

```
In [66]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [67]: .read_csv(r"C:\Users\user\Downloads\16_Sleep_health_and_lifestyle_dataset - 16_Sleep_health_and_lifestyle_dataset.csv")

| | | | | | | | | | | | | | |
|-----|-----|--------|-----|----------------------|-----|-----|-----|-----|------------|--------|-----|------|-------------|
| 3 | 4 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 | 3000 | Sleep Apnea |
| 4 | 5 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 | 3000 | Sleep Apnea |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 369 | 370 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | 68 | 7000 | Sleep Apnea |
| 370 | 371 | Female | 59 | Nurse | 8.0 | 9 | 75 | 3 | Overweight | 140/95 | 68 | 7000 | Sleep Apnea |
| 371 | 372 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | 68 | 7000 | Sleep Apnea |
| 372 | 373 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | 68 | 7000 | Sleep Apnea |
| 373 | 374 | Female | 59 | Nurse | 8.1 | 9 | 75 | 3 | Overweight | 140/95 | 68 | 7000 | Sleep Apnea |

374 rows × 13 columns

```
In [68]: df=data.head(100)
df
```

Out[68]:

| | 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 |
|-----|-----------|--------|-----|----------------------|----------------|------------------|-------------------------|--------------|--------------|----------------|------------|-------------|----------------|
| 0 | 1 | Male | 27 | Software Engineer | 6.1 | 6 | 42 | 6 | Overweight | 126/83 | 77 | 4200 | None |
| 1 | 2 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | 75 | 10000 | None |
| 2 | 3 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | 75 | 10000 | None |
| 3 | 4 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 | 3000 | Sleep Apnea |
| 4 | 5 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 | 3000 | Sleep Apnea |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 95 | 96 | Female | 36 | Accountant | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 96 | 97 | Female | 36 | Accountant | 7.2 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 97 | 98 | Female | 36 | Accountant | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 98 | 99 | Female | 36 | Teacher | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 99 | 100 | Female | 36 | Teacher | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |

100 rows × 13 columns

```
In [69]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 13 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Person ID                            100 non-null    int64
1   Gender                               100 non-null    object
2   Age                                   100 non-null    int64
3   Occupation                           100 non-null    object
4   Sleep Duration                       100 non-null    float64
5   Quality of Sleep                     100 non-null    int64
6   Physical Activity Level              100 non-null    int64
7   Stress Level                         100 non-null    int64
8   BMI Category                         100 non-null    object
9   Blood Pressure                       100 non-null    object
10  Heart Rate                           100 non-null    int64
11  Daily Steps                          100 non-null    int64
12  Sleep Disorder                       100 non-null    object
dtypes: float64(1), int64(7), object(5)
memory usage: 10.3+ KB
```

```
In [70]: df.describe()
```

Out[70]:

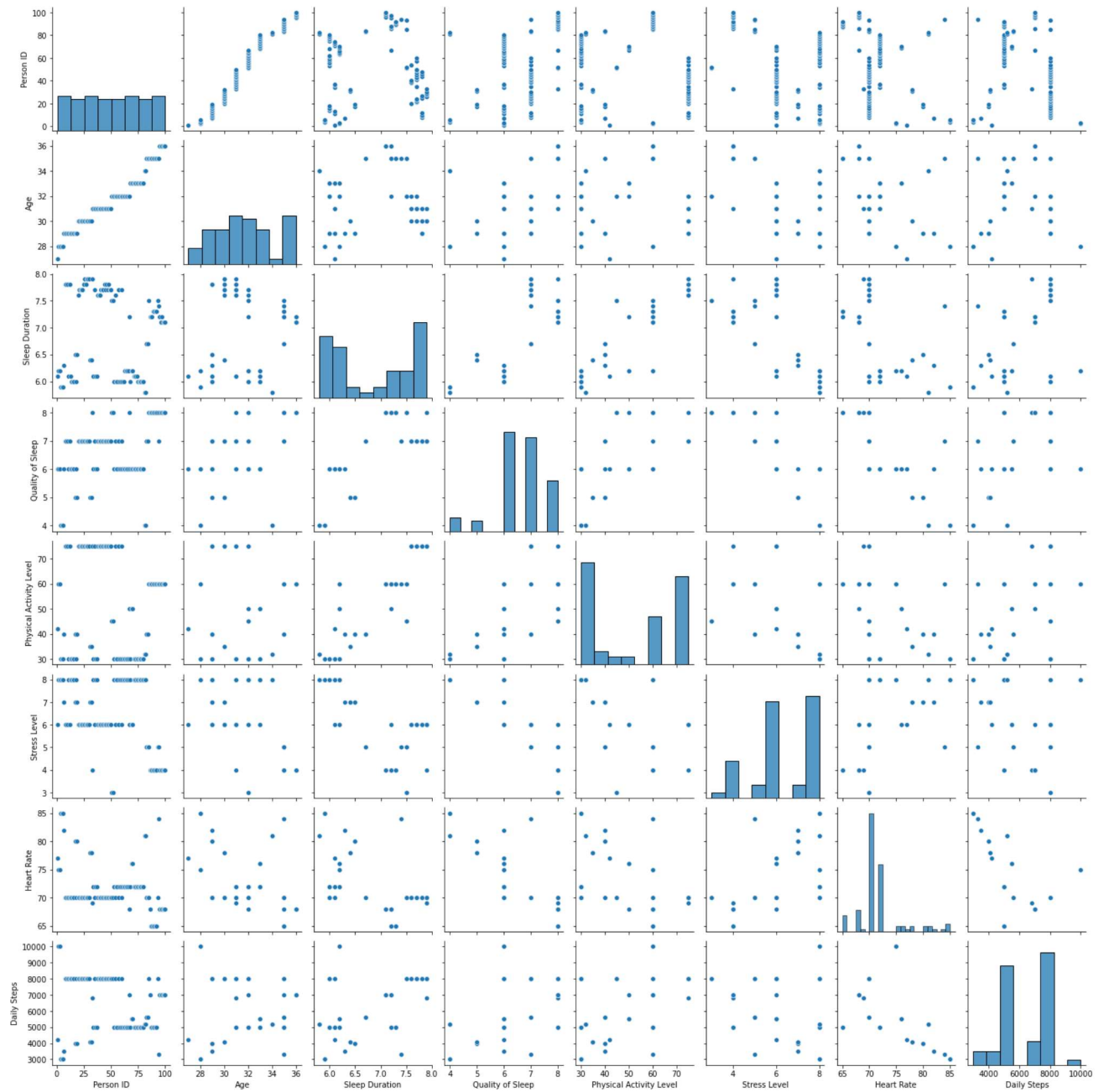
| | Person ID | Age | Sleep Duration | Quality of Sleep | Physical Activity Level | Stress Level | Heart Rate | Daily Steps |
|-------|------------|------------|----------------|------------------|-------------------------|--------------|------------|--------------|
| count | 100.000000 | 100.000000 | 100.000000 | 100.000000 | 100.000000 | 100.000000 | 100.000000 | 100.000000 |
| mean | 50.500000 | 31.69000 | 6.871000 | 6.590000 | 51.910000 | 6.420000 | 71.610000 | 6426.000000 |
| std | 29.011492 | 2.26388 | 0.766903 | 1.005992 | 19.429279 | 1.485145 | 4.240009 | 1689.517294 |
| min | 1.000000 | 27.00000 | 5.800000 | 4.000000 | 30.000000 | 3.000000 | 65.000000 | 3000.000000 |
| 25% | 25.750000 | 30.00000 | 6.100000 | 6.000000 | 30.000000 | 6.000000 | 70.000000 | 5000.000000 |
| 50% | 50.500000 | 31.50000 | 7.100000 | 7.000000 | 60.000000 | 6.000000 | 70.000000 | 7000.000000 |
| 75% | 75.250000 | 33.00000 | 7.700000 | 7.000000 | 75.000000 | 8.000000 | 72.000000 | 8000.000000 |
| max | 100.000000 | 36.00000 | 7.900000 | 8.000000 | 75.000000 | 8.000000 | 85.000000 | 10000.000000 |

```
In [71]: df.columns
```

Out[71]: 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 [72]: sns.pairplot(df)
```

```
Out[72]: <seaborn.axisgrid.PairGrid at 0x1a211c51580>
```



```
In [75]: da=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']]
da
```

