Hands-on Activity 11.1 Linear Regression Analysis

Name: Cuadra, Audrick Zander G.

Section: CPE22S3

Date: April 24, 2024

Objective(s):

- cThis activity aims to demonstrate how to apply simple linear regression analysis to solve regression problem

Intended Learning Outcomes (ILOs):

- Demonstrate how to solve regression problems using simple linear regression
- Use the linear regression model to predict the target value

Resources:

- Jupyter Notebook

Files:

- Life Expectancy Data.

Submission Requirements:

- PDF containing initial EDA and Data Wrangling
- PDF showing demonstration of simple linear regression.
- Submit a link to the colab file through the comment section.

```
import pandas as pd
import numpy as np

led_df = pd.read_csv("/content/Life Expectancy Data.csv")
led_df
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	D
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.000000	68.0	31	...	67.0	7.13	
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.000000	7.0	998	...	7.0	6.52	
2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.000000	73.0	304	...	73.0	6.53	
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.000000	76.0	529	...	76.0	6.16	
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.000000	79.0	1483	...	78.0	7.10	

2938 rows × 22 columns

```
def countDup(data):
    if data.duplicated().any():
        return data.duplicated().sum()
    else:
        return "No Duplicates Found!"
```

```
countDup(led_df)
```

```
'No Duplicates Found!'
```

```
1 led_df.isnull().sum()
```

```
Country                             0
Year                                0
Status                              0
Life expectancy                    10
Adult Mortality                    10
infant deaths                       0
Alcohol                           194
percentage expenditure              0
Hepatitis B                       553
Measles                             0
 BMI                               34
under-five deaths                   0
Polio                              19
Total expenditure                 226
Diphtheria                         19
 HIV/AIDS                           0
GDP                               448
Population                        652
 thinness  1-19 years              34
 thinness 5-9 years                34
Income composition of resources   167
Schooling                         163
dtype: int64
```

```
1 led_df.columns
```

```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

```
1 def rem_null(data):
2     for i in data.columns:
3         if data[i].dtype != 'object':
4            data[i].fillna(method='ffill', inplace=True)
```

```
1 rem_null(led_df)
```

```
1 led_df.isnull().sum()
```

```
Country                           0
Year                              0
Status                            0
Life expectancy                   0
Adult Mortality                   0
infant deaths                     0
Alcohol                           0
percentage expenditure            0
Hepatitis B                       0
Measles                           0
 BMI                              0
under-five deaths                 0
Polio                             0
Total expenditure                 0
Diphtheria                        0
 HIV/AIDS                         0
GDP                               0
Population                        0
 thinness  1-19 years             0
 thinness 5-9 years               0
Income composition of resources   0
Schooling                         0
dtype: int64
```

```
1 led_df.columns
```

```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

```
1 led_df.rename(columns={'Life expectancy ': 'Life expectancy'}, inplace=True)
```

```
1 led_df.rename(columns={'Measles ': 'Measles'}, inplace=True)
```

```
led_df.rename(columns={' BMI ': 'BMI'}, inplace=True)
```

```
led_df.rename(columns={'under-five deaths ': 'under-five deaths'}, inplace=True)
```

```
led_df.rename(columns={'Diphtheria ': 'Diphtheria'}, inplace=True)
```

```
led_df.rename(columns={' thinness  1-19 years': 'thinness  1-19 years'}, inplace=True)
```

```
led_df.rename(columns={' thinness 5-9 years': 'thinness 5-9 years'}, inplace=True)
```

```
led_df.rename(columns={' HIV/AIDS': 'HIV/AIDS'}, inplace=True)
```

```
led_df.isnull().sum()
```

```
Country                          0
Year                             0
Status                           0
Life expectancy                  0
Adult Mortality                  0
infant deaths                    0
Alcohol                          0
percentage expenditure           0
Hepatitis B                      0
Measles                          0
BMI                              0
under-five deaths                0
Polio                            0
Total expenditure                0
Diphtheria                       0
HIV/AIDS                         0
GDP                              0
Population                       0
thinness  1-19 years             0
thinness 5-9 years               0
Income composition of resources  0
Schooling                        0
dtype: int64
```

```
led_cat = led_df.copy()
```

```
def preprocessing(data, catlist):
    if data[catlist].dtypes == 'object':
        cat_val = data[catlist].unique()
        range_val = range(1, len(cat_val)+1)
        map = dict(zip(cat_val, range_val))
        print(f"{catlist}:", map)
        data[catlist] = data[catlist].map(map)
    return data
for i in led_df.select_dtypes(include=['object']).columns:
    preprocessing(led_df, i)
```

```
Country: {'Afghanistan': 1, 'Albania': 2, 'Algeria': 3, 'Angola': 4, 'Antigua and Barbuda': 5, 'Argentina': 6, 'Armenia': 7, 'Austra
Status: {'Developing': 1, 'Developed': 2}
```

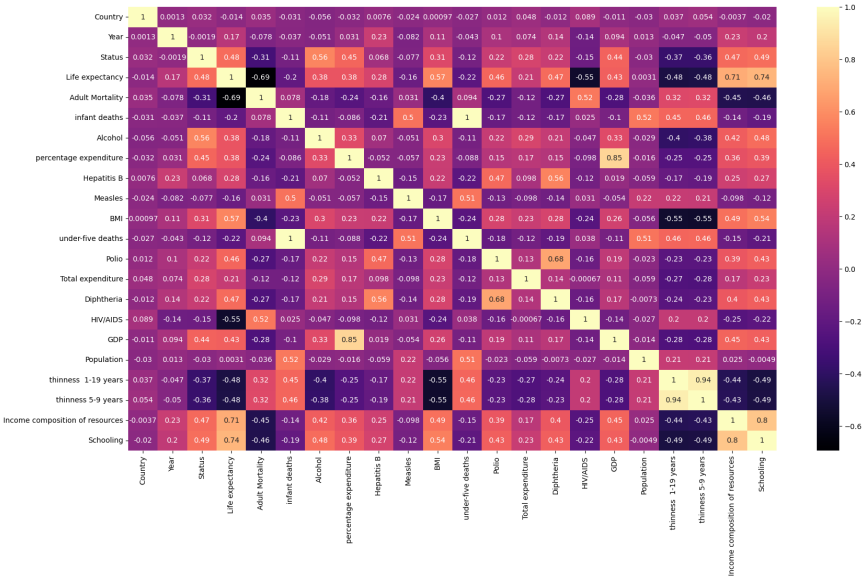
```
led_df.head()
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepati
0	1	2015	1	65.0	263.0	62	0.01	71.279624	6
1	1	2014	1	59.9	271.0	64	0.01	73.523582	6
2	1	2013	1	59.9	268.0	66	0.01	73.219243	6
3	1	2012	1	59.5	272.0	69	0.01	78.184215	6
4	1	2011	1	59.2	275.0	71	0.01	7.097109	6

5 rows × 22 columns

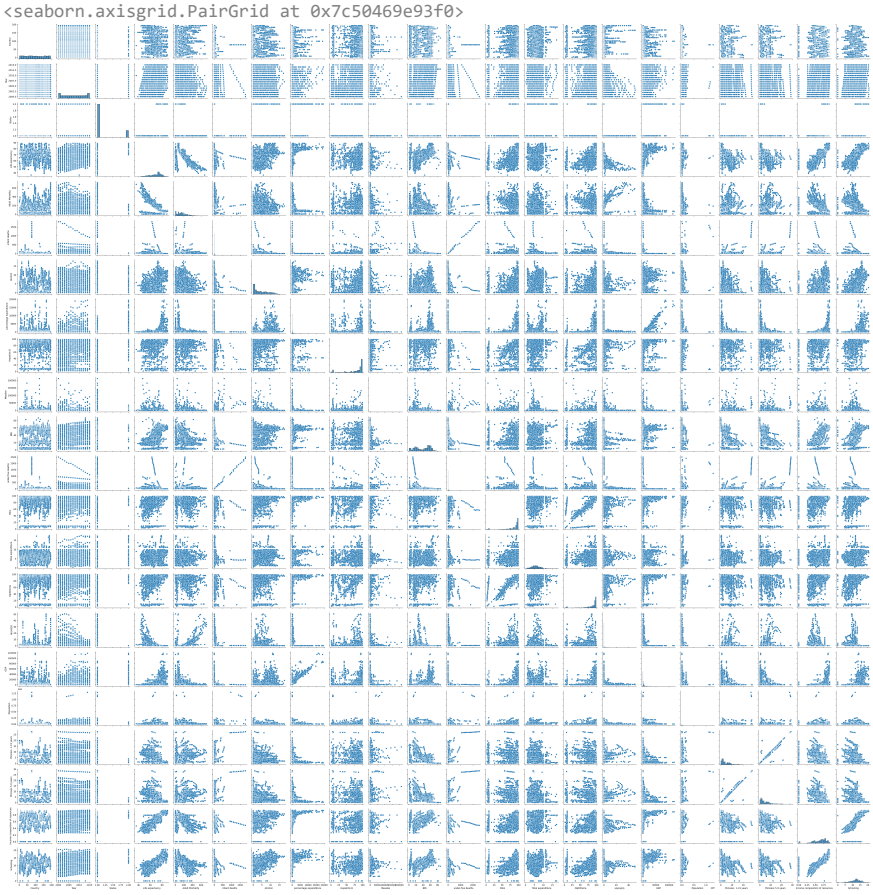
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(20,11))
sns.heatmap(led_df.corr(), annot=True, cmap='magma')
```

```
<Axes: >
```




```
sns.pairplot(led_df)
```

```
<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