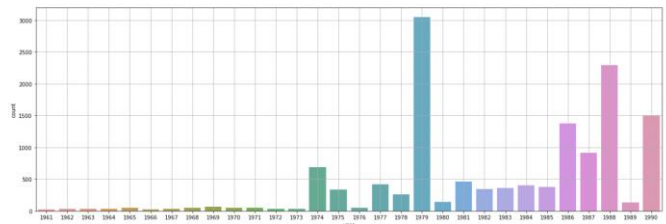


Figure 24: Meteorites observed between 1961-1990

As you can see, the number of meteorites falling on Earth started increasing in 1974. Before 1974, the number of meteorites fallen was consistently less than 500. This could also possibly mean that till 1974, we did not have the technology to precisely monitor the fall of the meteorites.

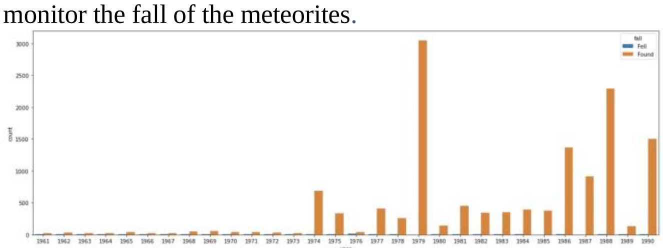


Figure 25: Meteorites found

Again, the Found meteorites are large in numbers for each year.

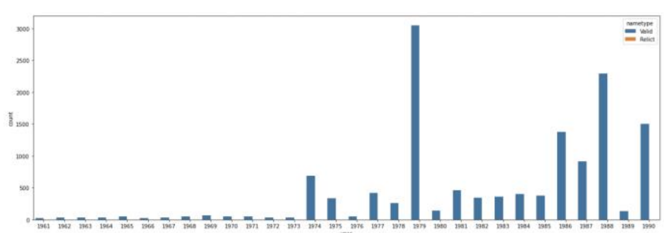


Figure 26: Valid Meteorites

Again, the Valid meteorites are large in numbers for each year.

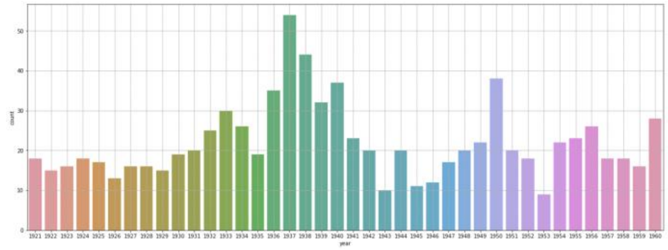


Figure 27: Countplot showing from 1921-1960

Again, there is no fixed pattern in the number of fallen meteorites between 1921 and 1960. In the period 1941 to 1970, the most numbers of meteorites were observed in the year 1937 but they all are less than 100 which is not so significant compared to the meteorites observed in the late 1900s till 2000s.

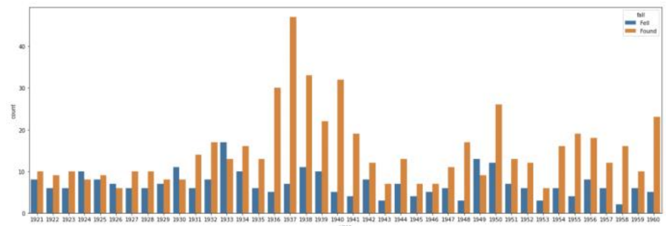


Figure 28: Fell and Found meteorites from 1921-1960

Between 1921 and 1960, the difference between the number of Fell meteorites and Found meteorites every year is not huge unlike between 1961 and 1990.

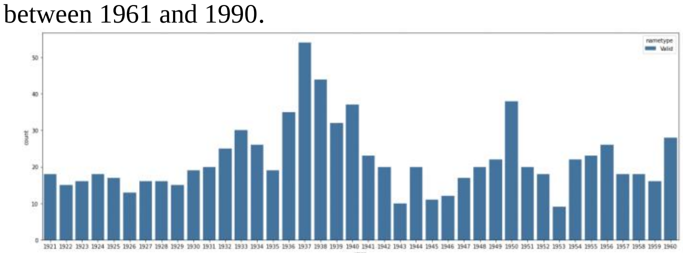


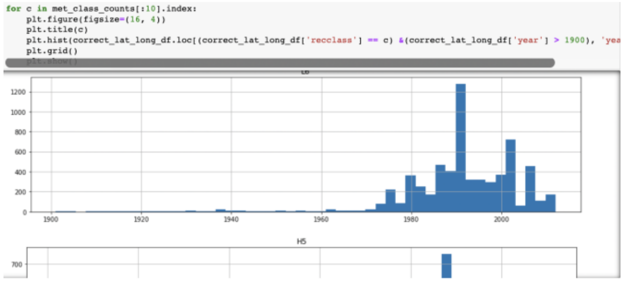
Figure 29: Checking for withered meteorites

So, there were no withered meteorites between 1921 to 1960. This could also mean that we did not have the capacity (or technology) to find the withered meteorites between 1921 and 1960. Additionally, it seems that as we go in the past, the number of meteorites observed decreased.

Figure 30: Meteorites observed before 1920

As we can see, the number of meteorites observed before 1921 drops significantly even less than 10 and it decreases as we gradually go backwards.

L. Histogram:



```
for c in met_class_counts[:10].index:
    plt.figure(figsize=(16, 4))
    plt.title(c)
    plt.hist(correct_lat_long_df.loc[(correct_lat_long_df['recclass'] == c) &(correct_lat_long_df['year'] > 1900), 'yea
    plt.grid()
    plt.show()
```

Figure 31: Histograms from 1900-2000 on meteorites

There is no definite recognizable pattern in the fall of the meteorites over the years for a class of a meteorite. The only concrete observation that we can make from the histograms is that the most number of LL5 class meteorites fell between the years 1995 and 2005 and the most number of the H4/5 class meteorites fell between the years 1975 and 1980.

M. Annotated Bar Graphs:

These are similar to bar graphs but they are annotated with the occurrences in each bar. We usitlised these visualisation and created these bar graphs.

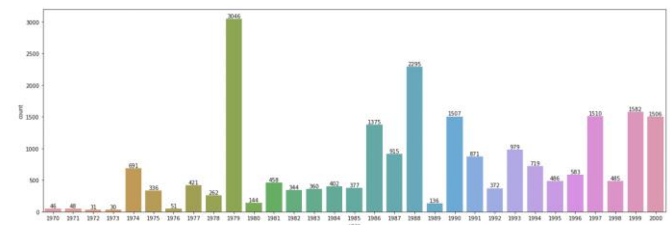


Figure 32: Annotated bar graphs from 1970-2000

N. Dist Plots:

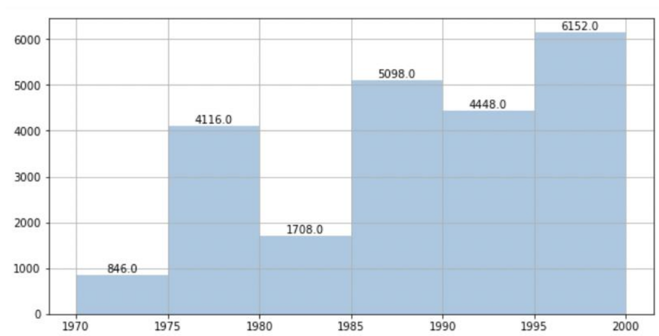
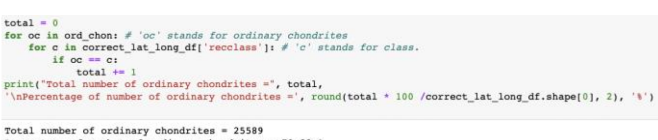


Figure 33: Showing the distplots



```
total = 0
for oc in ord_chon: # 'oc' stands for ordinary chondrites
    for c in correct_lat_long_df['recclass']: # 'c' stands for class.
        if oc == c:
            total += 1
print("Total number of ordinary chondrites =", total,
'\nPercentage of number of ordinary chondrites =', round(total * 100 /correct_lat_long_df.shape[0], 2), '%')

Total number of ordinary chondrites = 25589
Percentage of number of ordinary chondrites = 79.88 %
```

Figure 34: Result

We confirmed that at least 79% of the meteorites are ordinary chondrites in the correct_lat_long_df DataFrame.

IV. CONCLUSION

1. Most of the meteorites that fell on Earth are chondrites. According to Wikipedia, Chondrites are the most common type of meteorite that falls on Earth with estimates for the proportion of the total fall that they represent varying between 85.7% and 86.2%.
2. We are interested in ordinary chondrites because they contain chondrules. Any rock containing a chondrule is called a chondrite.

V. REFERENCES

[1] M. Islam and S. Jin, "An Overview of Data Visualization," *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, 2019, pp. 1-7, doi: 10.1109/ICISCT47635.2019.9012031./

[2] B. B. Becerra, S. V. Reyes, A. G. Hernández, P. V. Elizondo and A. M. González, "Good practice guide for data visualization in the area of descriptive statistics," 2021 Mexican International Conference on Computer Science (ENC), 2021, pp. 1-8, doi: 10.1109/ENC53357.2021.9534814.

[3] A. Nasser, D. Hamad and C. Nasr, "Visualization Methods for Exploratory Data Analysis," 2006 2nd International Conference on Information & Communication Technologies, 2006, pp. 1379-1384, doi: 10.1109/ICTTA.2006.1684582.

[4] Bisong, E. (2019). Matplotlib and Seaborn. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_12

[5] Nusrat, S. and Kobourov, S. (2016), The State of the Art in Cartograms. Computer Graphics Forum, 35: 619-642. https://doi.org/10.1111/cgf.12932https://en.wikipedia.org/wiki/Meteorite

[6] "Meteorite landings," NASA, 27-Jun-2018. [Online]. Available: https://data.nasa.gov/Space-Science/Meteorite-Landings/gh4g-9sfh. [Accessed: 16-Aug-2022].

[7] M. Matsushita and T. Kato, "Interactive visualization method for exploratory data analysis," Proceedings Fifth International Conference on Information Visualisation, 2001, pp. 671-676, doi: 10.1109/IV.2001.942128.

[8] D. Zhu, Y. Wang, B. Wei, Z. Guo and F. Wan, "Data Visualization Overview," 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), 2021, pp. 735-738, doi: 10.1109/ICCASIT53235.2021.9633610..