

In [1]:

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

# DATA COLLECTION

In [2]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\4_drug200 - 4_drug200.csv")
a
```

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]:

```
b=a.head(100)
b
```

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
...	...	...	...	...	...	...
95	36	M	LOW	NORMAL	11.424	drugX
96	58	F	LOW	HIGH	38.247	drugY
97	56	F	HIGH	HIGH	25.395	drugY
98	20	M	HIGH	NORMAL	35.639	drugY
99	15	F	HIGH	NORMAL	16.725	drugY

100 rows × 6 columns

# DATA CLEANING AND PRE-PROCESSING

In [4]:

```
b.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Age         100 non-null    int64
1   Sex         100 non-null    object
2   BP          100 non-null    object
3   Cholesterol 100 non-null    object
4   Na_to_K     100 non-null    float64
5   Drug        100 non-null    object
dtypes: float64(1), int64(1), object(4)
memory usage: 4.8+ KB
```

In [5]:

```
b.describe()
```

Out[5]:

	Age	Na_to_K
<b>count</b>	100.000000	100.000000
<b>mean</b>	43.770000	16.823000
<b>std</b>	16.367531	7.257723
<b>min</b>	15.000000	7.285000
<b>25%</b>	30.500000	11.031250
<b>50%</b>	43.000000	15.025500
<b>75%</b>	58.000000	20.020250
<b>max</b>	74.000000	38.247000

In [7]:

```
b.columns
```

Out[7]:

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

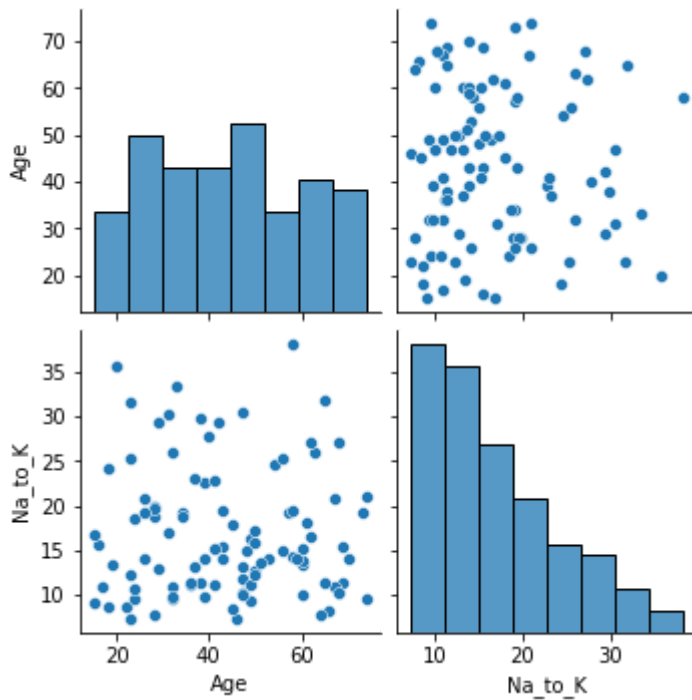
## EDA AND VISUALIZATION

In [8]:

```
sns.pairplot(b)
```

Out[8]:

&lt;seaborn.axisgrid.PairGrid at 0x2236421f670&gt;



In [10]:

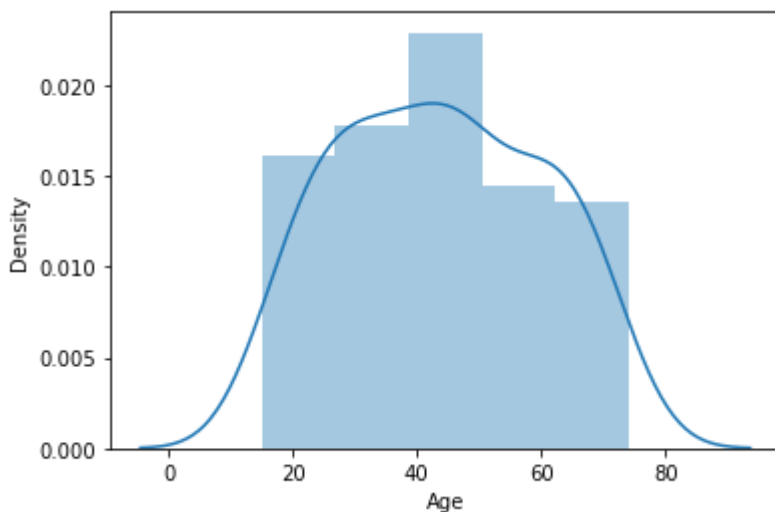
```
sns.distplot(b['Age'])
```

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[10]:

&lt;AxesSubplot:xlabel='Age', ylabel='Density'&gt;



In [25]:

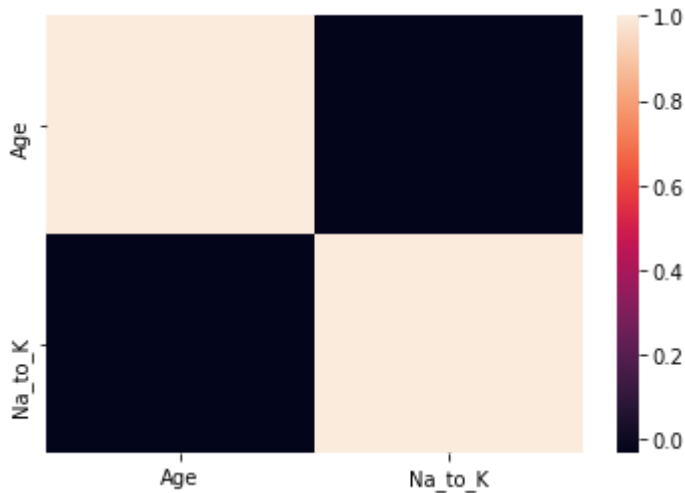
```
f=b[['Age', 'Na_to_K']]
```

In [26]:

```
sns.heatmap(f.corr())
```

Out[26]:

<AxesSubplot:>



In [27]:

```
x=f[['Age']]  
y=f[['Na_to_K']]
```

In [28]:

```
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
```

In [29]:

```
from sklearn.linear_model import LinearRegression  
  
lr=LinearRegression()  
lr.fit(x_train,y_train)
```

Out[29]:

LinearRegression()

In [30]:

```
print(lr.intercept_)
```

15.640017640932857

In [31]:

```
r=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])  
r
```

Out[31]:

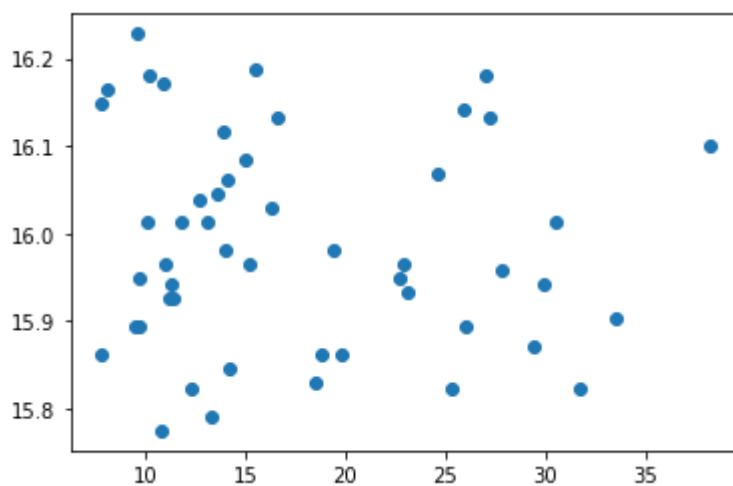
	Co-efficient
Age	0.007951

In [32]:

```
u=lr.predict(x_test)  
plt.scatter(y_test,u)
```

Out[32]:

&lt;matplotlib.collections.PathCollection at 0x2236a7eb250&gt;



In [33]:

```
print(lr.score(x_test,y_test))
```

-0.04803604987051768

In [ ]: