A real estate agent want help to predict the house price for regions in Usa.he gave us the dataset to work on to use linear Regression model.Create a model that helps him to estimate

Data Collection

In [141]:
```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [147]:
```
#import the dataset
data=pd.read_csv(r"C:\Users\user\Desktop\Vicky\4 drug200.csv")
```

In [148]:
```
#to display top 10 rows
data.head()
```

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

In [149]:
```
#to display null values
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 [150]:
```
data.shape
```

Out[150]: (200, 6)

In [151]:
```
#to display summary of statistics
data.describe()
```

Out[151]:

	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

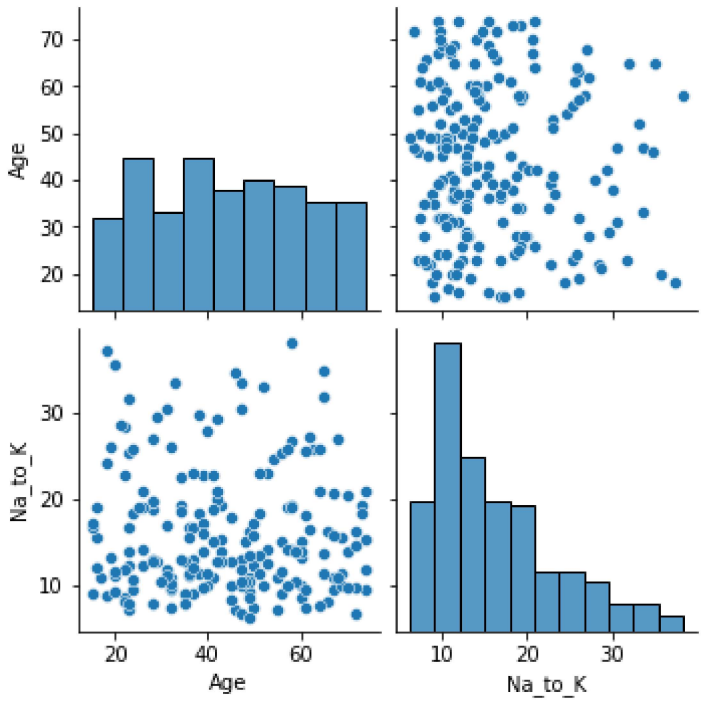
In [152]:
```
#to display columns name
data.columns
```

Out[152]: Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')

In [153]:
```
data1=data[['Age', 'Na to K']]
```

In [154]:
```
sns.pairplot(data1)
```

Out[154]: <seaborn.axisgrid.PairGrid at 0x250f6ecd2b0>

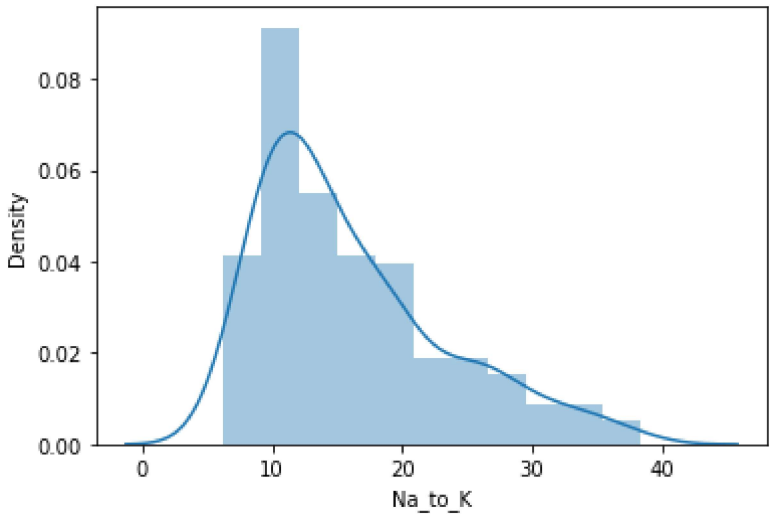


EDA and Visualization

In [156]:
```
sns.distplot(data['Na_to_K'])
```

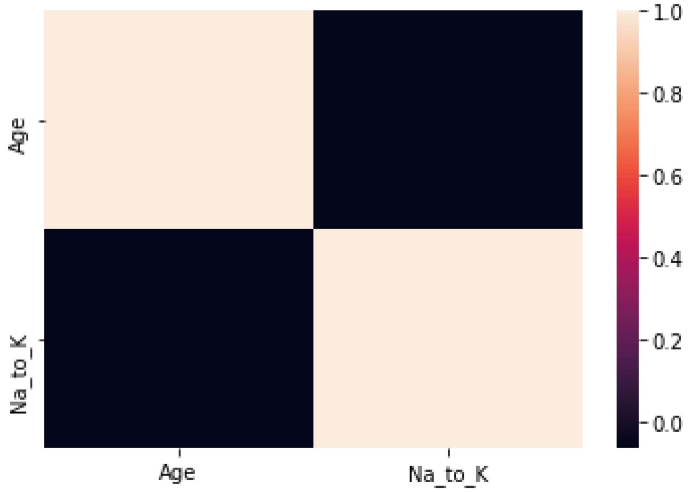
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function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

Out[156]:
```
<AxesSubplot:xlabel='Na_to_K', ylabel='Density'>
```



In [157]:
```
sns.heatmap(data1.corr())
```

Out[157]:
```
<AxesSubplot:>
```



To train the model

we are going to train the linear regression model ;We need to split the two variable x and y

In [158]:
```
x=data1[[ 'Age', 'Na_to_K']]
y=data1["Age"]
```

In [160]:
```
#To split test and train data
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.6)
```

In [161]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[161]:
```
LinearRegression()
```

In [162]:
```
lr.intercept
```

Out[162]:
```
7.105427357601002e-15
```

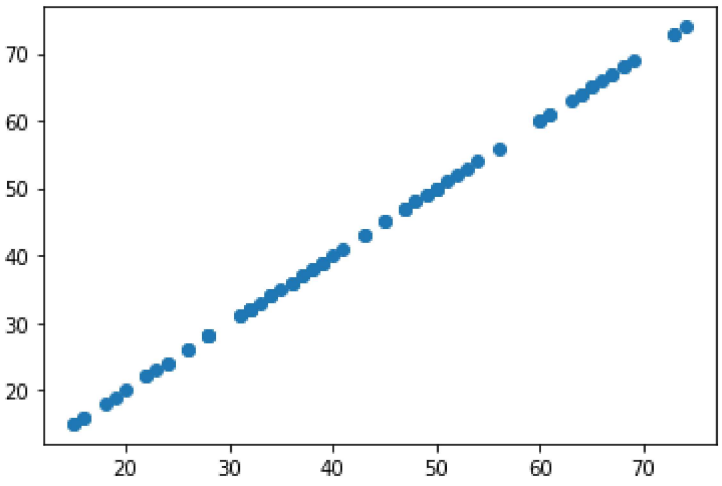
In [163]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])
coeff
```

Out[163]:

	Co-efficient
Age	1.000000e+00
Na_to_K	-1.237332e-17

In [164]:
```
prediction = lr.predict(x_train)
plt.scatter(y_train,prediction)
```

Out[164]:
```
<matplotlib.collections.PathCollection at 0x250f8cce340>
```



In [165]:
```
lr.score(x_test,y_test)
```

Out[165]:
```
1.0
```

In [166]:
```
lr.score(x_train,y_train)
```

Out[166]:
```
1.0
```

In [167]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [168]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[168]: 0.9999997579318577

In [169]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
la.score(x_test,y_test)
```

Out[169]: 0.998451174724841

In [ ]:

In [ ]:

In [ ]:

In [ ]: