In [77]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

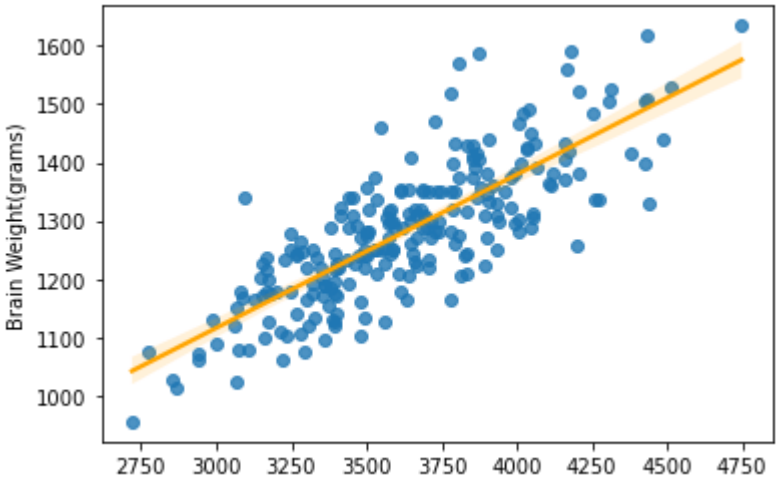
In [56]:
```
df = pd.read_csv('headbrain.csv')
```

In [57]:
```
X = df['Head Size(cm^3)'].values
Y = df['Brain Weight(grams)']
```

In [58]:
```
mean_x = X.mean()
mean_y = Y.mean()
n = len(X)
nume = 0
denom = 0

for i in range(n):
    nume += (X[i] - mean_x) * (Y[i] - mean_y)
    denom += (X[i] - mean_x) ** 2
b1 = nume / denom
b0 = mean_y  - (b1 * mean_x)

min_x = X.min() - 100
min_y = Y.min() + 100

x = np.linspace(min_x, min_y, 1000)
y = b0 + b1 * x
```

In [59]:
```
sns.regplot(x = X, y = Y, line_kws = {'color': 'orange'})
```

Out[59]:
```
<AxesSubplot:ylabel='Brain Weight(grams)'>
```



In [69]:
```
X  = X.reshape(-1,1)
model = LinearRegression()
model.fit(X,Y)
```

Out[69]:
```
LinearRegression()
```

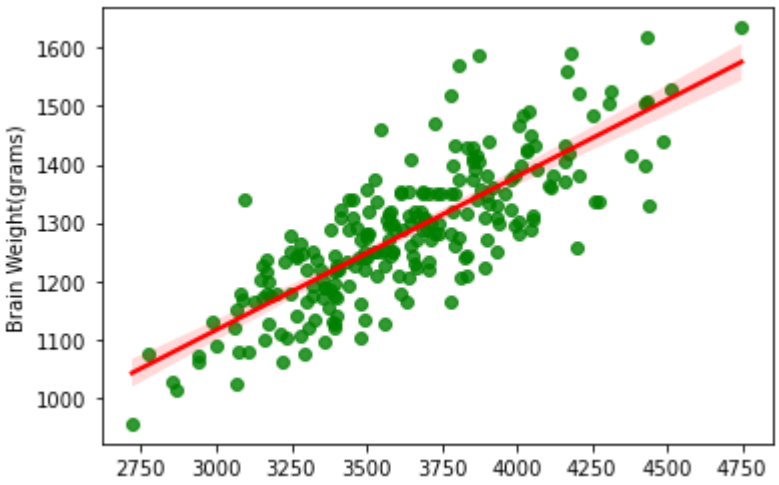
In [70]:
```
pred = model.predict(X)
```

In [71]:
```
mse = mean_squared_error(Y, pred)
rmse = np.sqrt(mse)
print(rmse)
```

```
72.1206213783709
```

In [72]:
```
sns.regplot(x = X, y = Y, color = 'g',line_kws = {'color': 'red'})
```

Out[72]:
```
<AxesSubplot:ylabel='Brain Weight(grams)'>
```



In [74]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 237 entries, 0 to 236
Data columns (total 4 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Gender               237 non-null    int64
 1   Age Range            237 non-null    int64
 2   Head Size(cm^3)      237 non-null    int64
 3   Brain Weight(grams)  237 non-null    int64
dtypes: int64(4)
memory usage: 7.5 KB
```

In [78]:
```
X = df[['Gender', 'Age Range', 'Head Size(cm^3)']].values
Y = df['Brain Weight(grams)'].values
```

```
Mean Squared Error (MSE): 4350.00
R-squared (R2): 0.73
Coefficients: [-14.44971698 -23.65458696   0.24795719]
Intercept: 436.4466363048025
```

In [79]:
```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,
```

In [80]:
```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[80]:
```
LinearRegression()
```

In [81]:
```
y_pred = model.predict(X_test)

# Calculate Mean Squared Error (MSE) and R-squared (R2) for evaluation
```

```
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error (MSE): {mse:.2f}')
print(f'R-squared (R2): {r2:.2f}')
```

```
Mean Squared Error (MSE): 4350.00
R-squared (R2): 0.73
```

In [82]:
```
coefficients = model.coef_
intercept = model.intercept_

print(f'Coefficients: {coefficients}')
print(f'Intercept: {intercept}')
```

```
Coefficients: [-14.44971698 -23.65458696   0.24795719]
Intercept: 436.4466363048025
```

In [ ]:
```

```