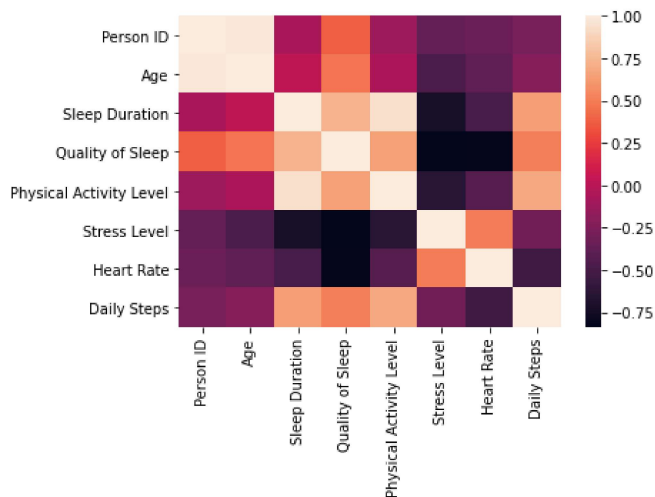
Out[75]:

| | 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 |
|-----|-----------|--------|-----|----------------------|----------------|------------------|-------------------------|--------------|--------------|----------------|------------|-------------|----------------|
| 0 | 1 | Male | 27 | Software Engineer | 6.1 | 6 | 42 | 6 | Overweight | 126/83 | 77 | 4200 | None |
| 1 | 2 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | 75 | 10000 | None |
| 2 | 3 | Male | 28 | Doctor | 6.2 | 6 | 60 | 8 | Normal | 125/80 | 75 | 10000 | None |
| 3 | 4 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 | 3000 | Sleep Apnea |
| 4 | 5 | Male | 28 | Sales Representative | 5.9 | 4 | 30 | 8 | Obese | 140/90 | 85 | 3000 | Sleep Apnea |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 95 | 96 | Female | 36 | Accountant | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 96 | 97 | Female | 36 | Accountant | 7.2 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 97 | 98 | Female | 36 | Accountant | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 98 | 99 | Female | 36 | Teacher | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |
| 99 | 100 | Female | 36 | Teacher | 7.1 | 8 | 60 | 4 | Normal | 115/75 | 68 | 7000 | None |

100 rows × 13 columns

```
In [76]: sns.heatmap(da.corr())
```

Out[76]: <AxesSubplot:>



```
In [77]: x=da[['Person ID','Sleep Duration',
               'Quality of Sleep', 'Physical Activity Level', 'Stress Level', 'Heart Rate', 'Daily Steps']]
y=da['Age']
```

```
In [78]: 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 [79]: from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[79]: LinearRegression()

```
In [80]: print(lr.intercept_)
```

32.502048496634245

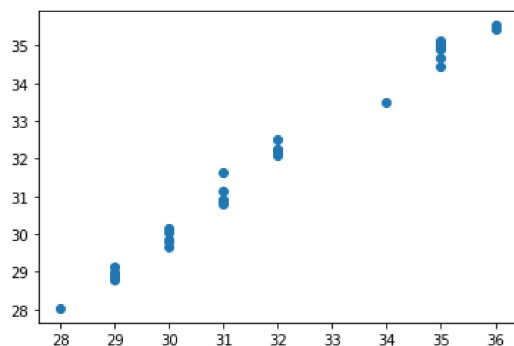
```
In [81]: coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[81]:

| | Co-efficient |
|-------------------------|--------------|
| Person ID | 0.069815 |
| Sleep Duration | -0.272682 |
| Quality of Sleep | 0.027579 |
| Physical Activity Level | 0.001087 |
| Stress Level | -0.259593 |
| Heart Rate | -0.016567 |
| Daily Steps | 0.000021 |

```
In [82]: prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[82]: <matplotlib.collections.PathCollection at 0x1a216495d00>



```
In [83]: print(lr.score(x_test,y_test))
```

0.9858614919293583

```
In [84]: print(lr.score(x_train,y_train))
```

0.9702134262087738

```
In [85]: from sklearn.linear_model import Ridge,Lasso
```

```
In [86]: rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[86]: Ridge(alpha=10)

```
In [87]: rr.score(x_test,y_test)
```

Out[87]: 0.9839418034552196

```
In [88]: la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[88]: Lasso(alpha=10)

```
In [89]: la.score(x_test,y_test)
```

Out[89]: 0.9213901261758228

In []: