statistical-analysis-using-python

September 8, 2024

Loading a dataset into python using pandas

[10]:
```
import pandas as pd
data=pd.read_csv('Sleep_health_and_lifestyle_dataset[1].csv')
df=pd.DataFrame(data)
print(df.head())
```

```
   Person ID Gender  Age            Occupation  Sleep Duration  \
0          1   Male   27     Software Engineer             6.1
1          2   Male   28                Doctor             6.2
2          3   Male   28                Doctor             6.2
3          4   Male   28  Sales Representative             5.9
4          5   Male   28  Sales Representative             5.9

   Quality of Sleep  Physical Activity Level  Stress Level BMI Category  \
0                 6                       42             6   Overweight
1                 6                       60             8       Normal
2                 6                       60             8       Normal
3                 4                       30             8        Obese
4                 4                       30             8        Obese

  Blood Pressure  Heart Rate  Daily Steps Sleep Disorder
0         126/83          77         4200            NaN
1         125/80          75        10000            NaN
2         125/80          75        10000            NaN
3         140/90          85         3000    Sleep Apnea
4         140/90          85         3000    Sleep Apnea
```

Performing Descriptive Statistics

[9]:
```
df.describe()
```

[9]:

	Person ID	Age	Sleep Duration	Quality of Sleep
count	374.000000	374.000000	374.000000	374.000000
mean	187.500000	42.184492	7.132086	7.312834
std	108.108742	8.673133	0.795657	1.196956
min	1.000000	27.000000	5.800000	4.000000
25%	94.250000	35.250000	6.400000	6.000000
50%	187.500000	43.000000	7.200000	7.000000

```
75%     280.750000   50.000000        7.800000          8.000000
max     374.000000   59.000000        8.500000          9.000000

       Physical Activity Level  Stress Level  Heart Rate   Daily Steps
count               374.000000    374.000000  374.000000    374.000000
mean                 59.171123      5.385027   70.165775   6816.844920
std                  20.830804      1.774526    4.135676   1617.915679
min                  30.000000      3.000000   65.000000   3000.000000
25%                  45.000000      4.000000   68.000000   5600.000000
50%                  60.000000      5.000000   70.000000   7000.000000
75%                  75.000000      7.000000   72.000000   8000.000000
max                  90.000000      8.000000   86.000000  10000.000000
```

[33]:
```
#CALCULATING MEAN
data2=pd.DataFrame(data,columns=['Age','Sleep Duration','Quality of␣
 ↪Sleep','Stress Level','Heart Rate'])
print("Mean is:",data2.mean())
```

```
Mean is: Age                 42.184492
Sleep Duration       7.132086
Quality of Sleep     7.312834
Stress Level         5.385027
Heart Rate          70.165775
dtype: float64
```

[34]:
```
#MEDIAN
print("Median is:",data2.median())
```

```
Median is: Age                 43.0
Sleep Duration       7.2
Quality of Sleep     7.0
Stress Level         5.0
Heart Rate          70.0
dtype: float64
```

[39]:
```
#MODE
print("Mode is:",data.mode().iloc[0])
```

```
Mode is: Person ID                         1
Gender                              Male
Age                                 43.0
Occupation                         Nurse
Sleep Duration                       7.2
Quality of Sleep                     8.0
Physical Activity Level             60.0
Stress Level                         3.0
BMI Category                      Normal
Blood Pressure                   130/85
```

```
Heart Rate                         68.0
Daily Steps                      8000.0
Sleep Disorder              Sleep Apnea
Name: 0, dtype: object
```

[40]:
```
print("Standard Deviation is:",data2.std())
```

```
Standard Deviation is: Age                 8.673133
Sleep Duration      0.795657
Quality of Sleep    1.196956
Stress Level        1.774526
Heart Rate          4.135676
dtype: float64
```

[ ]:
```
2.Performing Inferential Statistics
```

HYPOTHESIS TESTING

Let, Null Hypothesis (H0): The average heart rate is equal to 71. H0: =71 Alternative Hypothesis (H1): The average blood pressure is not equal to 120 mmHg. H1: 120

[5]:
```
from scipy import stats
import pandas as pd
data=pd.read_csv('Sleep_health_and_lifestyle_dataset[1].csv')
df=pd.DataFrame(data)
Heart_rate = df['Heart Rate']

# Hypothetical population mean for Heart rate
population_mean = 71

# Perform one-sample t-test
t_stat, p_value = stats.ttest_1samp(Heart_rate ,population_mean)

print(f"T-Statistic: {t_stat}")
print(f"P-Value: {p_value}")
```

```
T-Statistic: -3.900967518092964
P-Value: 0.00011363938512880992
```

[ ]:
```
CONFIDENCE INTERVALS
```

[6]:
```
import numpy as np
from scipy import stats

# Sample mean and standard error for heart rate
sample_mean = np.mean(Heart_rate )
standard_error = stats.sem(Heart_rate )
```

```
# Compute 95% confidence interval for heart rate
confidence_interval = stats.norm.interval(0.95, loc=sample_mean,␣
 ↪scale=standard_error)

print(f"95% Confidence Interval for Heart rate: {confidence_interval}")
```

```
95% Confidence Interval for Heart rate: (69.74663574883257, 70.58491505330647)
```

REGRESSION ANALYSIS

[11]:
```
import statsmodels.api as sm
import pandas as pd
data=pd.read_csv('Sleep_health_and_lifestyle_dataset[1].csv')
df=pd.DataFrame(data)
# Define independent variable (add constant for intercept)
X = sm.add_constant(df['Heart Rate'])

# Define dependent variable
y = df['Stress Level']

# Fit linear regression model
model = sm.OLS(y, X).fit()

# Print model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Stress Level   R-squared:                       0.449
Model:                            OLS   Adj. R-squared:                  0.447
Method:                 Least Squares   F-statistic:                     303.1
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           4.49e-50
Time:                        20:40:00   Log-Likelihood:                -633.25
No. Observations:                 374   AIC:                             1270.
Df Residuals:                     372   BIC:                             1278.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -14.7872      1.161    -12.739      0.000     -17.070     -12.505
Heart Rate     0.2875      0.017     17.409      0.000       0.255       0.320
==============================================================================
Omnibus:                       77.725   Durbin-Watson:                   0.906
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              251.816
Skew:                          -0.915   Prob(JB):                     2.08e-55
Kurtosis:                       6.579   Cond. No.                     1.20e+03
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.2e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

VISUALIZATION

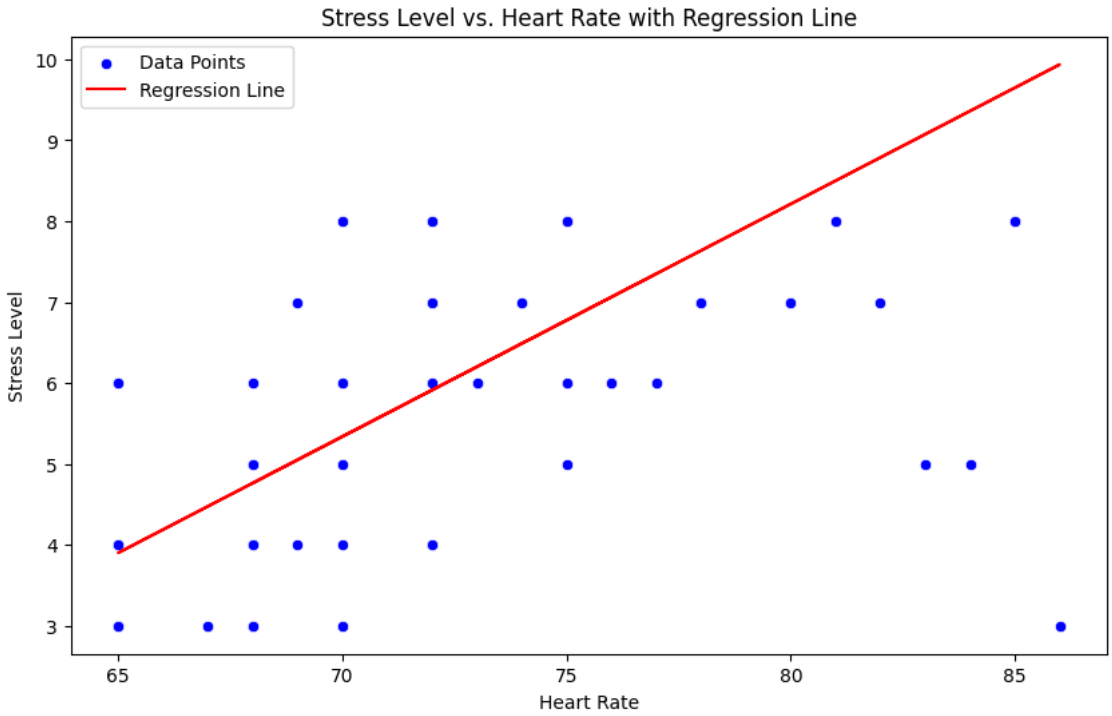
[20]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Assuming you have the data in variables `heart_rate` and `stress_level`
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['Heart Rate'], y=df['Stress Level'], color='blue',␣
 ↪label='Data Points')

# Add the regression line
slope = 0.2875
intercept = -14.7872
plt.plot(df['Heart Rate'], slope * df['Heart Rate'] + intercept, color='red',␣
 ↪label='Regression Line')

plt.xlabel('Heart Rate')
plt.ylabel('Stress Level')
plt.title('Stress Level vs. Heart Rate with Regression Line')
plt.legend()
```

[20]:
```
<matplotlib.legend.Legend at 0x21430f692b0>
```


Stress Level vs. Heart Rate with Regression Line
Data Points
Regression Line
Stress Level
Heart Rate
10
9
8
7
6
5
4
3
65
70
75
80
85