Type *Markdown* and LaTeX: α^2

In [4]:
```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
#import dataset
df=pd.read_csv(r"E:\154\16_Sleep_health_and_lifestyle_dataset.csv",low_memory=False).dropna(axis=
df
```

Out[5]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	S Disc
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	I
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	I
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	I
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	S A
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	S A
...	...	...	...	...	...	...	...	...	...	...	...	...	
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	S A
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	S A
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	S A
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	S A
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	S A

374 rows × 13 columns

In [6]:
```
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 [7]:
```
#to display top 5 rows
df.head()
```

Out[7]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure	Heart Rate	Daily Steps	Sle Disord
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	No
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	No
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	No
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sle Apn
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sle Apn

Data cleaning and Pre-Processing

In [8]:
```
#To find null values
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 [9]:
```
# To display summary of statistics
df.describe()
```

Out[9]:

	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

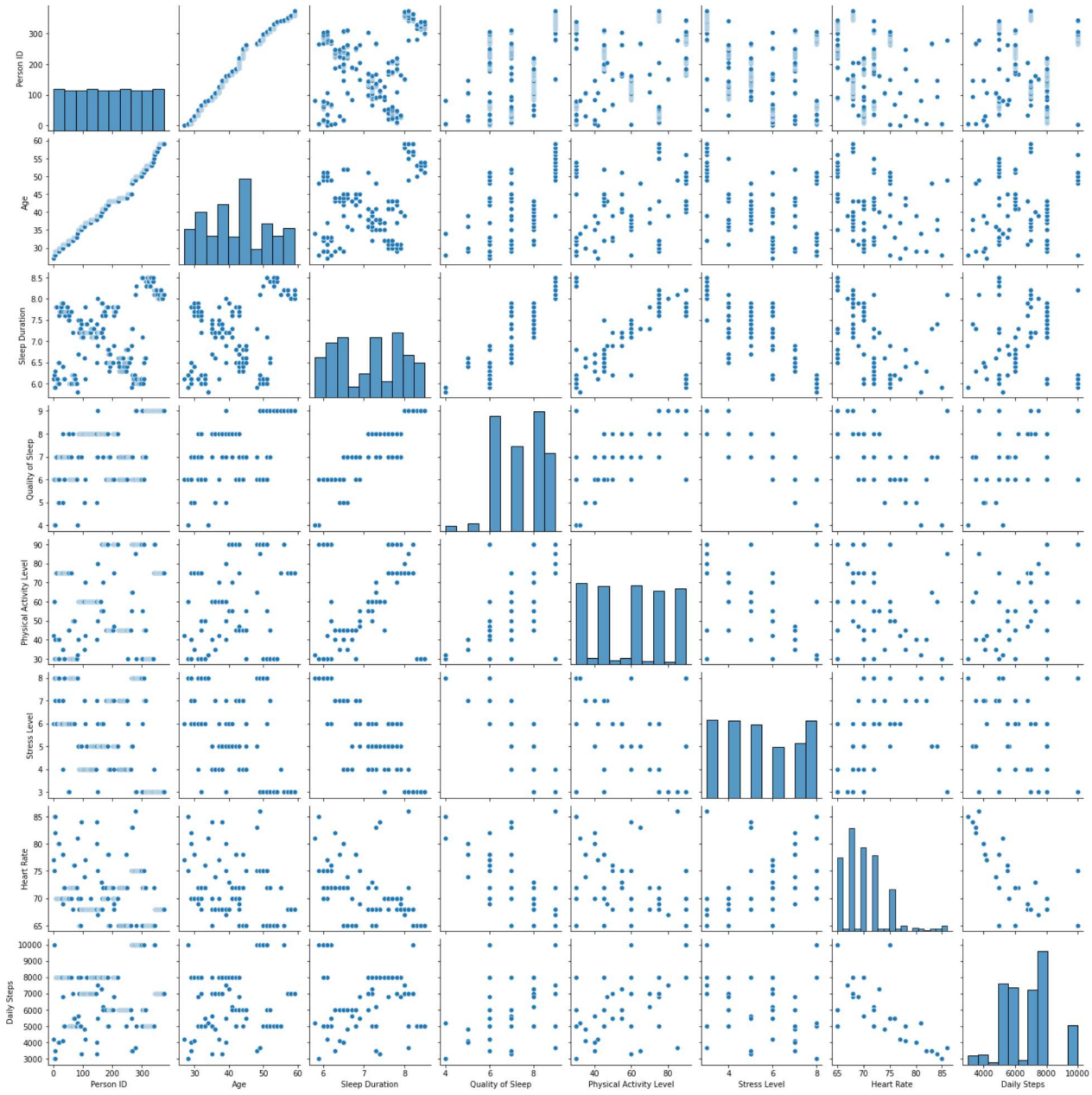
In [10]:
```
#To Display column heading
df.columns
```

