kaviyadevi 20106064

In [3]:
```
#to import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
#to import dataset
data=pd.read_csv(r"C:\Users\user\Downloads\16_Sleep_health_and_lifestyle_dataset
data
```

Out[4]:

| | Person ID | Gender | Age | Occupation | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | BMI Category | Blood Pressure |
|---|---|---|---|---|---|---|---|---|---|---|
| **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 [5]:
```
#to display top 5 rows
data.head()
```

Out[5]:

| | Person ID | Gender | Age | Occupation | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | BMI Category | Blood Pressure |
|---|---|---|---|---|---|---|---|---|---|---|
| **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 |

# DATA CLEANING AND PREPROCESSING

In [6]:
```
#
data.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 [7]:
```
#to display summary of statistics(here to know min max value)
data.describe()
```

Out[7]:

| | Person ID | Age | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | Heart Rate | Daily |
|---|---|---|---|---|---|---|---|---|
| **count** | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374.000000 | 374. |
| **mean** | 187.500000 | 42.184492 | 7.132086 | 7.312834 | 59.171123 | 5.385027 | 70.165775 | 6816. |
| **std** | 108.108742 | 8.673133 | 0.795657 | 1.196956 | 20.830804 | 1.774526 | 4.135676 | 1617. |
| **min** | 1.000000 | 27.000000 | 5.800000 | 4.000000 | 30.000000 | 3.000000 | 65.000000 | 3000. |
| **25%** | 94.250000 | 35.250000 | 6.400000 | 6.000000 | 45.000000 | 4.000000 | 68.000000 | 5600. |
| **50%** | 187.500000 | 43.000000 | 7.200000 | 7.000000 | 60.000000 | 5.000000 | 70.000000 | 7000. |
| **75%** | 280.750000 | 50.000000 | 7.800000 | 8.000000 | 75.000000 | 7.000000 | 72.000000 | 8000. |
| **max** | 374.000000 | 59.000000 | 8.500000 | 9.000000 | 90.000000 | 8.000000 | 86.000000 | 10000. |

In [8]:
```
#to display the column heading
data.columns
```

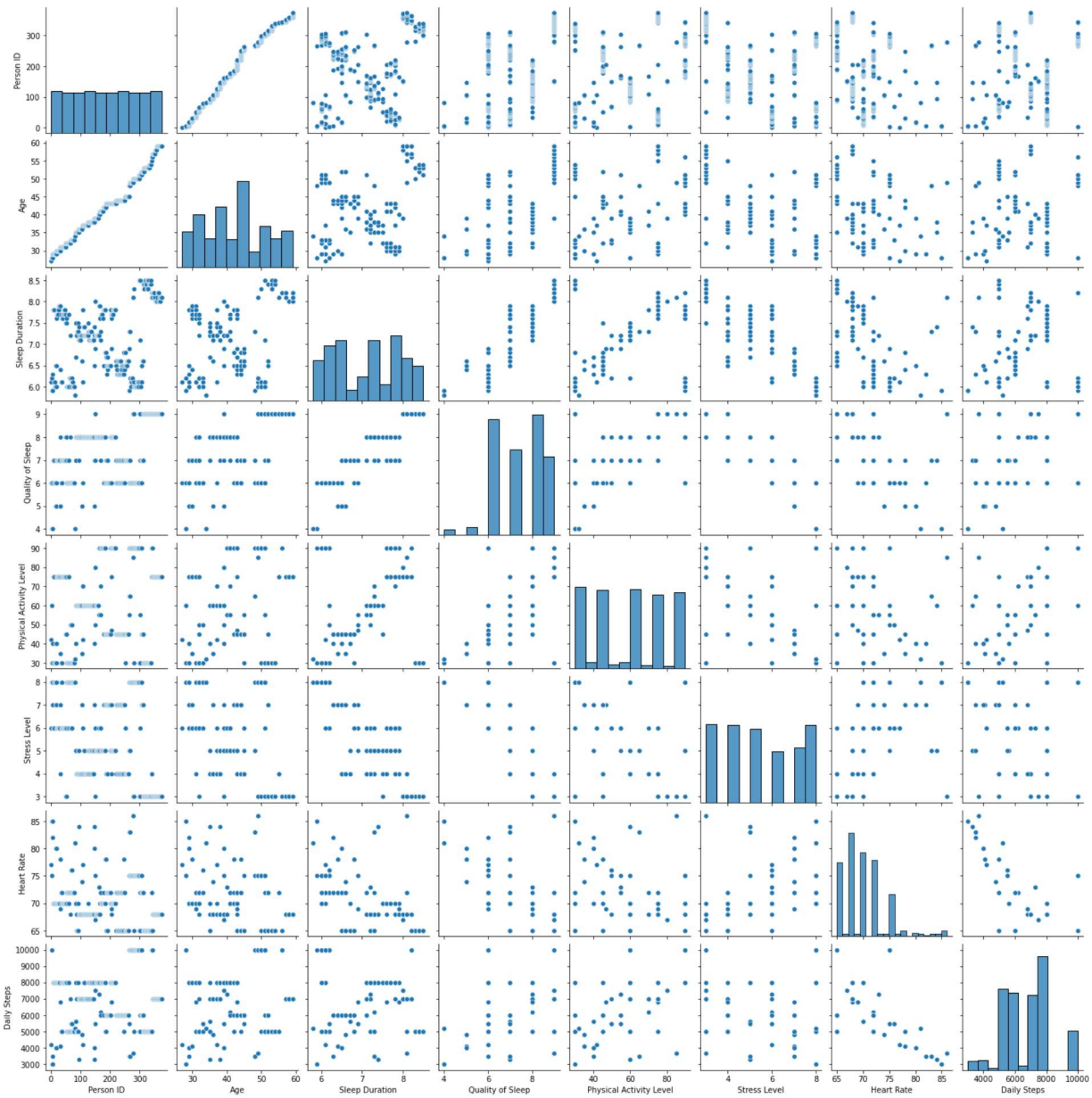
Out[8]:
```
Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
       'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
       'Sleep Disorder'],
      dtype='object')
```

In [9]:
```
#here there is no missing values (identified through info() 5000 data are descrit
```

# EDA and DATA VISUALIZATION

In [10]:
```
sns.pairplot(data)
```

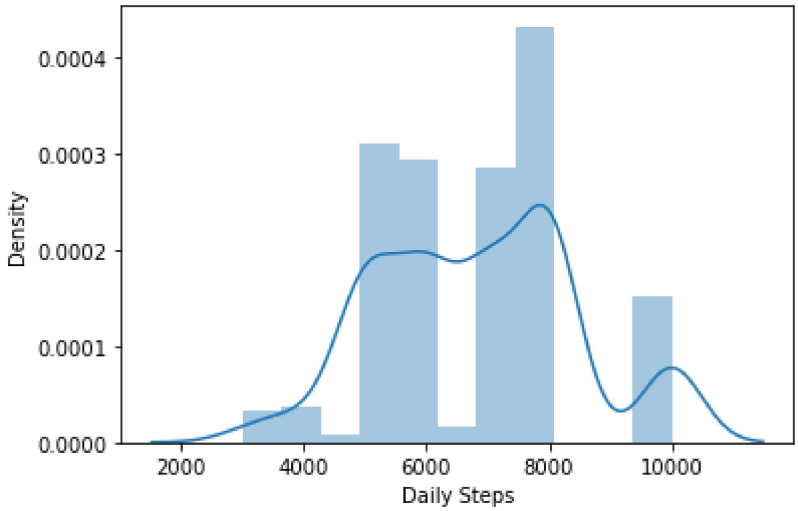
Out[10]:
```
<seaborn.axisgrid.PairGrid at 0x177c8b29c10>
```



In [12]:
```
sns.distplot(data['Daily Steps'])
```

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

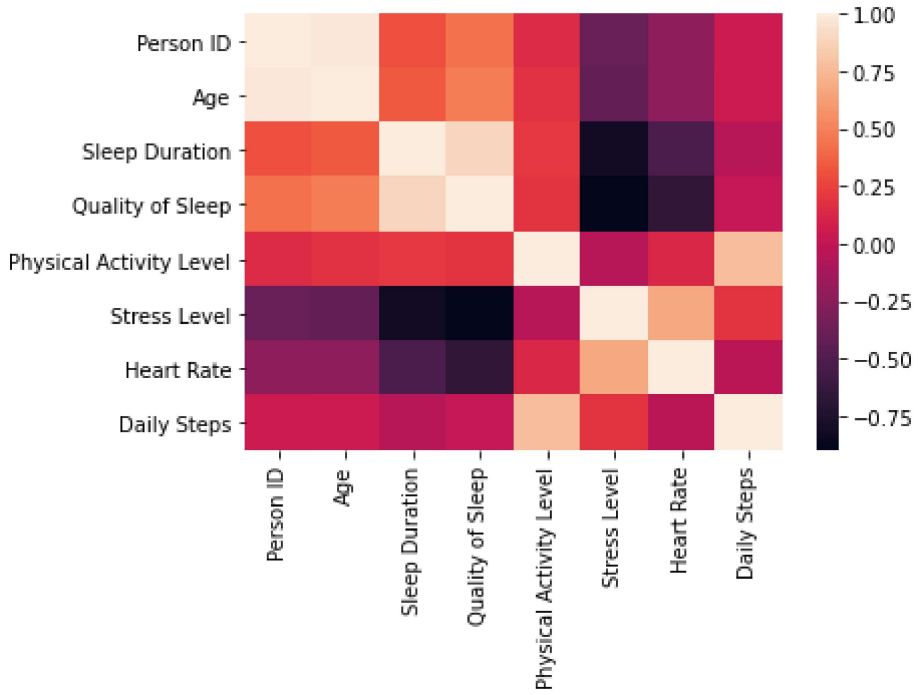
Out[12]: <AxesSubplot:xlabel='Daily Steps', ylabel='Density'>




In [18]:
```
df=data[['Person ID', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
       'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
       'Sleep Disorder']]
```

In [19]:
```
sns.heatmap(df.corr())
```

Out[19]: <AxesSubplot:>




# TRAINING MODEL

In [28]:
```
x=df[['Person ID', 'Age', 'Sleep Duration','Quality of Sleep', 'Physical Activity
y=df['Daily Steps']
```

In [29]:
```
#to split my dataset into trainning and test

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [30]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[30]: LinearRegression()

In [31]:
```
#to find intercept
print(lr.intercept_)
```

```
12080.778493615875
```

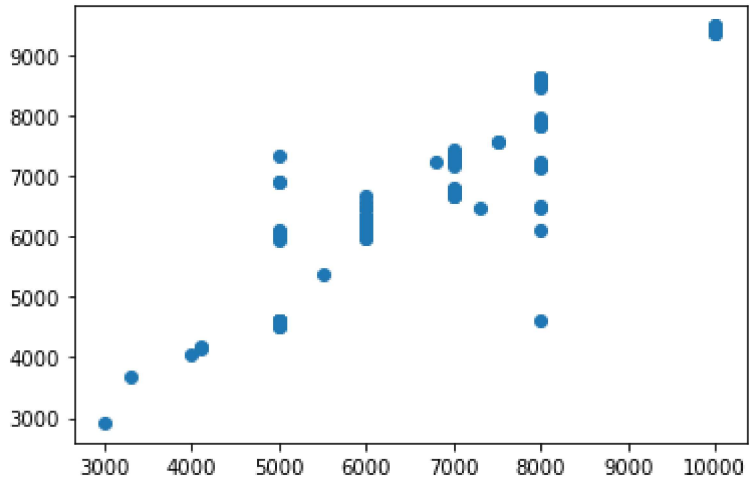
In [32]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[32]:

| | Co-efficient |
|---|---|
| **Person ID** | -7.223204 |
| **Age** | 94.477624 |
| **Sleep Duration** | -422.931489 |
| **Quality of Sleep** | 299.769730 |
| **Physical Activity Level** | 65.810137 |
| **Stress Level** | 567.093106 |
| **Heart Rate** | -199.864274 |

In [33]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[33]:
```
<matplotlib.collections.PathCollection at 0x177ce6173d0>
```




In [34]:
```
print(lr.score(x_test,y_test))
```

```
0.7988603716966736
```

# RIDGE AND LASSO REGRESSION

In [35]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [36]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[36]:
```
Ridge(alpha=10)
```

In [37]:
```
rr.score(x_test,y_test)
```

Out[37]:
```
0.7977506676371726
```

In [38]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[38]:
```
Lasso(alpha=10)
```

In [39]:
```
la.score(x_test,y_test)
```

Out[39]:
```
0.797706787365225
```