Texas A&M International University
Department of Mathematics and Physics

Predicting Sleep Efficiency using Multiple Linear Regression: A Model Based on Lifestyle Factors

Daniela Esparza

Dr. Runchang Lin

A report submitted in partial fulfilment of the requirements of
Texas A&M International University for the degree of
Master of Science in *Mathematics*

December 9, 2024

Abstract

This project investigates the relationship between lifestyle factors and sleep efficiency to develop a predictive model for estimating sleep efficiency based on individual lifestyle choices. The research utilizes a regression model derived from mathematical modeling applications and compares it with another regression model to determine the best model for prediction. The project uses Python to analyze sleeping patterns in multiple test subjects and aims to build a model that predicts an estimate of a person's sleep efficiency based on their lifestyle factors.

Key Words: *Sleep efficiency, linear regression, correlation, mean squared error, r-squared value*

1 Introduction

The efficiency of our sleep can be significantly influenced by our lifestyle choices, encompassing factors such as physical activity, smoking habits, sleep duration, and deep sleep percentage (Equilibriumm, 2023). Inefficient sleep can negatively impact both cognitive performance and overall health (Ikeda et al., 2022). Therefore, developing a predictive model that estimates sleep efficiency based on individual lifestyle choices is crucial for preventing potential health issues. Regression analysis, a statistical technique, can be employed to create such a model, applying principles learned in mathematical modeling. Subsequently, a second regression model can be developed and compared to the first, ultimately selecting the superior model for predicting sleep efficiency.

2 Methods

The dataset utilized in this project, originating from Kaggle, offers a rich collection of variables related to sleep patterns and lifestyle factors. It encompasses a total of 15 variables that provide a comprehensive view of individual sleep health and potential influencing factors. These variables include:

- Subject_ID: A unique identifier for each participant in the study.
- Age: The age of each participant.
- Gender : The gender of each participant.
- Bedtime: The recorded bedtime of each participant.
- Wakeup Time: The recorded wake-up time of each participant.
- Sleep Duration: The total duration of sleep for each participant.
- Sleep Efficiency: The percentage of time in bed that each participant spends asleep.
- REM Sleep percentage: The percentage of total sleep time spent in the Rapid Eye Movement (REM) stage.

- Deep Sleep percentage: The percentage of total sleep time spent in the Deep sleep stage.
- Light Sleep percentage: The percentage of total sleep time spent in the Light sleep stage.
- Awakenings: The number of times each participant woke up during the night.
- Alcohol Consumption: The amount of caffeine consumed by each participant within 24 hours of bedtime.
- Smoking Status: Whether or not each participant smokes.
- Exercise Frequency: The frequency with which participant exercises.

Data was collected from a total of 452 individuals, each uniquely identified by the *Subject_ID* variable. This dataset offers a wealth of information for exploring the relationships between sleep patterns, individual characteristics (age, gender), lifestyle choices (caffeine and alcohol consumption, smoking, exercise), and sleep quality metrics (sleep duration, sleep efficiency, sleep stages, awakenings).

In this analysis, we aim to predict sleep efficiency using multiple independent variables. We'll employ two regression models: Multiple Linear Regression (MLR) and Multivariate Polynomial Regression (MPR). MLR predicts the dependent variable using a linear combination of independent variables, while MPR captures non-linear relationships between variables by including polynomial terms. The equations for both models are provided.

Multiple Linear Regression:

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_n x_n \tag{1}$$

where:

- $b_0, b_1, ..., b_n$ are the coefficients of the independent variables,
- $x_1, x_2, ..., x_n$ are the independent variables,
- y is the dependent variable (Saturn Cloud, 2023).

Multivariate Polynomial Regression of degree 2:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \beta_{12} x_n + ... + \beta_{nm} x_{n+1} + \beta_{11} x^2 + \beta_{22} x^2 + ... + \beta_{nm} x^2 \tag{2}$$

where:

- β_0 is the intercept,
- $\beta_1, \beta_2, ..., \beta_{nm}$ are the coefficients,
- $x_1, x_2, ..., x_n$ are the independent variables,
- y is the dependent variable (Saturn Cloud, 2024).

To evaluate and compare the performance of these models, the Mean Squared Error (MSE) and R-squared (R2) will be used. MSE measures the average squared difference between predicted and actual values, indicating the model's accuracy. R2 quantifies the proportion of variance in the dependent variable explained by the model, reflecting its goodness-of-fit. By examining these metrics, we can determine which model better predicts sleep efficiency based on the given data.

3 Model Simulation

3.1 Importing necessary libraries

To perform the regression models, we'll utilize Python within Google Colaboratory, a user-friendly Jupyter Notebook environment well-suited for machine learning tasks like linear regressions. The initial step involves importing the essential libraries: numpy, scipy, pandas, and matplotlib.pyplot for general assistance, along with LinearRegression and PolynomialFeatures from sklearn.linear_model and sklearn.preprocessing, respectively, for our regression model.

3.2 Uploading the dataset

Following the library imports, we'll upload the dataset (sourced from Kaggle) and assign it to the variable **sleep_better**. After listing the dataset's variables for clarity, *Sleep Efficiency* will be designated as the dependent variable and *Subject_ID* as the index variable, which aids in identifying individual participants in the study. While the index variable will be temporarily removed from the dataset, it can be reintroduced later. The remaining variables will serve as our independent variables.

