Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

https://catalog.data.gov/dataset/meteorite-landings

Why I chose this dataset

- I chose this dataset because I've always wanted to be able to understand if there is any relevance in how meteorites cluster and land in the same areas. Later, you will see my mapping result of meteorite geolocations. I also use OLS regression to depict if a meteors mass varies based on its Geo Location.

Preprocessing

- I had to load the data, which was very big compared to the datasets I am used to.
- I filtered the relevant columns like Geo Location into GeoLat and GeoLong. I feel that this minimized the complexity of the cluster process.
- I handled NaN values by replacing them with the mean / or dropping them.
- I just learned why I need to choose my own "n_ clusters" during this assignment, we don't want overfitting or underfitting when dealing with our data.

Data Definitions

name:

- The name of the meteorite.

id:

- A unique identifier that is assigned to each meteorite entry in the dataset.

nametype:

- Indicates the type of name (e.g., "Valid" signifies that the meteorite name is officially recognized).

recclass:

- The classification of the meteorite is based on its chemical and mineral composition.

mass (g):

- The weight of the meteorite in grams, representing the total mass of the specimen.

Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

fall:

- Describes whether the meteorite fell to Earth (e.g., "Fell") or was found (e.g., "Found").

year:

- The year in which the meteorite was observed to fall or was discovered.

reclat:

- The latitude at which the meteorite was found or observed, given in decimal degrees.

reclong:

- The longitude at which the meteorite was found or observed, also in decimal degrees.

GeoLat:

- The extracted latitude value from the GeoLocation string, presented in decimal degrees.

GeoLong:

- The extracted longitude value from the GeoLocation string, also in decimal degrees.

Yes, I know some of the columns in this dataset may not be utilized such as: name, name type, etc. However, I decided to keep most columns so I can work on this more in-depth after the assignment due date.

Starting DF of the Meteorite Landings.

```
Initial DataFrame:
       name   id nametype     recclass  mass (g)  fall    year    reclat    reclong           GeoLocation
0    Aachen    1    Valid           L5      21.0  Fell  1880.0  50.77500    6.08333     (50.775, 6.08333)
1    Aarhus    2    Valid           H6     720.0  Fell  1951.0  56.18333   10.23333  (56.18333, 10.23333)
2      Abee    6    Valid          EH4  107000.0  Fell  1952.0  54.21667 -113.00000    (54.21667, -113.0)
3  Acapulco   10    Valid  Acapulcoite    1914.0  Fell  1976.0  16.88333  -99.90000     (16.88333, -99.9)
4   Achiras  370    Valid           L6     780.0  Fell  1902.0 -33.16667  -64.95000  (-33.16667, -64.95)
```

Showing the NA values, I later imputed them.

```
Missing Values:
name              0
id                0
nametype          0
recclass          0
mass (g)        131
fall              0
year            291
reclat         7315
reclong        7315
GeoLocation    7315
dtype: int64
```

Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

Cleansed DF

```
Cleansed DataFrame:
             name     id nametype              recclass  mass (g)   fall    year    reclat     reclong    GeoLat     GeoLong
0          Aachen      1    Valid                    L5      21.0   Fell  1880.0  50.77500     6.08333  50.77500     6.08333
1          Aarhus      2    Valid                    H6     720.0   Fell  1951.0  56.18333    10.23333  56.18333    10.23333
2            Abee      6    Valid                   EH4  107000.0   Fell  1952.0  54.21667  -113.00000  54.21667  -113.00000
3        Acapulco     10    Valid           Acapulcoite    1914.0   Fell  1976.0  16.88333   -99.90000  16.88333   -99.90000
4         Achiras    370    Valid                    L6     780.0   Fell  1902.0 -33.16667   -64.95000 -33.16667   -64.95000
...           ...    ...      ...                   ...       ...    ...     ...       ...         ...       ...         ...
45711  Zillah 002  31356    Valid               Eucrite     172.0  Found  1990.0  29.03700    17.01850  29.03700    17.01850
45712      Zinder  30409    Valid  Pallasite, ungrouped      46.0  Found  1999.0  13.78333     8.96667  13.78333     8.96667
45713        Zlin  30410    Valid                    H4       3.3  Found  1939.0  49.25000    17.66667  49.25000    17.66667
45714   Zubkovsky  31357    Valid                    L6    2167.0  Found  2003.0  49.78917    41.50460  49.78917    41.50460
45715  Zulu Queen  30414    Valid                  L3.7     200.0  Found  1976.0  33.98333  -115.68333  33.98333  -115.68333

[45716 rows x 11 columns]
PS C:\Users\gwalt\OneDrive\Desktop\Classes\Data Analytics\Self-Guided Proj>
```

OLS Regression Results

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               mass (g)   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     19.92
Date:                Sun, 03 Nov 2024   Prob (F-statistic):           2.24e-09
Time:                        14:48:04   Log-Likelihood:            -6.7107e+05
No. Observations:               45716   AIC:                         1.342e+06
Df Residuals:                   45713   BIC:                         1.342e+06
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       2.974e+04   3766.889      7.895      0.000    2.24e+04    3.71e+04
GeoLat       332.8197     78.323      4.249      0.000     179.305     486.334
GeoLong      -56.3071     45.042     -1.250      0.211    -144.590      31.976
==============================================================================
Omnibus:                   166648.304   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.000   Jarque-Bera (JB):      88523317787.188
Skew:                          77.002   Prob(JB):                         0.00
Kurtosis:                    6818.371   Cond. No.                         150.
==============================================================================
```

This OLS regression uses GeoLat & Geo Long as the independent variables and mass (g) as the dependent variable. If the coefficient for GeoLat is 2.5, this suggests that for each one degree increase in latitude, the mass is expected to increase by 2.5 grams, assuming longitude remains constant. This test suggests that GeoLat is statistically significant, however GeoLong is not significant at all, so we fail to reject that null hypothesis. In the end, mass in grams of a meteor tends to change when the Latitude of it differentiates.

Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

Contingency Table

```
#Contingency Table

import pandas as pd

contingency_table = pd.crosstab(df['fall'], df['recclass'])    "crosstab": Un
print(contingency_table)
✓ 0.0s

recclass  Acapulcoite  Acapulcoite/Lodranite  Acapulcoite/lodranite  \
fall
Fell                1                      0                      0
Found              53                      6                      3

recclass  Achondrite-prim  Achondrite-ung  Angrite  Aubrite  Aubrite-an  \
fall
Fell                    0               1        1        9           0
Found                   9              56       20       54           6

recclass  Brachinite  C  ...  Relict H  Relict OC  Relict iron  Stone-uncl  \
fall                     ...
Fell               0  1  ...         0          0            0          40
Found             33  7  ...         1         65            1          10

recclass  Stone-ung  Unknown  Ureilite  Ureilite-an  Ureilite-pmict  Winonaite
fall
Fell              0        2         5            1               0          1
Found             1        5       295            3              23         24
```

A contingency table here could show the distribution of meteorite falls across different meteorite classes, giving a clear picture of how often each meteorite class appears in "fell" versus "found" categories. This table can visually indicate possible dependencies between class and fall status, supplementing the Chi-square test results.

Each row under fall shows either "Fell" or "Found," with the values showing the count of each type of meteorite. For instance:

- **Acapulcoite**: 1 meteorite was observed falling, while 53 were discovered later.
- **Aubrite**: 9 were observed falling, and 54 were found without observation.
- **Ureilite**: 5 fell and were observed, while 295 were found without prior observation.

Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

I attempted to create a Chi Square Test for a different analysis from the one above, I chose to use "reclass" and "fall." This test is supposed to show significance in meteorite composition and whether the meteorite was found. It claims to be SUPER significant, which I believe is helped by the lack of variation in the data. However, I plan to investigate it more.

