4.1 Given the brain weight and body weight , its been seen that there exists a relation. You as a data scientist are required to fit the data to a line and plot a linear regression graph. Display the intercept and slope value. Perform prediction for new data - brain weight = 15.

4.2 Predict the the petrol consumption , when given the details average income, paved highways, petrol tax and population data. Apply linear regression to find the intercept.

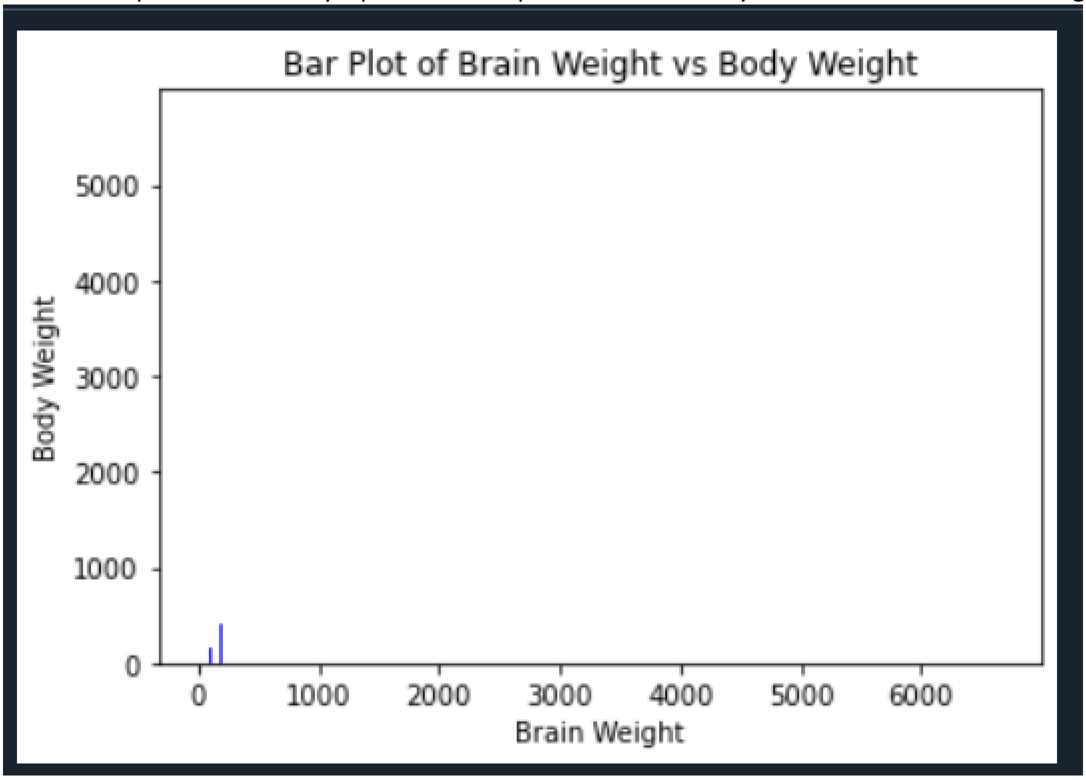
```
print("-----* Question 1 *-----")

#Import Some Required Libraries

import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
#%matplotlib inline
import seaborn as sns

#Read The Data
data = pd.read_csv(r"C:\Users\LENOVO\.spyder-py3\breain_weight.csv")
print(data)

#Drop Function
#data1 = data.drop(["Body Weight"])
#print(data1)

#Data Print top 5 Recors
print(data.head(5))

#Data Print last 5 Recors
print(data.tail(5))

#Slicing
print(data[20:30])

#Statistical Information
```

```
data.info()


#Statistical Information

data.describe()


#Single Value

print(data.count())


#Checking the null value

data.isna().sum()


data["Brain Weight"].fillna(0, inplace=True)

print(data.head(15))


data["Brain Weight"].isna().sum()


#Bar Plot

x = data["Brain Weight"]

y = data["Body Weight"]

plt.bar(x, y, color="Blue")

plt.title("Bar Plot of Brain Weight vs Body Weight")

plt.xlabel("Brain Weight")

plt.ylabel("Body Weight")

plt.show()


#Scatter Plot

plt.scatter(data["Brain Weight"], data["Body Weight"], color="Green")

plt.title("Scatter Plot of Brain Weight vs Body Weight")

plt.xlabel("Brain Weight")

plt.xlabel("Body Weight")

plt.show()

plt.savefig("ScatterPlot.png")


#Seaborn Use

sns.set(rc={"figure.figsize": (8, 4)}); np.random.seed(0)

x = np.random.randn(100)

ax = sns.distplot(x)
```

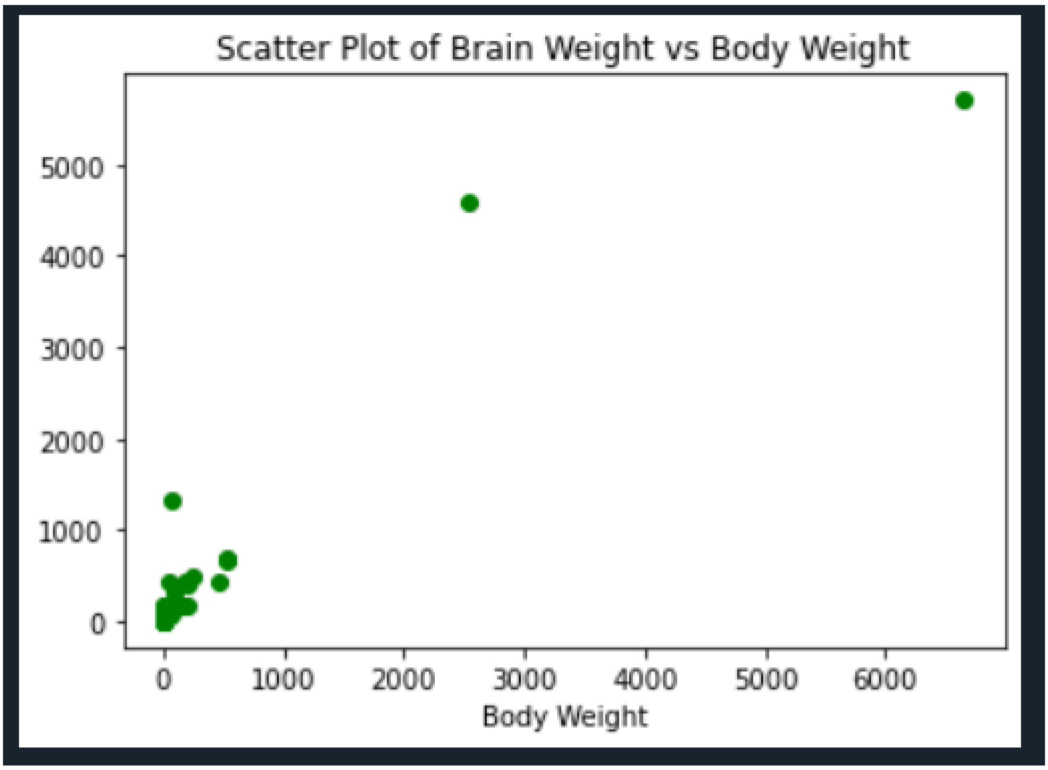
```
plt.show()


#box plot

sns.boxplot(data["Brain Weight"], data["Body Weight"])


#training and testing dataset.

train = data[:40]

print(train)


test = data[40:]

print(test)


#Model, Fitting

#Linear Regression

from sklearn.linear_model import LinearRegression


model = LinearRegression().fit(data[["Brain Weight"]], data[["Body Weight"]])


#Intercept

print("Intercept : ", model.intercept_)


#Slope - Coefficient

print("Slope : ", model.coef_)


#Determination of coefficient

print("Determination of Coefficient : ", model.score(data[["Brain Weight"]], data[["Body Weight"]]))


#Prediction : Brain Weight = 15

model.predict([[15]])



print("-----* Question 2 *-----")

# Predict the the petrol consumption , when given the details average income, paved highways, petrol tax and population data. Apply linear regression to find the intercept.


#Predict the the petrol consumption
```

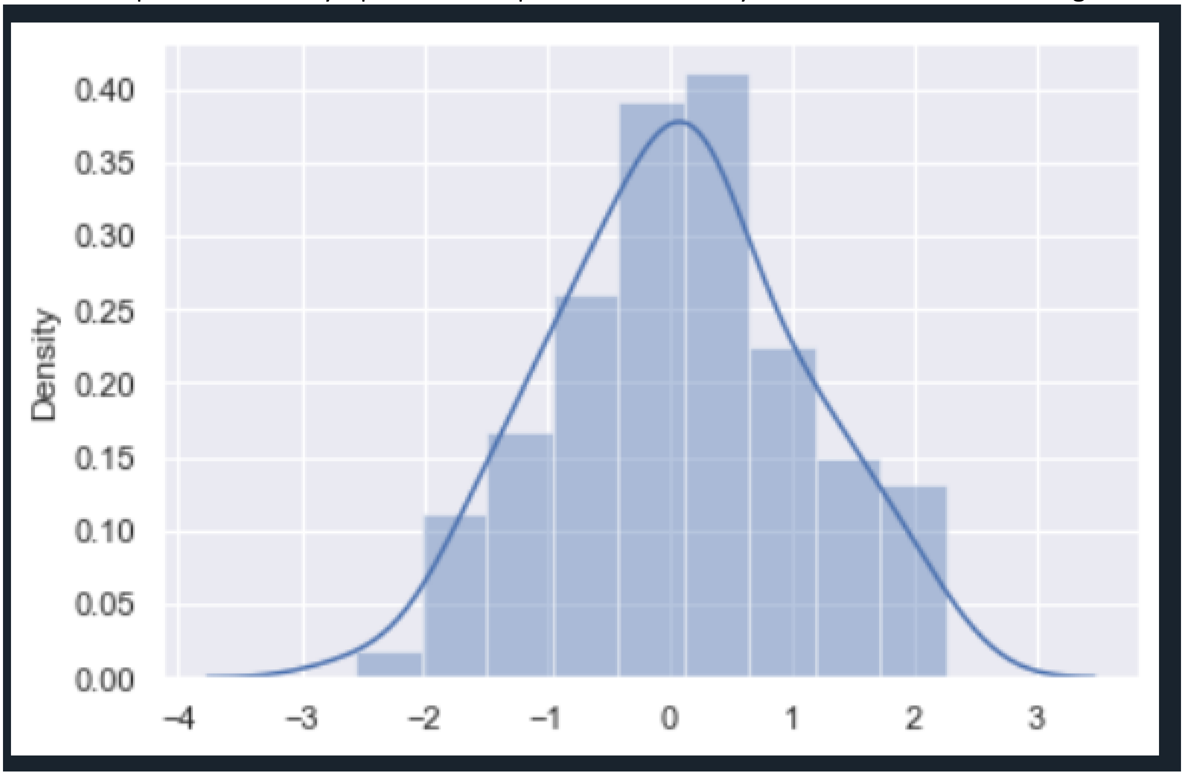
```
from sklearn.linear_model import LinearRegression


data1 = pd.read_csv(r"C:\Users\LENOVO\.spyder-py3\petrol_consumption.csv")

data1.columns=['Petrol_tax','Average_income','Paved_Highways','Population_Driver_licence','Petrol_Consumption']

print(data1.head(10))

print(data1.tail(10))


data1.describe()


model1 = LinearRegression().fit(data1[["Petrol_tax"]], data1[["Average_income"]])


#Prediction : Petrol Consumption

model1.predict(data1[['Petrol_Consumption']])


#Intercept

print("Intercept : ", model1.intercept_)


#Slooe - Coefficient

print("Slope : ", model1.coef_)
```

Output :

