In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df=pd.read_csv(r'C:\Users\rohit\Desktop\Data sets\wearable_tech_sleep_quality_1.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Heart_Rate_Variability	Body_Temperature	Movement_During_Sleep	Sleep_Duration_Hours	Sleep_Quality_Score	Caffeine_Intake_mg	Stress_Level	Bedtime_Consistency	Light_Exposure_hours
0	79.934283	37.199678	1.324822	4.638289	1.0	107.624032	2.771837	0.657037	7.933949
1	67.234714	36.962317	1.855481	6.209422	1.0	104.658589	3.738138	0.144464	6.992699
2	82.953771	36.529815	1.207580	6.879592	10.0	0.000000	3.115880	0.642949	7.655250
3	100.460597	36.176532	1.692038	10.331531	1.0	116.990981	3.904008	0.453255	9.429463
4	65.316933	36.849112	0.106385	8.334830	1.0	223.282908	4.571699	0.641492	10.555713

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Heart_Rate_Variability  1000 non-null   float64
 1   Body_Temperature        1000 non-null   float64
 2   Movement_During_Sleep   1000 non-null   float64
 3   Sleep_Duration_Hours    1000 non-null   float64
 4   Sleep_Quality_Score     1000 non-null   float64
 5   Caffeine_Intake_mg      1000 non-null   float64
 6   Stress_Level            1000 non-null   float64
 7   Bedtime_Consistency     1000 non-null   float64
 8   Light_Exposure_hours    1000 non-null   float64
dtypes: float64(9)
memory usage: 70.4 KB
```

DESCRIPTIVE STATAISTICS

mean,median,mode,stsndsrd deviation,variance,skewness,kurtosis

In [5]:
```
print('\nMean Heartrate variability :',df['Heart_Rate_Variability'].mean())
print('\nMedian Body temperature :',df['Body_Temperature'].median())
print('\nMode of sleep quality score :',df['Sleep_Quality_Score'].mode()[0])
print('\nstandard deviation of caffiene intake in mg :',df['Caffeine_Intake_mg'].std())
print('\nvariance of caffienne intake in mg :',df['Caffeine_Intake_mg'].var())
print('\nrange of caffiene intake in mg :',df['Caffeine_Intake_mg'].min(),'mg  to ',df['Caffeine_Intake_mg'].max(),'mg')
print('\nskewness of sleep duration hours :',df['Sleep_Duration_Hours'].skew())
print('\nkurtosis of light exposure hours :',df['Light_Exposure_hours'].kurt())
```

```
Mean Heartrate variability : 70.38664111644651

Median Body temperature : 36.53153856616453

Mode of sleep quality score : 1.0

standard deviation of caffiene intake in mg : 94.03175964072372

variance of caffienne intake in mg : 8841.971821130837

range of caffiene intake in mg : 0.0 mg  to  400.0 mg

skewness of sleep duration hours : -0.0021214179470118435

kurtosis of light exposure hours : 0.16353162415826716
```

INTERPRETAION OF SKEWNESS AND KURTOSIS:
skewness value is almost equal to zero
=> data points follow normal distribution in sleep duration hours
kurtosis value is slightly positive
=> data points has heavy tailed normal distribution i.e has more extreme values in light exposure hours

INFERENTIAL STATISTICS

In [6]:
```
from scipy import stats

#LET US CONSIDER HEART RATE VARIABILITY
heart_rates = df['Heart_Rate_Variability']

#WE SAW THAT MEAN HEART RATE VARIABILITY IS 70.38664111644651
#HYPOTHETICAL MEAN HEART RATE VARIABILITY
heart_rate_mean_assumed = 71.0008056

#LET US USE STUDENTS DRISTRIBUTION FOR STSTISTIC TEST
#LET US PERFORM ONE SAMPLE T-TEST
t_stat , p_value = stats.ttest_1samp(heart_rates , heart_rate_mean_assumed)

print('T-Statistic :',t_stat)
print('\nP-Value :',p_value)
```

```
T-Statistic : -0.9916906732648079

P-Value : 0.32158857439873356
```

INTERPRETATION
t-statistic represents that sample mean is 0.99 standard errors below the population mean
p-value is larger than 0.05 so,we fail to reject the null hypothesis:heart_rate_mean_assumed = 71.0008056

CONFIDENCE INTERVAL

In [7]:
```
from scipy import stats

# sample mean and standard error for heart rate variability
sample_mean = np.mean(df['Heart_Rate_Variability'])
standard_error = stats.sem(df['Heart_Rate_Variability'])

#computing 95% confidence interval for heart rate variability
confidence_interval = stats.norm.interval(0.95 , loc=sample_mean , scale=standard_error)
print('95% confidence interval for heart rate variability :',confidence_interval)
```

```
95% confidence interval for heart rate variability : (69.17281476837209, 71.60046746452093)
```

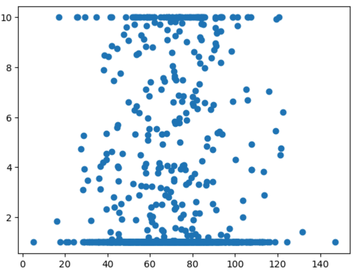
REGEESSION ANALYSIS(using ststsmodels)

In [8]:
```
import statsmodels.api as sm

#this is independent variable add constant is to add intercept to the line of regression
x = sm.add_constant(df['Heart_Rate_Variability'])

#this is dependent variable
y = df['Sleep_Quality_Score']

#fit linear regression model(ordinary least squares[OLS])
model = sm.OLS(y , x).fit()

print(model.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:     Sleep_Quality_Score   R-squared:                       0.001
Model:                             OLS   Adj. R-squared:                 -0.000
Method:                  Least Squares   F-statistic:                    0.7233
Date:                 Sun, 08 Sep 2024   Prob (F-statistic):              0.395
Time:                         19:33:05   Log-Likelihood:                -2509.8
No. Observations:                 1000   AIC:                             5024.
Df Residuals:                      998   BIC:                             5033.
Df Model:                            1                                         
Covariance Type:             nonrobust                                         
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                      2.3048      0.352      6.553      0.000       1.615       2.995
Heart_Rate_Variability     0.0041      0.005      0.850      0.395      -0.005       0.014
==============================================================================
Omnibus:                      268.675   Durbin-Watson:                   2.103
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              519.041
Skew:                           1.664   Prob(JB):                    1.96e-113
Kurtosis:                       4.177   Cond. No.                          273.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [9]:
```
import matplotlib.pyplot as plt
```

In [10]:
```
plt.scatter(df['Heart_Rate_Variability'] , y)
plt.show()
```



CONCLUSION

we have **R-squared = 0.001**
It means that only 0.1% of the variance in sleep quality score is defined by heart variabilty rate

-> we can see the meaninig of R-squared = 0.001 in the above plot
-> there isn't any correlation between heart rate varaibility and sleep quality score

SUGGESTION

we can perform this regression analysis with other features of the data also
->body temperature
->movement during sleep
->sleep duration hours
->caffiene intake mg
->stress level
->bedtime consistency
->light exposure hours

check the R-squared , F-statistic and interpret their values
also plot scatter plots to observe any observable relation between **pedictor variable** and **target variable** e_hours

In [ ]: