In [76]:
```
# import libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [144]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\\4_drug200 - 4_drug200.csv")
```

Out[144]:

	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 [145]:
```
x=x.head(10)
```

Out[145]:

	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 [146]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          10 non-null     int64  
 1   Sex          10 non-null     object 
 2   BP           10 non-null     object 
 3   Cholesterol  10 non-null     object 
 4   Na_to_K      10 non-null     float64
 5   Drug         10 non-null     object 
dtypes: float64(1), int64(1), object(4)
memory usage: 608.0+ bytes
```

In [147]:

Out[147]:
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

In [148]:
```
d=x[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']]
```

Out[148]:

	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

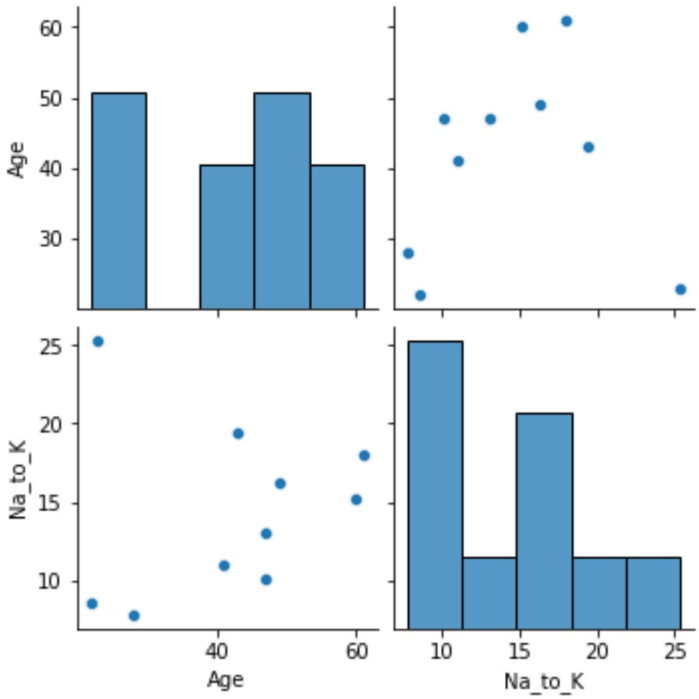
In [149]:

Out[149]:

	Age	Na_to_K
count	10.000000	10.000000
mean	42.100000	14.486100
std	13.916018	5.482634
min	22.000000	7.798000
25%	31.250000	10.344750
50%	45.000000	14.132000
75%	48.500000	17.601000
max	61.000000	25.355000

In [150]:

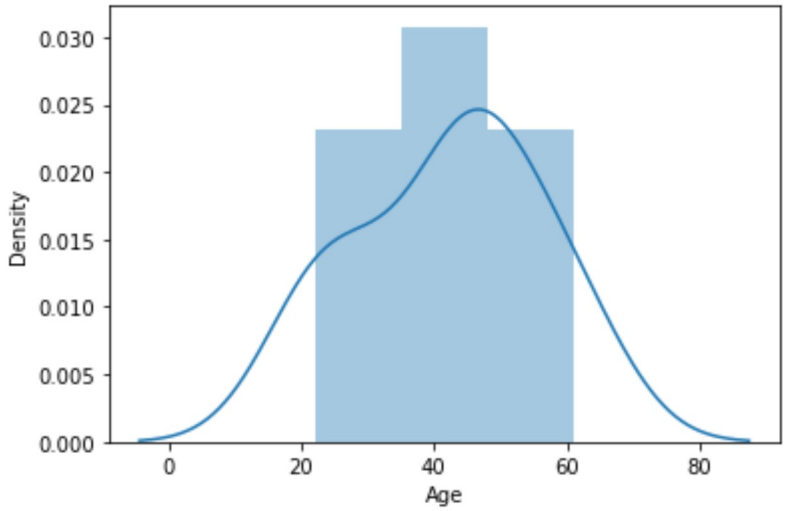
Out[150]: <seaborn.axisgrid.PairGrid at 0x190c2987880>



In [151]:

```
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)
```

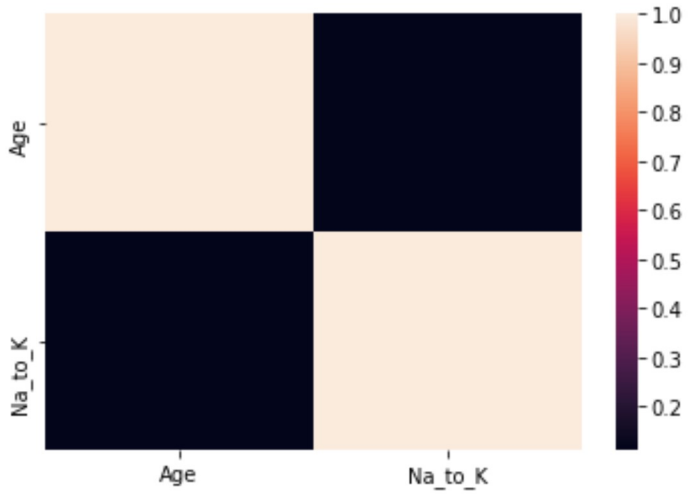
Out[151]: <AxesSubplot:xlabel='Age', ylabel='Density'>



In [153]:

In [154]:

Out[154]: <AxesSubplot:>



In [155]:

```
x=x1[['Age']]
```

In [156]:
```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [157]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[157]:
```
LinearRegression()
```

In [158]:

```
0.0
```

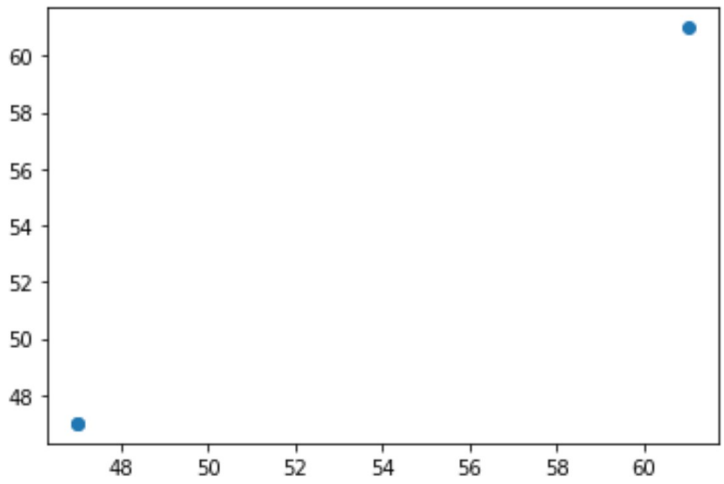
In [159]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[159]:

	Co-efficient
Age	1.0

In [160]:
```
prediction=lr.predict(x_test)
```

Out[160]:
```
<matplotlib.collections.PathCollection at 0x190c779d0d0>
```



In [161]:

Out[161]:
```
1.0
```

In [162]:

Out[162]:
```
1.0
```

In [163]:

In [164]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[164]: 0.999650455066024

In [165]:
```
la=Lasso(alpha=10)
```

Out[165]: Lasso(alpha=10)

In [166]:

Out[166]: 0.982590365493147

In [167]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
```

Out[167]: ElasticNet()

In [168]:

Out[168]: array([0.99427871])

In [169]:

Out[169]: array([46.94850838, 46.94850838, 60.8684103])

In [170]:

Out[170]: 0.21740907233346718

In [171]:

Out[171]: 0.9998268982861678

In [172]:

In [173]:

Mean Absolute Error 0.0

In [174]:

Mean Squared Error 0.0

In [175]:

Root Mean Squared Error 0.0

In [ ]: