```
<seaborn.axisgrid.PairGrid at 0x7c50469e93f0>
```

Linear Regression Model

```
1 X = led_df.drop('BMI', axis=1)
2 y = led_df['BMI']
```

```
1 print("X",X.shape,"\ny=",y.shape)
```

```
X (2938, 21)
y= (2938,)
```

Train Test Splitting

```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.4.0)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import hvplot.pandas
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.linear_model import LinearRegression
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

```
X_train.shape
```

```
(2056, 21)
```

```
X_test.shape
```

```
(882, 21)
```

```
model = LinearRegression()
```

```
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Model Evaluation

```
model.coef_
```

```
array([ 7.57056771e-03, -4.63411687e-02, -2.07939848e+00,  5.56411501e-01,
       -6.24185007e-03, -3.41735872e-03, -1.14684826e-01,  3.06103000e-04,
        2.83703325e-02, -1.07978944e-04,  9.05328887e-03, -7.11044654e-04,
        3.56594950e-01, -1.75437562e-02,  1.36511969e-01, -8.45656970e-05,
        6.05489000e-09, -4.91879831e-01, -1.07257902e+00,  4.62048803e+00,
        1.01030665e+00])
```

```
pd.DataFrame(model.coef_, X.columns, columns=['Coedicients'])
```

	Coedicients
Country	7.570568e-03
Year	-4.634117e-02
Status	-2.079398e+00
Life expectancy	5.564115e-01
Adult Mortality	-6.241850e-03
infant deaths	-3.417359e-03
Alcohol	-1.146848e-01
percentage expenditure	3.061030e-04
Hepatitis B	2.837033e-02
Measles	-1.079789e-04
under-five deaths	9.053289e-03
Polio	-7.110447e-04
Total expenditure	3.565949e-01
Diphtheria	-1.754376e-02
HIV/AIDS	1.365120e-01
GDP	-8.456570e-05
Population	6.054890e-09
thinness 1-19 years	-4.918798e-01
thinness 5-9 years	-1.072579e+00
Income composition of resources	4.620488e+00
Schooling	1.010307e+00

Predictions from our Model

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

Regression Evaluation Metrics

```
MAE = metrics.mean_absolute_error(y_test, y_pred)
MSE = metrics.mean_squared_error(y_test, y_pred)
RMSE = np.sqrt(MSE)
```

```
MAE
```

```
10.684448166792613
```

```
MSE
```

```
212.90158397376152
```

```
RMSE
```

```
14.591147452265758
```

```
led_df['BMI'].mean()
```

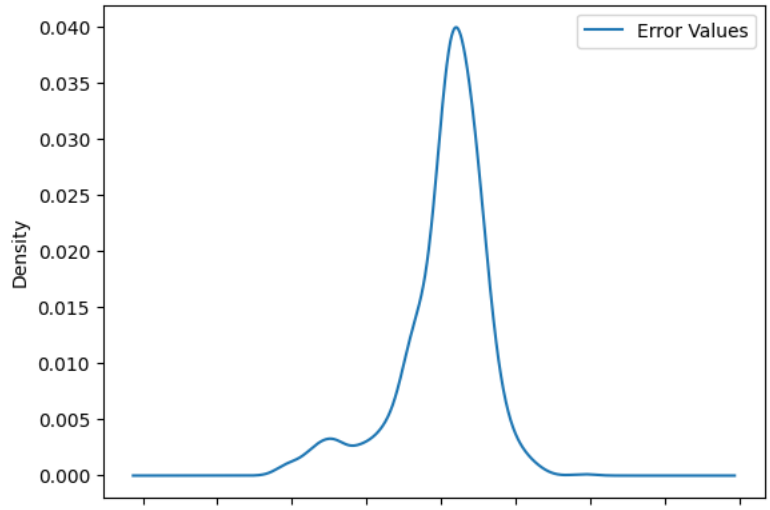
```
38.02014976174268
```

Residual Histogram

```
test_residual = y_test - y_pred
```

```
pd.DataFrame({'Error Values': (test_residual)}).plot.kde()
```

```
<Axes: ylabel='Density'>
```




```
1 sns.displot(test_residual, bins=25, kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7c501fc41990>
```