statistical-analysis

September 8, 2024

[4]:

```
import pandas as pd

# Load the dataset
df = pd.read_csv("C:/Users/user/OneDrive/Documents/sleep_health_dataset.csv")

# Show the first few rows to confirm it loaded correctly
df.head()
```

[4]:

```
   Person ID  Age  Sleep Duration  Quality of Sleep  Physical Activity Level  \
0          1   27             6.1                 6                       42
1          2   28             6.2                 6                       60
2          3   28             6.2                 6                       60
3          4   28             5.9                 4                       30
4          5   28             5.9                 4                       30

   Stress Level  High BP  Low BP  Daily Steps
0             6      126      83         4200
1             8      125      80        10000
2             8      125      80        10000
3             8      140      90         3000
4             8      140      90         3000
```

[5]:

```
# Calculate descriptive statistics for numerical columns
relevant_features = ['Age', 'Sleep Duration', 'Quality of Sleep', 'Physical␣
 ↪Activity Level',
                     'Stress Level', 'High BP', 'Low BP', 'Daily Steps']

# Mean
mean_values = df[relevant_features].mean()

# Median
median_values = df[relevant_features].median()

# Mode
mode_values = df[relevant_features].mode().iloc[0]  # mode() returns a␣
 ↪dataframe, selecting the first mode
```

```python
# Standard Deviation
std_values = df[relevant_features].std()

# Variance
variance_values = df[relevant_features].var()

# Print the results
print("Mean Values:\n", mean_values)
print("\nMedian Values:\n", median_values)
print("\nMode Values:\n", mode_values)
print("\nStandard Deviation:\n", std_values)
print("\nVariance:\n", variance_values)
```

```
Mean Values:
 Age                          42.184492
Sleep Duration                7.132086
Quality of Sleep              7.312834
Physical Activity Level      59.171123
Stress Level                  5.385027
High BP                     128.553476
Low BP                       84.649733
Daily Steps                6816.844920
dtype: float64

Median Values:
 Age                          43.0
Sleep Duration                7.2
Quality of Sleep              7.0
Physical Activity Level      60.0
Stress Level                  5.0
High BP                     130.0
Low BP                       85.0
Daily Steps                7000.0
dtype: float64

Mode Values:
 Age                          43.0
Sleep Duration                7.2
Quality of Sleep              8.0
Physical Activity Level      60.0
Stress Level                  3.0
High BP                     130.0
Low BP                       80.0
Daily Steps                8000.0
Name: 0, dtype: float64

Standard Deviation:
 Age                           8.673133
Sleep Duration                 0.795657
Quality of Sleep               1.196956
Physical Activity Level       20.830804
Stress Level                   1.774526
High BP                        7.748118
Low BP                         6.161611
Daily Steps                 1617.915679
dtype: float64

Variance:
 Age                        7.522324e+01
Sleep Duration             6.330696e-01
Quality of Sleep           1.432703e+00
Physical Activity Level    4.339224e+02
Stress Level               3.148944e+00
High BP                    6.003333e+01
Low BP                     3.796546e+01
Daily Steps                2.617651e+06
dtype: float64
```

[6]:

```python
from scipy import stats

# Null hypothesis: mean sleep duration = 7 hours
# Alternative hypothesis: mean sleep duration != 7 hours
sleep_duration = df['Sleep Duration']
t_stat, p_value = stats.ttest_1samp(sleep_duration, 7)

print(f"T-statistic: {t_stat}, P-value: {p_value}")

# Interpret the result
alpha = 0.05  # significance level
if p_value < alpha:
    print("Reject the null hypothesis - The mean sleep duration is␣
 ↪significantly different from 7 hours.")
else:
    print("Fail to reject the null hypothesis - The mean sleep duration is not␣
 ↪significantly different from 7 hours.")
```

```
T-statistic: 3.2104462758942, P-value: 0.0014402421900475528
Reject the null hypothesis - The mean sleep duration is significantly different
from 7 hours.
```

[7]:

```python
import numpy as np

# Function to compute confidence interval
def confidence_interval(data, confidence=0.95):
    mean = np.mean(data)
    std_err = stats.sem(data)  # standard error
    margin_of_error = std_err * stats.t.ppf((1 + confidence) / 2, len(data) - 1)
    return mean - margin_of_error, mean + margin_of_error

# Compute 95% confidence interval for daily steps
ci_lower, ci_upper = confidence_interval(df['Daily Steps'])

print(f"95% Confidence Interval for Daily Steps: ({ci_lower}, {ci_upper})")
```

```
95% Confidence Interval for Daily Steps: (6652.339714058175, 6981.350125514018)
```

[8]:

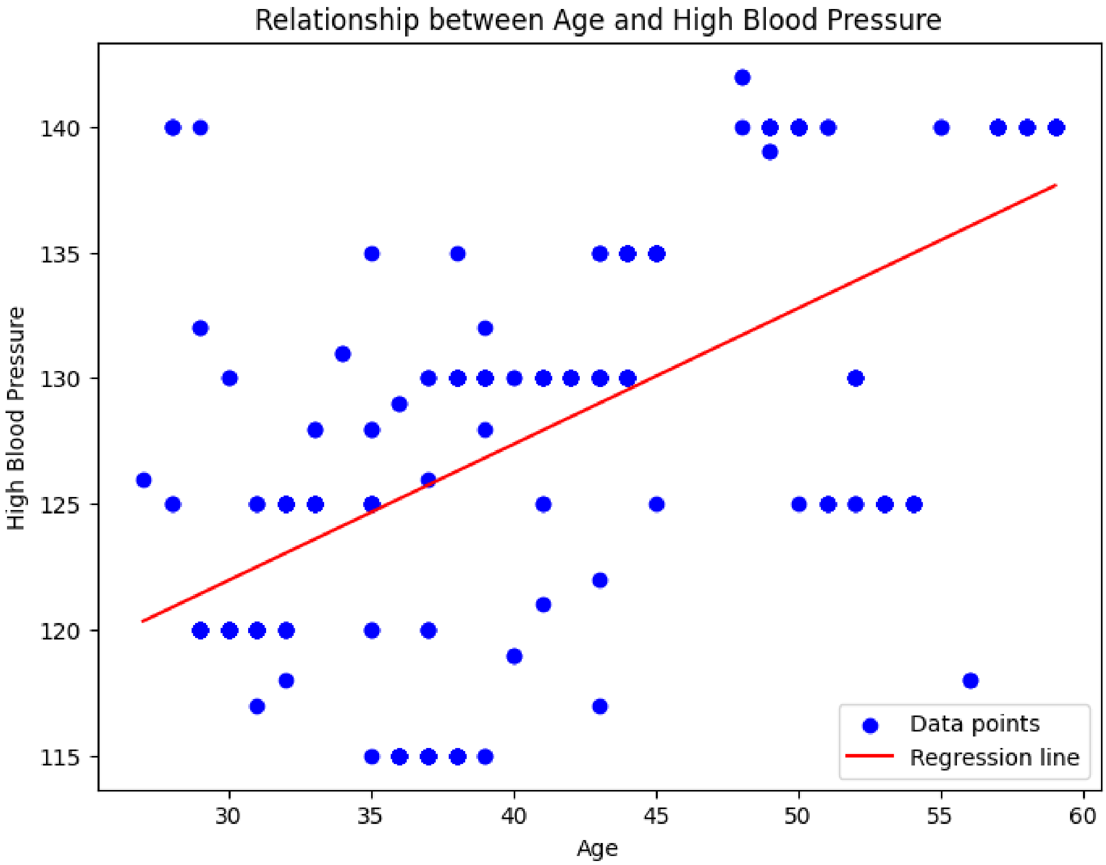
```
import statsmodels.api as sm

# Define independent and dependent variables
X = df['Age']  # Independent variable
y = df['High BP']  # Dependent variable

# Add a constant to the independent variable (intercept)
X = sm.add_constant(X)

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

# Output the summary of the regression
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                High BP   R-squared:                       0.367
Model:                            OLS   Adj. R-squared:                  0.365
Method:                 Least Squares   F-statistic:                     215.8
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           7.62e-39
Time:                        21:32:55   Log-Likelihood:                -1210.4
No. Observations:                 374   AIC:                             2425.
Df Residuals:                     372   BIC:                             2433.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        105.7207      1.587     66.622      0.000     102.600     108.841
Age            0.5413      0.037     14.689      0.000       0.469       0.614
==============================================================================
Omnibus:                        7.988   Durbin-Watson:                   0.871
Prob(Omnibus):                  0.018   Jarque-Bera (JB):                7.892
Skew:                          -0.347   Prob(JB):                       0.0193
Kurtosis:                       3.153   Cond. No.                         214.
```

```
==============================================================================

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

[10]:

```python
import matplotlib.pyplot as plt
import numpy as np

# Scatter plot using matplotlib
plt.figure(figsize=(8,6))
plt.scatter(df['Age'], df['High BP'], color='blue', label='Data points')

# Fit the regression line
slope, intercept = np.polyfit(df['Age'], df['High BP'], 1)  # 1 represents a linear fit

# Create the regression line
reg_line = slope * df['Age'] + intercept

# Plot the regression line
plt.plot(df['Age'], reg_line, color='red', label='Regression line')

# Add title and labels
plt.title('Relationship between Age and High Blood Pressure')
plt.xlabel('Age')
plt.ylabel('High Blood Pressure')

# Add a legend
plt.legend()

# Show the plot
plt.show()
```


Relationship between Age and High Blood Pressure
High Blood Pressure
115
120
125
130
135
140
Age
30
35
40
45
50
55
60
● Data points
Regression line