In [1]:
```python
#Task:1)To draw graph,calculate the r2 score and to predict the daily cases in the month of October
#Program By:Ayush Pandey
#Email Id:1805290@kiit.ac.in
#DATE:18-Oct-2021
#Python Version:3.7
#CAVEATS:None
#LICENSE:None
```

In [2]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```python
df=pd.read_csv('case_time_series.csv')
```

In [4]:
```python
df.head()
```

Out[4]:

	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased
0	30 January 2020	2020-01-30	1	1	0	0	0	0
1	31 January 2020	2020-01-31	0	1	0	0	0	0
2	1 February 2020	2020-02-01	0	1	0	0	0	0
3	2 February 2020	2020-02-02	1	2	0	0	0	0
4	3 February 2020	2020-02-03	1	3	0	0	0	0

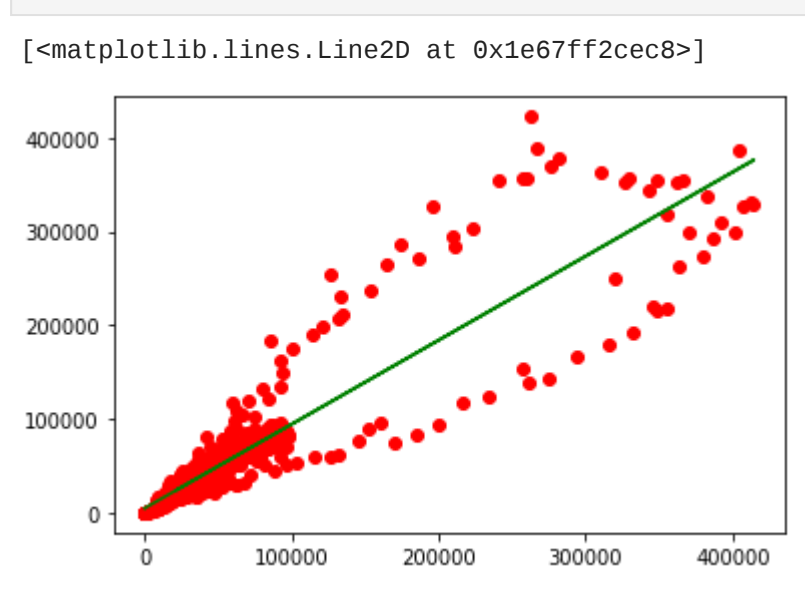
In [5]:
```python
df.isna().sum()
```

Out[5]:
```
Date               0
Date_YMD           0
Daily Confirmed    0
Total Confirmed    0
Daily Recovered    0
Total Recovered    0
Daily Deceased     0
Total Deceased     0
dtype: int64
```

In [6]:
```python
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import r2_score
```

In [7]:
```python
#A graph between Daily Confirmed and Daily Recovered Using both Linear and Polynomail Regression
model=LinearRegression()
plt.scatter(df['Daily Confirmed'],df['Daily Recovered'],color="red")
x=np.array(list(df['Daily Confirmed']))
x=x.reshape(-1,1)
model.fit(x,df['Daily Recovered'])
plt.plot(df['Daily Confirmed'],model.predict(x),color="green")
```

Out[7]: [<matplotlib.lines.Line2D at 0x1e67ff2cec8>]



In [8]:
```python
#R2 score between Daily Confirmed and Daily Recovered
r2_score(df['Daily Recovered'],model.predict(x))
```

Out[8]: 0.8445335431378322

In [9]:
```python
model1=PolynomialFeatures(degree=4)
x=np.array(list(df['Daily Confirmed']))
x=x.reshape(-1,1)
X=model1.fit_transform(x)
```

In [10]:
```python
pd.DataFrame(X)
```

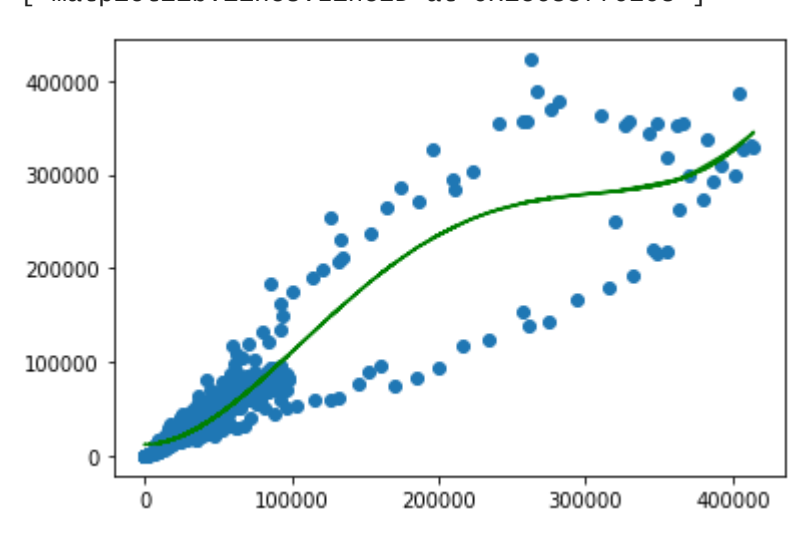
Out[10]:

	0	1	2	3	4
0	1.0	1.0	1.0	1.000000e+00	1.000000e+00
1	1.0	0.0	0.0	0.000000e+00	0.000000e+00
2	1.0	0.0	0.0	0.000000e+00	0.000000e+00
3	1.0	1.0	1.0	1.000000e+00	1.000000e+00
4	1.0	1.0	1.0	1.000000e+00	1.000000e+00
...	...	...	...	...	...