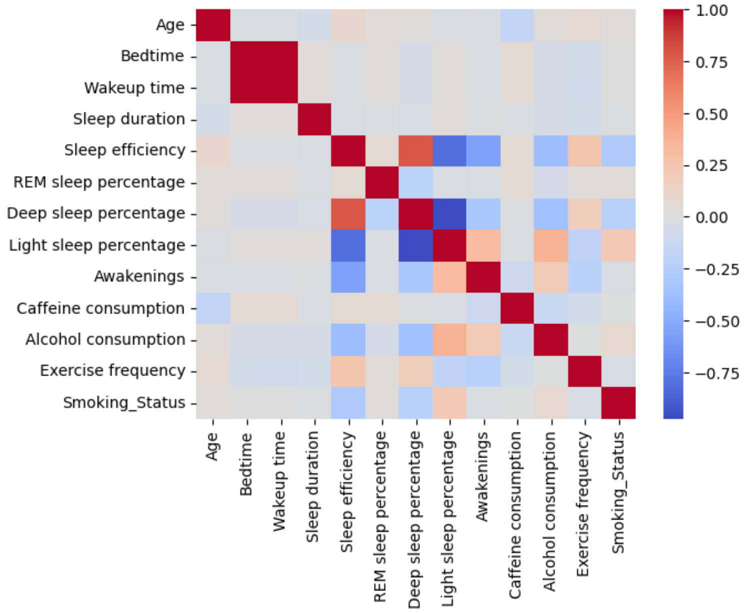
3.3 Data Cleaning

The data cleaning process focused on three variables: *Wakeup Time*, *Bedtime*, and *Smoking Status*. Both *Wakeup Time* and *Bedtime* were converted from strings to datetime objects with the format *"YYYY-mm-dd H:M:S"*. The updated data was saved in the **sleep_better2** dataframe. The *Smoking Status* variable was converted from *"Yes/No"* to binary *"1/0"* and concatenated with **sleep_better2**, resulting in the **sleep_update** variable. Duplicate *Smoking Status* columns and unnecessary columns (*Age* and the index) were dropped from **sleep_update**. Finally, a new *Smoking_Status* column was created in **sleep_update**, allowing the program to read the dataset and run successfully.

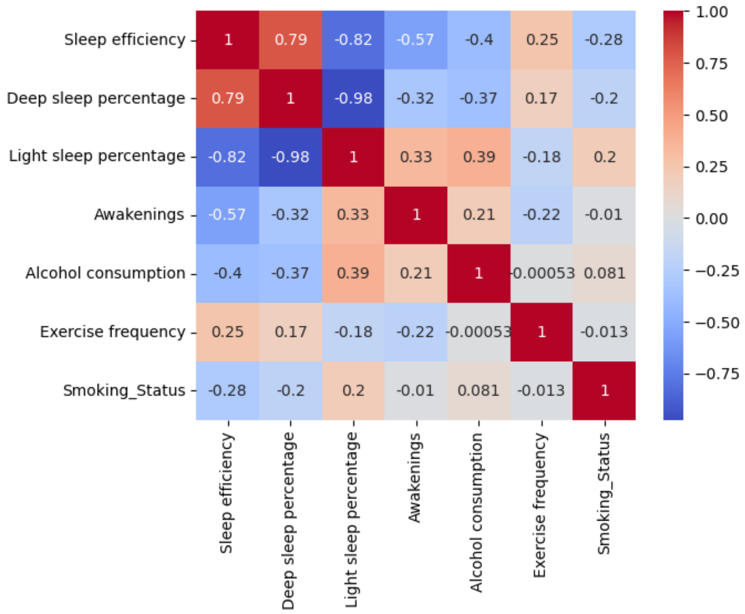
3.4 Correlation

The next step after successfully running the dataset is to determine the correlation between the independent and dependent variables. This correlation, denoted by ρ (rho), is calculated from the **sleep_update** dataframe and ranges from -1 to 1. A value of -1 indicates a perfect negative correlation, +1 signifies a strong positive association, and 0 implies no association. The closer ρ is to zero, the weaker the association. Each independent variable is assessed in relation to the dependent variable, *Sleep Efficiency*. Variables with a ρ value closer to 0 will be excluded from the dataframe. To visualize these correlations, heatmaps are generated. The initial heatmap displays all independent variables, with lighter colors indicating variables to be removed. The second heatmap presents the updated dataframe after removing the non-correlating variables.

Correlation of all the variables:



Correlation of the updated variables:



3.5 Separating features and Target variables

The process begins with the separation of features and the target variable from the updated dataframe, **sleep_update2**. The independent variables (*Deep sleep percentage, Light sleep percentage, Awakenings, Alcohol consumption, Exercise frequency*, and *Smoking_Status*) are assigned to **X**, while the dependent variable, *Sleep Efficiency*, is assigned to **y**. The data is then split into training and testing sets, with 20% allocated to the training set and the remaining 80% to the testing set. This allows for the development of the regression model using the training data and its subsequent evaluation on the unseen testing data.

3.6 Multiple Linear Regression Model

The initial step involves splitting the data into training and testing sets. Subsequently, a linear regression model is generated using the 'fit' method to establish the optimal line representing the relationship between the data points. The trained model can then predict target values for new data by plugging in the independent variable values into the linear equation. These predictions are stored in the **y_pred_test** variable. The coefficients reveal the influence of each independent variable on the target variable, while the intercept acts as the starting point for predictions. The linear regression equation is formulated using the intercept and coefficients.

Equation:

$$y = -0.00118153x_1 - 0.00706125x_2 - 0.0318915x_3 - 0.00549086x_4 + 0.00696941x_5 \quad (3)$$

$$-0.046282x_6 + 1.0881036294222948$$

Following this, the Mean Squared Error (MSE) and R-squared values are calculated. MSE quantifies the average discrepancy between predicted and actual values, whereas R-squared indicates the proportion of the target variable's variance explained by the model. These metrics are crucial for model comparison.

3.7 Multivariate Polynomial Regression Model

Next, a Multivariate Polynomial Regression (MPR) model is constructed using scikit-learn's *PolynomialFeatures* library with a default degree of 2. The training data undergoes transformation, and the same transformation is applied to the testing data. A linear regression model is then created and fitted using the transformed training data. Predictions are generated for the MPR model, and the coefficients and intercept are extracted. The MSE and R-squared values are also computed for the MPR model.

Finally, the performance of both regression models is compared based on their MSE and R-squared values to determine the most suitable model for prediction.

4 Discussion & Conclusion

Now that we have obtained both regressions, we can compare the models by examining their Mean Squared Error (MSE) and R-squared (R2) values. A lower MSE indicates a better fit, as it represents the average squared difference between predicted and actual values. A higher R-squared value, which represents the proportion of variance in the dependent variable explained by the model, also signifies a better fit.

In the context of our analysis, the Linear Regression model exhibits a slightly lower MSE and a slightly higher R-squared value compared to the Polynomial Regression model. This suggests that the Linear Regression model provides marginally more accurate predictions and explains a slightly larger proportion of the variance in the dependent variable. Although the performance difference between the two models is not substantial, it indicates that the **Linear Regression model offers a slightly superior fit** to the data in this specific scenario.

It is important to note that the choice between a linear and a polynomial model often depends on the specific characteristics of the data and the underlying relationships between the variables. While a linear model may be sufficient in some cases, a polynomial model may be more appropriate when the relationship between the variables is non-linear. The evaluation of MSE and R-squared, along with other diagnostic tools, aids in selecting the model that best balances complexity and explanatory power for the given dataset.

5 Scripts

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, cal
l drive.mount("/content/drive", force_remount=True).
```

In [ ]:

```
# Import necessary libraries
import seaborn as sns
import numpy as np
from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import spearmanr
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder
```

In [ ]:

```
# The path is read and then assigned the variable sleep_better
sleep_better = pd.read_csv("/content/drive/MyDrive/Sleep_Efficiency.csv")
print(sleep_better)

# Print the columns for the variable sleep_better
print(sleep_better.columns)
```

```
      ID  Age  Gender             Bedtime         Wakeup time  \
0      1   65  Female 2021-03-06 01:00:00 2021-03-06 07:00:00
1      2   69    Male 2021-12-05 02:00:00 2021-12-05 09:00:00
2      3   40  Female 2021-05-25 21:30:00 2021-05-25 05:30:00
3      4   40  Female 2021-11-03 02:30:00 2021-11-03 08:30:00
4      5   57    Male 2021-03-13 01:00:00 2021-03-13 09:00:00
..   ...  ...     ...                 ...                 ...
447  448   27  Female 2021-11-13 22:00:00 2021-11-13 05:30:00
448  449   52    Male 2021-03-31 21:00:00 2021-03-31 03:00:00
449  450   40  Female 2021-09-07 23:00:00 2021-09-07 07:30:00
450  451   45    Male 2021-07-29 21:00:00 2021-07-29 04:00:00
451  452   18    Male 2021-03-17 02:30:00 2021-03-17 10:00:00

     Sleep duration  Sleep efficiency  REM sleep percentage  \
0               6.0              0.88                    18
1               7.0              0.66                    19
2               8.0              0.89                    20
3               6.0              0.51                    23
4               8.0              0.76                    27
..              ...               ...                   ...
447             7.5              0.91                    22
448             6.0              0.74                    28
449             8.5              0.55                    20
450             7.0              0.76                    18
451             7.5              0.63                    22

     Deep sleep percentage  Light sleep percentage  Awakenings  \
0                       70                      12         0.0
1                       28                      53         3.0
2                       70                      10         1.0
3                       25                      52         3.0
4                       55                      18         3.0
..                     ...                     ...         ...
447                     57                      21         0.0
448                     57                      15         4.0
449                     32                      48         1.0
450                     72                      10         3.0
451                     23                      55         1.0

     Caffeine consumption  Alcohol consumption Smoking status  \
0                     0.0                  0.0            Yes
1                     0.0                  3.0            Yes
2                     0.0                  0.0             No
3                    50.0                  5.0            Yes
4                     0.0                  3.0             No
..                    ...                  ...            ...
447                   0.0                  0.0             No
448                  25.0                  0.0             No
449                   NaN                  3.0            Yes
450                   0.0                  0.0             No
451                  50.0                  0.0             No

     Exercise frequency
0                   3.0
1                   3.0
2                   3.0
```

```
3                    1.0
4                    3.0
..                   ...
447                  5.0
448                  3.0
449                  0.0
450                  3.0
451                  1.0

[452 rows x 15 columns]
Index(['ID', 'Age', 'Gender', 'Bedtime', 'Wakeup time', 'Sleep duration',
       'Sleep efficiency', 'REM sleep percentage', 'Deep sleep percentage',
       'Light sleep percentage', 'Awakenings', 'Caffeine consumption',
       'Alcohol consumption', 'Smoking status', 'Exercise frequency'],
      dtype='object')
```

In [ ]:

```
def date_formatting(dataframe3):
# Converting 'Wakeup time' to datetime
  dataframe3['Wakeup time'] = pd.to_datetime(dataframe3['Wakeup time'], form

# Converting 'Bedtime' to datetime
  dataframe3['Bedtime'] = pd.to_datetime(dataframe3['Bedtime'], format='%Y-%
  return dataframe3

# The date_formatting fcn is called sleep_better, and the
# modified dataframe is assigned to the variable sleep_better2
sleep_better2 = date_formatting(sleep_better)
```

In [ ]:

```
smoking = sleep_better['Smoking status']
# Replacing 'Yes'/ 'No' to 1/0
smoking_st = smoking.replace({'Yes': 1, 'No': 0})
print(smoking_st)
```

```
0      1
1      1
2      0
3      1
4      0
      ..
447    0
448    0
449    1
450    0
451    0
Name: Smoking status, Length: 452, dtype: int64
```

```
<ipython-input-5-5b4255d5ba6d>:3: FutureWarning: Downcasting behavior in `re
place` is deprecated and will be removed in a future version. To retain the
old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in
to the future behavior, set `pd.set_option('future.no_silent_downcasting', T
rue)`
  smoking_st = smoking.replace({'Yes': 1, 'No': 0})
```

In [ ]:

```
import pandas as pd
#sleep_better2 = pd.concat([sleep_update, smoking_st], axis=1)
# Combines sleep_better2 and smoking_st
sleep_update = pd.concat([sleep_better2, smoking_st], axis=1)
```

```
print(sleep_update)

sleep_update.drop('Smoking status', axis=1, inplace=True)
print(sleep_update)
```

```
      ID  Age  Gender             Bedtime         Wakeup time  Sleep duratio
n  \
0      1   65  Female 2021-03-06 01:00:00 2021-03-06 07:00:00             6.
0
1      2   69    Male 2021-12-05 02:00:00 2021-12-05 09:00:00             7.
0
2      3   40  Female 2021-05-25 21:30:00 2021-05-25 05:30:00             8.
0
3      4   40  Female 2021-11-03 02:30:00 2021-11-03 08:30:00             6.
0
4      5   57    Male 2021-03-13 01:00:00 2021-03-13 09:00:00             8.
0
..   ...  ...     ...                 ...                 ...
...
447  448   27  Female 2021-11-13 22:00:00 2021-11-13 05:30:00             7.
5
448  449   52    Male 2021-03-31 21:00:00 2021-03-31 03:00:00             6.
0
449  450   40  Female 2021-09-07 23:00:00 2021-09-07 07:30:00             8.
5
450  451   45    Male 2021-07-29 21:00:00 2021-07-29 04:00:00             7.
0
451  452   18    Male 2021-03-17 02:30:00 2021-03-17 10:00:00             7.
5

     Sleep efficiency  REM sleep percentage  Deep sleep percentage  \
0                0.88                    18                     70
1                0.66                    19                     28
2                0.89                    20                     70
3                0.51                    23                     25
4                0.76                    27                     55
..                ...                   ...                    ...
447              0.91                    22                     57
448              0.74                    28                     57
449              0.55                    20                     32
450              0.76                    18                     72
451              0.63                    22                     23

     Light sleep percentage  Awakenings  Caffeine consumption  \
0                        12         0.0                   0.0
1                        53         3.0                   0.0
2                        10         1.0                   0.0
3                        52         3.0                  50.0
4                        18         3.0                   0.0
..                      ...         ...                   ...
447                      21         0.0                   0.0
448                      15         4.0                  25.0
449                      48         1.0                   NaN
450                      10         3.0                   0.0
451                      55         1.0                  50.0

     Alcohol consumption  Smoking status  Exercise frequency  Smoking status
0                    0.0             Yes                 3.0               1
1                    3.0             Yes                 3.0               1
2                    0.0              No                 3.0               0
3                    5.0             Yes                 1.0               1
```

```
4                    3.0             No                 3.0              0
..                   ...            ...                 ...            ...
447                  0.0             No                 5.0              0
448                  0.0             No                 3.0              0
449                  3.0            Yes                 0.0              1
450                  0.0             No                 3.0              0
451                  0.0             No                 1.0              0

[452 rows x 16 columns]
      ID  Age  Gender             Bedtime         Wakeup time  Sleep duratio
n  \
0      1   65  Female 2021-03-06 01:00:00 2021-03-06 07:00:00             6.
0
1      2   69    Male 2021-12-05 02:00:00 2021-12-05 09:00:00             7.
0
2      3   40  Female 2021-05-25 21:30:00 2021-05-25 05:30:00             8.
0
3      4   40  Female 2021-11-03 02:30:00 2021-11-03 08:30:00             6.
0
4      5   57    Male 2021-03-13 01:00:00 2021-03-13 09:00:00             8.
0
..   ...  ...     ...                 ...                 ...
...
447  448   27  Female 2021-11-13 22:00:00 2021-11-13 05:30:00             7.
5
448  449   52    Male 2021-03-31 21:00:00 2021-03-31 03:00:00             6.
0
449  450   40  Female 2021-09-07 23:00:00 2021-09-07 07:30:00             8.
5
450  451   45    Male 2021-07-29 21:00:00 2021-07-29 04:00:00             7.
0
451  452   18    Male 2021-03-17 02:30:00 2021-03-17 10:00:00             7.
5

     Sleep efficiency  REM sleep percentage  Deep sleep percentage  \
0                0.88                    18                     70
1                0.66                    19                     28
2                0.89                    20                     70
3                0.51                    23                     25
4                0.76                    27                     55
..                ...                   ...                    ...
447              0.91                    22                     57
448              0.74                    28                     57
449              0.55                    20                     32
450              0.76                    18                     72
451              0.63                    22                     23

     Light sleep percentage  Awakenings  Caffeine consumption  \
0                        12         0.0                   0.0
1                        53         3.0                   0.0
2                        10         1.0                   0.0
3                        52         3.0                  50.0
4                        18         3.0                   0.0
..                      ...         ...                   ...
447                      21         0.0                   0.0
448                      15         4.0                  25.0
```

```
449                        48          1.0                      NaN
450                        10          3.0                      0.0
451                        55          1.0                     50.0

     Alcohol consumption  Exercise frequency
0                    0.0                 3.0
1                    3.0                 3.0
2                    0.0                 3.0
3                    5.0                 1.0
4                    3.0                 3.0
..                   ...                 ...
447                  0.0                 5.0
448                  0.0                 3.0
449                  3.0                 0.0
450                  0.0                 3.0
451                  0.0                 1.0

[452 rows x 14 columns]
```

In [ ]:

```
# Drops the following columns from the dataset
sleep_update = sleep_better.drop(['ID', 'Gender', 'Smoking status'], axis=1)
print(sleep_update)
```

```
     Age            Bedtime         Wakeup time  Sleep duration  \
0     65 2021-03-06 01:00:00 2021-03-06 07:00:00             6.0
1     69 2021-12-05 02:00:00 2021-12-05 09:00:00             7.0
2     40 2021-05-25 21:30:00 2021-05-25 05:30:00             8.0
3     40 2021-11-03 02:30:00 2021-11-03 08:30:00             6.0
4     57 2021-03-13 01:00:00 2021-03-13 09:00:00             8.0
..   ...                 ...                 ...             ...
447   27 2021-11-13 22:00:00 2021-11-13 05:30:00             7.5
448   52 2021-03-31 21:00:00 2021-03-31 03:00:00             6.0
449   40 2021-09-07 23:00:00 2021-09-07 07:30:00             8.5
450   45 2021-07-29 21:00:00 2021-07-29 04:00:00             7.0
451   18 2021-03-17 02:30:00 2021-03-17 10:00:00             7.5

     Sleep efficiency  REM sleep percentage  Deep sleep percentage  \
0                0.88                    18                     70
1                0.66                    19                     28
2                0.89                    20                     70
3                0.51                    23                     25
4                0.76                    27                     55
..                ...                   ...                    ...
447              0.91                    22                     57
448              0.74                    28                     57
449              0.55                    20                     32
450              0.76                    18                     72
451              0.63                    22                     23

     Light sleep percentage  Awakenings  Caffeine consumption  \
0                        12         0.0                   0.0
1                        53         3.0                   0.0
2                        10         1.0                   0.0
3                        52         3.0                  50.0
4                        18         3.0                   0.0
..                      ...         ...                   ...
447                      21         0.0                   0.0
448                      15         4.0                  25.0
449                      48         1.0                   NaN
450                      10         3.0                   0.0
451                      55         1.0                  50.0

     Alcohol consumption  Exercise frequency
0                    0.0                 3.0
1                    3.0                 3.0
2                    0.0                 3.0
3                    5.0                 1.0
4                    3.0                 3.0
..                   ...                 ...
447                  0.0                 5.0
448                  0.0                 3.0
449                  3.0                 0.0
450                  0.0                 3.0
451                  0.0                 1.0

[452 rows x 12 columns]
```

In [ ]:

```
# New column is created in the sleep_update dataframe
sleep_update['Smoking_Status'] = smoking_st
```

```python
print(sleep_update)
```

```
     Age             Bedtime         Wakeup time  Sleep duration  \
0     65 2021-03-06 01:00:00 2021-03-06 07:00:00             6.0
1     69 2021-12-05 02:00:00 2021-12-05 09:00:00             7.0
2     40 2021-05-25 21:30:00 2021-05-25 05:30:00             8.0
3     40 2021-11-03 02:30:00 2021-11-03 08:30:00             6.0
4     57 2021-03-13 01:00:00 2021-03-13 09:00:00             8.0
..   ...                 ...                 ...             ...
447   27 2021-11-13 22:00:00 2021-11-13 05:30:00             7.5
448   52 2021-03-31 21:00:00 2021-03-31 03:00:00             6.0
449   40 2021-09-07 23:00:00 2021-09-07 07:30:00             8.5
450   45 2021-07-29 21:00:00 2021-07-29 04:00:00             7.0
451   18 2021-03-17 02:30:00 2021-03-17 10:00:00             7.5

     Sleep efficiency  REM sleep percentage  Deep sleep percentage  \
0                0.88                    18                     70
1                0.66                    19                     28
2                0.89                    20                     70
3                0.51                    23                     25
4                0.76                    27                     55
..                ...                   ...                    ...
447              0.91                    22                     57
448              0.74                    28                     57
449              0.55                    20                     32
450              0.76                    18                     72
451              0.63                    22                     23

     Light sleep percentage  Awakenings  Caffeine consumption  \
0                        12         0.0                   0.0
1                        53         3.0                   0.0
2                        10         1.0                   0.0
3                        52         3.0                  50.0
4                        18         3.0                   0.0
..                      ...         ...                   ...
447                      21         0.0                   0.0
448                      15         4.0                  25.0
449                      48         1.0                   NaN
450                      10         3.0                   0.0
451                      55         1.0                  50.0

     Alcohol consumption  Exercise frequency  Smoking_Status
0                    0.0                 3.0               1
1                    3.0                 3.0               1
2                    0.0                 3.0               0
3                    5.0                 1.0               1
4                    3.0                 3.0               0
..                   ...                 ...             ...
447                  0.0                 5.0               0
448                  0.0                 3.0               0
449                  3.0                 0.0               1
450                  0.0                 3.0               0
451                  0.0                 1.0               0

[452 rows x 13 columns]
```

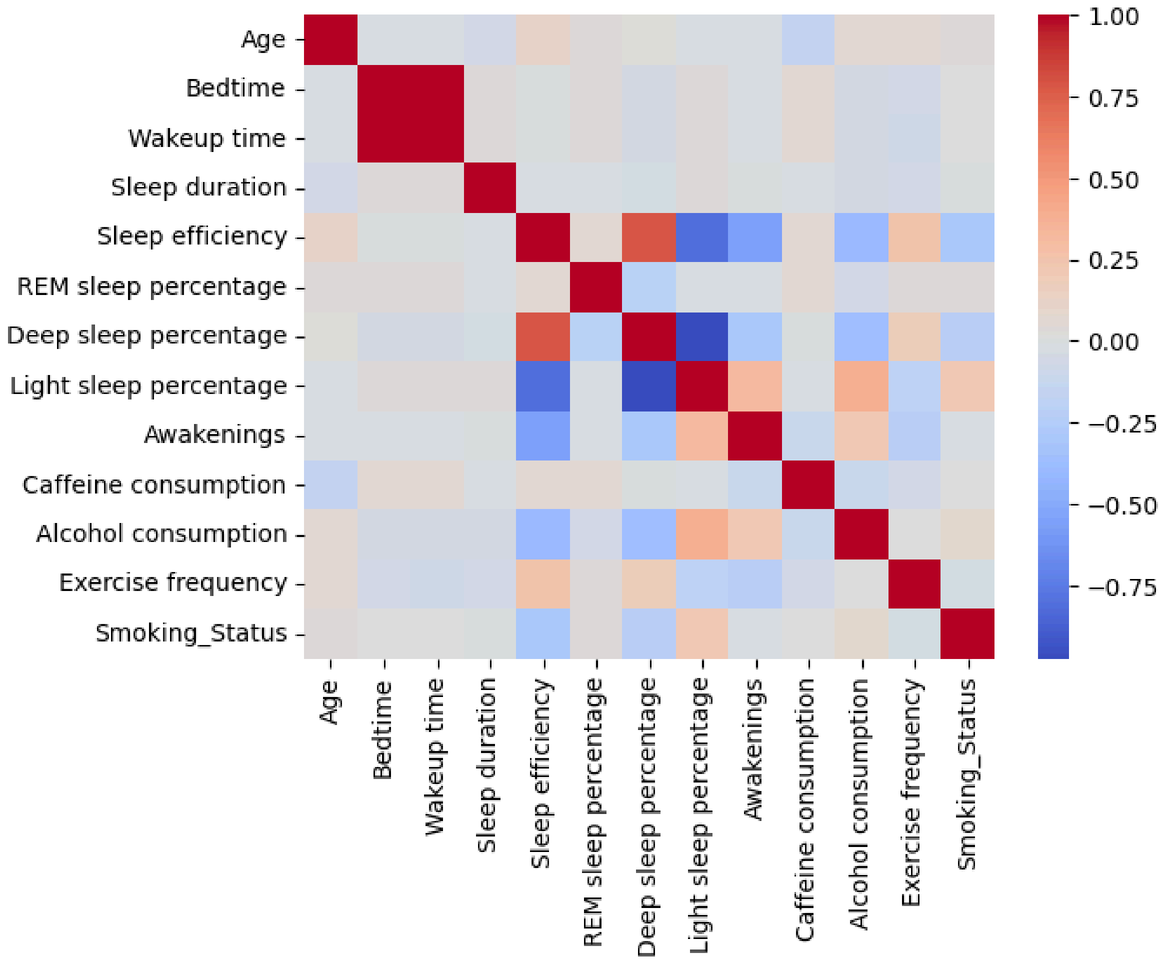
In [ ]:
```
# Obtaining the correlation for the dataframe sleep_update
sleep_update.corr()
```

Out[ ]:

	Age	Bedtime	Wakeup time	Sleep duration	Sleep efficiency	REM sl percent
Age	1.000000	-0.026877	-0.026533	-0.062462	0.098357	0.042
Bedtime	-0.026877	1.000000	0.999988	0.029673	-0.007848	0.038
Wakeup time	-0.026533	0.999988	1.000000	0.030566	-0.008488	0.038
Sleep duration	-0.062462	0.029673	0.030566	1.000000	-0.027467	-0.015
Sleep efficiency	0.098357	-0.007848	-0.008488	-0.027467	1.000000	0.062
REM sleep percentage	0.042091	0.038341	0.038283	-0.015940	0.062362	1.000
Deep sleep percentage	0.021730	-0.042009	-0.042446	-0.037304	0.787335	-0.208
Light sleep percentage	-0.031905	0.034116	0.034576	0.041804	-0.819204	-0.017
Awakenings	-0.017789	-0.015050	-0.014534	0.004939	-0.564979	-0.025
Caffeine consumption	-0.171460	0.059698	0.059541	-0.014802	0.065082	0.060
Alcohol consumption	0.047188	-0.041902	-0.041752	-0.046243	-0.389624	-0.053
Exercise frequency	0.072308	-0.071735	-0.073693	-0.068272	0.259563	0.031
Smoking_Status	0.031237	0.017287	0.018280	0.004211	-0.290026	0.032

Visualizing the Corrrelations between the original datast

In [ ]:
```
# Restate the correlation matrix for the primary dataframe, sleep_update
correlation_matrix = sleep_update.corr()
# Create a heatmap
sns.heatmap(correlation_matrix, annot=False, cmap='coolwarm')
plt.show()
```



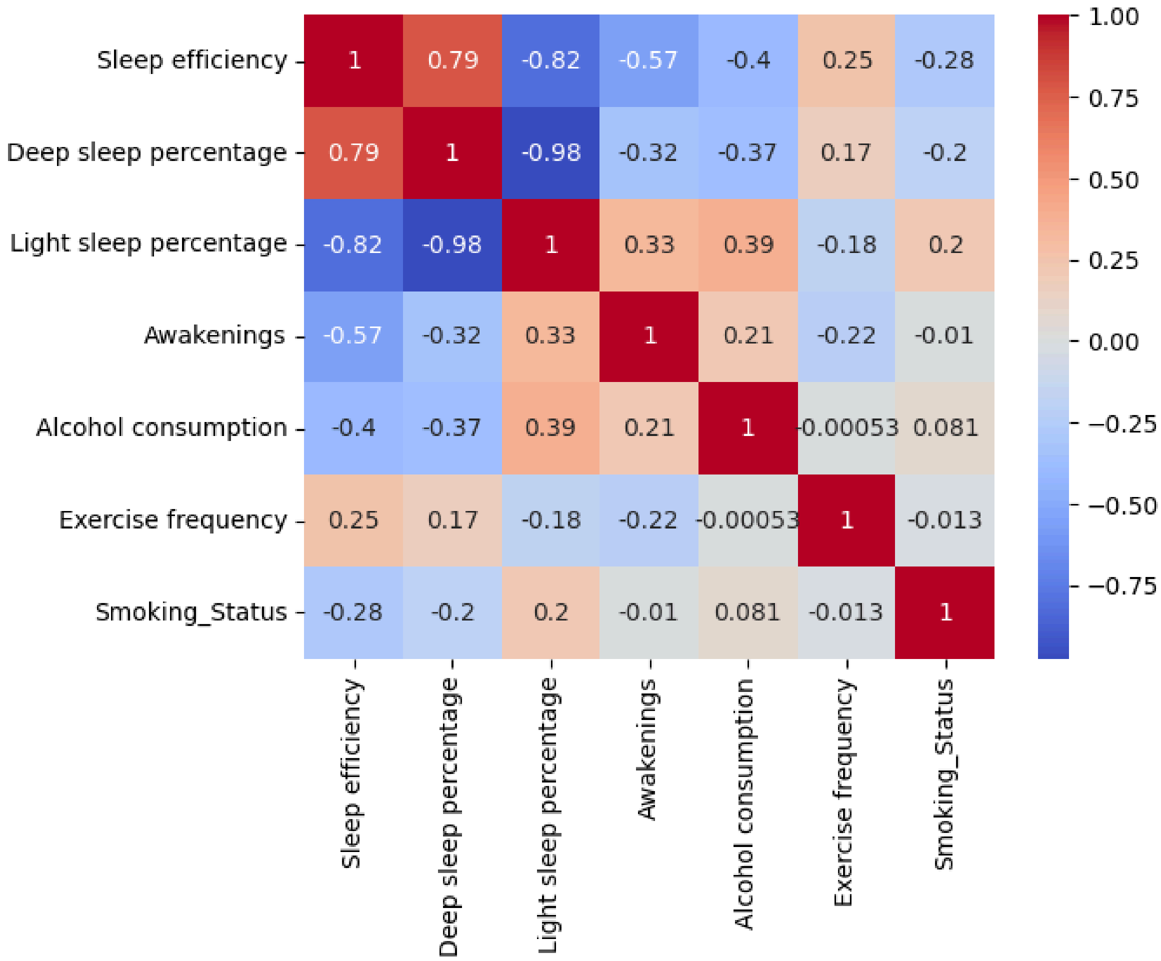
In [ ]:
```
# The following columns are dropped from the sleep_update dataframe
sleep_update2 = sleep_update.drop(['Age', 'Bedtime', 'Wakeup time', 'Sleep c
print(sleep_update2)
```

```
     Sleep efficiency  Deep sleep percentage  Light sleep percentage  \
0                0.88                     70                      12   
1                0.66                     28                      53   
2                0.89                     70                      10   
3                0.51                     25                      52   
4                0.76                     55                      18   
..                ...                    ...                     ...   
447              0.91                     57                      21   
448              0.74                     57                      15   
449              0.55                     32                      48   
450              0.76                     72                      10   
451              0.63                     23                      55   

     Awakenings  Alcohol consumption  Exercise frequency  Smoking_Status  
0           0.0                  0.0                 3.0               1  
1           3.0                  3.0                 3.0               1  
2           1.0                  0.0                 3.0               0  
3           3.0                  5.0                 1.0               1  
4           3.0                  3.0                 3.0               0  
..          ...                  ...                 ...             ...  
447         0.0                  0.0                 5.0               0  
448         4.0                  0.0                 3.0               0  
449         1.0                  3.0                 0.0               1  
450         3.0                  0.0                 3.0               0  
451         1.0                  0.0                 1.0               0  

[452 rows x 7 columns]
```

Visualizing the correlation of the updated dataset

In [ ]:

```
# Restate the correlation matrix for the updated dataframe, sleep_update2
corr_matrix = sleep_update2.corr()
# Create a Heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.show()
```



In [ ]:

```
# Drop rows with any NaN values in either X or Y
sleep_update2 = sleep_update2.dropna()

# Separate features (independent variables) and target (dependent variable)
X = sleep_update2.drop('Sleep efficiency', axis=1)
y = sleep_update2['Sleep efficiency']
print(X)
print(y)
```

```
     Deep sleep percentage  Light sleep percentage  Awakenings  \
0                       70                      12         0.0
1                       28                      53         3.0
2                       70                      10         1.0
3                       25                      52         3.0
4                       55                      18         3.0
..                     ...                     ...         ...
447                     57                      21         0.0
448                     57                      15         4.0
449                     32                      48         1.0
450                     72                      10         3.0
451                     23                      55         1.0

     Alcohol consumption  Exercise frequency  Smoking_Status
0                    0.0                 3.0               1
1                    3.0                 3.0               1
2                    0.0                 3.0               0
3                    5.0                 1.0               1
4                    3.0                 3.0               0
..                   ...                 ...             ...
447                  0.0                 5.0               0
448                  0.0                 3.0               0
449                  3.0                 0.0               1
450                  0.0                 3.0               0
451                  0.0                 1.0               0

[412 rows x 6 columns]
0      0.88
1      0.66
2      0.89
3      0.51
4      0.76
       ...
447    0.91
448    0.74
449    0.55
450    0.76
451    0.63
Name: Sleep efficiency, Length: 412, dtype: float64
```

Training and Testing sets

In [ ]:

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rar
```

Regression

After checking the correlation of the dependent variables with the independent variable, then it is time to begin the regression model.

In [ ]:
```
# Regressor model
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[ ]:

```
▾ LinearRegression
LinearRegression()
```

Making Predictions

Predict any value of y dependent on x with the trained model using ***regressor.predict***

In [ ]:
```
# Regressor model
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predict
y_pred_test = regressor.predict(X_test)
#y_pred_train = regressor.predict(X_train)

# Print the coefficients and intercept
print('Coefficients:', regressor.coef_)
print('Intercept:', regressor.intercept_)

# Evaluate the model, obtain the mse and R2
from sklearn.metrics import mean_squared_error, r2_score
mse_linear = mean_squared_error(y_test, y_pred_test)
r2_linear = r2_score(y_test, y_pred_test)
print("Mean squared error:",  mean_squared_error(y_test, y_pred_test))
print("R-squared:", r2_score(y_test, y_pred_test))
```

```
Coefficients: [-0.00118153 -0.00706125 -0.0318915  -0.00549086  0.00696941 -
0.046282  ]
Intercept: 1.0881036294222948
Mean squared error: 0.0031909500116446767
R-squared: 0.803128328172789
```

Multivariate Polynomial Regression - another model

In [ ]:
```
# Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures
# Create a PolynomialFeatures obj with degree 2
poly = PolynomialFeatures(degree = 2)
# Transform the training data
X_train_poly = poly.fit_transform(X_train)
# Transforms test data using same transformation learned from training data
X_test_poly = poly.transform(X_test)

# Create and fit LinearRegression model using the transformed training data
poly_reg = LinearRegression()
poly_reg.fit(X_train_poly, y_train)

# Predict the model
predict = poly_reg.predict(X_test_poly)
print(len(predict))
print(predict)
```

```
83
[0.93900505 0.80619283 0.55205285 0.89669043 0.89406791 0.90312677
 0.65627672 0.82664329 0.69855075 0.88007239 0.48814413 0.6625744
 0.50814435 0.91766221 0.78279802 0.78742431 0.88246902 0.5352969
 0.86940063 0.61758035 0.60681029 0.67654777 0.79536675 0.78659696
 0.77144919 0.84187955 0.60881221 0.81039139 0.73946622 0.88108845
 0.92751346 0.83400704 0.76064681 0.66477684 0.83666255 0.79371682
 0.89180523 0.52469205 0.97364817 0.90108738 0.94073133 0.81341182
 0.79145538 0.95671779 0.90121066 0.86477048 0.92842539 0.9523928
 0.86165846 0.83205429 0.9566745  0.94063922 0.78279802 0.84491827
 0.64539814 0.76787675 0.53672113 0.8743241  0.82004003 0.68982536
 0.86587685 0.91309002 0.92677014 0.90676972 0.87803409 0.92337115
 0.77635309 0.63041967 0.88898653 0.79246158 0.88510342 0.77290793
 0.93183647 0.40243217 0.90737654 0.94063922 0.95961823 0.86734253
 0.95125099 0.87901641 0.78834889 0.53219672 0.79669956]
```

In [ ]:
```
# Get the coefficients and intercept
coefficients = poly_reg.coef_
intercept = poly_reg.intercept_
print("Coefficients:", coefficients)
print("Intercept:", intercept)
```

```
Coefficients: [ 3.70727233e-13  6.47437228e-02  7.16362232e-02 -1.19489270e-
01
 -1.24913722e-01  3.02656035e-02 -7.20748379e-02 -4.28784873e-04
 -9.74900156e-04  6.07110854e-04  1.23220606e-03 -1.48110158e-04
  2.09906010e-03 -6.02336811e-04  1.25356793e-03  1.55488016e-03
 -4.80193360e-04 -9.32456057e-04  7.94058174e-03 -6.84358022e-04
 -4.24280851e-03  8.00208672e-03  1.49527175e-03  7.41758594e-03
  2.41680357e-04 -9.19523409e-04  2.41602851e-03 -7.20748379e-02]
Intercept: -1.4772395889009562
```

In [ ]:
```
from sklearn.metrics import mean_squared_error, r2_score
# Obtain the mse and r2 for the polynomial regression model
mse_poly = mean_squared_error(y_test, predict)
r2_poly = r2_score(y_test, predict)
```

```
print("Mean squared error:",  mean_squared_error(y_test, predict))
print("R-squared:", r2_score(y_test, predict))
```

```
Mean squared error: 0.003407118843662460
R-squared: 0.7897913848797345
```

In [ ]:

```
# Comparison of both models
# mse
print("Linear Regression MSE:", mse_linear)
print("Polynomial Regression MSE:", mse_poly)

# r2
print("Linear Regression R2:", r2_linear)
print("Polynomial Regression R2:", r2_poly)
```

```
Linear Regression MSE: 0.0031909500116446767
Polynomial Regression MSE: 0.003407118843662460
Linear Regression R2: 0.803128328172789
Polynomial Regression R2: 0.7897913848797345
```

In [ ]:

```
# Choose the model which the lowest MSE
if mse_linear < mse_poly:
    print("Linear Regression performs better")
else:
    print("Polynomial Regression performs better")

# Choose the model with the highest R2
if r2_linear > r2_poly:
    print("Linear Regression performs better")
else:
    print("Polynomial Regression performs better")
```

```
Linear Regression performs better
Linear Regression performs better
```

6 References

Equilibriumm. (2023, February 21). Sleep efficiency dataset. Kaggle. https://www.kaggle.com/datasets/equilibriumm/sleep-efficiency

Ikeda, Y., Morita, E., Muroi, K., Arai, Y., Ikeda, T., Takahashi, T., Shiraki, N., Doki, S., Hori, D., Oi, Y., Sasahara, S. I., Ishihara, A., Matsumoto, S., Yanagisawa, M., Satoh, M., & Matsuzaki, I. (2022). Relationships between sleep efficiency and lifestyle evaluated by objective sleep assessment: SLeep Epidemiology Project at University of Tsukuba. Nagoya journal of medical science, 84(3), 554–569. https://doi.org/10.18999/nagjms.84.3.554

How to plot for multiple linear regression model using matplotlib. Saturn Cloud Blog. (2023, December 20). https://saturncloud.io/blog/how-to-plot-for-multiple-linear-regression-model-using-matplotlib/

Multivariate polynomial regression with python. Saturn Cloud Blog. (2024, February 1). https://saturncloud.io/blog/multivariate-polynomial-regression-with-python/ :~̈text=Multivariate%20polynomial%20regressionnewline%20is%20an,just%20like %20in%20linear%20regression.

Multivariate regression: Definition, example and steps. Voxco. (2024, July 5). https://www.voxco.com/blog/multivariate-regression-definition-example-and steps/ #:~̈text=of%20computer%20sciences.-,What%20is%20multivariate%20regression?,based %20on%20multiple%20independent%20variables.