In [23]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [24]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\4_drug200.csv")
data
```

Out[24]:

	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 [25]:
```
df=data.head(100)
df
```

Out[25]:

	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

In [26]:
```
df.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 [27]:
```
df.describe()
```

Out[27]:

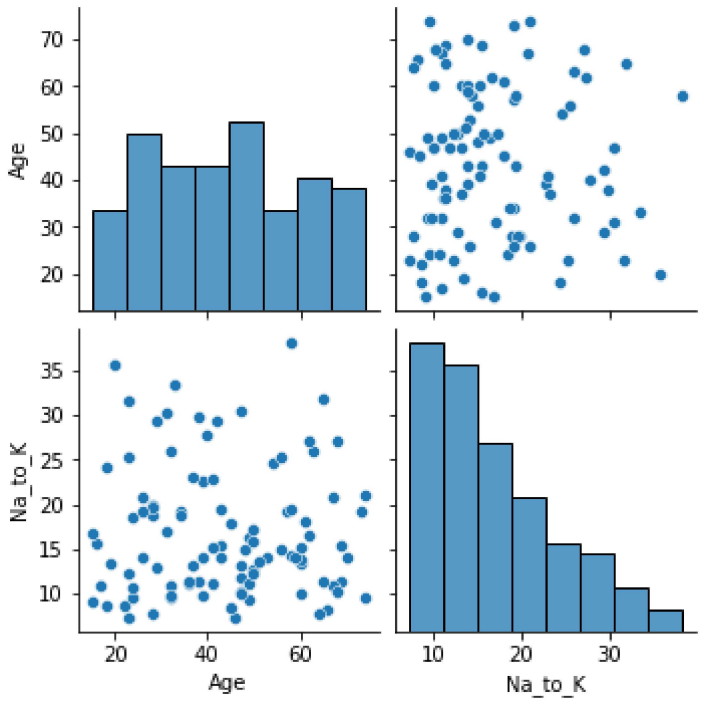
	Age	Na_to_K
count	100.000000	100.000000
mean	43.770000	16.823000
std	16.367531	7.257723
min	15.000000	7.285000
25%	30.500000	11.031250
50%	43.000000	15.025500
75%	58.000000	20.020250
max	74.000000	38.247000

In [28]:
```
df.columns
```

Out[28]:
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

In [29]:
```
sns.pairplot(df)
```

Out[29]:
```
<seaborn.axisgrid.PairGrid at 0x1ed590d4c70>
```



In [30]:
```
da=df[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']]
da
```

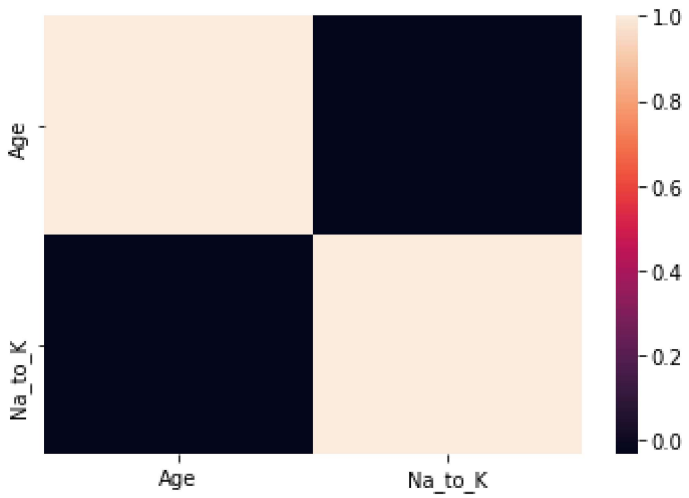
Out[30]:

	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

In [31]:
```
sns.heatmap(da.corr())
```

Out[31]: <AxesSubplot:>



In [32]:
```
x=df[['Na_to_K']]
y=df['Age']
```

In [33]:
```
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 [34]:
```
from sklearn.linear_model import LinearRegression

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

Out[34]:
```
LinearRegression()
```

In [35]:
```
print(lr.intercept_)
```

```
48.23899261528304
```

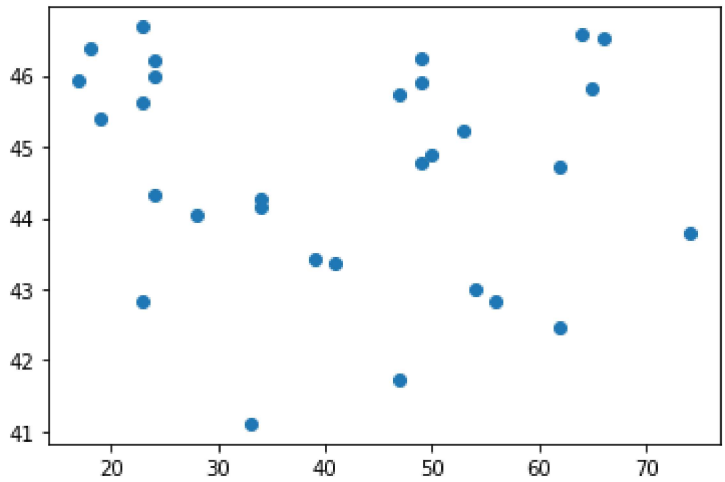
In [36]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[36]:

	Co-efficient
Na_to_K	-0.212774

In [37]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[37]:
```
<matplotlib.collections.PathCollection at 0x1ed5a35a7f0>
```



In [38]:
```
print(lr.score(x_test,y_test))
```

```
-0.0613309781995135
```

In [39]:
```
print(lr.score(x_train,y_train))
```

```
0.009292226156223271
```

In [40]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [41]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[41]:
```
Ridge(alpha=10)
```

In [42]:
```
rr.score(x_test,y_test)
```

Out[42]:
```
-0.0612282222195808
```

In [43]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[43]:
```
Lasso(alpha=10)
```

In [44]:
```
la.score(x_test,y_test)
```

Out[44]:
```
-0.033955212786694666
```

In [ ]:
```

```