Problem Statement

In [62]:

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

In [63]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\drug.csv")
d
```

Out[63]:

	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 [64]:

```
d.head(10)
```

Out[64]:

	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
5	22	F	NORMAL	HIGH	8.607	drugX
6	49	F	NORMAL	HIGH	16.275	drugY
7	41	M	LOW	HIGH	11.037	drugC
8	60	M	NORMAL	HIGH	15.171	drugY
9	43	M	LOW	NORMAL	19.368	drugY

In [65]:

```
d.describe()
```

Out[65]:

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

In [66]:

```
d.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 [67]:

```
d.columns
```

Out[67]:

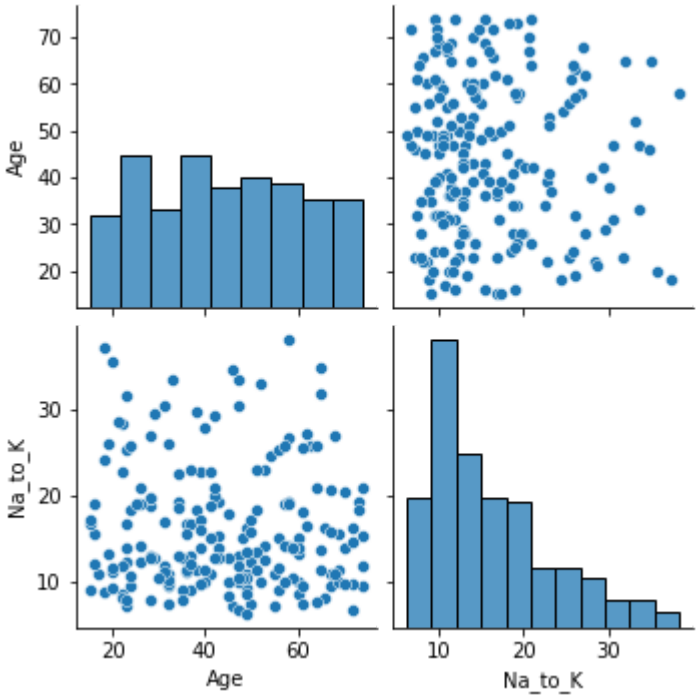
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='objec
t')
```

In [68]:

```
sns.pairplot(d)
```

Out[68]:

```
<seaborn.axisgrid.PairGrid at 0x171970e2f10>
```



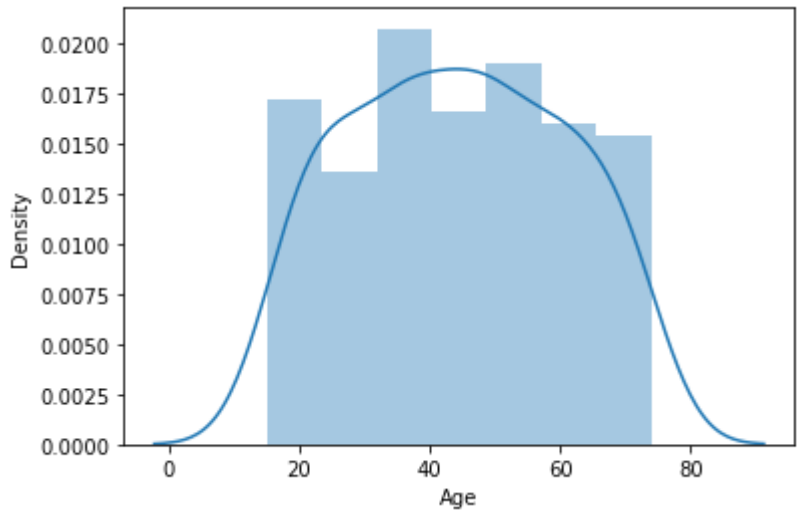
In [69]:

```
sns.distplot(d['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 fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[69]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```



In [70]:

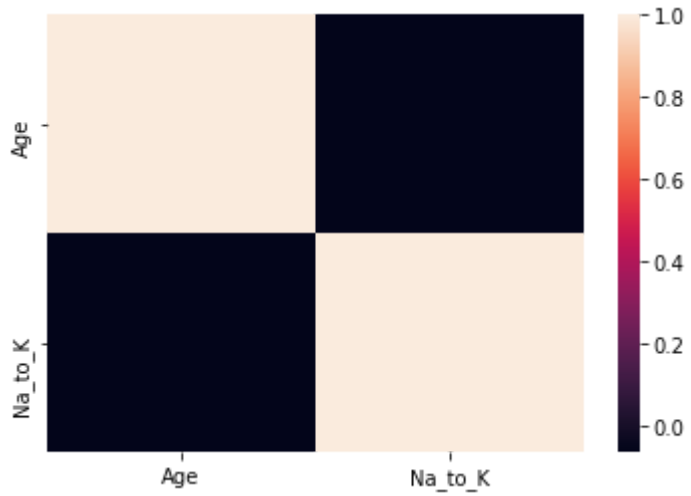
```
da=d[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']]
```

In [71]:

```
# relation
sns.heatmap(da.corr())
```

Out[71]:

```
<AxesSubplot:>
```



to train the model

we are going to train linear regresssion model;we need to split out data into two values varible x and y where x is independent(input) and y is depdendent on x (output)we could ignore adrees column as it not requird foer model

In [84]:

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

In [85]:

```
# to split my dataset into test and train data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [86]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[86]:

```
LinearRegression()
```

In [87]:

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

```
18.0669868939996
```

In [88]:

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

Out[88]:

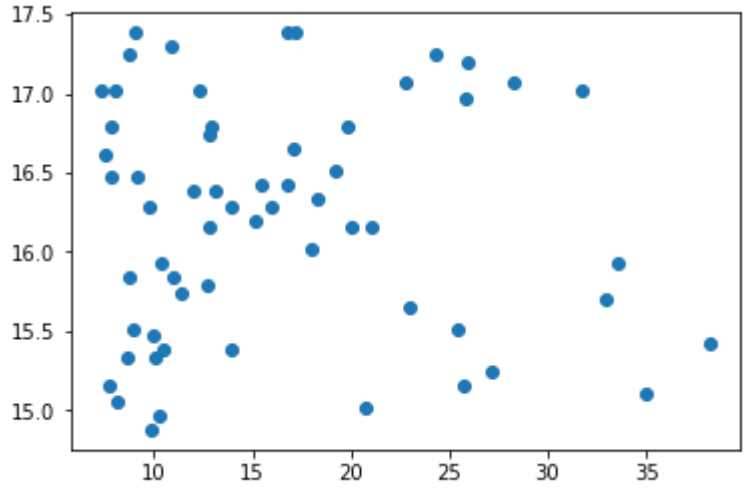
	Co-effecient
Age	-0.045533

In [89]:

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

Out[89]:

```
<matplotlib.collections.PathCollection at 0x1719837e820>
```



In [90]:

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

```
-0.015814744052748075
```

In [ ]:

```

```