```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn import metrics


# loading the data from csv file to a Pandas DataFrame
calories = pd.read_csv('/content/calories.csv')


# print the first 5 rows of the dataframe
calories.head()
```

	User_ID	Calories
0	14733363	231.0
1	14861698	66.0
2	11179863	26.0
3	16180408	71.0
4	17771927	35.0

Next steps: Generate code with calories View recommended plots

```
exercise_data = pd.read_csv('/content/exercise.csv')


exercise_data.head()
```

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8

Next steps: Generate code with exercise_data View recommended plots

```
calories_data = pd.concat([exercise_data, calories['Calories']], axis=1)


calories_data.head()
```

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8	35.0

Next steps: Generate code with calories_data View recommended plots

```
# checking the number of rows and columns
calories_data.shape
```

```
(15000, 9)
```

```
# checking the number of rows and columns
calories_data.shape
```

```
(15000, 9)
```

```
# checking for missing values
calories_data.isnull().sum()
```

```
User_ID       0
Gender        0
Age           0
Height        0
Weight        0
Duration      0
Heart_Rate    0
Body_Temp     0
Calories      0
dtype: int64
```

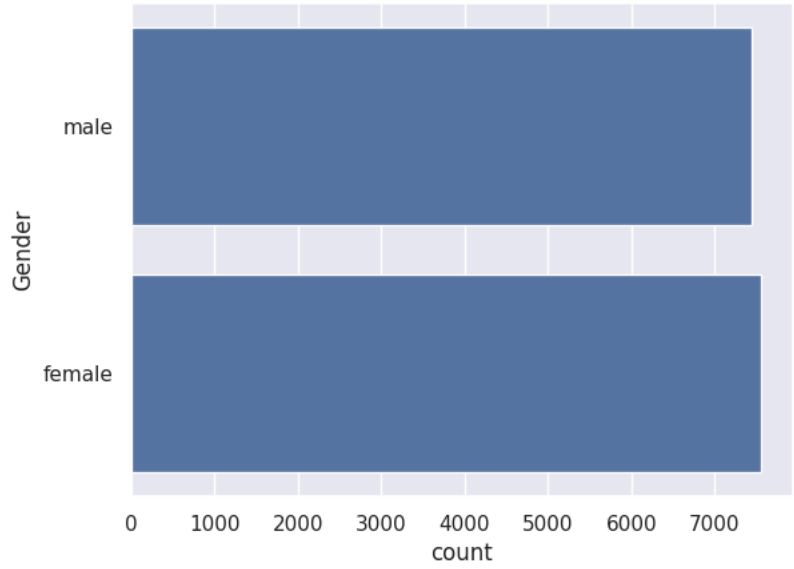
```
# get some statistical measures about the data
calories_data.describe()
```

	User_ID	Age	Height	Weight	Duration	Heart_Ra
count	1.500000e+04	15000.000000	15000.000000	15000.000000	15000.000000	15000.0000
mean	1.497736e+07	42.789800	174.465133	74.966867	15.530600	95.5185
std	2.872851e+06	16.980264	14.258114	15.035657	8.319203	9.5833
min	1.000116e+07	20.000000	123.000000	36.000000	1.000000	67.0000
25%	1.247419e+07	28.000000	164.000000	63.000000	8.000000	88.0000
50%	1.499728e+07	39.000000	175.000000	74.000000	16.000000	96.0000
75%	1.744928e+07	56.000000	185.000000	87.000000	23.000000	103.0000
max	1.999965e+07	79.000000	222.000000	132.000000	30.000000	128.0000

```
sns.set()
```

```
# plotting the gender column in count plot
sns.countplot(calories_data['Gender'])
```

```
<Axes: xlabel='count', ylabel='Gender'>
```




```
# finding the distribution of "Age" column
sns.distplot(calories_data['Age'])
```

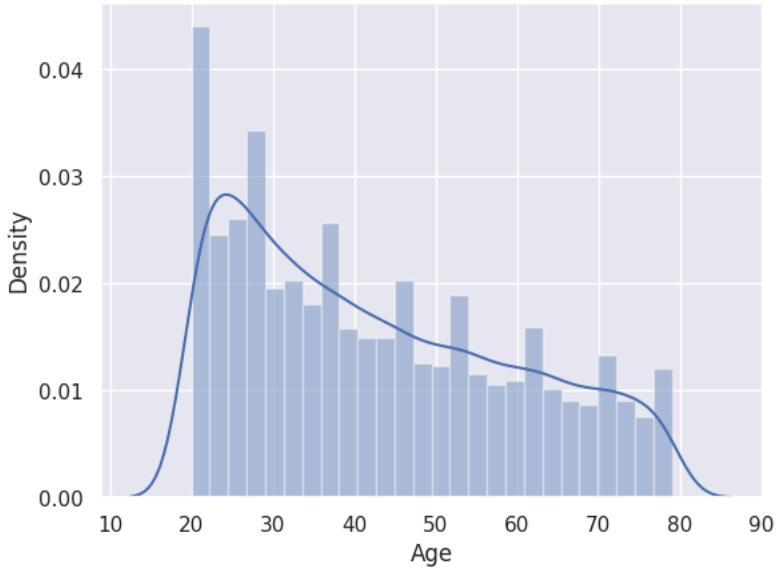
```
<ipython-input-14-6cbf196d4d06>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(calories_data['Age'])
<Axes: xlabel='Age', ylabel='Density'>
```




```
# finding the distribution of "Height" column
sns.distplot(calories_data['Height'])
```

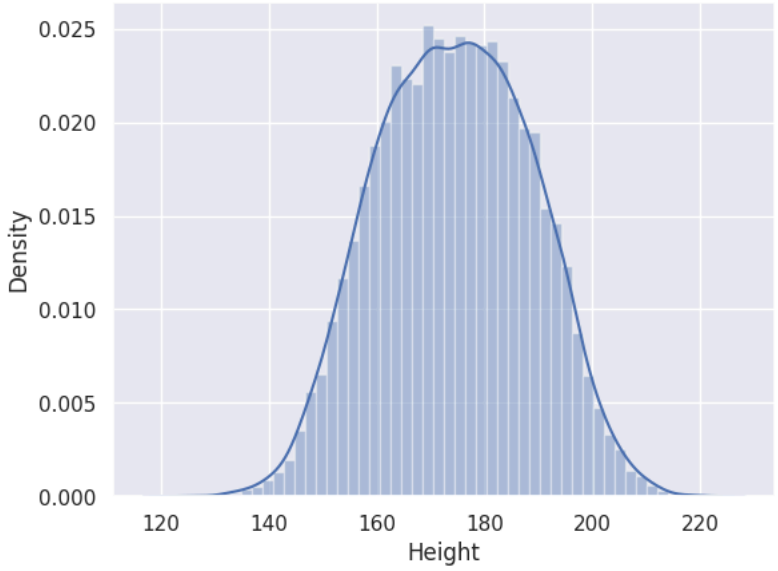
```
<ipython-input-15-fdc2a1fecb6d>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(calories_data['Height'])
<Axes: xlabel='Height', ylabel='Density'>
```




```
# finding the distribution of "Weight" column
sns.distplot(calories_data['Weight'])
```

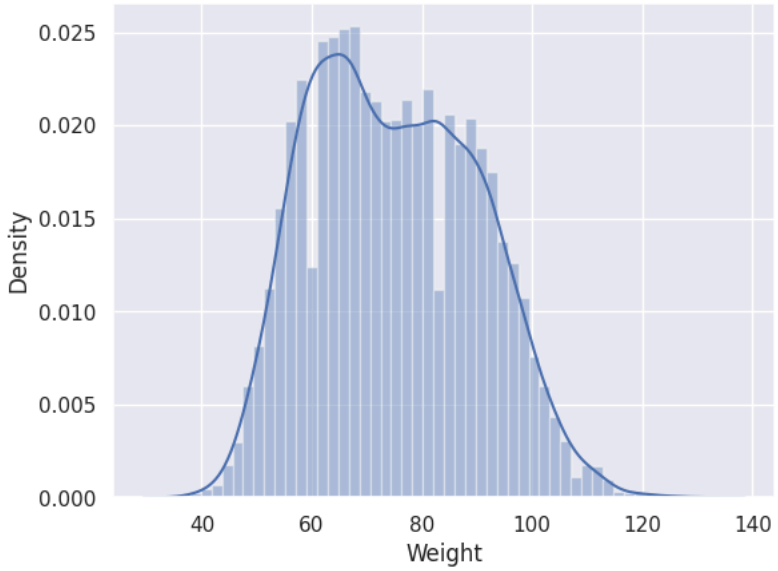
```
<ipython-input-16-ac6457c483b4>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(calories_data['Weight'])
<Axes: xlabel='Weight', ylabel='Density'>
```



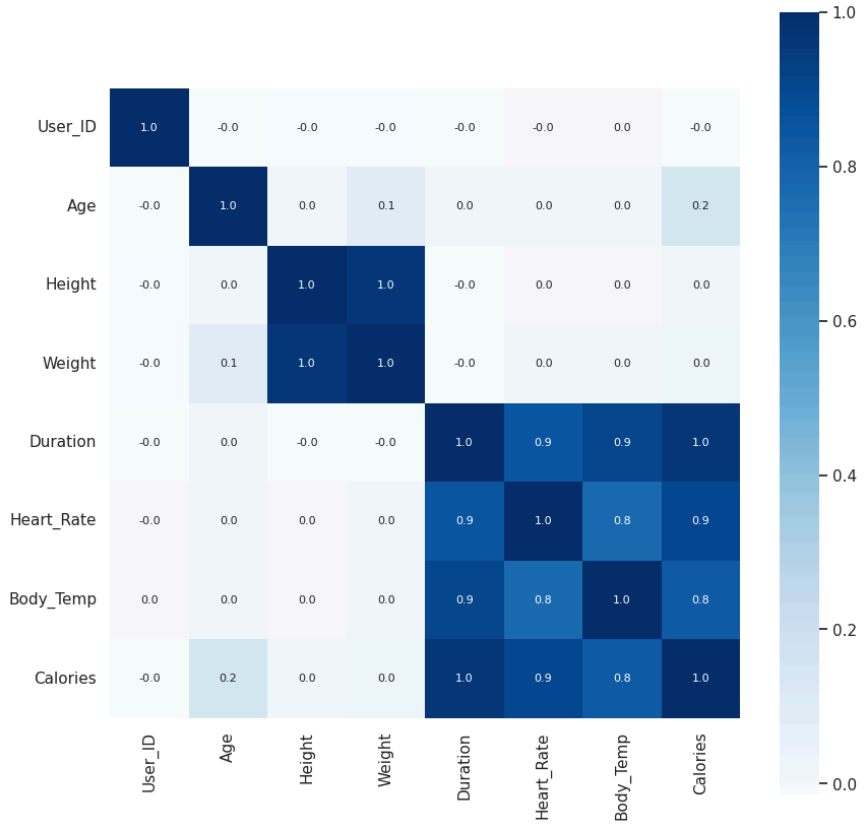

```
correlation = calories_data.corr()
```

```
<ipython-input-17-65125daba8ee>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  correlation = calories_data.corr()
```

```
# constructing a heatmap to understand the correlation

plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':8}, cmap='Blues')
```

```
<Axes: >
```




```
calories_data.replace({"Gender":{'male':0,'female':1}}, inplace=True)
```

```
calories_data.head()
```

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	0	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	1	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	0	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	1	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	1	27	154.0	58.0	10.0	81.0	39.8	35.0

Next steps: Generate code with `calories_data` View recommended plots

```
X = calories_data.drop(columns=['User_ID','Calories'], axis=1)
Y = calories_data['Calories']
```

```
print(X)
```

```
       Gender  Age  Height  Weight  Duration  Heart_Rate  Body_Temp
0           0   68   190.0    94.0      29.0       105.0       40.8
1           1   20   166.0    60.0      14.0        94.0       40.3
2           0   69   179.0    79.0       5.0        88.0       38.7
3           1   34   179.0    71.0      13.0       100.0       40.5
4           1   27   154.0    58.0      10.0        81.0       39.8
...       ...  ...     ...     ...       ...         ...        ...
14995       1   20   193.0    86.0      11.0        92.0       40.4
14996       1   27   165.0    65.0       6.0        85.0       39.2
14997       1   43   159.0    58.0      16.0        90.0       40.1
14998       0   78   193.0    97.0       2.0        84.0       38.3
14999       0   63   173.0    79.0      18.0        92.0       40.5
```

```
[15000 rows x 7 columns]
```

```
print(Y)
```

```
0        231.0
1         66.0
2         26.0
3         71.0
4         35.0
         ...
14995     45.0
14996     23.0
14997     75.0
14998     11.0
14999     98.0
Name: Calories, Length: 15000, dtype: float64
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```