In [6]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
pd.set_option('display.max_column',None)
from random import sample
from sklearn import preprocessing
from sklearn.preprocessing import scale
import seaborn as sns
from scipy import stats
import scipy
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import statsmodels.formula.api as smf
from statsmodels.stats.outliers_influence import variance_inflation_factor
pd.set_option('display.max_column',None)
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
import plotly.graph_objects as go
import plotly.io as pio

pio.templates.default = 'plotly_white'
```

In [2]:

```
df=pd.read_csv('Asteroid.csv')
```

```
C:\Users\saina\anaconda3\lib\site-packages\IPython\core\interactiveshel
l.py:3146: DtypeWarning: Columns (11,14,15,22,23) have mixed types.Speci
fy dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [3]:

```
df
```

Out[3]:

	full_name	a	e	G	i	om	w	q
0	1 Ceres	2.769165	0.076009	0.12	10.594067	80.305532	73.597694	2.558684
1	2 Pallas	2.772466	0.230337	0.11	34.836234	173.080063	310.048857	2.133865
2	3 Juno	2.669150	0.256942	0.32	12.988919	169.852760	248.138626	1.983332
3	4 Vesta	2.361418	0.088721	0.32	7.141771	103.810804	150.728541	2.151909
4	5 Astraea	2.574249	0.191095	NaN	5.366988	141.576604	358.687608	2.082324
...	...	...	...	...	...	...	...	...
839731	(6344 P-L)	2.812945	0.664688	NaN	4.695700	183.310012	234.618352	0.943214
839732	(1168 T-2)	2.645238	0.259376	NaN	12.574937	1.620020	339.568072	1.959126
839733	(2060 T-2)	2.373137	0.202053	NaN	0.732484	176.499082	198.026527	1.893638
839734	(2678 T-3)	2.260404	0.258348	NaN	9.661947	204.512448	148.496988	1.676433
839735	(4571 T-3)	2.546442	0.287672	NaN	5.356238	70.709555	273.483265	1.813901

839736 rows × 27 columns

In [7]:

```
!pip install tensorflow_decision_forests --upgrade -q
```

```
ERROR: Could not find a version that satisfies the requirement tensorflo
w_decision_forests (from versions: none)
ERROR: No matching distribution found for tensorflow_decision_forests
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 839736 entries, 0 to 839735
Data columns (total 27 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   full_name       839736 non-null  object 
 1   a               839734 non-null  float64
 2   e               839736 non-null  float64
 3   G               119 non-null     float64
 4   i               839736 non-null  float64
 5   om              839736 non-null  float64
 6   w               839736 non-null  float64
 7   q               839736 non-null  float64
 8   ad              839730 non-null  float64
 9   per_y           839735 non-null  float64
 10  data_arc        823947 non-null  float64
 11  condition_code  838743 non-null  object 
 12  n_obs_used      839736 non-null  int64  
 13  H               837042 non-null  float64
 14  diameter        137681 non-null  object 
 15  extent          18 non-null      object 
 16  albedo          136452 non-null  float64
 17  rot_per         18796 non-null   float64
 18  GM              14 non-null      float64
 19  BV              1021 non-null    float64
 20  UB              979 non-null     float64
 21  IR              1 non-null       float64
 22  spec_B          1666 non-null    object 
 23  spec_T          980 non-null     object 
 24  neo             839730 non-null  object 
 25  pha             822814 non-null  object 
 26  moid            822814 non-null  float64
dtypes: float64(18), int64(1), object(8)
memory usage: 173.0+ MB
```

