

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

```
In [2]: df=pd.read_csv("/Users/bob/Downloads/4_drug200.csv")
df.fillna(0,inplace=True)
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.head()
```

Out [3]:

	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 [4]: `df.info()`

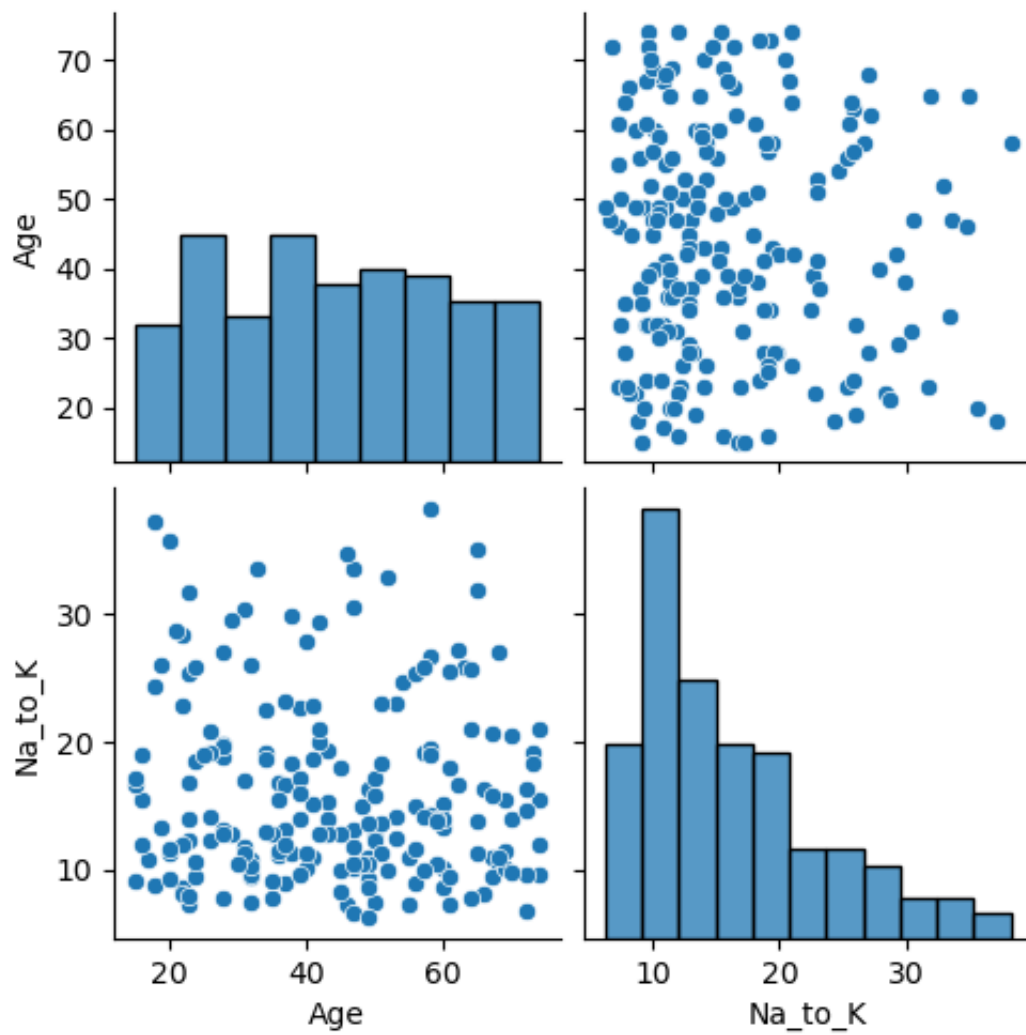
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Age             200 non-null    int64
 1   Sex             200 non-null    object
 2   BP              200 non-null    object
 3   Cholesterol     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 [5]: `df.columns`

Out[5]: `Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')`

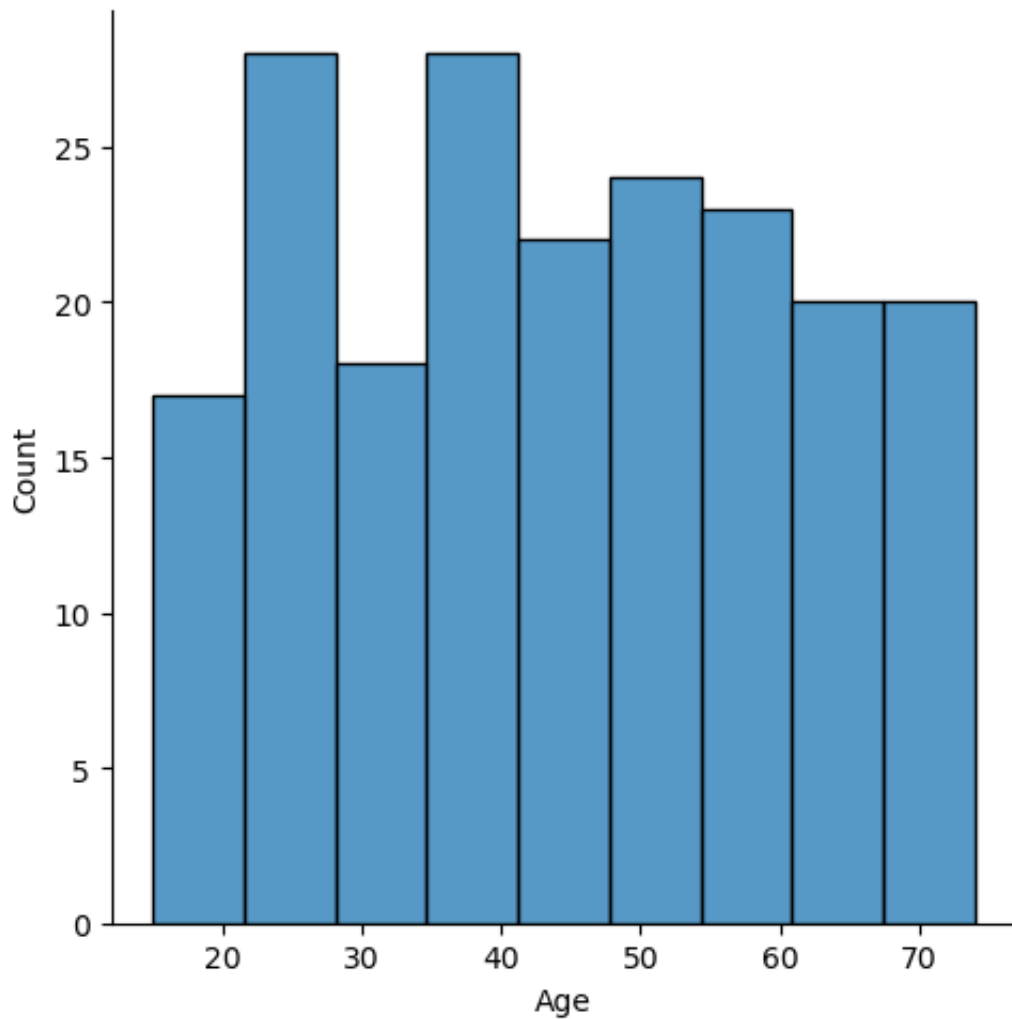
```
In [6]: sns.pairplot(df)
```

```
Out[6]: <seaborn.axisgrid.PairGrid at 0x7f8d33325780>
```



```
In [7]: sns.displot(df['Age'])
```

```
Out[7]: <seaborn.axisgrid.FacetGrid at 0x7f8d50447340>
```



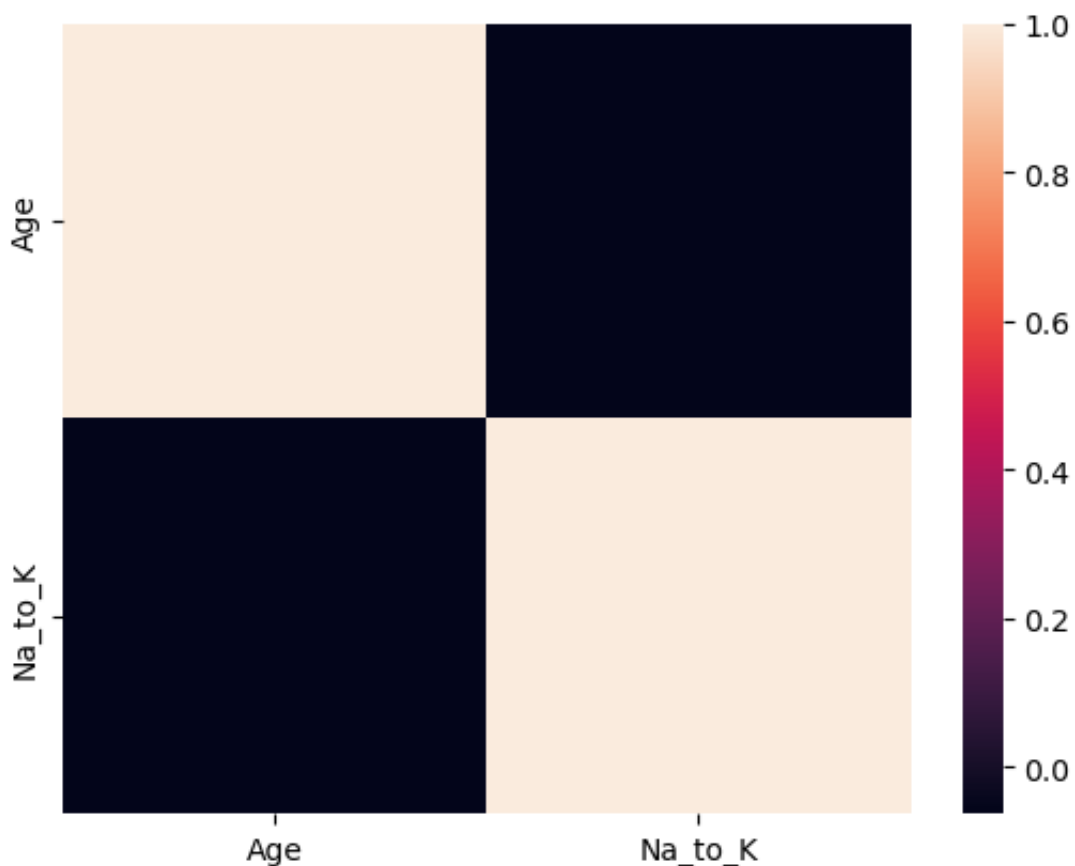
```
In [8]: df1=df.drop(['Sex'],axis=1)
df1
df1=df1.drop(df1.index[1537:])
df1.isna().sum()
```

```
Out[8]: Age      0
BP      0
Cholesterol  0
Na_to_K  0
Drug     0
dtype: int64
```

```
In [9]: sns.heatmap(df1.corr())
```

```
/var/folders/2n/rrl24lws3pb1nz8_t911srvm0000gn/T/ipykernel_15998/781785195.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.  
sns.heatmap(df1.corr())
```

```
Out [9]: <Axes: >
```



```
In [10]: from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LinearRegression
```

```
In [11]: df1.isna().sum()
```

```
Out [11]: Age          0  
BP          0  
Cholesterol  0  
Na_to_K     0  
Drug        0  
dtype: int64
```

```
In [12]: y=df1['Age']
x=df1.drop(['BP', 'Cholesterol', 'Drug'],axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
print(x_train)
```

	Age	Na_to_K
110	50	7.490
149	22	8.151
138	51	11.343
118	32	10.292
63	45	8.370
..
7	41	11.037
69	18	24.276
169	20	11.262
194	46	34.686
23	48	15.036

[140 rows x 2 columns]

```
In [13]: model=LinearRegression()
model.fit(x_train,y_train)
model.intercept_
```

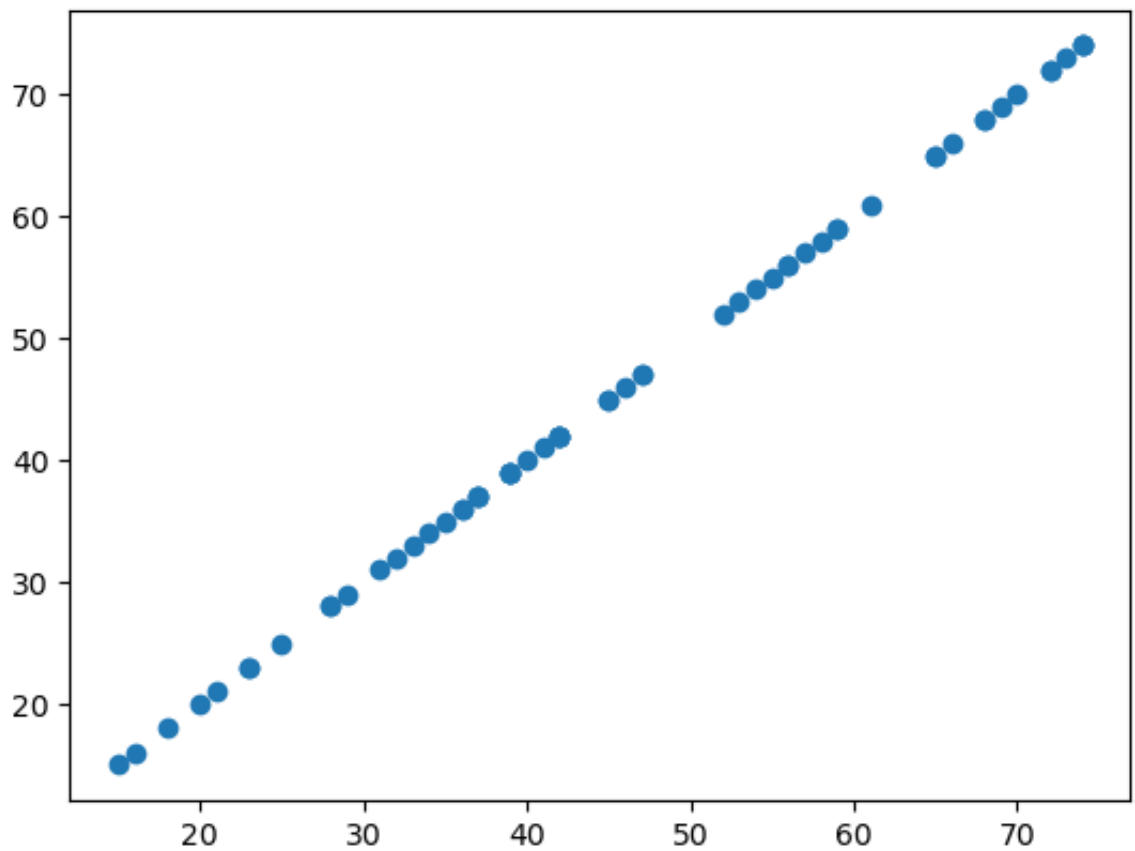
Out[13]: -2.1316282072803006e-14

```
In [14]: model.coef_
```

Out[14]: array([1.00000000e+00, 1.83854971e-17])

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

```
Out[15]: <matplotlib.collections.PathCollection at 0x7f8d337735e0>
```



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

```
Out[16]: 1.0
```

```
In [ ]:
```