603	1.0	29565.0	874089225.0	2.584245e+13	7.640320e+17
604	1.0	28169.0	793492561.0	2.235189e+13	6.296304e+17
605	1.0	26999.0	728946001.0	1.968081e+13	5.313623e+17
606	1.0	14907.0	222218649.0	3.312613e+12	4.938113e+16
607	1.0	21898.0	479522404.0	1.050058e+13	2.299417e+17

608 rows × 5 columns

In [11]:
```python
plt.scatter(df['Daily Confirmed'],df['Daily Recovered'])
model.fit(X,df['Daily Recovered'])
plt.plot(df['Daily Confirmed'],model.predict(X),color="green")
```

Out[11]: [<matplotlib.lines.Line2D at 0x1e638ff0208>]



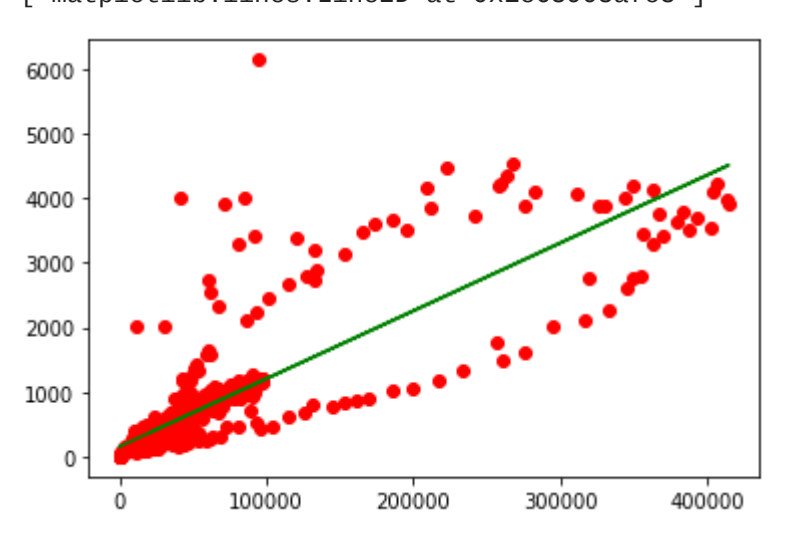
In [12]:
```python
#R2 score between Daily Confirmed and Daily Recovered
r2_score(df['Daily Recovered'],model.predict(X))
```

Out[12]: 0.8543277394343837

In [ ]:

In [13]:
```python
#A graph between Daily Confirmed and Daily Deceased Using both Linear and Polynomail Regression
model=LinearRegression()
plt.scatter(df['Daily Confirmed'],df['Daily Deceased'],color="red")
x=np.array(list(df['Daily Confirmed']))
x=x.reshape(-1,1)
model.fit(x,df['Daily Deceased'])
plt.plot(df['Daily Confirmed'],model.predict(x),color="green")
```

Out[13]: [<matplotlib.lines.Line2D at 0x1e63908afc8>]



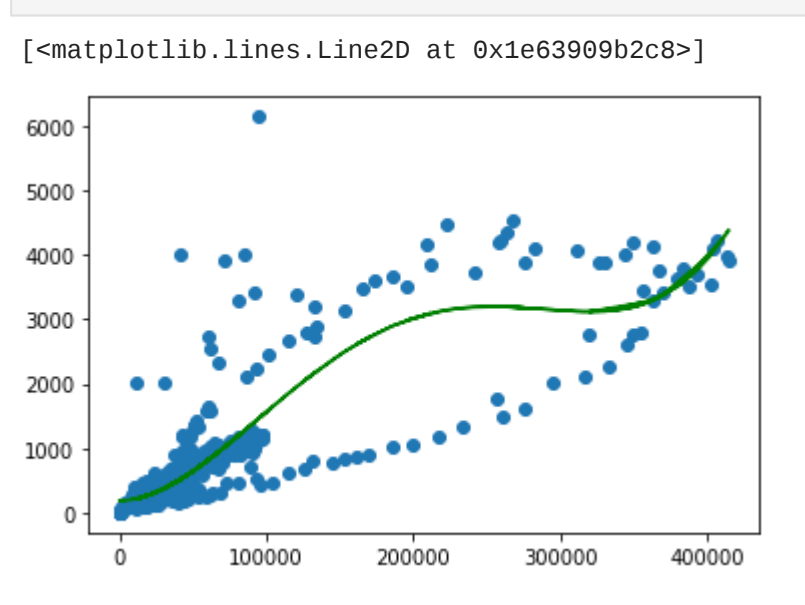
In [14]:
```python
#R2 score between Daily Confirmed and Daily Deceased
r2_score(df['Daily Recovered'],model.predict(x))
```

Out[14]: -0.45005287459197363

In [15]:
```python
model1=PolynomialFeatures(degree=4)
x=np.array(list(df['Daily Confirmed']))
x=x.reshape(-1,1)
X=model1.fit_transform(x)

plt.scatter(df['Daily Confirmed'],df['Daily Deceased'])
model.fit(X,df['Daily Deceased'])
plt.plot(df['Daily Confirmed'],model.predict(X),color="green")
```

Out[15]: [<matplotlib.lines.Line2D at 0x1e63909b2c8>]



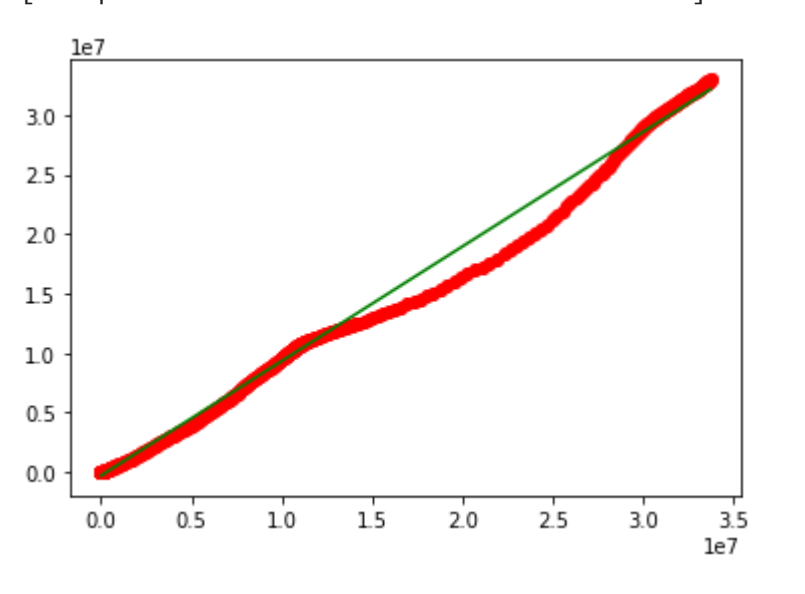
In [16]:
```python
#R2 score between Daily Confirmed and Daily Deceased
r2_score(df['Daily Deceased'],model.predict(X))
```

Out[16]: 0.7296976056092532

In [ ]:

In [17]:
```python
#A graph between Total Confirmed and Total Recovered Using both Linear and Polynomail Regression
model=LinearRegression()
plt.scatter(df['Total Confirmed'],df['Total Recovered'],color="red")
x=np.array(list(df['Total Confirmed']))
x=x.reshape(-1,1)
model.fit(x,df['Total Recovered'])
plt.plot(df['Total Confirmed'],model.predict(x),color="green")
```

Out[17]: [<matplotlib.lines.Line2D at 0x1e639130d48>]



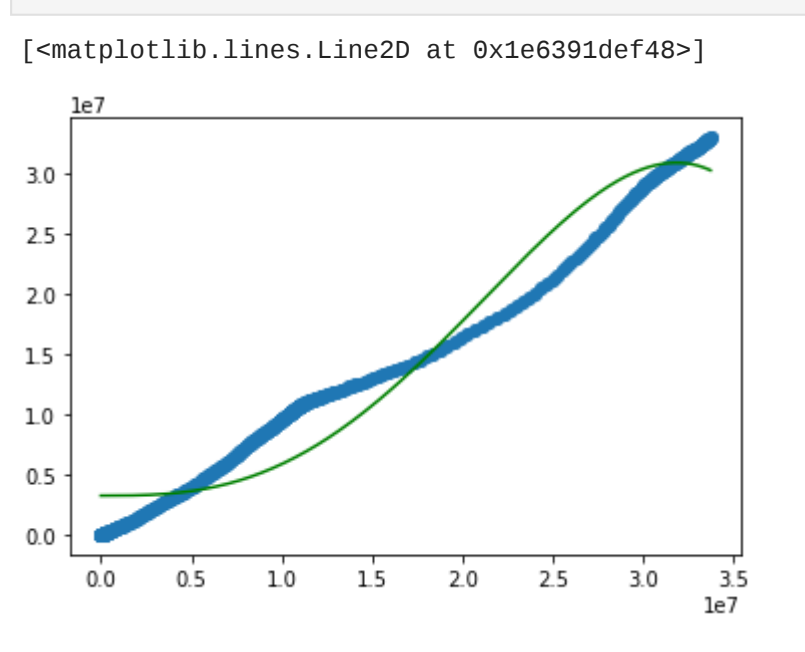
In [18]:
```python
#R2 score between Total Confirmed and Total Recovered
r2_score(df['Total Recovered'],model.predict(x))
```

Out[18]: 0.9960664838910648

In [19]:
```python
model1=PolynomialFeatures(degree=4)
x=np.array(list(df['Total Confirmed']))
x=x.reshape(-1,1)
X=model1.fit_transform(x)

plt.scatter(df['Total Confirmed'],df['Total Recovered'])
model.fit(X,df['Total Recovered'])
plt.plot(df['Total Confirmed'],model.predict(X),color="green")
```

Out[19]: [<matplotlib.lines.Line2D at 0x1e6391def48>]



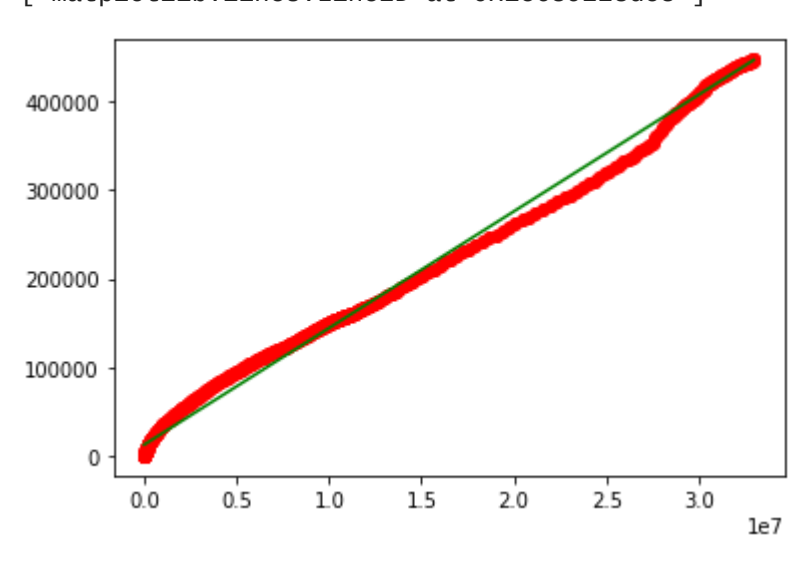
In [20]:
```python
#R2 score between Total Confirmed and Total Recovered
r2_score(df['Total Recovered'],model.predict(X))
```

Out[20]: 0.9355944339804353

In [ ]:

In [21]:
```python
#A graph between Total Recovered and Total Deceased using both linear and polynomial regression
model=LinearRegression()
x=np.array(list(df['Total Recovered']))
x=x.reshape(-1,1)
model.fit(x,df['Total Deceased'])
plt.scatter(df['Total Recovered'],df['Total Deceased'],color="red")
plt.plot(df['Total Recovered'],model.predict(x),color="green")
```

Out[21]: [<matplotlib.lines.Line2D at 0x1e639223dc8>]



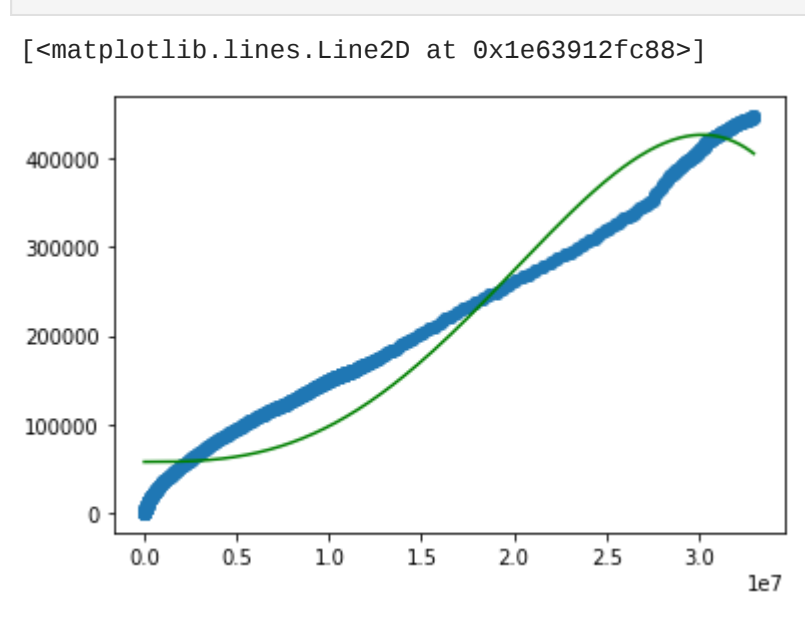
In [22]:
```python
#R2 score between Total Confirmed and Total Deceased
r2_score(df['Total Deceased'],model.predict(x))
```

Out[22]: 0.9955158458262664

In [23]:
```python
model1=PolynomialFeatures(degree=4)
x=np.array(list(df['Total Recovered']))
x=x.reshape(-1,1)
X=model1.fit_transform(x)

plt.scatter(df['Total Recovered'],df['Total Deceased'])
model.fit(X,df['Total Deceased'])
plt.plot(df['Total Recovered'],model.predict(X),color="green")
```

Out[23]: [<matplotlib.lines.Line2D at 0x1e63912fc88>]



In [24]:
```python
#R2 score between Total Confirmed and Total Deceased
r2_score(df['Total Deceased'],model.predict(X))
```

Out[24]: 0.9150780844264632

In [ ]:

In [25]:
```python
df.dtypes
```

Out[25]:
```
Date               object
Date_YMD           object
Daily Confirmed     int64
Total Confirmed     int64
Daily Recovered     int64
Total Recovered     int64
Daily Deceased      int64
Total Deceased      int64
dtype: object
```

In [26]:
```python
import datetime as dt
df['Date_YMD']=pd.to_datetime(df['Date_YMD'])
df['Month']=df['Date_YMD'].dt.month
df['Year']=df['Date_YMD'].dt.year
```

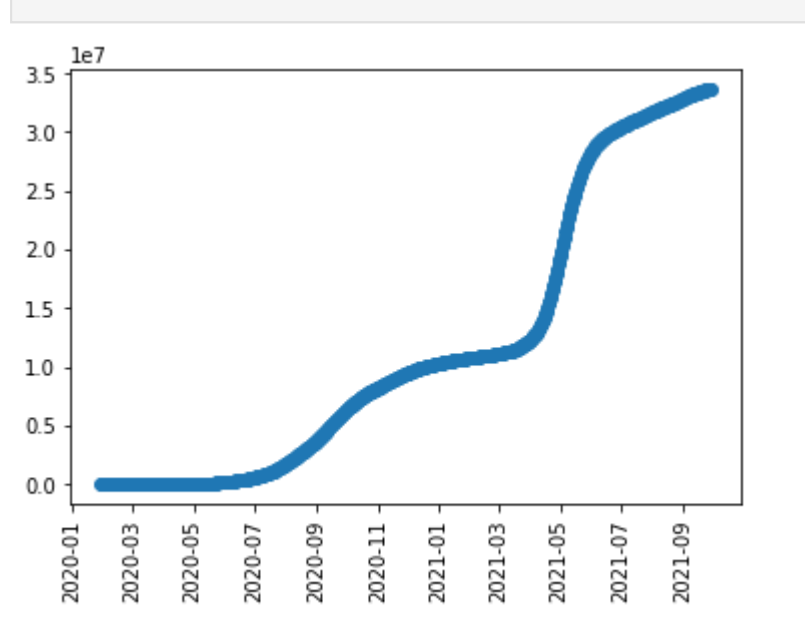
In [27]:
```python
df.head(2)
```

Out[27]:

	Date	Date_YMD	Daily Confirmed	Total Confirmed	Daily Recovered	Total Recovered	Daily Deceased	Total Deceased	Month	Year
0	30 January 2020	2020-01-30	1	1	0	0	0	0	1	2020
1	31 January 2020	2020-01-31	0	1	0	0	0	0	1	2020

In [28]:
```python
df1=df.groupby(['Month','Year'],as_index=False).agg({'Daily Confirmed':'sum','Total Confirmed':'sum','Daily Recovered':'sum','Total Recovered':'su
```

In [29]:
```python
plt.scatter(df['Date_YMD'],df['Total Confirmed'])
plt.xticks(rotation=90)
plt.show()
```



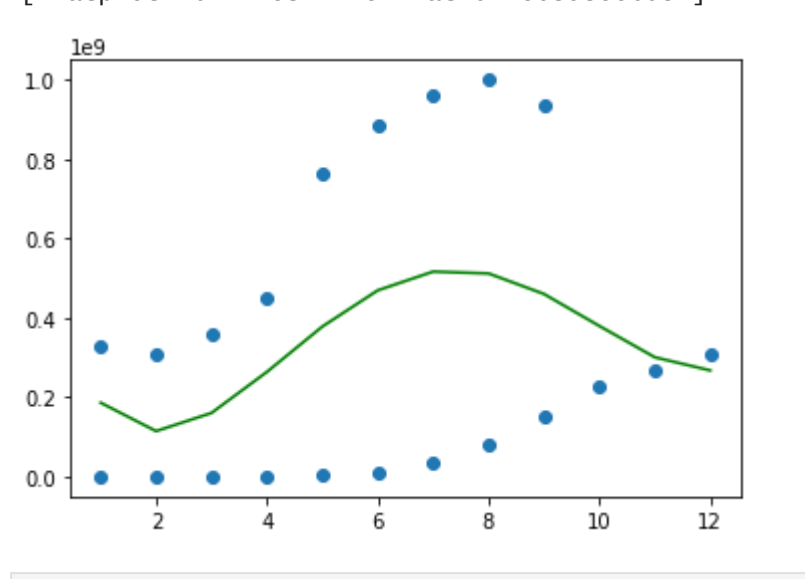
In [30]:
```python
x=np.array(list(df1['Month']))
x=x.reshape(-1,1)
```

In [31]:
```python
#Total Confirmed and Month Graph
model1=PolynomialFeatures(degree=4)
X=model1.fit_transform(x)

model=LinearRegression()
#model.fit(X,df['Daily Confirmed'])

plt.scatter(df1['Month'],df1['Total Confirmed'])
model.fit(X,df1['Total Confirmed'])
plt.plot(df1['Month'],model.predict(X),color="green")
```

Out[31]: [<matplotlib.lines.Line2D at 0x1e6393ccb08>]



In [32]:
```python
r2_score(df1['Total Confirmed'],model.predict(X))
```

Out[32]: 0.16371990124124902

In [33]:
```python
#Prediction of Total Confirmed in the Month of October
ans=model.predict(model1.transform([[10]]))
ans
```

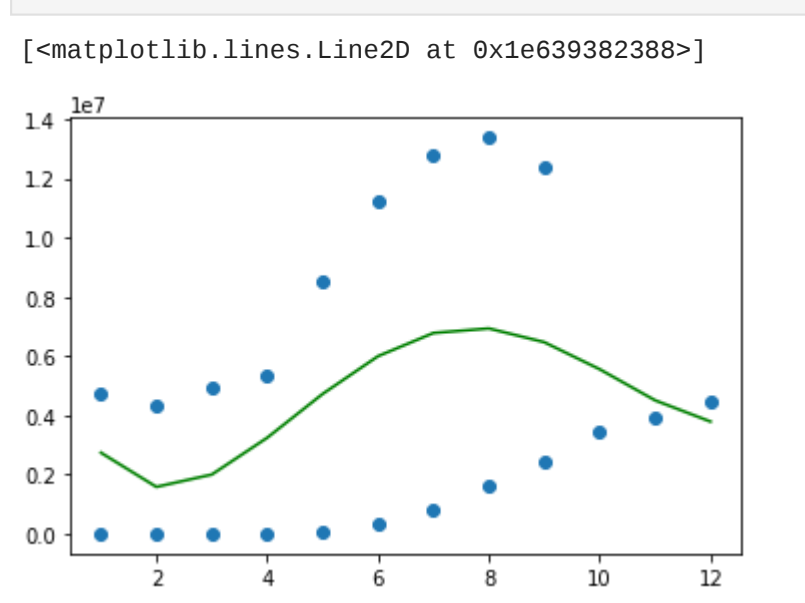
Out[33]: array([3.79291387e+08])

In [ ]:

In [34]:
```python
#Regression between Month and Total Deceased