In [9]:

```
df = df.dropna(subset=['diameter'])
```

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137681 entries, 0 to 810411
Data columns (total 27 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   full_name       137681 non-null  object 
 1   a               137681 non-null  float64
 2   e               137681 non-null  float64
 3   G               119 non-null     float64
 4   i               137681 non-null  float64
 5   om              137681 non-null  float64
 6   w               137681 non-null  float64
 7   q               137681 non-null  float64
 8   ad              137681 non-null  float64
 9   per_y           137681 non-null  float64
 10  data_arc        137541 non-null  float64
 11  condition_code  137681 non-null  object 
 12  n_obs_used      137681 non-null  int64  
 13  H               136930 non-null  float64
 14  diameter        137681 non-null  object 
 15  extent          16 non-null      object 
 16  albedo          136449 non-null  float64
 17  rot_per         11188 non-null   float64
 18  GM              14 non-null      float64
 19  BV              1005 non-null    float64
 20  UB              965 non-null     float64
 21  IR              1 non-null       float64
 22  spec_B          1370 non-null    object 
 23  spec_T          965 non-null     object 
 24  neo             137681 non-null  object 
 25  pha             137681 non-null  object 
 26  moid            137681 non-null  float64
dtypes: float64(18), int64(1), object(8)
memory usage: 29.4+ MB
```

In [11]:

```
df = df.drop(columns=['full_name', 'H', 'albedo', 'G', 'extent', 'rot_per', 'GM', 'BV',
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137681 entries, 0 to 810411
Data columns (total 15 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   a               137681 non-null  float64
 1   e               137681 non-null  float64
 2   i               137681 non-null  float64
 3   om              137681 non-null  float64
 4   w               137681 non-null  float64
 5   q               137681 non-null  float64
 6   ad              137681 non-null  float64
 7   per_y           137681 non-null  float64
 8   data_arc        137541 non-null  float64
 9   condition_code  137681 non-null  object 
 10  n_obs_used      137681 non-null  int64  
 11  diameter        137681 non-null  object 
 12  neo             137681 non-null  object 
 13  pha             137681 non-null  object 
 14  moid            137681 non-null  float64
dtypes: float64(10), int64(1), object(4)
memory usage: 16.8+ MB
```

In [12]:

```
df.diameter.dtype
```

Out[12]:

```
dtype('O')
```

In [13]:

```
df.diameter = pd.to_numeric(df.diameter, errors='coerce')
```

In [14]:

```
np.sum(df.diameter.isna())
```

Out[14]:

```
1
```

In [15]:

```
df = df.dropna(subset=['diameter'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137680 entries, 0 to 810411
Data columns (total 15 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   a               137680 non-null  float64
 1   e               137680 non-null  float64
 2   i               137680 non-null  float64
 3   om              137680 non-null  float64
 4   w               137680 non-null  float64
 5   q               137680 non-null  float64
 6   ad              137680 non-null  float64
 7   per_y           137680 non-null  float64
 8   data_arc        137540 non-null  float64
 9   condition_code  137680 non-null  object 
 10  n_obs_used      137680 non-null  int64  
 11  diameter        137680 non-null  float64
 12  neo             137680 non-null  object 
 13  pha             137680 non-null  object 
 14  moid            137680 non-null  float64
dtypes: float64(11), int64(1), object(3)
memory usage: 16.8+ MB
```

In [16]:

```
df.diameter.describe()
```

Out[16]:

```
count    137680.000000
mean          5.480873
std           9.365499
min           0.002500
25%           2.770000
50%           3.956000
75%           5.741000
max         939.400000
Name: diameter, dtype: float64
```

In [17]:

```
df = df.dropna(subset=['data_arc'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137540 entries, 0 to 810411
Data columns (total 15 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   a               137540 non-null  float64
 1   e               137540 non-null  float64
 2   i               137540 non-null  float64
 3   om              137540 non-null  float64
 4   w               137540 non-null  float64
 5   q               137540 non-null  float64
 6   ad              137540 non-null  float64
 7   per_y           137540 non-null  float64
 8   data_arc        137540 non-null  float64
 9   condition_code  137540 non-null  object 
 10  n_obs_used      137540 non-null  int64  
 11  diameter        137540 non-null  float64
 12  neo             137540 non-null  object 
 13  pha             137540 non-null  object 
 14  moid            137540 non-null  float64
dtypes: float64(11), int64(1), object(3)
memory usage: 16.8+ MB
```

In [18]:

```
df.condition_code.value_counts()
```

Out[18]:

```
0      126192
9        5584
0        2017
9.0       827
1         809
1         563
2         302
5         242
6         152
4         144
3         135
7         129
2          95
5.0        92
7.0        78
8          49
6.0        36
4.0        33
3          33
8.0        28
Name: condition_code, dtype: int64
```

In [19]:

