

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [2]: df = pd.read_csv("drug.csv")
# .dropna(axis="columns")
df
```

```
Out[2]:
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

```
In [3]: df.head()
```

```
Out[3]:
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

## Data cleaning and pre processing

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Age              200 non-null    int64
1   Sex              200 non-null    object
2   BP               200 non-null    object
3   Cholesterol      200 non-null    object
4   Na_to_K          200 non-null    float64
5   Drug             200 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [5]: `df.describe()`

```
Out[5]:
```

	Age	Na_to_K
<b>count</b>	200.000000	200.000000
<b>mean</b>	44.315000	16.084485
<b>std</b>	16.544315	7.223956
<b>min</b>	15.000000	6.269000
<b>25%</b>	31.000000	10.445500
<b>50%</b>	45.000000	13.936500
<b>75%</b>	58.000000	19.380000
<b>max</b>	74.000000	38.247000

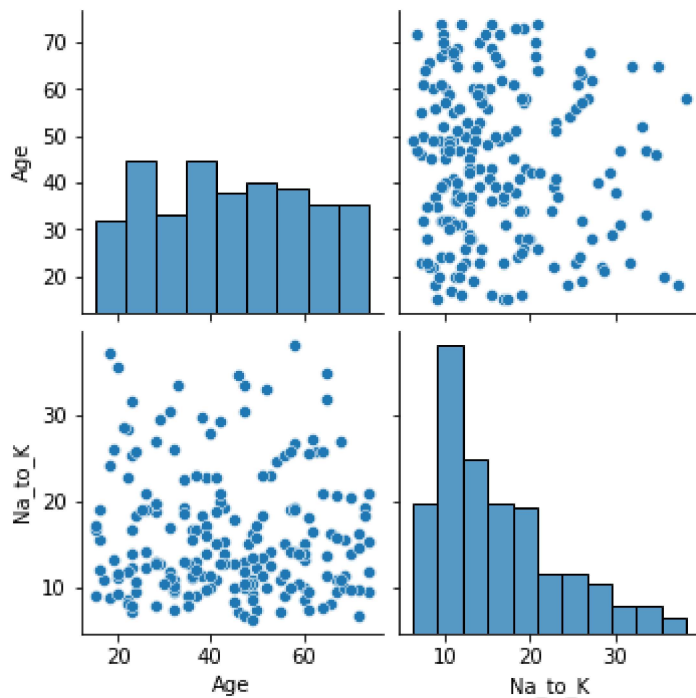
In [6]: `df.columns`

Out[6]: `Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')`

## EDA and VISUALIZATION

In [7]: `sns.pairplot(df)`

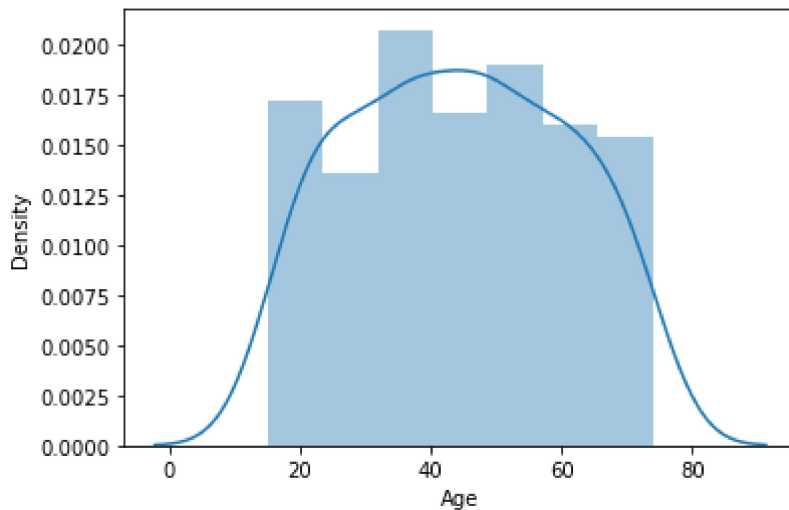
Out[7]: `<seaborn.axisgrid.PairGrid at 0x28fab0471c0>`



```
In [8]: sns.distplot(df["Age"])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

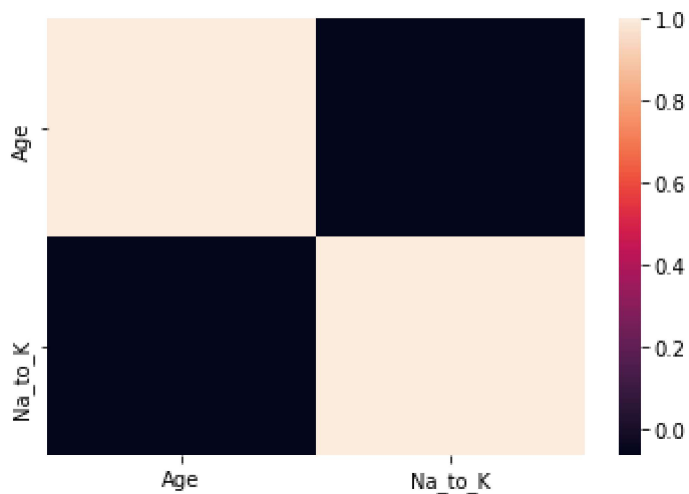
```
Out[8]: <AxesSubplot:xlabel='Age', ylabel='Density'>
```



```
In [9]: df1 = df[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']]
```

```
In [10]: sns.heatmap(df1.corr())
```

```
Out[10]: <AxesSubplot:>
```



```
In [11]: x = df1[['Age', 'Na_to_K']]
         y = df1['Age']
```

## split the data into training and test data

```
In [12]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

```
In [13]: lr = LinearRegression()
         lr.fit(x_train, y_train)
```

Out[13]: LinearRegression()

```
In [14]: lr.intercept_
```

Out[14]: 0.0

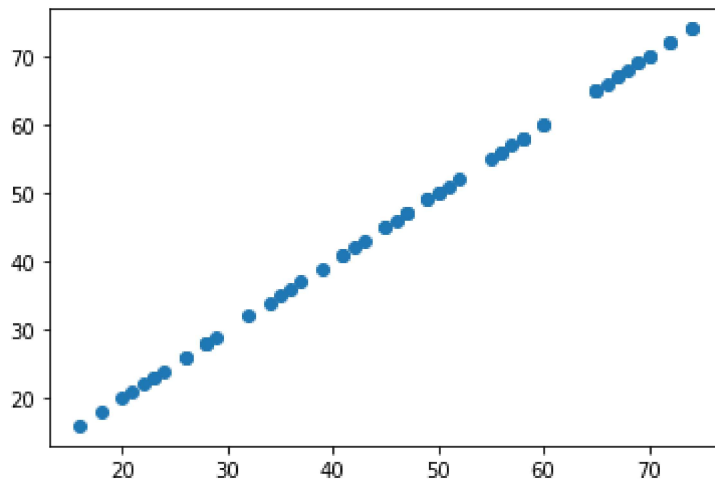
```
In [15]: coeff = pd.DataFrame(lr.coef_, x.columns, columns =['Co-efficient'])
         coeff
```

```
Out[15]:
```

	Co-efficient
Age	1.000000e+00
Na_to_K	-4.646522e-18

```
In [16]: prediction = lr.predict(x_test)
         plt.scatter(y_test, prediction)
```

Out[16]: <matplotlib.collections.PathCollection at 0x28faba92c10>



```
In [17]: lr.score(x_test,y_test)
```

```
Out[17]: 1.0
```