model1=PolynomialFeatures(degree=4)
X=model1.fit_transform(x)

model=LinearRegression()
plt.scatter(df1['Month'],df1['Total Deceased'])
model.fit(X,df1['Total Deceased'])
plt.plot(df1['Month'],model.predict(X),color="green")
```

Out[34]: [<matplotlib.lines.Line2D at 0x1e639382388>]



In [35]:
```python
r2_score(df1['Total Deceased'],model.predict(X))
```

Out[35]: 0.1799843736015646

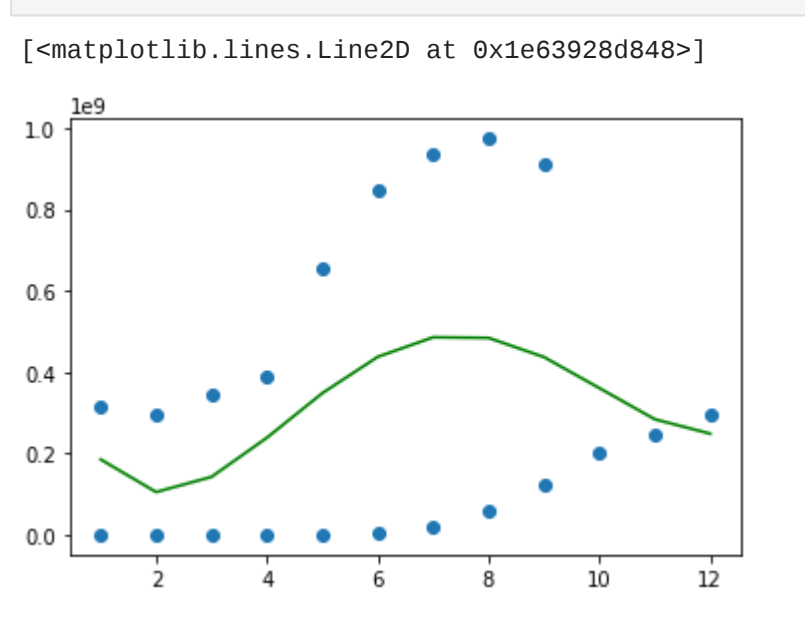
In [36]:
```python
#Predicting the Total Deceased in the month of October
ans=model.predict(model1.transform([[10]]))
ans
```

Out[36]: array([5553160.03657116])

In [37]:
```python
#Regression between Month and Total Recovered
model1=PolynomialFeatures(degree=4)
X=model1.fit_transform(x)

model=LinearRegression()
plt.scatter(df1['Month'],df1['Total Recovered'])
model.fit(X,df1['Total Recovered'])
plt.plot(df1['Month'],model.predict(X),color="green")
```

Out[37]: [<matplotlib.lines.Line2D at 0x1e63928d848>]



In [38]:
```python
r2_score(df1['Total Recovered'],model.predict(X))
```

Out[38]: 0.1590906744798436

In [39]:
```python
#Predicting the Total Recovered in the month of October
ans=model.predict(model1.transform([[10]]))
ans
```

Out[39]: array([3.60559382e+08])

In [ ]: