In [1]:
```python
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 sklearn.linear_model import LinearRegression
```

In [2]:
```python
df = pd.read_csv("Sleep.csv")
# .dropna(axis="columns")
df
```

Out[2]:

| | Person ID | Gender | Age | Occupation | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | BMI Category | Blood Pressure | Hea Ra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Male | 27 | Software Engineer | 6.1 | 6 | 42 | 6 | Overweight | 126/83 | |
| **1** | 2 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | |
| **2** | 3 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | |
| **3** | 4 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | |
| **4** | 5 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **369** | 370 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | |
| **370** | 371 | Female | 59 | Nurse | 8.0 | 9 | 75 | 3 | Overweight | 140/95 | |
| **371** | 372 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | |
| **372** | 373 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | |
| **373** | 374 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | |

374 rows × 13 columns

In [3]:
```python
df.head()
```

Out[3]:

| | Person ID | Gender | Age | Occupation | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | BMI Category | Blood Pressure | Heart Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | Male | 27 | Software Engineer | 6.1 | 6 | 42 | 6 | Overweight | 126/83 | 77 |

| | Person ID | Gender | Age | Occupation | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | BMI Category | Blood Pressure | Heart Rate |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 2 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | 75 |
| **2** | 3 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | 75 |
| **3** | 4 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 |
| **4** | 5 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 |

# Data cleaning and pre processing

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Person ID                374 non-null    int64
 1   Gender                   374 non-null    object
 2   Age                      374 non-null    int64
 3   Occupation               374 non-null    object
 4   Sleep Duration           374 non-null    float64
 5   Quality of Sleep         374 non-null    int64
 6   Physical Activity Level  374 non-null    int64
 7   Stress Level             374 non-null    int64
 8   BMI Category             374 non-null    object
 9   Blood Pressure           374 non-null    object
 10  Heart Rate               374 non-null    int64
 11  Daily Steps              374 non-null    int64
 12  Sleep Disorder           374 non-null    object
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

In [5]:
```python
df.describe()
```

Out[5]:

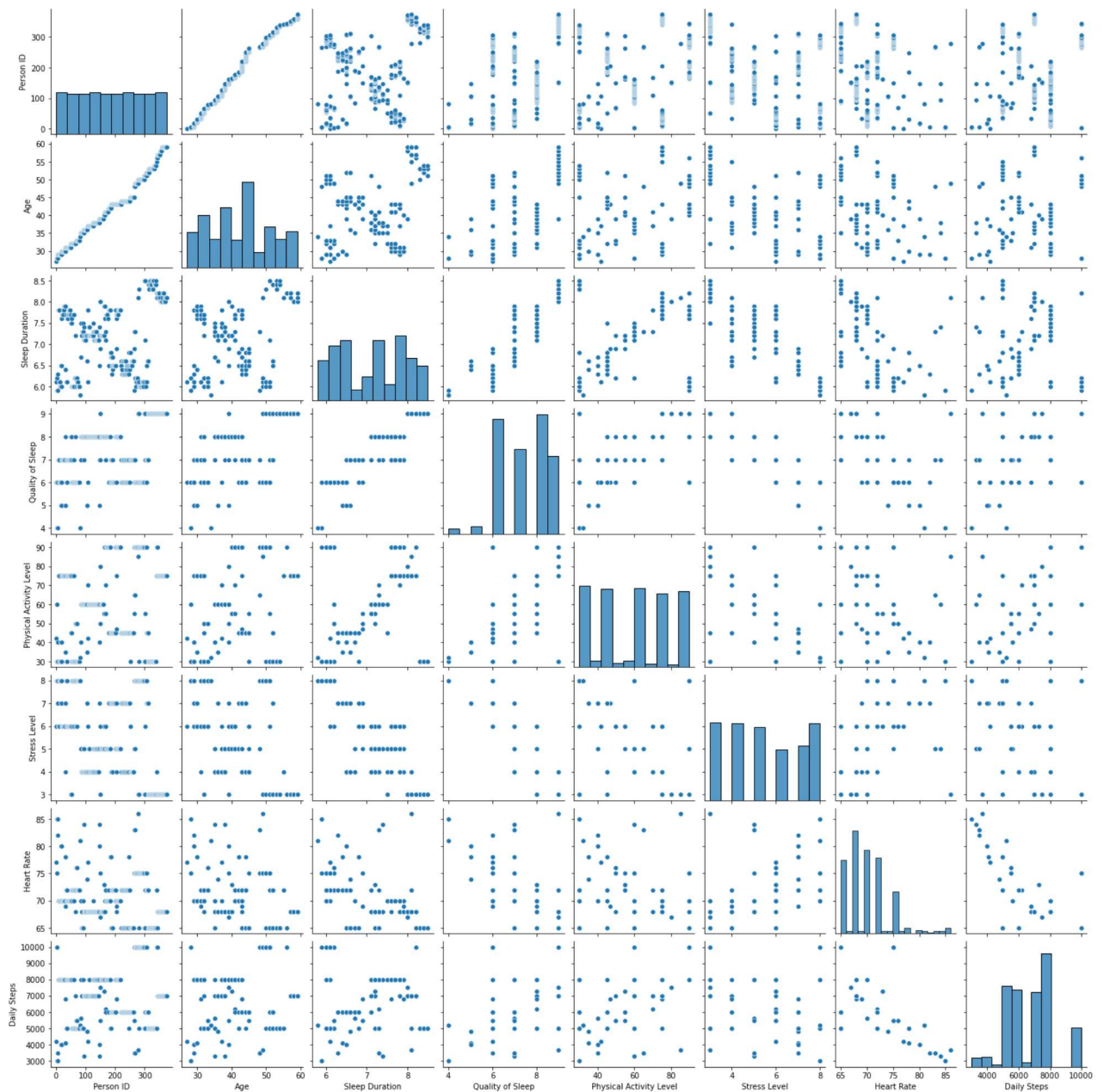| | Person ID | Age | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | Heart Rate | Daily Steps |
|---|---|---|---|---|---|---|---|---|
| **count** | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 |
| **mean** | 187.500000 | 42.184492 | 7.132086 | 7.312834 | 59.171123 | 5.385027 | 70.165775 | 6816.844920 |
| **std** | 108.108742 | 8.673133 | 0.795657 | 1.196956 | 20.830804 | 1.774526 | 4.135676 | 1617.915679 |
| **min** | 1.000000 | 27.000000 | 5.800000 | 4.000000 | 30.000000 | 3.000000 | 65.000000 | 3000.000000 |
| **25%** | 94.250000 | 35.250000 | 6.400000 | 6.000000 | 45.000000 | 4.000000 | 68.000000 | 5600.000000 |
| **50%** | 187.500000 | 43.000000 | 7.200000 | 7.000000 | 60.000000 | 5.000000 | 70.000000 | 7000.000000 |
| **75%** | 280.750000 | 50.000000 | 7.800000 | 8.000000 | 75.000000 | 7.000000 | 72.000000 | 8000.000000 |
| **max** | 374.000000 | 59.000000 | 8.500000 | 9.000000 | 90.000000 | 8.000000 | 86.000000 | 10000.000000 |

In [6]:
```python
df.columns
```

Out[6]: Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
              'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
              'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
              'Sleep Disorder'],
            dtype='object')

# EDA and VISUALIZATION

In [7]:
```python
sns.pairplot(df)
```

Out[7]: <seaborn.axisgrid.PairGrid at 0x1f053131ca0>

In [8]:
```python
sns.distplot(df['Sleep Duration'])
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
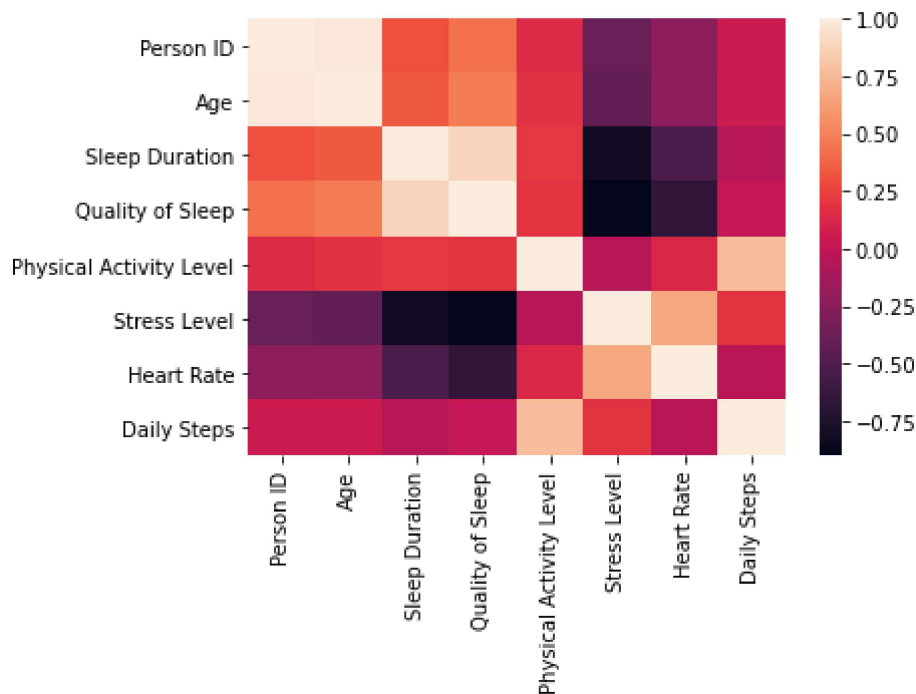  warnings.warn(msg, FutureWarning)

Out[8]:  <AxesSubplot:xlabel='Sleep Duration', ylabel='Density'>
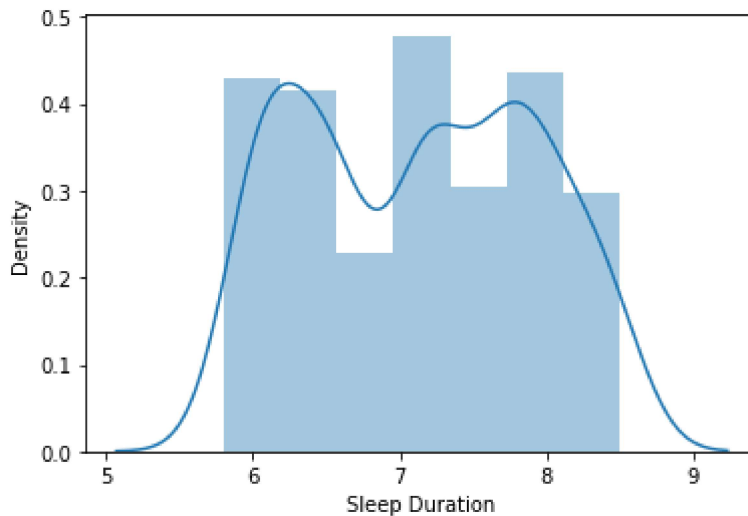


In [9]:
```python
df1 = df[['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
          'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
          'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
          'Sleep Disorder']]
```

In [10]:
```python
sns.heatmap(df1.corr())
```

Out[10]:  <AxesSubplot:>

```
In [11]:   x = df1[['Person ID',  'Sleep Duration',
               'Quality of Sleep','Stress Level',
                'Heart Rate', 'Daily Steps']]
           y = df1[ 'Age']
```

## split the data into training and test data

```
In [12]:   x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

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

Out[13]:   LinearRegression()

```
In [14]:   lr.intercept_
```

Out[14]:   11.899286331704307

```
In [15]:   coeff = pd.DataFrame(lr.coef_, x.columns, columns =['Co-efficient'])
           coeff
```
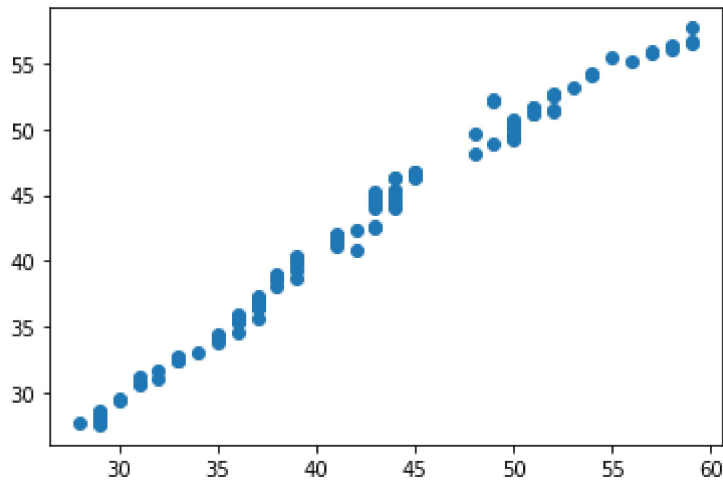
Out[15]:

|                   | Co-efficient |
|-------------------|--------------|
| Person ID         | 0.077905     |
| Sleep Duration    | 0.287208     |
| Quality of Sleep  | 0.591638     |
| Stress Level      | 0.031027     |

|  | Co-efficient |
| --- | --- |
| **Heart Rate** | 0.123934 |
| **Daily Steps** | 0.000072 |

In [16]:
```python
prediction = lr.predict(x_test)
plt.scatter(y_test, prediction)
```

Out[16]:  <matplotlib.collections.PathCollection at 0x1f05513ed00>

In [17]:
```python
lr.score(x_test,y_test)
```

Out[17]:  0.9819159155441144

# ACURACY

In [18]:
```python
from sklearn.linear_model import Ridge,Lasso
```

In [19]:
```python
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
rr.score(x_train,y_train)
```

Out[19]:  0.9868875766465539

In [20]:
```python
rr.score(x_test,y_test)
```

Out[20]:  0.9819248705055501

In [21]:
```python
la = Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[21]: Lasso(alpha=10)

In [22]:
```python
la.score(x_test,y_test)
```

Out[22]: 0.9779820947092206