```
import pandas as pd
import scipy.stats as stats

data = pd.read_csv('Meteorite_Landings2.csv')

data = data[['recclass', 'fall']].dropna()    "recclass": Unknown word.

contingency_table = pd.crosstab(data['recclass'], data['fall'])    "crosstab":

print("Contingency Table:")
print(contingency_table)

chi2_stat, p_val, dof, expected = stats.chi2_contingency(contingency_table)

print(f"\nChi-square Statistic: {chi2_stat}")
print(f"p-value: {p_val}")
print(f"Degrees of Freedom: {dof}")
print("Expected Frequencies:")
print(expected)
```

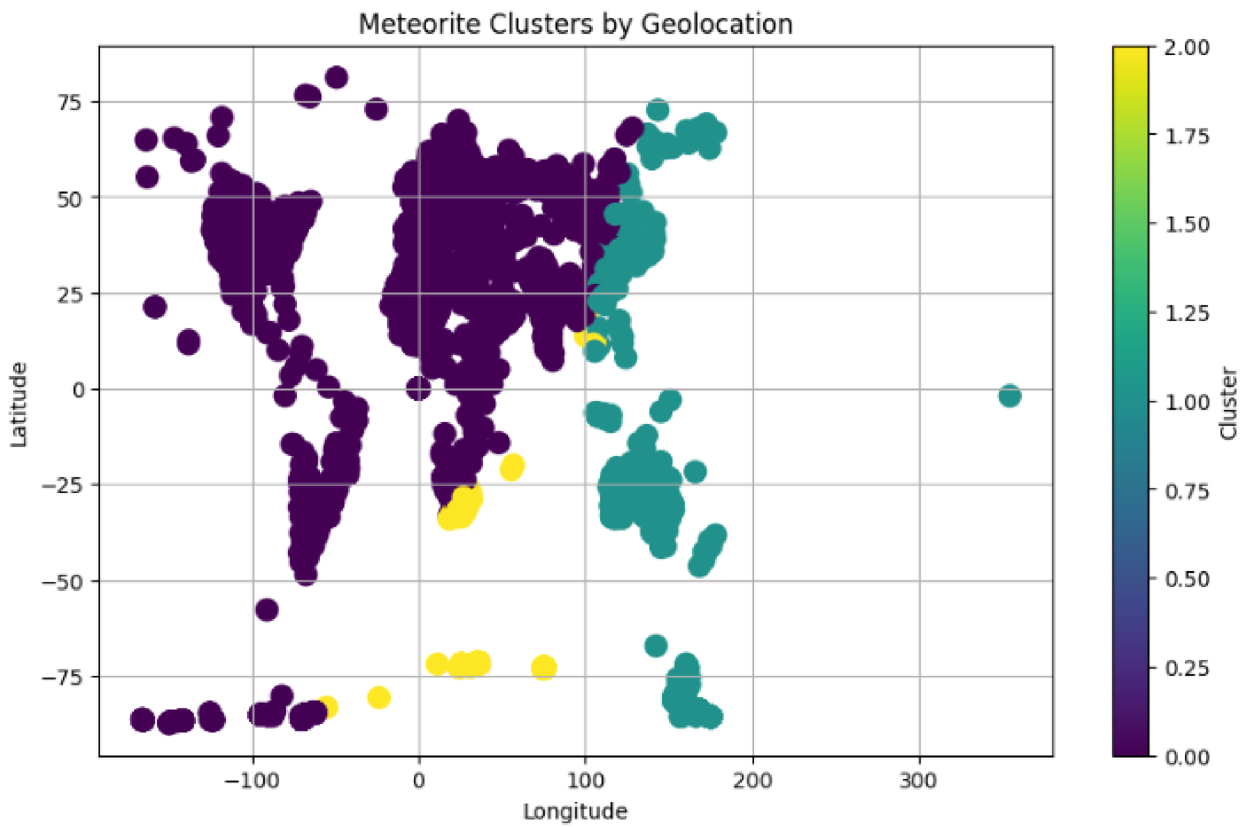
```
Chi-square Statistic: 3181.030655586693
p-value: 0.0
Degrees of Freedom: 465
Expected Frequencies:
[[1.30759472e+00 5.26924053e+01]
 [1.45288302e-01 5.85471170e+00]
 [7.26441508e-02 2.92735585e+00]
 [2.17932453e-01 8.78206755e+00]
 ...
 [7.26441508e+00 2.92735585e+02]
 [9.68588678e-02 3.90314113e+00]
 [5.56938490e-01 2.24430615e+01]
 [6.05367924e-01 2.43946321e+01]]
```

Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

Here is an analysis I made to show meteorite clustering using GeoLat & GeoLong.

Notice anything familiar?

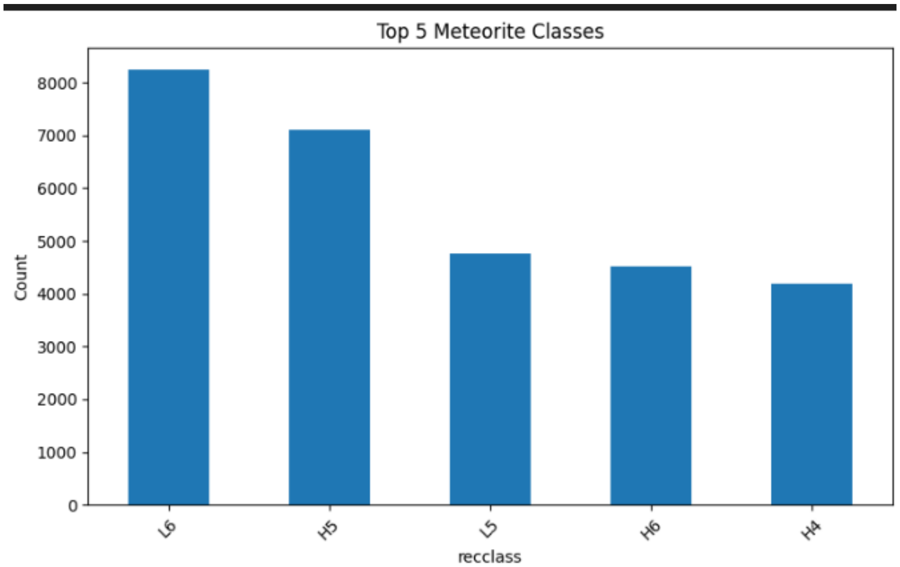


Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

This is an analysis used to measure the top 5 meteorite classes captured in the CSV.

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

df = pd.read_csv('Meteorite_Landings2.csv')

df['year'] = pd.to_numeric(df['year'], errors='coerce')

df = df.dropna(subset=['year', 'recclass', 'fall'])    "dropna": Ur

plt.figure(figsize=(15, 5))    "figsize": Unknown word.

plt.subplot(1, 2, 2)
df['recclass'].value_counts().nlargest(5).plot(kind='bar')    "recc
plt.title('Top 5 Meteorite Classes')
plt.ylabel('Count')    "ylabel": Unknown word.
plt.xticks(rotation=45)    "xticks": Unknown word.

plt.tight_layout()
plt.show()
```



Summary: All in all, I did many tests on this Meteorite Landing data set. I learned that there are many factors that can affect how things happen, and using data to support this is

Gavin Walter
Prof. Rochelle
10/30/2024

Meteorite Landings

very fun. I plan to look back on this assignment and code to see if I can do any other tests that could be a little more accurate, etc. Looking into the GeoLocation of meteorite clusters and where they are found is super cool to me and I love graphing this data to view it. Cleansing my own DF showed me that it’s really important to understand your data and what YOU need to change in it to fit your needs.