Deliverable 2

Descriptive Statistics :

Dataset contains 958524 rows and 45 columns.

```
df.shape
```

```
(958524, 45)
```

Detailed Info

```
In [6]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 958524 entries, 0 to 958523
Data columns (total 45 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   id              958524 non-null  object
 1   spkid           958524 non-null  int64
 2   full_name       958524 non-null  object
 3   pdes            958524 non-null  object
 4   name            22064 non-null   object
 5   prefix          18 non-null      object
 6   neo             958520 non-null  object
 7   pha             938603 non-null  object
 8   H               952261 non-null  float64
 9   diameter        136209 non-null  float64
 10  albedo          135103 non-null  float64
 11  diameter_sigma  136081 non-null  float64
 12  orbit_id        958524 non-null  object
 13  epoch           958524 non-null  float64
 14  epoch_mjd       958524 non-null  int64
 15  epoch_cal       958524 non-null  float64
 16  equinox         958524 non-null  object
 17  e               958524 non-null  float64
 18  a               958524 non-null  float64
 19  q               958524 non-null  float64
 20  i               958524 non-null  float64
 21  om              958524 non-null  float64
 22  w               958524 non-null  float64
 23  ma              958523 non-null  float64
 24  ad              958520 non-null  float64
 25  n               958524 non-null  float64
 26  tp              958524 non-null  float64
 27  tp_cal          958524 non-null  float64
 28  per             958520 non-null  float64
 29  per_y           958523 non-null  float64
 30  moid            938603 non-null  float64
 31  moid_ld         958397 non-null  float64
 32  sigma_e         938602 non-null  float64
 33  sigma_a         938602 non-null  float64
 34  sigma_q         938602 non-null  float64
 35  sigma_i         938602 non-null  float64
 36  sigma_om        938602 non-null  float64
 37  sigma_w         938602 non-null  float64
 38  sigma_ma        938602 non-null  float64
 39  sigma_ad        938598 non-null  float64
 40  sigma_n         938602 non-null  float64
 41  sigma_tp        938602 non-null  float64
 42  sigma_per       938598 non-null  float64
 43  class           958524 non-null  object
 44  rms             958522 non-null  float64
dtypes: float64(33), int64(2), object(10)
memory usage: 329.1+ MB
```

Sample rows :

```
In [7]: df.describe()
```

Out[7]:

	spkid	H	diameter	albedo	diameter_sigma	epoch	epoch_mjd	epoch_cal	e	
count	9.585240e+05	952261.000000	136209.000000	135103.000000	136081.000000	9.585240e+05	958524.000000	9.585240e+05	958524.000000	958524.000
mean	3.810114e+06	16.906411	5.506429	0.130627	0.479184	2.458869e+06	58868.781950	2.019693e+07	0.156116	2.902
std	6.831541e+06	1.790405	9.425164	0.110323	0.782895	7.016716e+02	701.671573	1.930354e+04	0.092643	39.716
min	2.000001e+06	-1.100000	0.002500	0.001000	0.000500	2.425052e+06	25051.000000	1.927062e+07	0.000000	-14702.447
25%	2.239632e+06	16.100000	2.780000	0.053000	0.180000	2.459000e+06	59000.000000	2.020053e+07	0.092193	2.387
50%	2.479262e+06	16.900000	3.972000	0.079000	0.332000	2.459000e+06	59000.000000	2.020053e+07	0.145002	2.646
75%	3.752518e+06	17.714000	5.765000	0.190000	0.620000	2.459000e+06	59000.000000	2.020053e+07	0.200650	3.001
max	5.401723e+07	33.200000	939.400000	1.000000	140.000000	2.459000e+06	59000.000000	2.020053e+07	1.855356	33488.895

8 rows × 35 columns

Data Preprocessing:

Checking For Total Number of Null values :

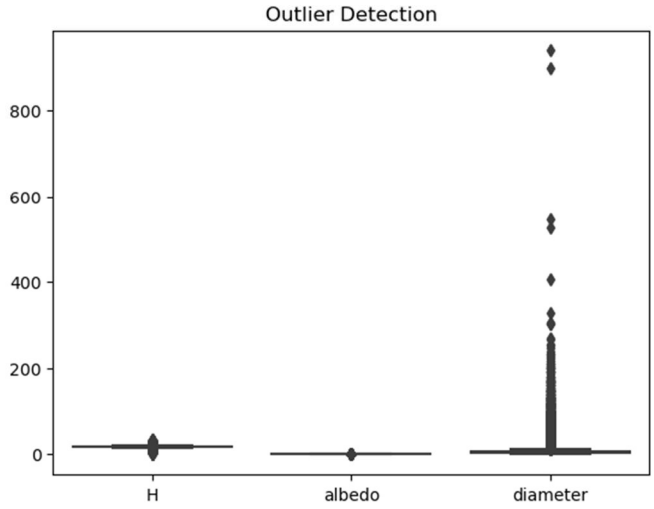
```
In [8]: df.isnull().sum()
```

Out[8]:

```
id                  0
spkid               0
full_name           0
pdes                0
name           936460
prefix         958506
neo                 4
pha             19921
H                6263
diameter       822315
albedo         823421
diameter_sigma 822443
orbit_id            0
epoch               0
epoch_mjd           0
epoch_cal           0
equinox             0
e                   0
a                   0
q                   0
i                   0
om                  0
w                   0
ma                  1
ad                  4
n                   0
tp                  0
tp_cal              0
per                 4
per_y               1
moid            19921
moid_ld           127
sigma_e         19922
sigma_a         19922
sigma_q         19922
sigma_i         19922
sigma_om        19922
sigma_w         19922
sigma_ma        19922
sigma_ad        19926
sigma_n         19922
sigma_tp        19922
sigma_per       19926
class               0
rms                 2
dtype: int64
```

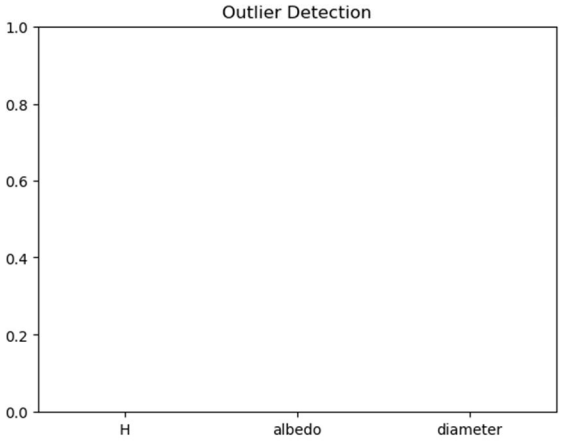
Outlier Detection :

In [7]:
```
# Identify outliers using box plots
sns.boxplot(data=df[['H', 'albedo', 'diameter']])
plt.title('Outlier Detection')
plt.show()
```



In [8]:
```
z_scores = zscore(df[['H', 'albedo', 'diameter']])
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3).all(axis=1)
df = df[filtered_entries]
```

In [9]:
```
sns.boxplot(data=df[['H', 'albedo', 'diameter']])
plt.title('Outlier Detection')
plt.show()
```



There are no null values.

```
In [10]: print(df.isnull().sum())
```

```
id                0.0
spkid             0.0
full_name         0.0
pdes              0.0
name              0.0
prefix            0.0
neo               0.0
pha               0.0
H                 0.0
diameter          0.0
albedo            0.0
diameter_sigma    0.0
orbit_id          0.0
epoch             0.0
epoch_mjd         0.0
epoch_cal         0.0
equinox           0.0
e                 0.0
a                 0.0
q                 0.0
i                 0.0
om                0.0
w                 0.0
ma                0.0
ad                0.0
n                 0.0
tp                0.0
tp_cal            0.0
per               0.0
per_y             0.0
moid              0.0
moid_ld           0.0
sigma_e           0.0
sigma_a           0.0
sigma_q           0.0
sigma_i           0.0
sigma_om          0.0
sigma_w           0.0
sigma_ma          0.0
sigma_ad          0.0
sigma_n           0.0
sigma_tp          0.0
sigma_per         0.0
class             0.0
rms               0.0
dtype: float64
```

Checking for duplicate rows:

```
In [12]: duplicate = df[df.duplicated()]
         print("Duplicate Rows : ", len(duplicate))
         duplicate
```

```
Duplicate Rows :  0
```

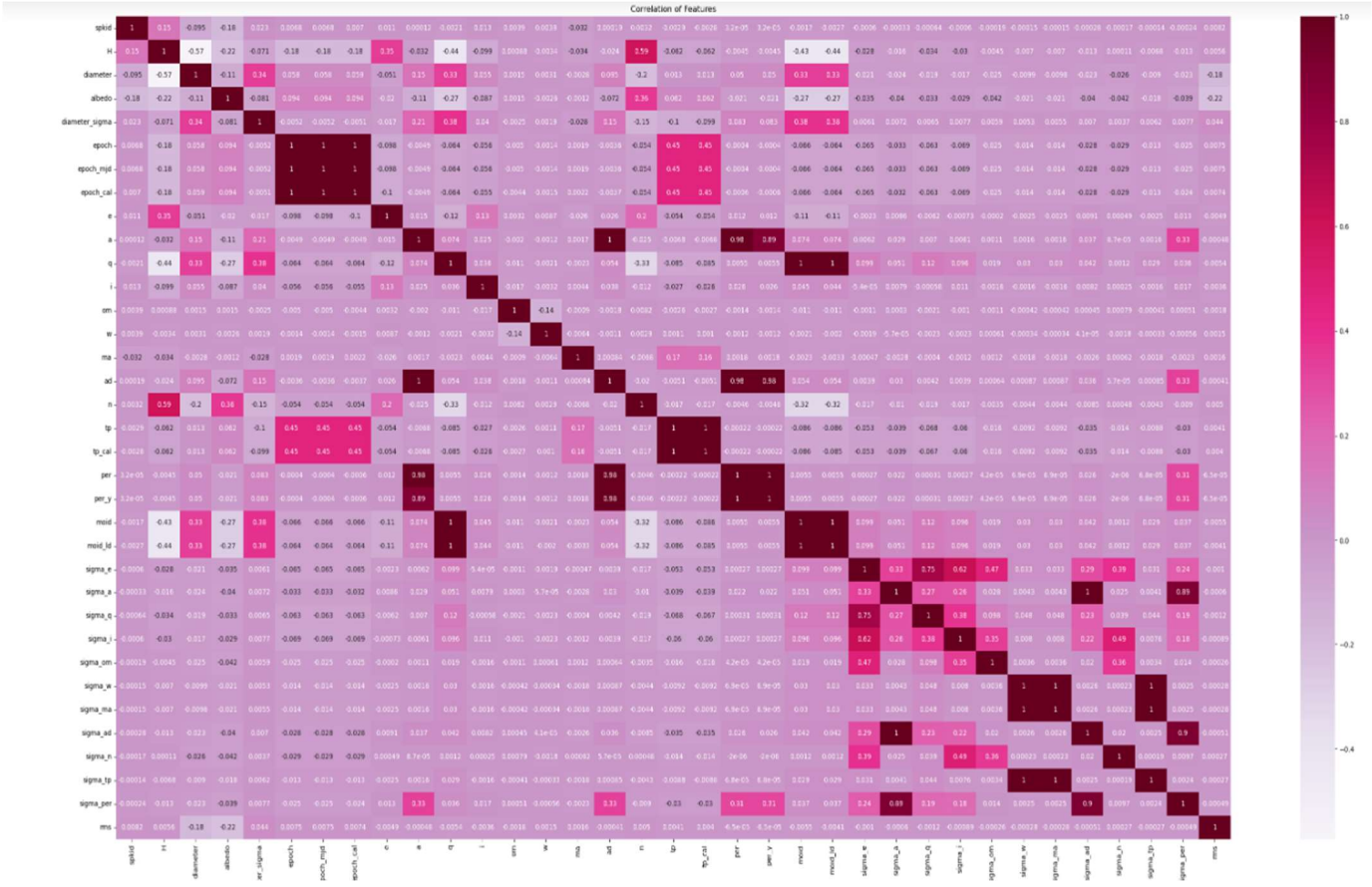
Out[12]:

id	spkid	full_name	pdes	name	prefix	neo	pha	H	diameter	...	sigma_i	sigma_om	sigma_w	sigma_ma	sigma_ad	sigma_n	sigma_tp	sigma_pe

0 rows × 45 columns

Feature correlations:

```
plt.figure(figsize = (40,20))
sns.heatmap(df.corr(), annot=True, cmap='PuRd')
plt.title("Correlation of Features")
plt.show()
```



Correlation with Decision Feature

```
df.corr()['diameter']
```

```
spkid            -0.095362
H                -0.572648
diameter          1.000000
albedo           -0.108880
diameter_sigma    0.337145
epoch             0.058475
epoch_mjd         0.058475
epoch_cal         0.058539
e                -0.050649
a                 0.146799
q                 0.329223
i                 0.054963
om                0.001530
w                 0.003115
ma               -0.002811
ad                0.094735
n                -0.199425
tp                0.013128
tp_cal            0.013350
per               0.050282
per_y             0.050282
moid              0.331983
moid_ld           0.331983
sigma_e          -0.020864
sigma_a          -0.023531
sigma_q          -0.019447
sigma_i          -0.017378
sigma_om         -0.024879
sigma_w          -0.009857
sigma_ma         -0.009806
sigma_ad         -0.023295
sigma_n          -0.025989
sigma_tp         -0.009043
sigma_per        -0.022720
rms              -0.182322
Name: diameter, dtype: float64
```

Strong Correlations:

The strong correlations with diameter in the asteroid dataset are:

H: Absolute magnitude parameter, which is an indicator of the asteroid's brightness.

diameter_sigma: Standard error of the asteroid diameter measurement.

q: Perihelion distance, which is the closest distance of an asteroid's orbit to the Sun.

ad: Aphelion distance, which is the farthest distance of an asteroid's orbit to the Sun.

moid: Minimum orbit intersection distance, which is the closest distance between the asteroid's orbit and the Earth's orbit.

moid_ld: Minimum orbit intersection distance in lunar distances, which is the closest distance between the asteroid's orbit and the Moon's orbit.

rms: Root mean square residual of the asteroid's orbit solution.

Weak correlation:

spkid (-0.095362)

e (-0.050649)

om (0.001530)

w (0.003115)

ma (-0.002811)

tp (0.013128)

tp_cal (0.013350)

per (0.050282)

per_y (0.050282)

sigma_e (-0.020864)

sigma_a (-0.023531)

sigma_q (-0.019447)

sigma_i (-0.017378)

sigma_om (-0.024879)

sigma_w (-0.009857)

sigma_ma (-0.009806)

sigma_ad (-0.023295)

sigma_n (-0.025989)

sigma_tp (-0.009043)

sigma_per (-0.022720)

Data Preparation:
Eliminating the unnecessary columns from dataset:

```
colomns = ["per_y" , "per" , "ma" , "tp"]
```

```
df.drop(colomns, axis = 1, inplace=True)
```

```
df.describe
```

```
[958524 rows x 41 columns]>
```

Removed columns that were found to be weak correlations.

Renaming Columns:

In [31]:
```
df = df.rename(columns = {
    'e':'Eccentricity',
    'a':'Semi_Major_Axis',
    'q':'PerihelionDistance',
    'i':'Inclination',
    'M':'MeanAnomaly',
    'tp': 'Time_Perihelion_Passage',
    'n':'Mean_Motion',
    'Q':'Aphelion_Distance',
    'full_name':'FullName',
    'spkid':'SPKID',
    'class':'Classification',
    'per':'Period_Days',
    'per_y':'Period_Years'
}).copy()
df.head(10)
```

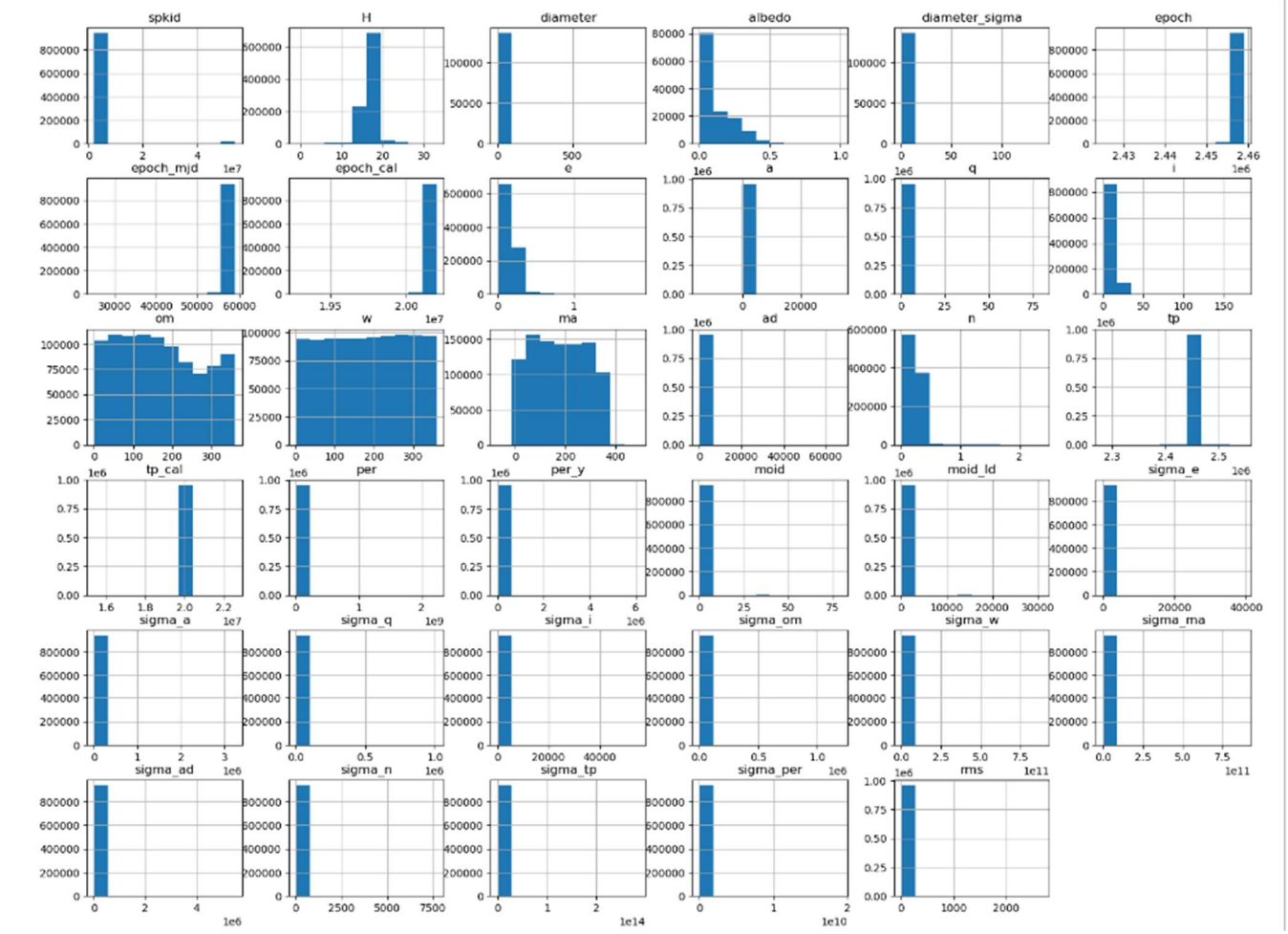
Out[31]:

	SPKID	FullName	orbit_id	Eccentricity	Semi_Major_Axis	PerihelionDistance	Inclination	Mean_Motion	Time_Perihelion_Passage	Period_Days	Period_
0	2000001	1 Ceres	JPL 47	0.076009	2.769165	2.558684	10.594067	0.213885	2.458239e+06	1683.145703	4.6
1	2000002	2 Pallas	JPL 37	0.229972	2.773841	2.135935	34.832932	0.213345	2.458321e+06	1687.410992	4.6
2	2000003	3 Juno	JPL 112	0.256936	2.668285	1.982706	12.991043	0.226129	2.458446e+06	1592.013769	4.3
3	2000004	4 Vesta	JPL 35	0.088721	2.361418	2.151909	7.141771	0.271609	2.458248e+06	1325.432763	3.6
4	2000005	5 Astraea	JPL 114	0.190913	2.574037	2.082619	5.367427	0.238661	2.458926e+06	1508.414421	4.1
5	2000006	6 Hebe	JPL 89	0.203219	2.424533	1.931822	14.739653	0.261073	2.459649e+06	1378.924506	3.7
6	2000007	7 Iris	110	0.230145	2.387375	1.837933	5.521598	0.267192	2.459422e+06	1347.347071	3.6
7	2000008	8 Flora	JPL 118	0.155833	2.201415	1.858362	5.889081	0.301753	2.459149e+06	1193.029574	3.2
8	2000009	9 Metis	JPL 116	0.123300	2.386189	2.091972	5.576494	0.267391	2.458911e+06	1346.343282	3.6
9	2000010	10 Hygiea	JPL 96	0.112117	3.142435	2.790114	3.831786	0.176931	2.459776e+06	2034.688644	5.5

Data Visualizations :

Histogram for each column:

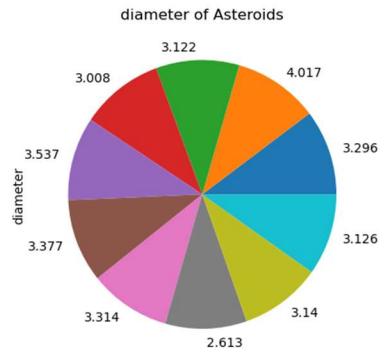
```
df.hist(figsize=(20,15));
```



Retrieving the top 10 values of diameter:

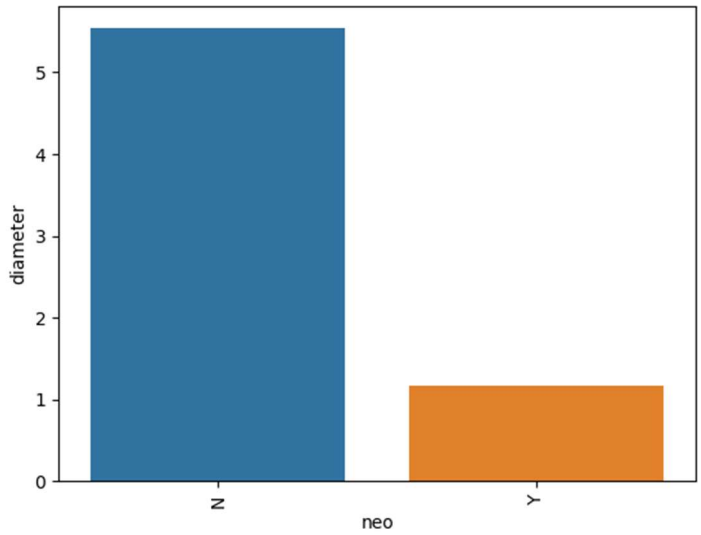
```
In [35]: df['diameter'].value_counts() \
             .head(10) \
             .plot(kind='pie', title='diameter of Asteroids')
```

```
Out[35]: <AxesSubplot: title={'center': 'diameter of Asteroids'}, ylabel='diameter'>
```



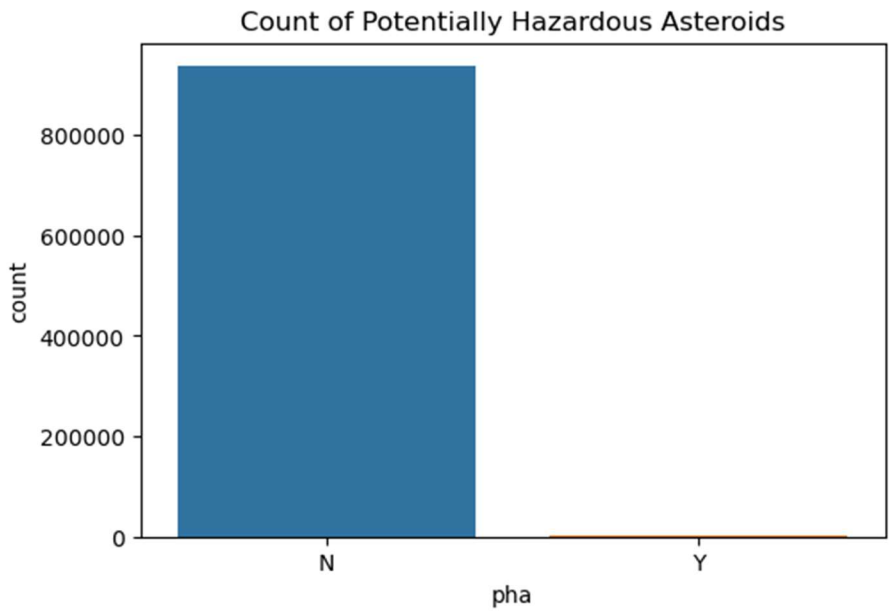
Visualization of[YES-NO] Columns and their relationship with the target:

```
In [29]: # Create a bar plot showing the mean value of the target for each category of a categorical column
sns.barplot(x='neo', y='diameter', data=df, estimator=np.mean)
plt.xticks(rotation=90)
plt.show()
```



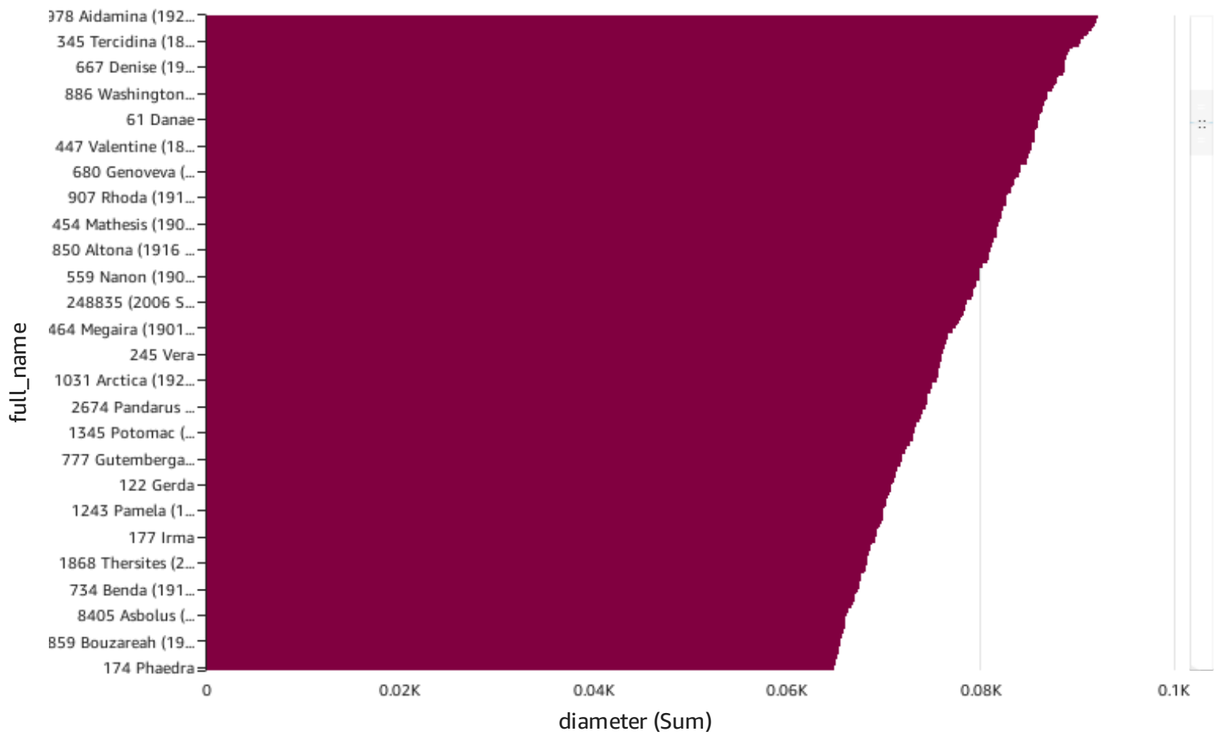
Distribution of variable that indicates hazard:

```
plt.figure(figsize=(6,4))
sns.countplot(x='pha', data=df)
plt.title('Count of Potentially Hazardous Asteroids')
plt.show()
```

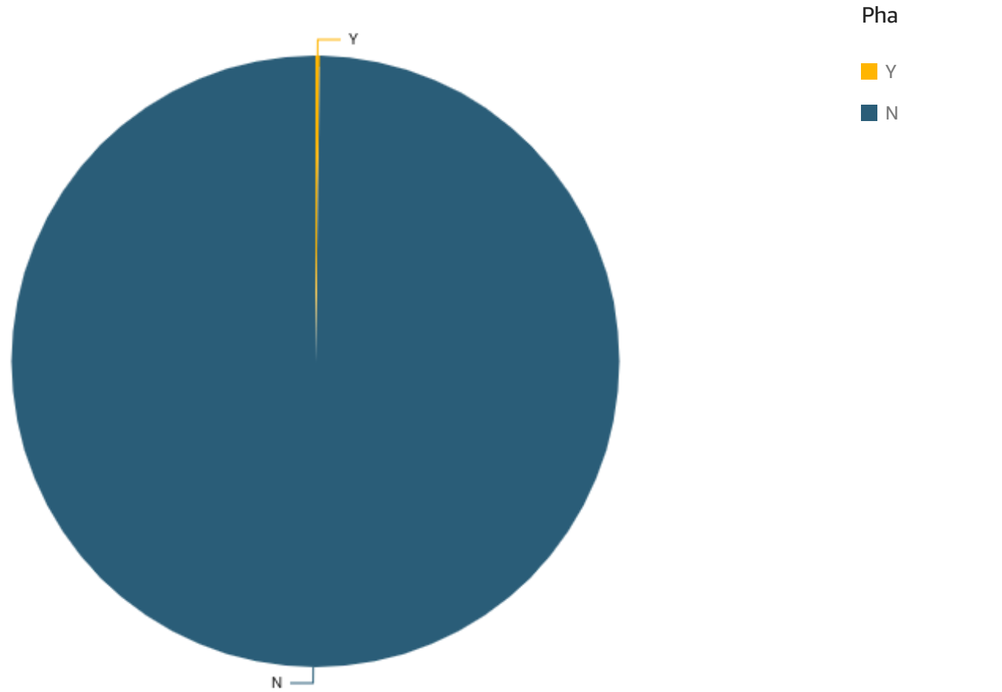


Diameter of Asteroids

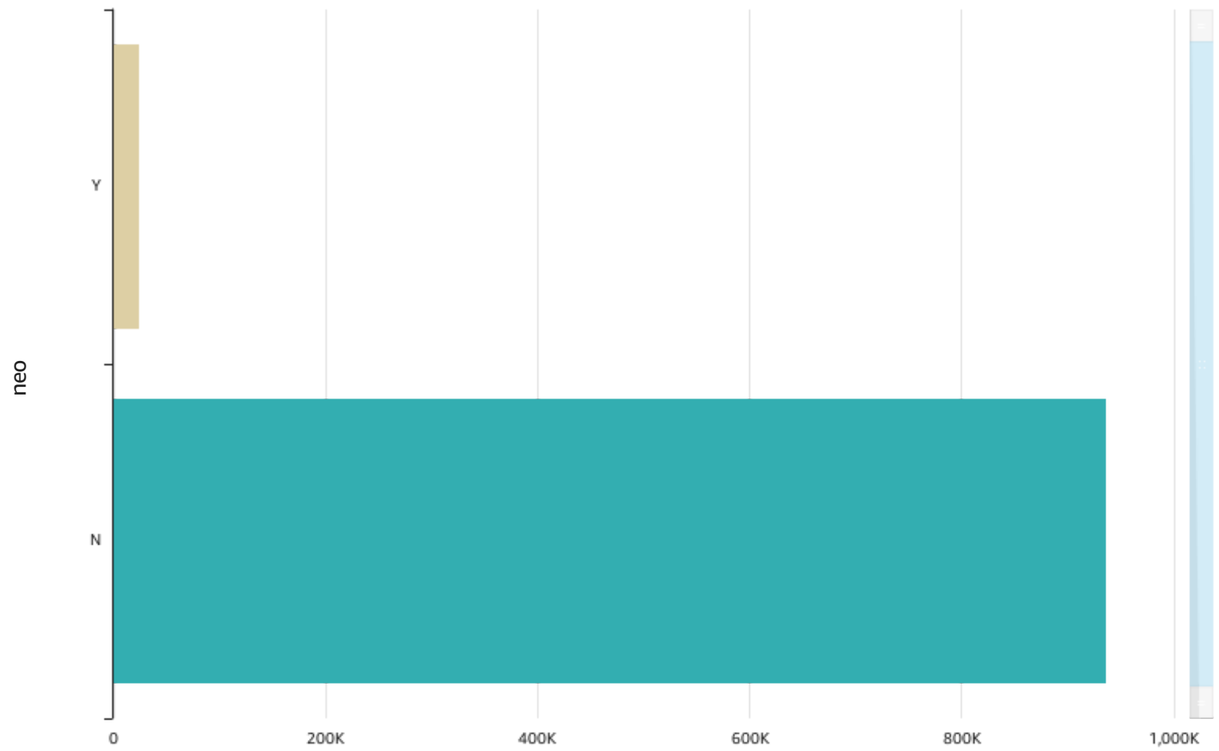
SHOWING TOP 2500 IN FULL_NAME



Count of Potentially Hazardous Asteroids



Count of Near earth objects



Eccentricity of Asteroids

SHOWING TOP 20 IN E