kaviyadevi 20106064

In [51]:
```
#to import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [54]:
```
#to import dataset
data=pd.read_csv(r"C:\Users\user\Downloads\4_drug200 - 4_drug200.csv")
data
```

Out[54]:

	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 [55]:
```
data.head()
```

Out[55]:

	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

DATA CLEANING AND PREPROCESSING

In [56]:
```
data.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 [57]:
```
data.describe()
```

Out[57]:

	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

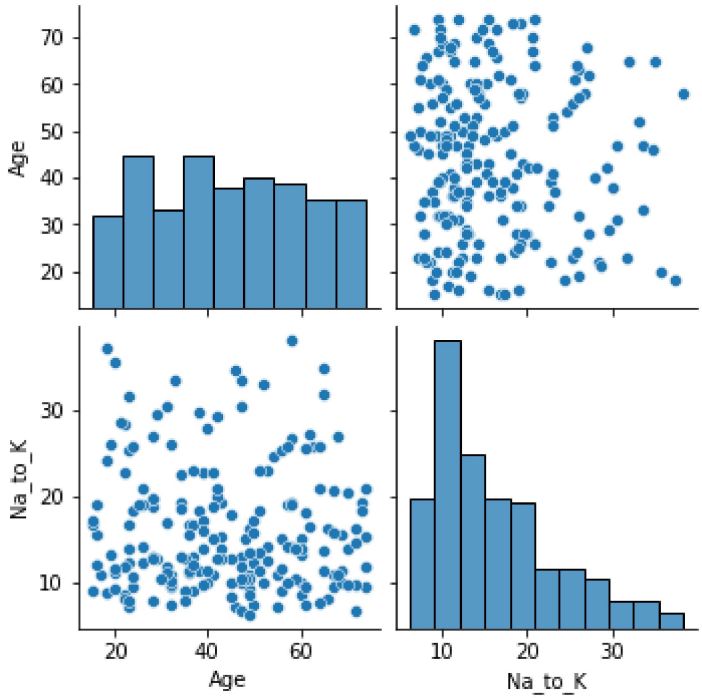
In [58]:
```
data.columns
```

Out[58]:
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

EDA and DATA VISUALIZATION

In [59]:
```
sns.pairplot(data)
```

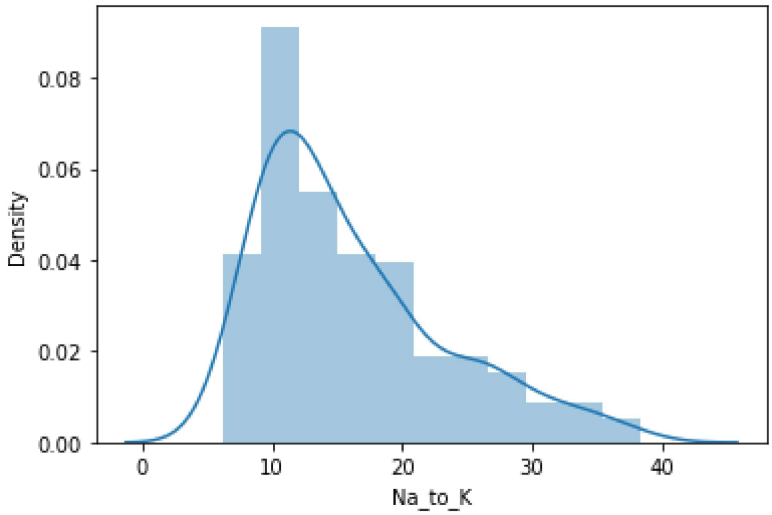
Out[59]:
```
<seaborn.axisgrid.PairGrid at 0x241d536b910>
```



In [62]:
```
sns.distplot(data['Na_to_K'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Futur
eWarning: `distplot` is a deprecated function and will be removed in a future v
ersion. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```

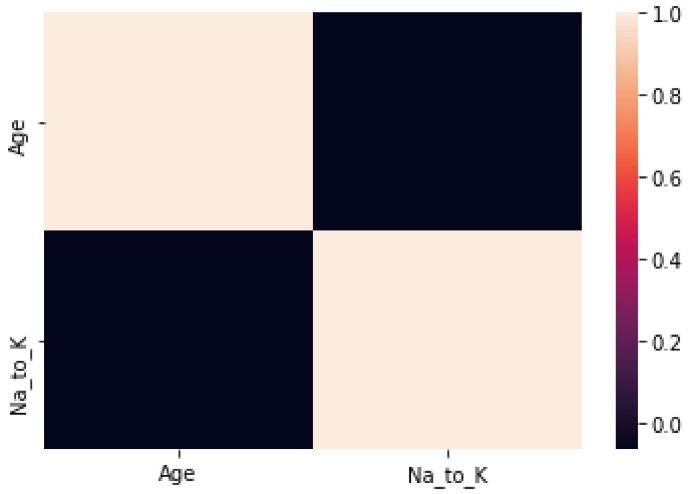
Out[62]: <AxesSubplot:xlabel='Na_to_K', ylabel='Density'>



In [63]:
```
df=data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug']]
```

In [64]:
```
sns.heatmap(df.corr())
```

Out[64]: <AxesSubplot:>



TRAINNING MODEL

In [69]:
```
x=df[['Age']]
y=df[['Na_to_K']]
```

In [70]:
```
#to split my dataset into trainning and test

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 [71]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

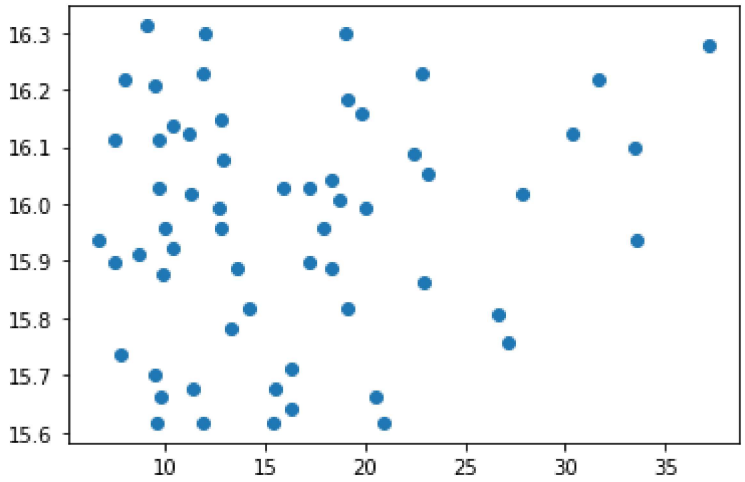
Out[71]:
```
LinearRegression()
```

In [72]:
```
#to find intercept
print(lr.intercept_)
```

```
[16.48883429]
```

In [73]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[73]:
```
<matplotlib.collections.PathCollection at 0x241d57f8880>
```



In [74]:
```
print(lr.score(x_test,y_test))
```

```
0.0042465716188196945
```

RIDGE AND LASSO REGRESSION

In [75]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [76]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[76]:
```
Ridge(alpha=10)
```

In [77]:
```
rr.score(x_test,y_test)
```

Out[77]:
```
0.0042448844866279645
```

In [78]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[78]:
```
Lasso(alpha=10)
```

In [79]:
```
la.score(x_test,y_test)
```

Out[79]:
```
-0.0027678708157250487
```