In [1]:

```
# Importing Necessary Libraries
```

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [3]:

```
# Reading the CSV File which is stored in local
```

In [4]:

```
df = pd.read_csv('C:/Users/sanjith/Desktop/suicide_rate.csv')
```

In [5]:

```
# Understanding the Data
```

In [6]:

```
df.shape
```

Out[6]:

```
(140, 3)
```

In [7]:

```
df.head(10)
```

Out[7]:

	Country	GDP per capita	Suicide rate
0	Afghanistan	1,105	20.6
1	Albania	4,932	32.4
2	Algeria	4,534	15.6
3	Andorra	46,622	18.9
4	Angola	2,077	20.5
5	Antigua and Barbuda	14,447	18.8
6	Argentina	14,647	18.5
7	Armenia	3,456	39.7
8	Australia	59,542	10.7
9	Austria	61,906	11.5

In [8]:

```
df.tail(10)
```

Out[8]:

	Country	GDP per capita	Suicide rate
130	Paraguay	4,527	15.0
131	Peru	6,927	12.7
132	Philippines	3,150	11.3
133	Poland	18,746	20.3
134	Portugal	24,027	13.2
135	Spain	29,542	11.5
136	Switzerland	83,832	12.9
137	United Kingdom	39,720	11.1
138	United States	65,112	14.0
139	Thailand	6,792	27.8

In [9]:

```
df.dtypes
```

Out[9]:

```
Country            object
GDP per capita     object
Suicide rate      float64
dtype: object
```

In [10]:

```
df.describe()
```

Out[10]:

	Suicide rate
count	140.000000
mean	15.982143
std	5.454182
min	9.600000
25%	12.500000
50%	14.400000
75%	18.325000
max	39.700000

In [11]:

```
df.columns
```

Out[11]:

```
Index(['Country', 'GDP per capita', 'Suicide rate'], dtype='object')
```

In [12]:

```
# Cleaning and Pre-Processing the data
```

In [13]:

```
df['GDP per capita'] = df['GDP per capita'].str.replace(',', '').astype(float)
```

In [14]:

```
df.dtypes
```

Out[14]:

```
Country            object
GDP per capita    float64
Suicide rate      float64
dtype: object
```

In [15]:

```
df = df.rename(columns = {'GDP per capita': 'GDP/capita', 'Suicide rate':'Suicide_rate' }
```

In [16]:

```
df.isna().sum()
```

Out[16]:

```
Country         0
GDP/capita      0
Suicide_rate    0
dtype: int64
```

In [17]:

```
# Defining features and Target values
```

In [18]:

```
X = df['GDP/capita']
y = df['Suicide_rate']
```

In [19]:

```
# Splitting data for train and test
```

In [20]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [21]:

```
# Model fitting
```

In [22]:

```
X_train = X_train.values.reshape(-1, 1)
X_test = X_test.values.reshape(-1, 1)

# Create an instance of the LinearRegression model
model = LinearRegression()

# Train the model using the training data
model.fit(X_train, y_train)
```

Out[22]:

```
▾ LinearRegression
LinearRegression()
```

In [23]:

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

In [24]:

```
# Finding the metrics of the obtained model
```

In [25]:

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


print("Mean Squared Error (MSE):", mse)
print("R-squared Score:", r2)


coefficient = model.coef_[0]


intercept = model.intercept_

print("Coefficient:", coefficient)
print("Intercept:", intercept)
```

```
Mean Squared Error (MSE): 18.833346508804684
R-squared Score: -0.014563171692035803
Coefficient: -4.5899479276487904e-05
Intercept: 16.87811205410332
```

In [26]:

```
# Giving sample data to the model
```

In [27]:

```
new_data = np.array([100000, 80301, 15007])
new_data = new_data.reshape(-1, 1)


predictions = model.predict(new_data)


for gdp, pred in zip(new_data, predictions):
    print("GDP/Capita:", gdp[0])
    print("Predicted Suicide Rate:", pred)
    print()
```

```
GDP/Capita: 100000
Predicted Suicide Rate: 12.288164126454529

GDP/Capita: 80301
Predicted Suicide Rate: 13.192337968722065

GDP/Capita: 15007
Predicted Suicide Rate: 16.189298568601064
```