lets practice normalization

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

import a datasets firstly

```
df =pd.read_csv('/content/weight-height.csv')
df
```

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801
...	...	...	...
9995	Female	66.172652	136.777454
9996	Female	67.067155	170.867906
9997	Female	63.867992	128.475319
9998	Female	69.034243	163.852461
9999	Female	61.944246	113.649103

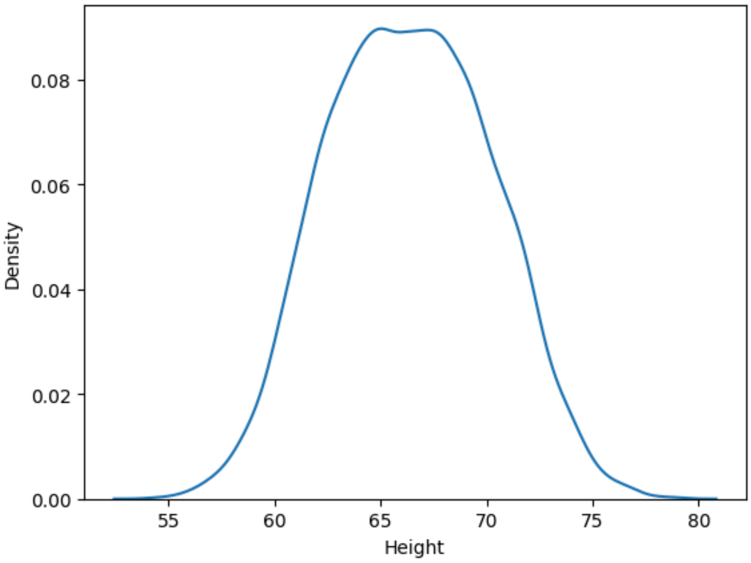
10000 rows × 3 columns

Next steps: Generate code with df New interactive sheet

now try to findout whats the distributions of the features

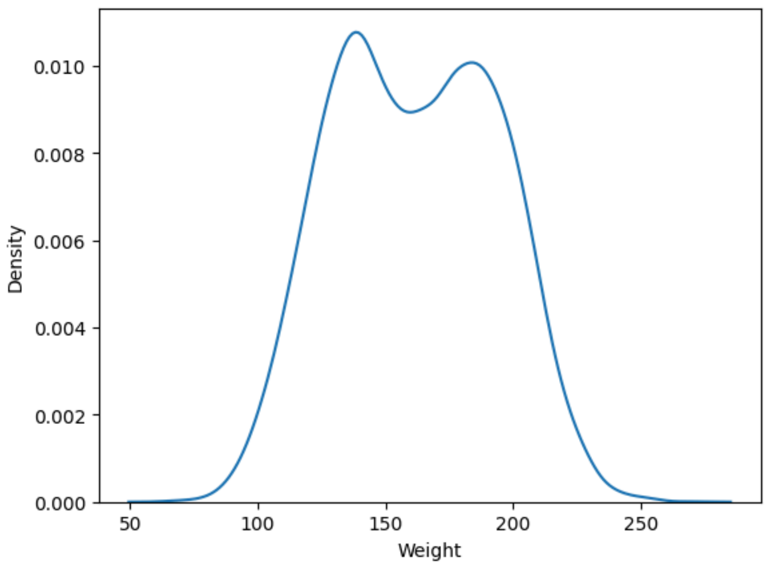
```
sns.kdeplot(df['Height'])
```

```
<Axes: xlabel='Height', ylabel='Density'>
```



```
sns.kdeplot(df['Weight'])
```

```
<Axes: xlabel='Weight', ylabel='Density'>
```



now we will implement normalizations(MinMaxScaler)

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
height_scaled = scaler.fit_transform(df[['Height']])
weight_scaled = scaler.fit_transform(df[['Weight']])
```

```
height_scaled
```

```
array([[0.79172838],
       [0.58695829],
       [0.8023644 ],
       ...,
       [0.38830089],
       [0.59715974],
       [0.31052854]])
```

```
weight_scaled
```

```
array([[0.863139  ],
       [0.4754764 ],
       [0.72113127],
       ...,
       [0.31065968],
       [0.48298768],
       [0.23843869]])
```

we have got the results in numpy arrays & now we will convert that into a data frame & concat them

```
height_scaled = pd.DataFrame(height_scaled , columns=['height_scaled'])
weight_scaled = pd.DataFrame(weight_scaled , columns=['weight_scaled'])
```

```
df = pd.concat([height_scaled,weight_scaled], axis = 1)
df
```

	height_scaled	weight_scaled
0	0.791728	0.863139
1	0.586958	0.475476
2	0.802364	0.721131
3	0.706182	0.756699
4	0.631424	0.689999
...	...	...
9995	0.481473	0.351101
9996	0.517635	0.517161
9997	0.388301	0.310660
9998	0.597160	0.482988
9999	0.310529	0.238439

10000 rows × 2 columns

Next steps: Generate code with df New interactive sheet

here we will see the statics of the feautures

```
df.describe()
```

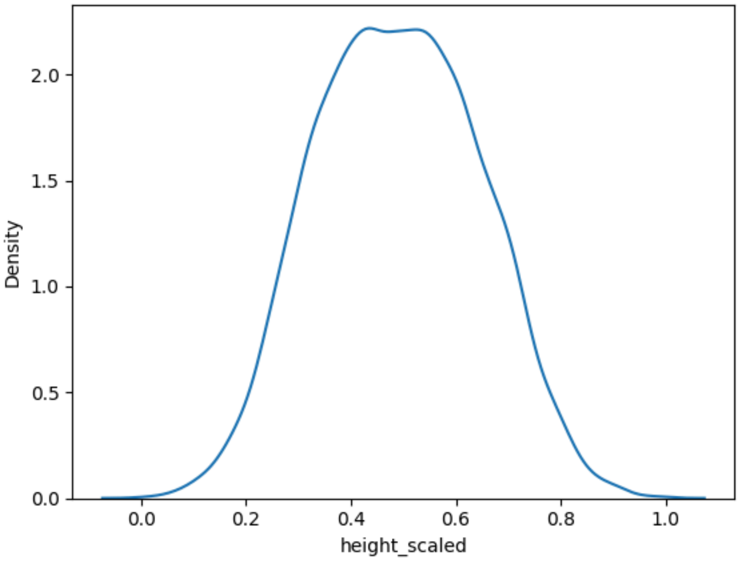
	height_scaled	weight_scaled
count	10000.000000	10000.000000
mean	0.489352	0.471238
std	0.155546	0.156406
min	0.000000	0.000000
25%	0.373651	0.346427
50%	0.487352	0.470130
75%	0.602820	0.596569
max	1.000000	1.000000

now we can see that the min value is 0 & max value is 1

lets see if there is any changes in the distributions after scaling?

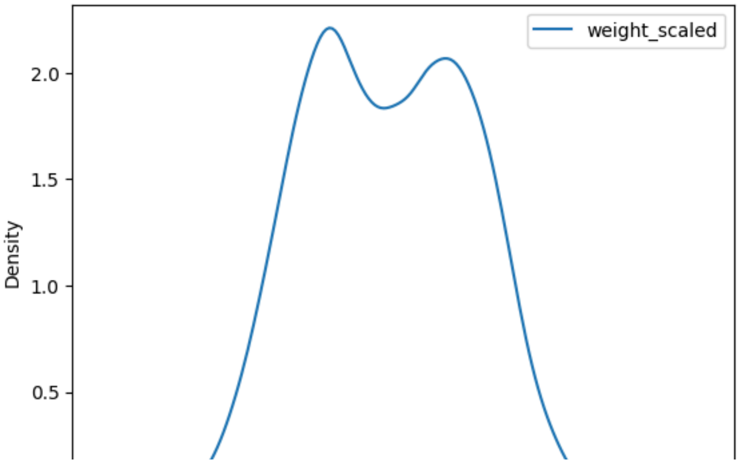
```
sns.kdeplot(df['height_scaled'])
```

```
<Axes: xlabel='height_scaled', ylabel='Density'>
```



```
sns.kdeplot(weight_scaled)
```

```
<Axes: ylabel='Density'>
```


weight_scaled
2.0
1.5
1.0
0.5
Density