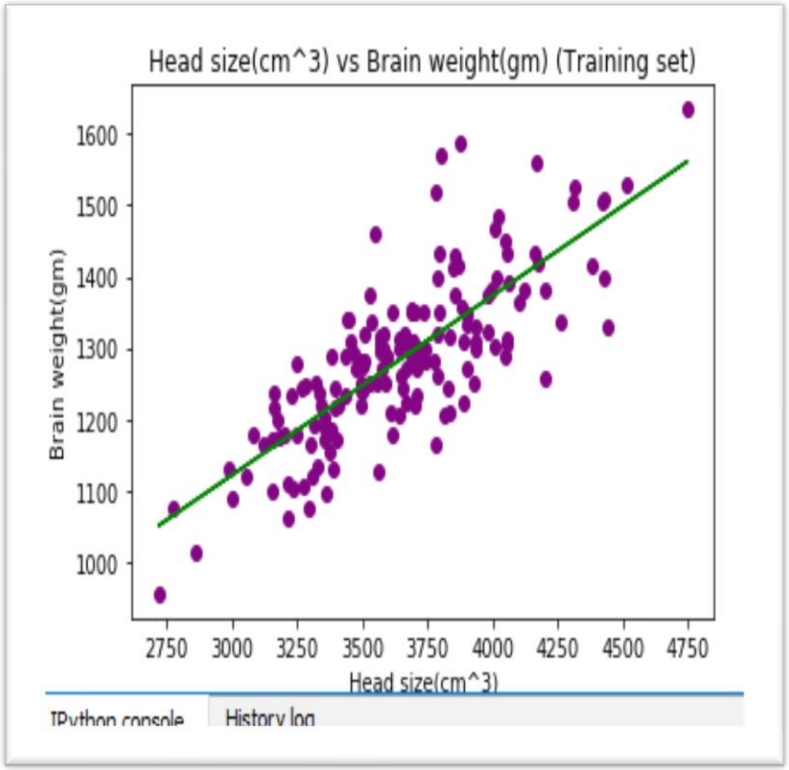
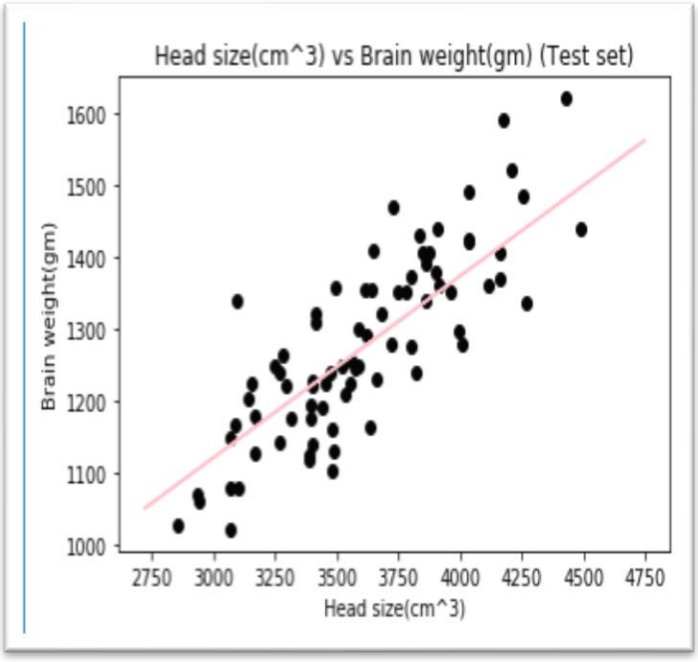
Perform linear and polynomial regression is up to you on the given dataset and predict brain weight from head size.

```
# simple Linear Regression

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
data = pd.read_csv('dataset.csv')
X_head_size = data.iloc[:, 2:3].values
y_brain_wgt = data.iloc[:, 3].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_head_size_train, X_head_size_test, y_brain_wgt_train, y_brain_wgt_test = train_test_split(X_head_size, y_brain_wgt, test_size = 1/3, random_state = 0)

# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_head_size_train, y_brain_wgt_train)

# Predicting the Test set results
y_brain_wgt_pred = regressor.predict(X_head_size_test)

# Visualising the Training set results
plt.scatter(X_head_size_train, y_brain_wgt_train, color = 'purple')
plt.plot(X_head_size_train, regressor.predict(X_head_size_train), color = 'green')
plt.title('Head size(cm^3) vs Brain weight(gm) (Training set)')
plt.xlabel('Head size(cm^3)')
plt.ylabel('Brain weight(gm)')
plt.show()

# Visualising the Test set results
plt.scatter(X_head_size_test, y_brain_wgt_test, color = 'black')
plt.plot(X_head_size_train, regressor.predict(X_head_size_train), color = 'pink')
plt.title('Head size(cm^3) vs Brain weight(gm) (Test set)')
plt.xlabel('Head size(cm^3)')
plt.ylabel('Brain weight(gm)')
plt.show()

```

Name	Type	Size	Value
X_head_size	int64	(237, 1)	[[4512] [3738]
X_head_size_test	int64	(79, 1)	[[3724] [3680]
X_head_size_train	int64	(158, 1)	[[3777] [3302]
data	DataFrame	(237, 4)	Column names: Gender, Age Range, Head Size(cm^3), Brain Weight(grams)
y_brain_wgt	int64	(237,)	[1530 1297 1335 ... 1104 1170 1120]
y_brain_wgt_pred	float64	(79,)	[1303.83322923 1292.73537163 1381.5182324 ... 1105.3329127 1363.8625
y_brain_wgt_test	int64	(79,)	[1280 1321 1425 ... 1070 1350 1522]
y_brain_wgt_train	int64	(158,)	[1282 1165 1635 ... 1270 1215 1316]

Variable explorer | File explorer | Help





CLASS TASK:

Create two random arrays A and B, and multiply them. Get their result in C and add 1 to every element of C.

```
: import numpy as np
  # creating random arrays
  A=np.random.randn (4,2)
  B=np.random.randn (4,2)
  print ("ARRAY A")
  print(A)
  print ("ARRAY B")
  print(B)

  #multiplication
  print ("MULTIPLICATION OF TWO ARRAYS:")
  C=np.multiply(A,B)
  print(C)

  #defining an array
  ones=np.ones((4,2))

  #adding array "one" to array C
  result=np.add(ones,C)
  print (result)
```

```
ARRAY A
[[-1.97320748 -0.4816845 ]
 [-2.10225053  1.47313205]
 [ 1.16031092  0.4676745 ]
 [ 2.03293034  0.05263926]]
ARRAY B
[[ 0.38269065 -0.44936192]
 [ 0.46846166 -1.24521762]
 [-0.82526597  1.72159511]
 [ 0.15910822  1.11927411]]
MULTIPLICATION OF TWO ARRAYS:
[[-0.75512806  0.21645067]
 [-0.98482377 -1.83436998]
 [-0.95756511  0.80514614]
 [ 0.32345592  0.05891777]]
[[ 0.24487194  1.21645067]
 [ 0.01517623 -0.83436998]
 [ 0.04243489  1.80514614]
 [ 1.32345592  1.05891777]]
```