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

In [289]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\16_Sleep_health_and_lifestyle_dataset.csv")
a
```

Out[289]:

	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
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
...	...	...	...	...	...	...	...	...	...	...	...
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68

374 rows × 13 columns

In [290]:
```
a=a.head(10)
a
```

Out[290]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	D St
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3
5	6	Male	28	Software Engineer	5.9	4	30	8	Obese	140/90	85	3
6	7	Male	29	Teacher	6.3	6	40	7	Obese	140/90	82	3
7	8	Male	29	Doctor	7.8	7	75	6	Normal	120/80	70	8
8	9	Male	29	Doctor	7.8	7	75	6	Normal	120/80	70	8
9	10	Male	29	Doctor	7.8	7	75	6	Normal	120/80	70	8

In [291]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Person ID                10 non-null     int64  
 1   Gender                   10 non-null     object 
 2   Age                      10 non-null     int64  
 3   Occupation               10 non-null     object 
 4   Sleep Duration           10 non-null     float64
 5   Quality of Sleep         10 non-null     int64  
 6   Physical Activity Level  10 non-null     int64  
 7   Stress Level             10 non-null     int64  
 8   BMI Category             10 non-null     object 
 9   Blood Pressure           10 non-null     object 
 10  Heart Rate               10 non-null     int64  
 11  Daily Steps              10 non-null     int64  
 12  Sleep Disorder           10 non-null     object 
dtypes: float64(1), int64(7), object(5)
memory usage: 1.1+ KB
```

In [292]:
```
a.columns
```

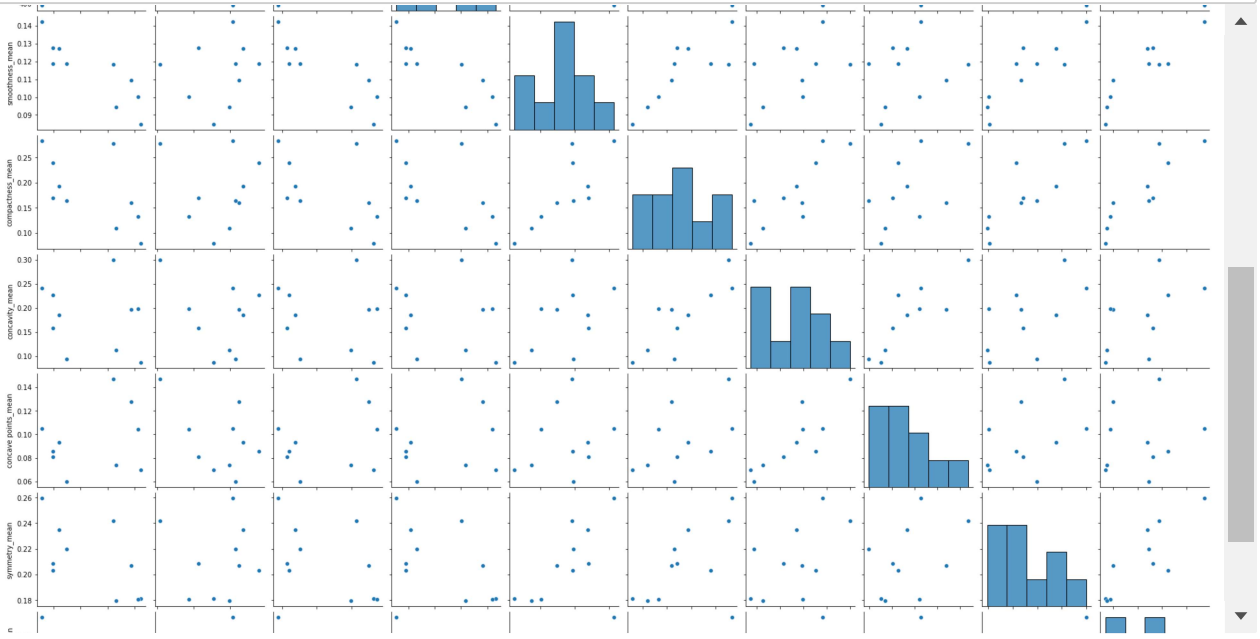
Out[292]:
```
Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level',
       'BMI Category', 'Blood Pressure', 'Heart Rate', 'Daily Steps',
       'Sleep Disorder'],
      dtype='object')
```

In [293]:
```
a.describe()
```

Out[293]:

	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
count	10.00000	10.000000	10.000000	10.000000	10.000000	10.000000	10.00000	10.000000
mean	5.50000	28.300000	6.590000	5.700000	51.700000	7.100000	77.40000	6070.000000
std	3.02765	0.674949	0.846496	1.251666	19.465354	0.994429	6.41526	2989.630226
min	1.00000	27.000000	5.900000	4.000000	30.000000	6.000000	70.00000	3000.000000
25%	3.25000	28.000000	5.950000	4.500000	32.500000	6.000000	71.25000	3125.000000
50%	5.50000	28.000000	6.200000	6.000000	51.000000	7.500000	76.00000	6100.000000
75%	7.75000	29.000000	7.425000	6.750000	71.250000	8.000000	84.25000	8000.000000
max	10.00000	29.000000	7.800000	7.000000	75.000000	8.000000	85.00000	10000.000000

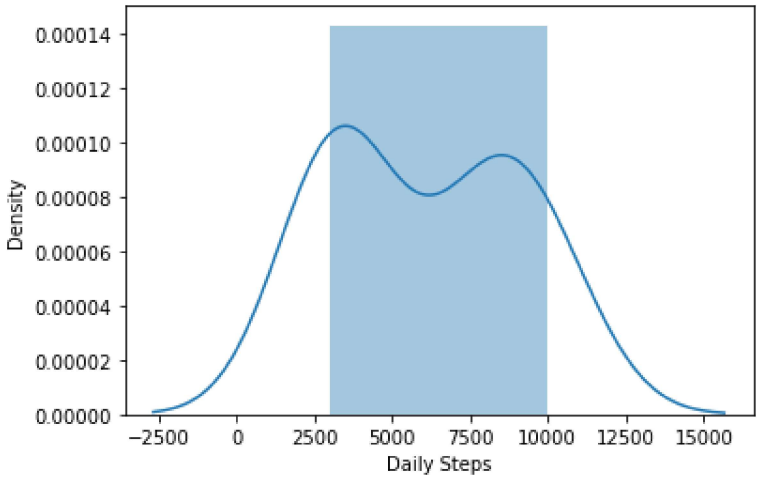
In [294]:
```
sns.pairplot(d)
```



In [295]:
```
sns.distplot(a['Daily Steps'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarnin
g: `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 flexibil
ity) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

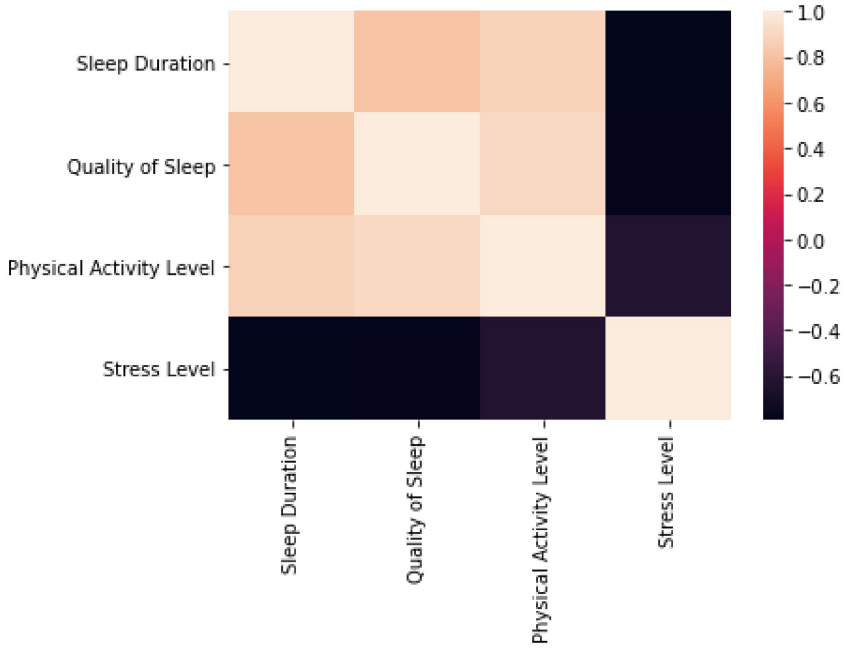
Out[295]: <AxesSubplot:xlabel='Daily Steps', ylabel='Density'>



In [296]:
```
x1=a[['Sleep Duration', 'Quality of Sleep', 'Physical Activity Level', 'Stress Level']]
```

In [297]:
```
sns.heatmap(x1.corr())
```

Out[297]: <AxesSubplot:>



In [298]:
```
x=a[['Sleep Duration','Quality of Sleep', 'Physical Activity Level', 'Stress Level']]
y=a['Daily Steps']
```

In [299]:
```
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 [300]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[300]:
```
LinearRegression()
```

In [301]:
```
print(lr.intercept_)
```

```
-11153.042765840984
```

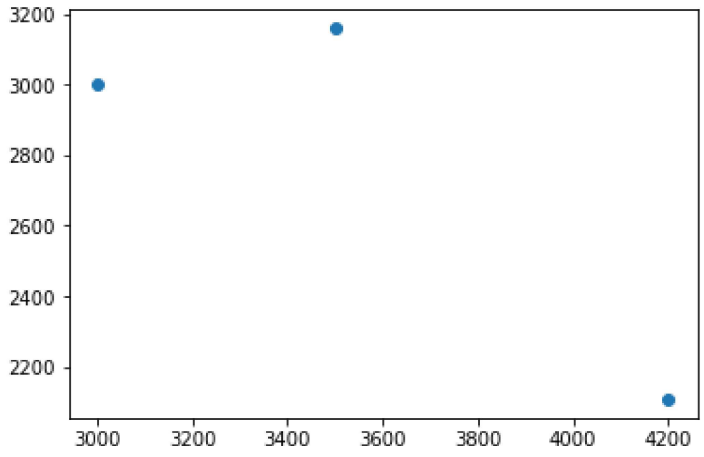
In [302]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[302]:

	Co-efficient
Sleep Duration	-1305.233622
Quality of Sleep	16.353031
Physical Activity Level	245.295467
Stress Level	1803.705624

In [303]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[303]:
```
<matplotlib.collections.PathCollection at 0x190c40e81f0>
```



In [304]:
```
print(lr.score(x_test,y_test))
```

```
-5.183056818542676
```

In [305]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [306]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[306]:
```
Ridge(alpha=10)
```

In [307]:
```
rr.score(x_test,y_test)
```

Out[307]:
```
-1.1412286635195228
```

In [308]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[308]:
```
Lasso(alpha=10)
```

In [309]:
```
la.score(x_test,y_test)
```

Out[309]:
```
-20.459946593410077
```

In [310]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[310]:
```
ElasticNet()
```

In [311]:
```
print(en.coef_)
```

```
[-680.12572035   11.56994357  183.6932392   941.03047253]
```

In [312]:
```
print(en.intercept_)
```

```
-5649.6669855904165
```

In [313]:
```
print(en.predict(x_test))
```

```
[3632.28466316 3422.91199479 4069.90351322]
```

In [314]:
```
en.score(x_test,y_test)
```

Out[314]:
```
-0.1366219358816212
```

In [315]:
```
from sklearn import metrics
```

In [316]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 810.7815451094957
```

In [317]:
```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 1497673.7627136705
```

In [318]:
```
print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
 Root Mean Squared Error 1223.7948205126831
```