In [1]:

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

In [2]:

```
df = pd.read_csv(r'C:\Users\user\Downloads\4_drug200 (1).csv')
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.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 [4]:

```
df.describe()
```

Out[4]:

	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 [5]:

```
df.columns
```

Out[5]:

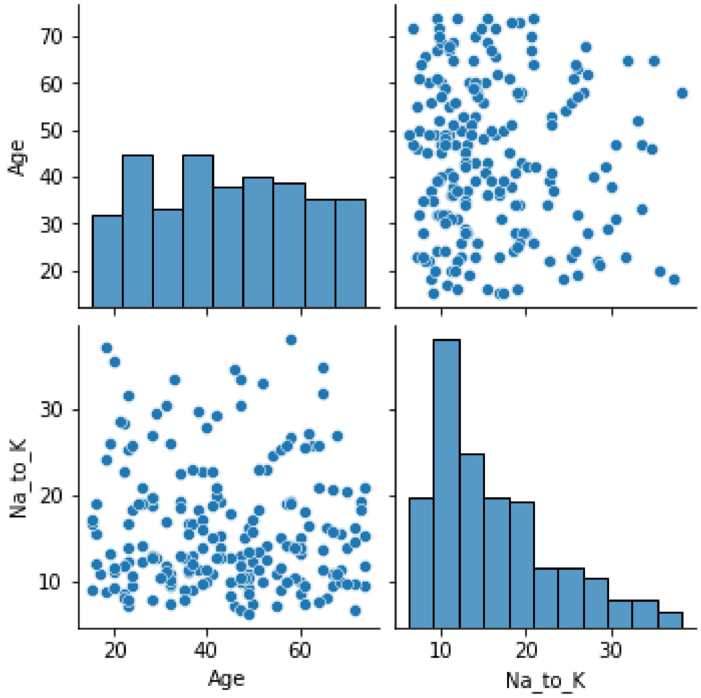
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

In [6]:

```
sns.pairplot(df)
```

Out[6]:

```
<seaborn.axisgrid.PairGrid at 0x2485e433520>
```



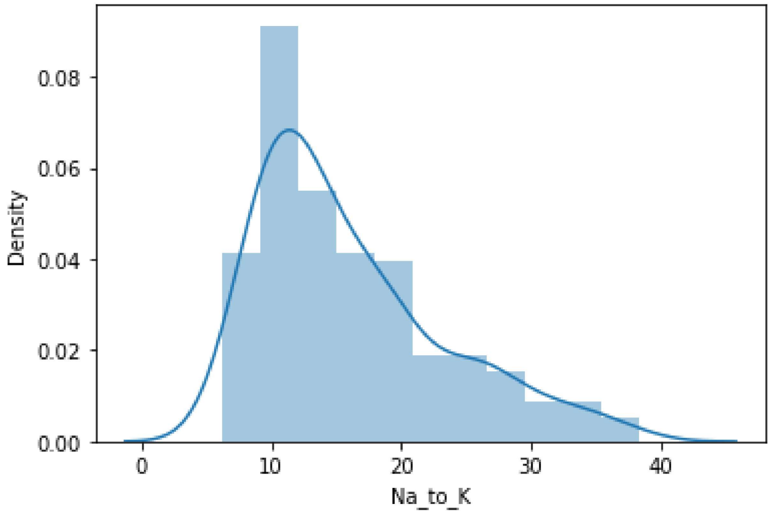
In [7]:

```
sns.distplot(df['Na_to_K'])
```

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

Out[7]:

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



In [8]:

```
df1 = df[['Age','Na_to_K']]
df1
```

Out[8]:

	Age	Na_to_K
0	23	25.355
1	47	13.093
2	47	10.114
3	28	7.798
4	61	18.043
...	...	...
195	56	11.567
196	16	12.006
197	52	9.894
198	23	14.020
199	40	11.349

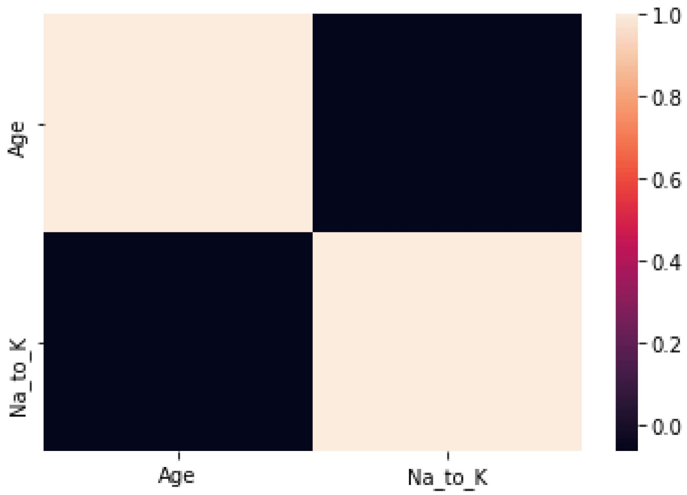
200 rows × 2 columns

In [9]:

```
sns.heatmap(df1.corr())
```

Out[9]:

```
<AxesSubplot:>
```



model building

In [10]:

```
x = df1[['Na_to_K']]
y = df1['Age']
```

In [11]:

```
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 [12]:

```
from sklearn.linear_model import LinearRegression

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

Out[12]:

```
LinearRegression()
```

In [13]:

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

```
46.89430157394965
```

In [14]:

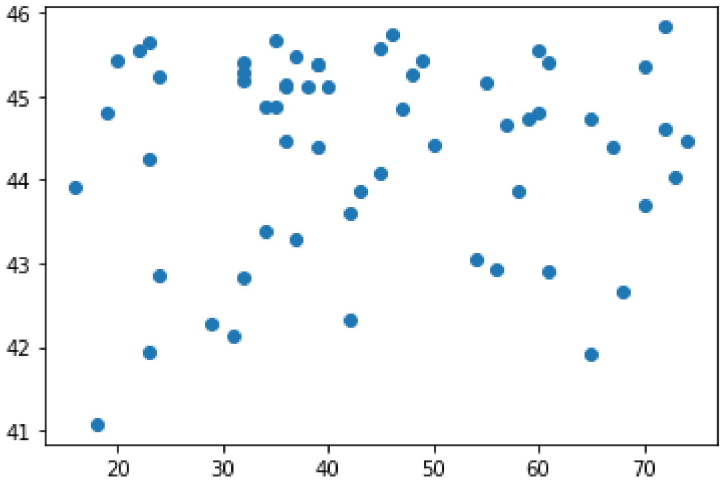
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[14]:

	Co-efficient
Na_to_K	-0.15648

In [15]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[15]:
```
<matplotlib.collections.PathCollection at 0x248606d4580>
```

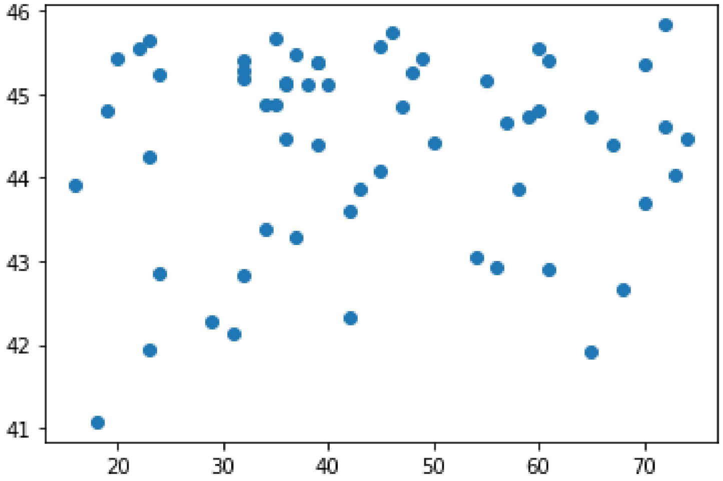


In [16]:
```
print(lr.score(x_test,y_test))
```

```
0.0025386490222192304
```

In [17]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[17]:
```
<matplotlib.collections.PathCollection at 0x24860726eb0>
```



In [18]:
```
lr.score(x_test,y_test)
```

Out[18]:
```
0.0025386490222192304
```

In [19]:
```
lr.score(x_train,y_train)
```

Out[19]:
```
0.004456740906168655
```

In [20]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [22]:
```
rr = Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_train,y_train)
```

Out[22]:
```
0.0044567319293381535
```

In [23]:
```
rr.score(x_test,y_test)
```

Out[23]:
```
0.0025420987280241336
```

In [24]:
```
ls = Lasso(alpha=10)
ls.fit(x_train,y_train)
ls.score(x_train,y_train)
```

Out[24]:
```
0.0
```

In [25]:
```
ls.score(x_test,y_test)
```

Out[25]:
```
-0.0002108632050292769
```

In [ ]:
```

```