

```
In [1]: # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [71]: x=pd.read_csv(r"C:\Users\user\Downloads\4_drug200 - 4_drug200.csv")
```

Out[71]:

	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 [72]: x=x.head(100)
```

Out[72]:

	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 [73]:

```
<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 [74]:

Out[74]: Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')

In [75]: d=x[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']]

Out[75]:

	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

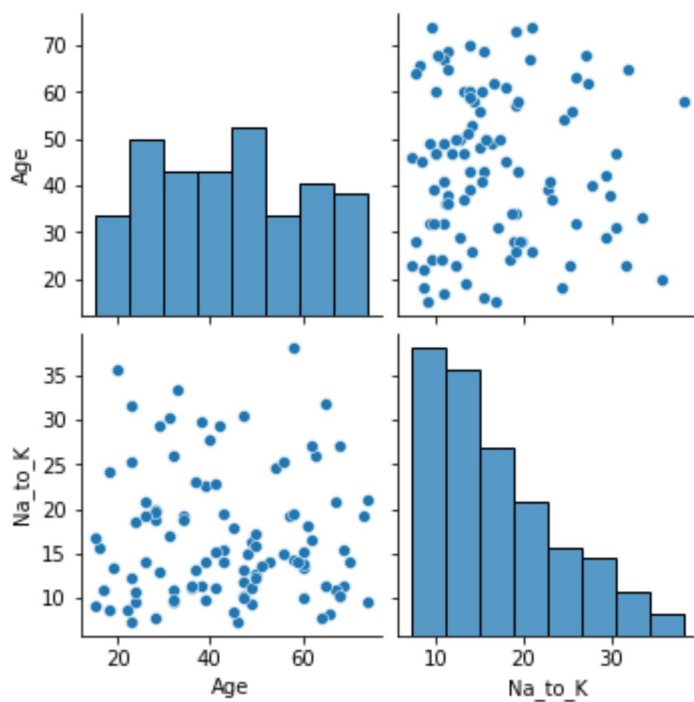
In [76]:

Out[76]:

	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 [77]:

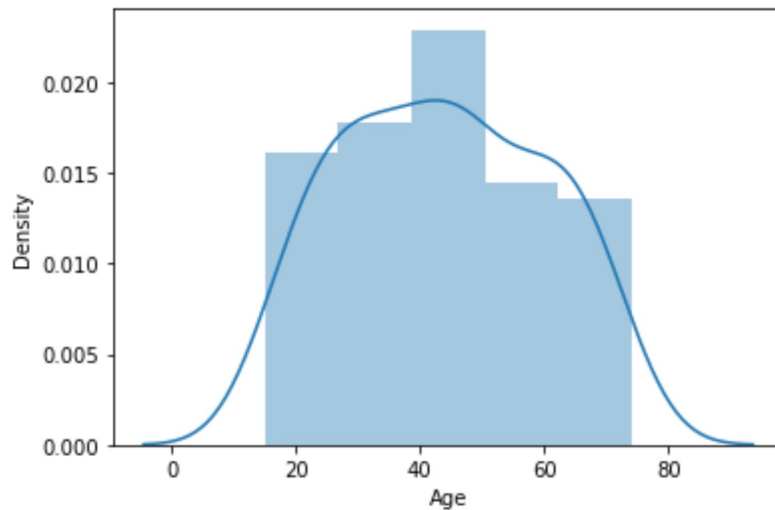
Out[77]: <seaborn.axisgrid.PairGrid at 0x22d2ca8d730>



In [78]:

```
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)
```

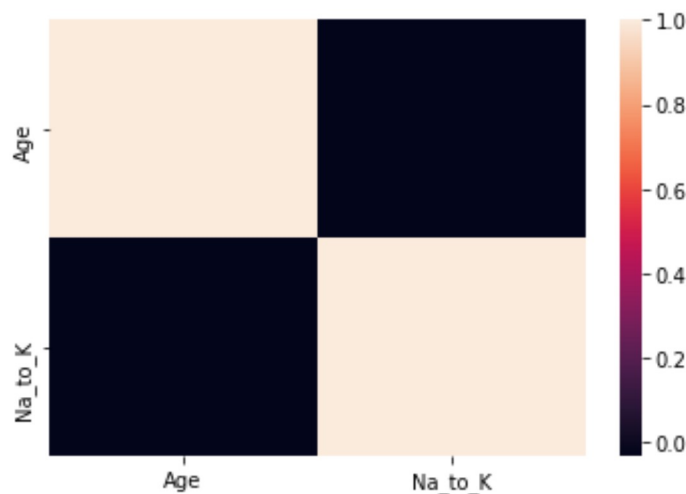
Out[78]: <AxesSubplot:xlabel='Age', ylabel='Density'>



In [79]:

In [80]:

Out[80]: <AxesSubplot:>

In [82]: `x=x1[['Age', 'Na_to_K']]`

In [83]: *# to split my dataset into training and test data*

```
from sklearn.model_selection import train_test_split
```

In [84]: **from** sklearn.linear_model **import** LinearRegression

```
lr=LinearRegression()
```

Out[84]: LinearRegression()

In [85]:

```
-7.105427357601002e-15
```

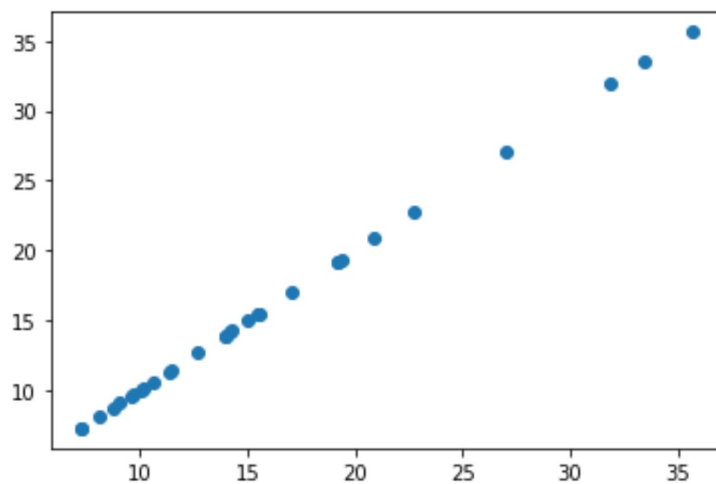
In [86]: `coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])`

Out[86]:

	Co-efficient
Age	6.733537e-18
Na_to_K	1.000000e+00

In [87]: `prediction=lr.predict(x_test)`

Out[87]: <matplotlib.collections.PathCollection at 0x22d2d1944f0>



In [88]:

Out[88]: 1.0

In [89]:

Out[89]: 1.0

In [90]:

```
In [91]: rr=Ridge(alpha=10)
         rr.fit(x_train,y_train)
```

```
Out[91]: 0.9999913048703735
```

```
In [92]: la=Lasso(alpha=10)
```

```
Out[92]: Lasso(alpha=10)
```

```
In [93]:
```

```
Out[93]: 0.9571797827752848
```

```
In [ ]:
```