D13 ¶

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

In [2]:
```
df=pd.read_csv(r"C:\Users\user\Downloads\16_Sleep_health_and_lifestyle_dataset
df
```

Out[2]:

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

374 rows × 13 columns

In [3]:

```
df.head(10)
```

Out[3]:

	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
5	6	Male	28	Software Engineer	5.9	4	30	8	Obese	140/90
6	7	Male	29	Teacher	6.3	6	40	7	Obese	140/90
7	8	Male	29	Doctor	7.8	7	75	6	Normal	120/80
8	9	Male	29	Doctor	7.8	7	75	6	Normal	120/80
9	10	Male	29	Doctor	7.8	7	75	6	Normal	120/80

In [4]:

```
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]:
```
df.describe()
```

Out[5]:

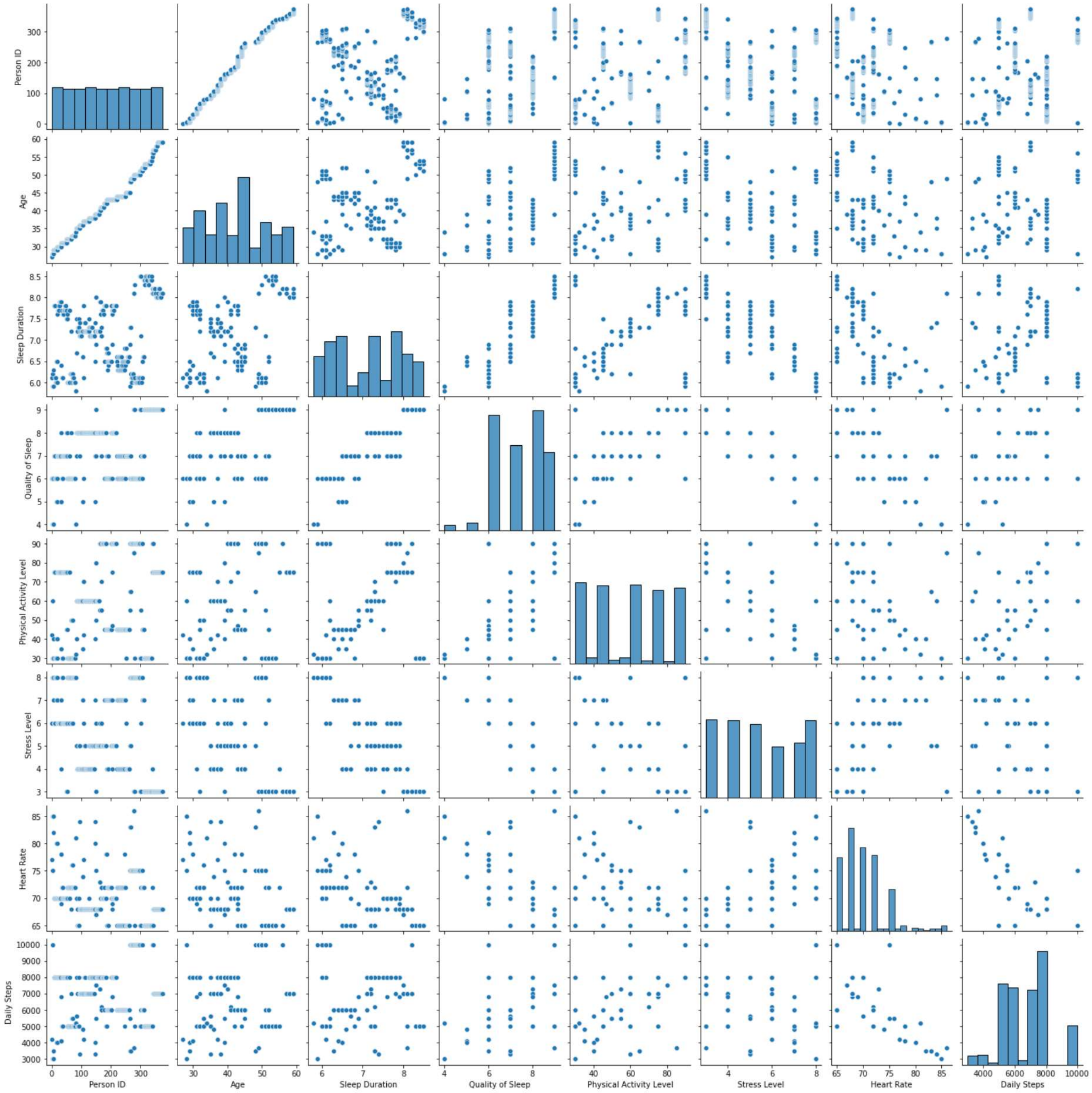
	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Da
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	37
mean	187.500000	42.184492	7.132086	7.312834	59.171123	5.385027	70.165775	681
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676	161
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000	300
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000	560
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000	700
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000	800
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000	1000

In [6]:
```
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')
```

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

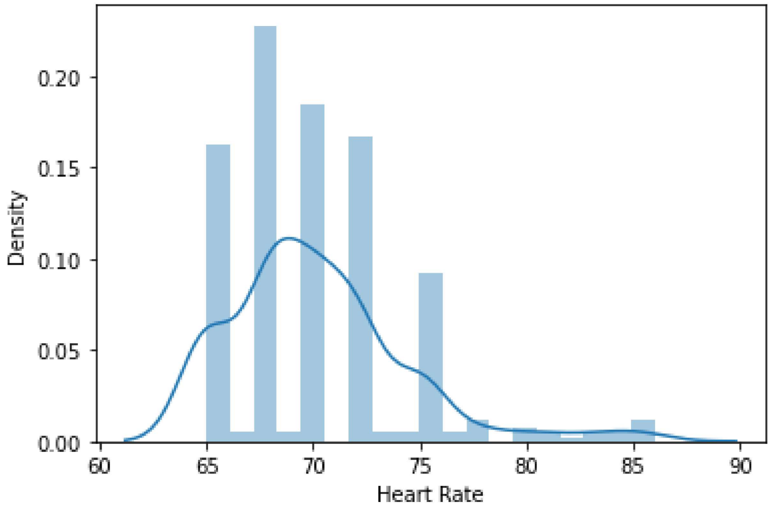
Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x29bc7cc7880>
```



In [8]:
```
sns.distplot(df["Heart Rate"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

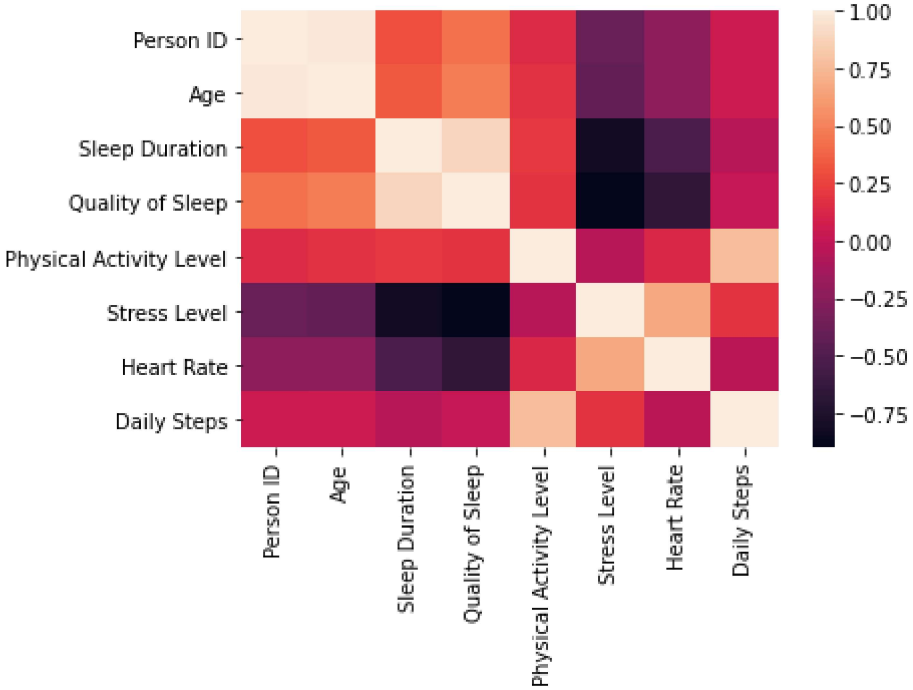
Out[8]:
```
<AxesSubplot:xlabel='Heart Rate', ylabel='Density'>
```



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

```
Out[10]: <AxesSubplot:>
```



```
In [13]: x=df1[['Person ID', 'Age', 'Sleep Duration',
                'Quality of Sleep', 'Physical Activity Level', 'Stress Level'
                , 'Daily Steps']]
         y=df1['Heart Rate']
```

```
In [14]: 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 [15]: from sklearn.linear_model import LinearRegression
         lr=LinearRegression()
         lr.fit(x_train,y_train)
```

```
Out[15]: LinearRegression()
```

```
In [16]: print(lr.intercept_)
```

```
62.1764893326392
```

In [17]:

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

Out[17]:

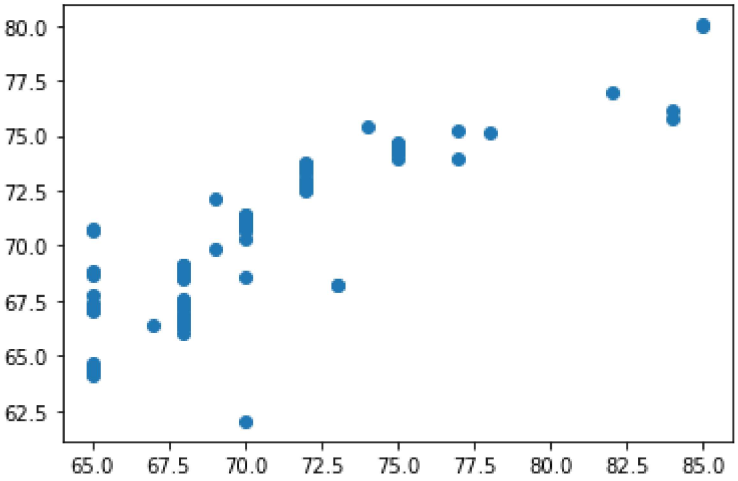
	Co-efficient
Person ID	-0.039314
Age	0.544544
Sleep Duration	0.037423
Quality of Sleep	-1.483060
Physical Activity Level	0.143170
Stress Level	1.181992
Daily Steps	-0.001742

In [18]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[18]:

```
<matplotlib.collections.PathCollection at 0x29bd106c7c0>
```



In [19]:

```
print(lr.score(x_test,y_test))
```

```
0.7370726621562377
```

In [20]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [21]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[21]:

```
Ridge(alpha=10)
```

In [22]:

```
rr.score(x_test,y_test)
```

Out[22]:

```
0.73806387225675
```

```
In [23]: la=Lasso(alpha=10)
         la.fit(x_train,y_train)

Out[23]: Lasso(alpha=10)

In [24]: la.score(x_test,y_test)

Out[24]: 0.11165647371768561

In [ ]:
```