Out[10]:
```
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 [11]:
```
sns.pairplot(df)
```

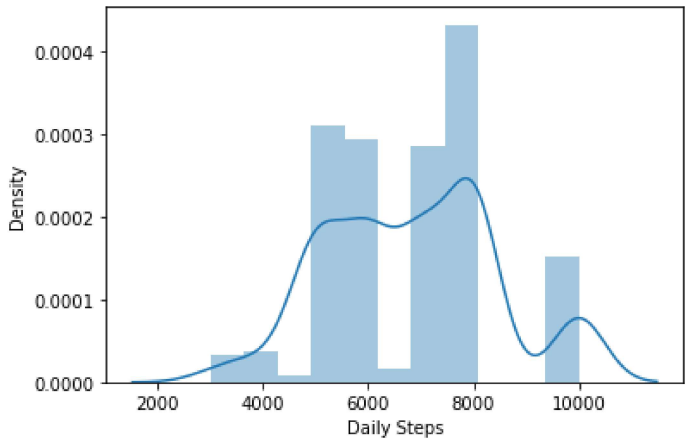
Out[11]:
```
<seaborn.axisgrid.PairGrid at 0x29aa6a8f490>
```



In [12]:
```
sns.distplot(df["Daily Steps"])
```

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

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

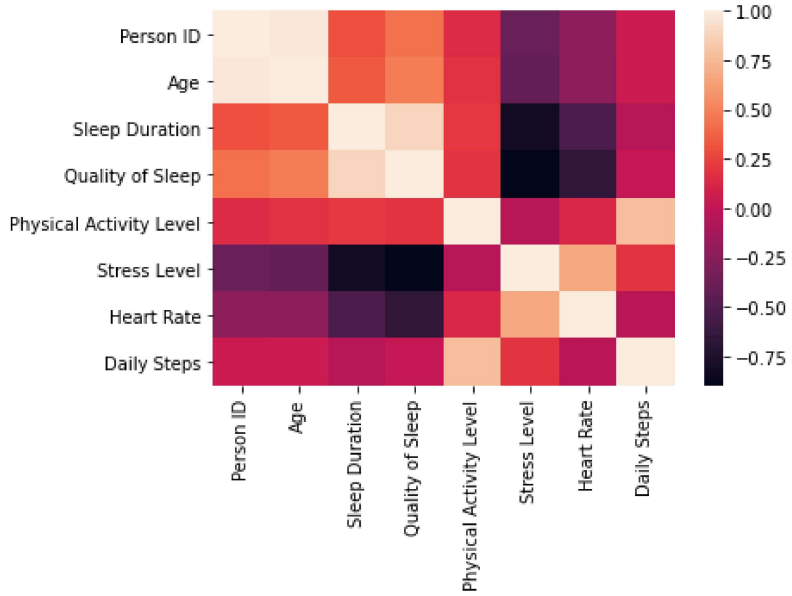


In [18]:
```
df1=df[['Person ID' , 'Age',  'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level', 'Heart Rate', 'Daily Steps
```

Plot Using Heat Map

In [19]:
```
sns.heatmap(df1.corr())
```

Out[19]: <AxesSubplot:>



To Train The Model-Model Building

we are going to train Linera Regression Model:We need to split out data into two variables x and y where x is

In [20]:
```
x=df1[['Person ID' , 'Age',  'Sleep Duration',
       'Quality of Sleep', 'Physical Activity Level', 'Stress Level', 'Heart Rate'
       ]]
y=df1["Daily Steps"]
```

To Split my dataset into training and test data

In [21]:
```
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 [22]:
```
from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

Out[22]:
```
LinearRegression()
```

In [23]:
```
lr.intercept_
```

Out[23]:
```
12854.703857844714
```

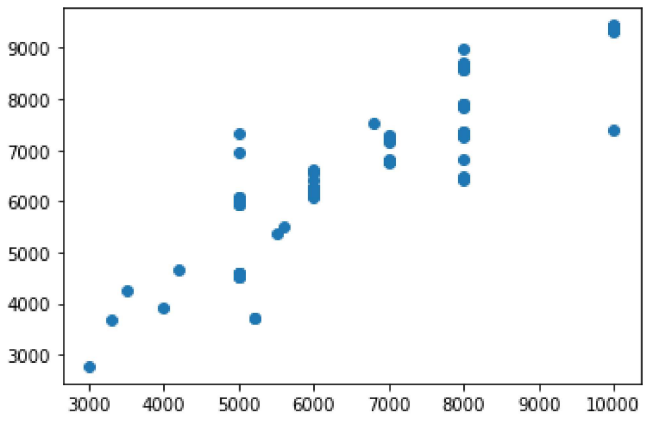
In [24]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[24]:

	Co-efficient
Person ID	-3.029367
Age	36.306371
Sleep Duration	-470.142893
Quality of Sleep	408.372612
Physical Activity Level	66.197846
Stress Level	575.512361
Heart Rate	-194.492959

In [25]:
```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[25]:
```
<matplotlib.collections.PathCollection at 0x29aac0be220>
```



Accuracy

In [26]:
```
lr.score(x_test,y_test)
```

Out[26]: 0.8131063167014344

In [27]:
```
lr.score(x_train,y_train)
```

Out[27]: 0.7991990965726312

In [28]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [29]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[29]: Ridge(alpha=10)

In [30]:
```
rr.score(x_test,y_test)
```

Out[30]: 0.8146601630043693

In [31]:
```
la =Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[31]: Lasso(alpha=10)

In [32]:
```
la.score(x_test,y_test)
```

Out[32]: 0.8139528507351983

In [ ]:
```

```