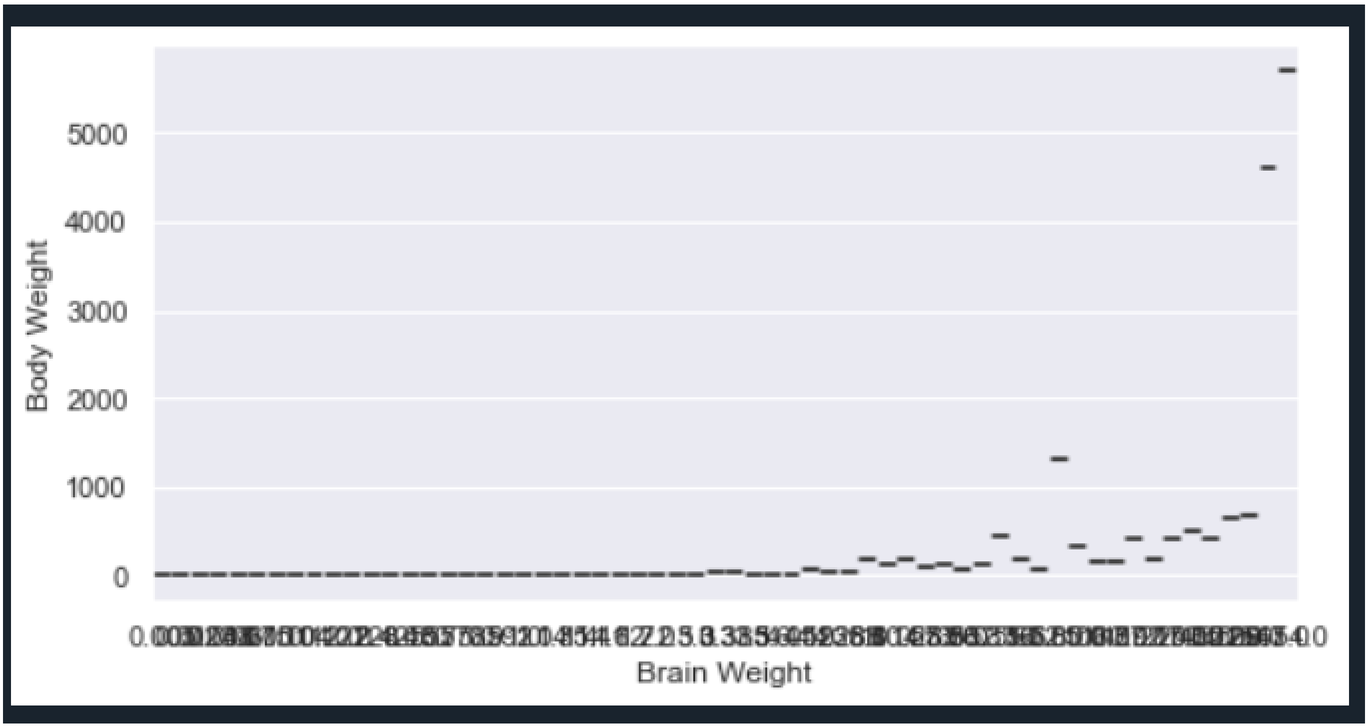

Bar Plot of Brain Weight vs Body Weight
Body Weight
0
1000
2000
3000
4000
5000
0
1000
2000
3000
4000
5000
6000
Brain Weight

4


Scatter Plot of Brain Weight vs Body Weight
0
1000
2000
3000
4000
5000
0
1000
2000
3000
4000
5000
6000
Body Weight



Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−3
−2
−1
0
1
2
3



Body Weight
5000
4000
3000
2000
1000
0
Brain Weight

	Index	Brain Weight	Body Weight
0	1	3.385	44.50
1	2	0.480	15.50
2	3	1.350	8.10
3	4	465.000	423.00
4	5	36.330	119.50
5	6	27.660	115.00
6	7	14.830	98.20
7	8	1.040	5.50
8	9	4.190	58.00
9	10	0.425	6.40
10	11	0.101	4.00
11	12	0.920	5.70
12	13	1.000	6.60
13	14	0.005	0.14
14	15	0.060	1.00
15	16	3.500	10.80
16	17	2.000	12.30
17	18	1.700	6.30
18	19	2547.000	4603.00
19	20	0.023	0.30
20	21	187.100	419.00
21	22	521.000	655.00
22	23	0.785	3.50
23	24	10.000	115.00
24	25	3.300	25.60
25	26	0.200	5.00
26	27	1.410	17.50
27	28	529.000	680.00
28	29	207.000	406.00
29	30	85.000	325.00
30	31	0.750	12.30
31	32	62.000	1320.00