```
df.condition_code = pd.to_numeric(df.condition_code, errors='coerce')
```

In [20]:

```
df.neo.value_counts()
```

Out[20]:

```
N    136691
Y       849
Name: neo, dtype: int64
```

In [21]:

```
df.neo = df.neo.replace({'N':0, 'Y':1})
```

In [22]:

```
df.pha.value_counts()
```

Out[22]:

```
N    137320
Y       220
Name: pha, dtype: int64
```

In [23]:

```
df.pha = df.pha.replace({'N':0, 'Y':1})
```

In [24]:

```
df.info()
```

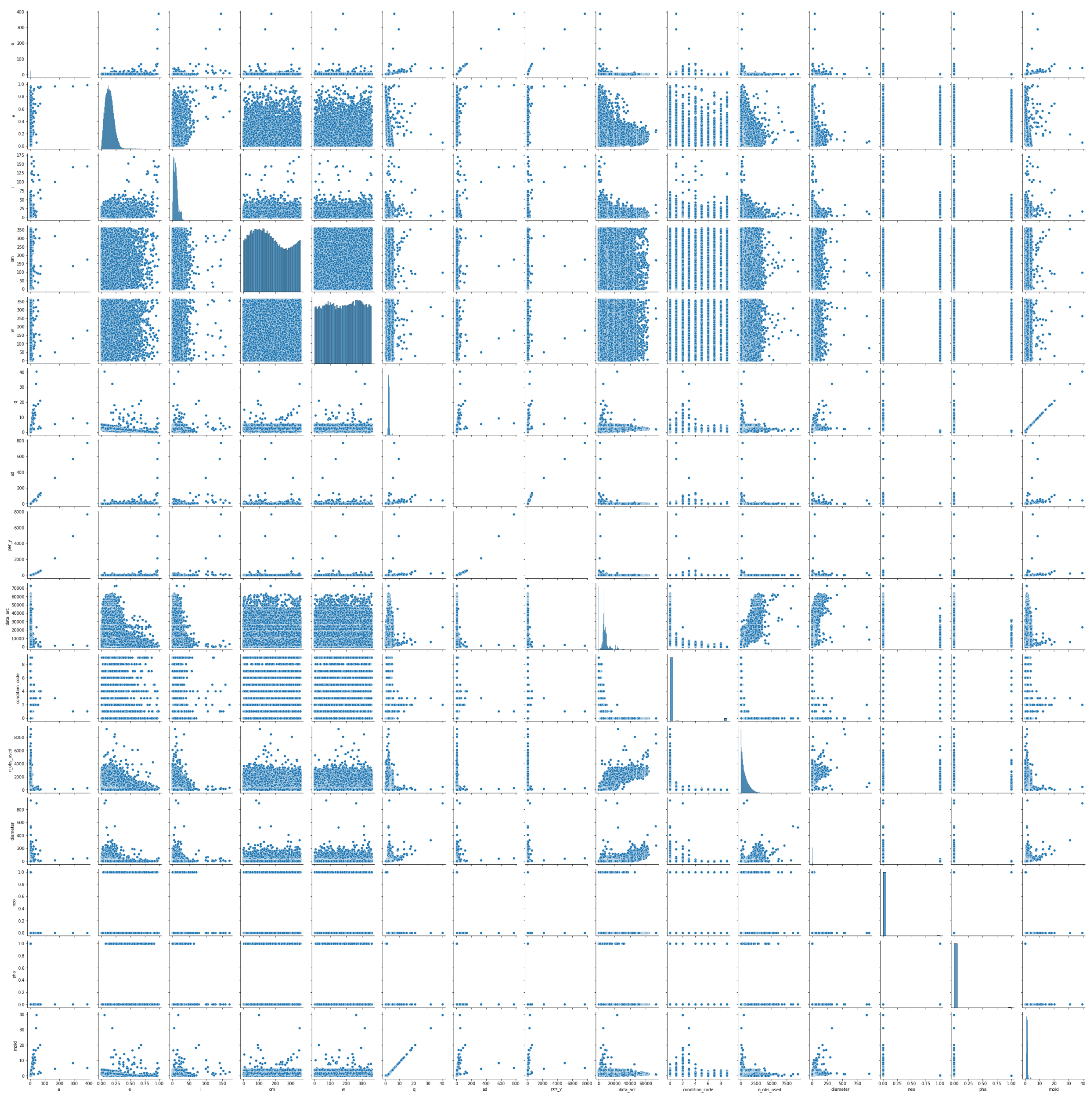
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 137540 entries, 0 to 810411
Data columns (total 15 columns):
 #   Column          Non-Null Count   Dtype  
---  ------          --------------   -----  
 0   a               137540 non-null  float64
 1   e               137540 non-null  float64
 2   i               137540 non-null  float64
 3   om              137540 non-null  float64
 4   w               137540 non-null  float64
 5   q               137540 non-null  float64
 6   ad              137540 non-null  float64
 7   per_y           137540 non-null  float64
 8   data_arc        137540 non-null  float64
 9   condition_code  137540 non-null  float64
 10  n_obs_used      137540 non-null  int64  
 11  diameter        137540 non-null  float64
 12  neo             137540 non-null  int64  
 13  pha             137540 non-null  int64  
 14  moid            137540 non-null  float64
dtypes: float64(12), int64(3)
memory usage: 16.8 MB
```

In [37]:

```
sns.pairplot(df)
```

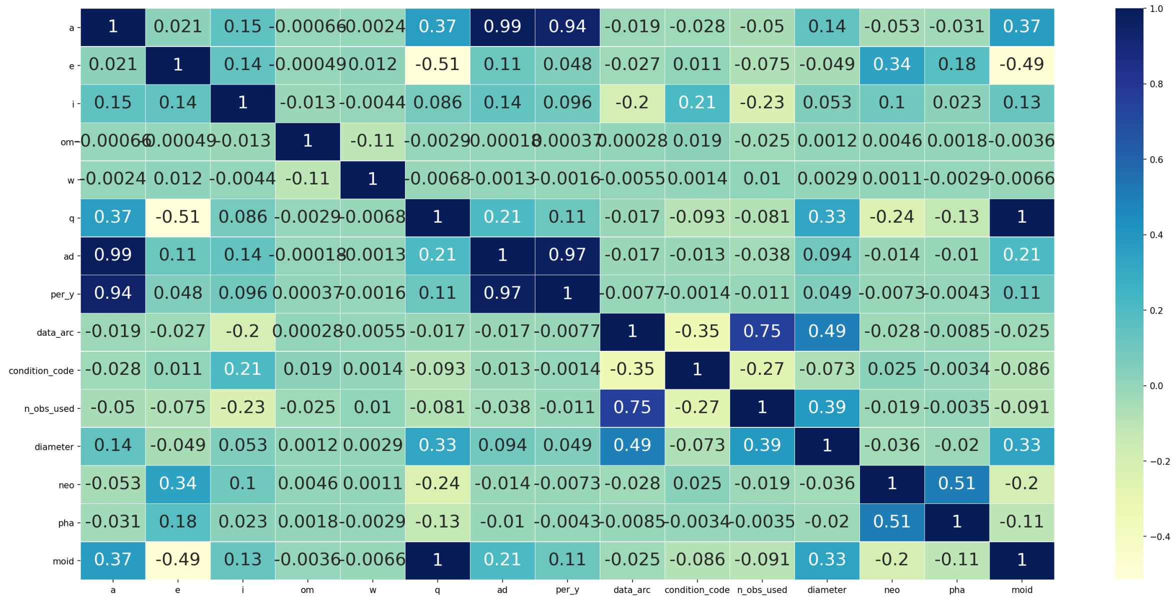
Out[37]:

```
<seaborn.axisgrid.PairGrid at 0x17c85a7f760>
```



In [57]:

```
plt.figure(figsize=(25,12),dpi=150)
sns.heatmap(df.corr(),annot=True,cmap='YlGnBu',lw=0.2,annot_kws={'size':20})
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.show()
```



	a	e	i	om	w	q	ad	per_y	data_arc	condition_code	n_obs_used	diameter	neo	pha	moid
a	1	0.021	0.15	-0.00066	-0.0024	0.37	0.99	0.94	-0.019	-0.028	-0.05	0.14	-0.053	-0.031	0.37
e	0.021	1	0.14	-0.00049	0.012	-0.51	0.11	0.048	-0.027	0.011	-0.075	-0.049	0.34	0.18	-0.49
i	0.15	0.14	1	-0.013	-0.0044	0.086	0.14	0.096	-0.2	0.21	-0.23	0.053	0.1	0.023	0.13
om	-0.00066	-0.00049	-0.013	1	-0.11	-0.0029	-0.00018	0.00037	0.00028	0.019	-0.025	0.0012	0.0046	0.0018	-0.0036
w	-0.0024	0.012	-0.0044	-0.11	1	-0.0068	-0.0013	-0.0016	-0.0055	0.0014	0.01	0.0029	0.0011	-0.0029	-0.0066
q	0.37	-0.51	0.086	-0.0029	-0.0068	1	0.21	0.11	-0.017	-0.093	-0.081	0.33	-0.24	-0.13	1
ad	0.99	0.11	0.14	-0.00018	-0.0013	0.21	1	0.97	-0.017	-0.013	-0.038	0.094	-0.014	-0.01	0.21
per_y	0.94	0.048	0.096	0.00037	-0.0016	0.11	0.97	1	-0.0077	-0.0014	-0.011	0.049	-0.0073	-0.0043	0.11
data_arc	-0.019	-0.027	-0.2	0.00028	-0.0055	-0.017	-0.017	-0.0077	1	-0.35	0.75	0.49	-0.028	-0.0085	-0.025
condition_code	-0.028	0.011	0.21	0.019	0.0014	-0.093	-0.013	-0.0014	-0.35	1	-0.27	-0.073	0.025	-0.0034	-0.086
n_obs_used	-0.05	-0.075	-0.23	-0.025	0.01	-0.081	-0.038	-0.011	0.75	-0.27	1	0.39	-0.019	-0.0035	-0.091
diameter	0.14	-0.049	0.053	0.0012	0.0029	0.33	0.094	0.049	0.49	-0.073	0.39	1	-0.036	-0.02	0.33
neo	-0.053	0.34	0.1	0.0046	0.0011	-0.24	-0.014	-0.0073	-0.028	0.025	-0.019	-0.036	1	0.51	-0.2
pha	-0.031	0.18	0.023	0.0018	-0.0029	-0.13	-0.01	-0.0043	-0.0085	-0.0034	-0.0035	-0.02	0.51	1	-0.11
moid	0.37	-0.49	0.13	-0.0036	-0.0066	1	0.21	0.11	-0.025	-0.086	-0.091	0.33	-0.2	-0.11	1

In [25]:

```
features = df.drop('diameter', axis=1)
target = df.diameter
```

In [26]:

```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, ra
```

In [27]:

```
forest = RandomForestRegressor(max_depth=32,n_estimators=50)
```

In [28]:

```
for column in df.columns:
    print(column, np.sum(df[column].isna()))
```

```
a 0
e 0
i 0
om 0
w 0
q 0
ad 0
per_y 0
data_arc 0
condition_code 0
n_obs_used 0
diameter 0
neo 0
pha 0
moid 0
```

In [ ]:

```
#Random Forsert Regresssor
```

In [34]:

```
forest.fit(X_train, np.ravel(y_train))
```

Out[34]:

```
RandomForestRegressor(max_depth=32, n_estimators=50)
```

In [35]:

```
y_pred = forest.predict(X_test)
```

In [36]:

```
r2_score(y_test,  y_pred)
```

Out[36]:

```
0.804780635717621
```

In [55]:

```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 1.2124214970776614
Mean Squared Error: 15.0206748969232
Root Mean Squared Error: 3.8756515448274245
```

In [56]:

```
df.diameter.mean()
```

Out[56]:

```
5.483639584848152
```

In [ ]:

```
# it looks like my model is good  because rmse is less than target variable mean
```