Data Analytics

[Sleep Health and Lifestyle]

REDA RERIZANI

April, 2024

Table of content

Introduction

1. Business Use Case

Understanding the trends and impacts of healthy sleep and lifestyle habits is crucial for companies in the nutritional supplement industry to develop effective strategies for promoting overall wellness.
Identifying what's important for good sleep and lifestyle helps companies create products that satisfy customers.

2. Goal

The goal of my project is to:
_ Analyze trends in the quality of sleep across individuals' lifestyles.

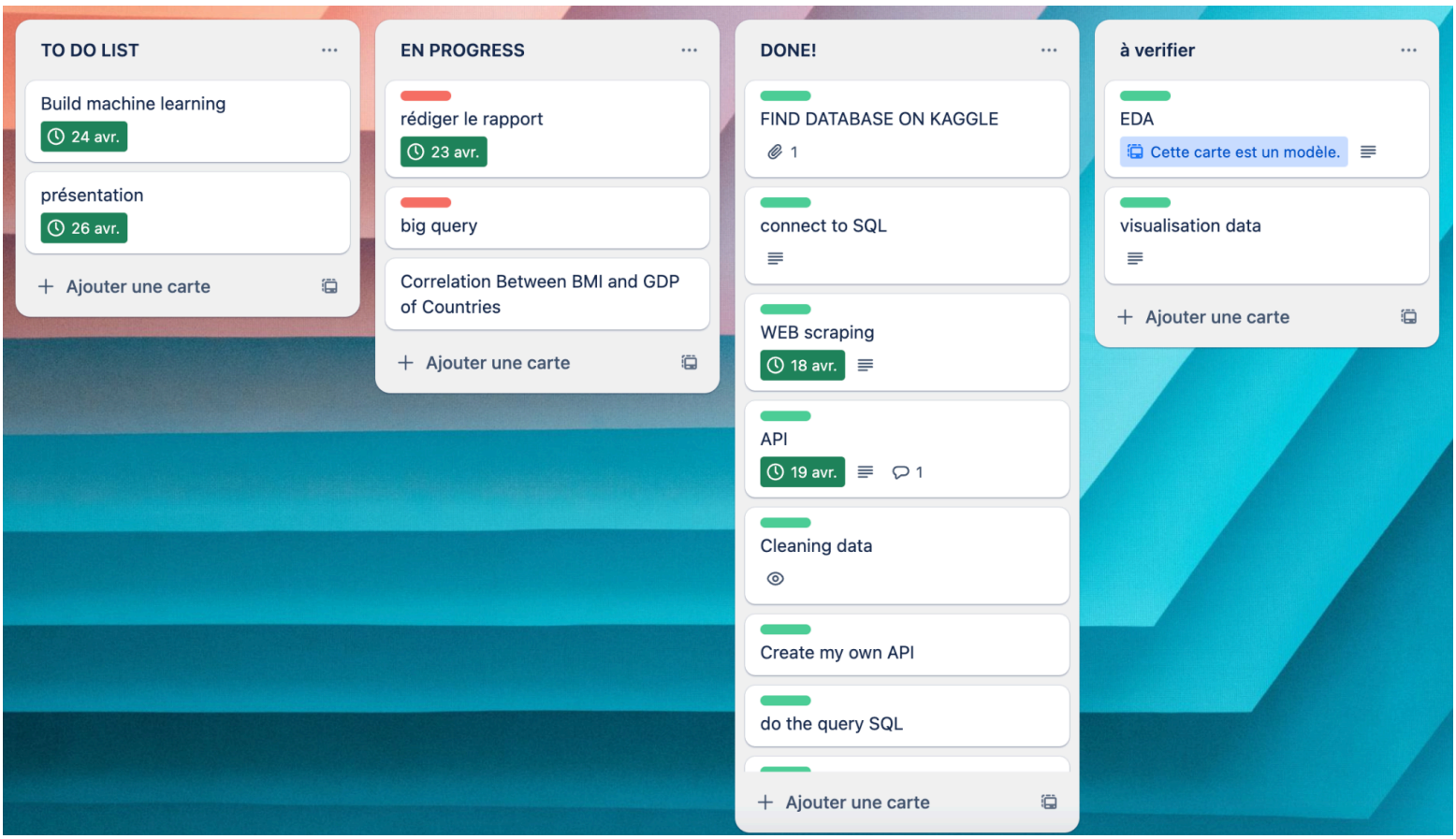
_ Identify key indicators of sleep quality and duration, and to create a prediction model based on available data.

3. Plan

_ Research about project topic
_ Data collection
_ Project planning in Trello
_ Selection and creation of a database using MySQL
_ Adding data to database and create Entity Relationship Diagram
_ Data manipulation in SQL
_ Exploratory data analysis in Python (data wrangling, data cleaning & data visualization
_ Exposing data via API
_ Visualization data

Project Management

Trello board to manage daily project tasks:
https://trello.com/b/ByEKV71C/projet-final

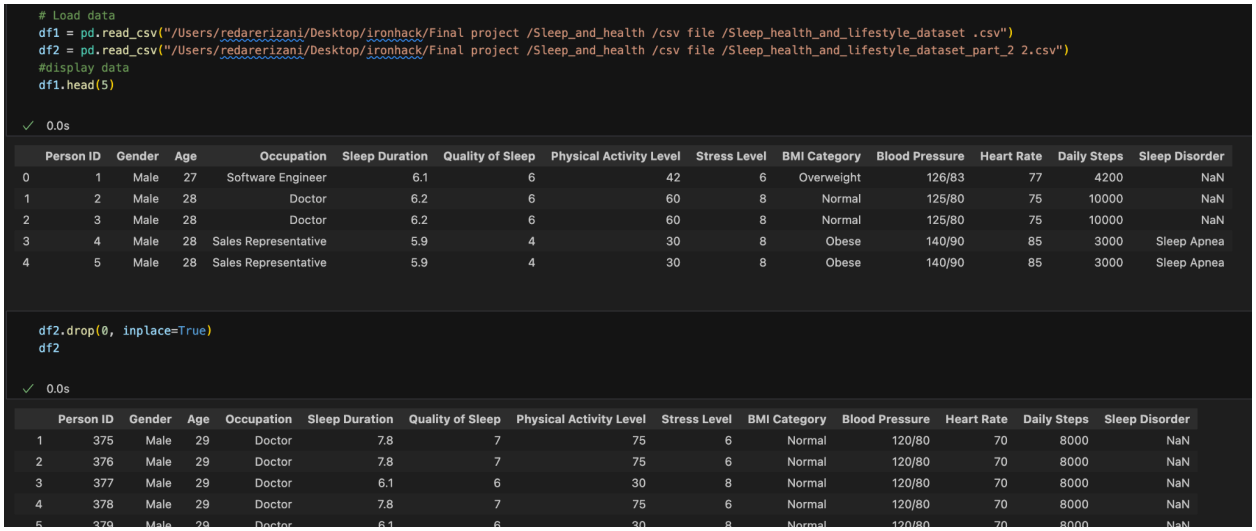


Data and data sources

For my last project, I gathered information from different places to create my final dataset. My main data was collected from Kaggle. In addition to that, I used methods like APIs, BigQuery, and web scraping to extract data from websites.

1. Flat data

I chose these two tables from Kaggle : https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset and https://www.kaggle.com/datasets/caymansmith/sleep-health-and-lifestyle-data-set-part-2 because they contain three crucial pieces of information for my project: details about people, their sleep quality, and factors affecting their sleep. This dataset will help me understand how various factors impact the quality of sleep.



```
# Load data
df1 = pd.read_csv("/Users/redarerizani/Desktop/ironhack/Final project /Sleep_and_health /csv file /Sleep_health_and_lifestyle_dataset .csv")
df2 = pd.read_csv("/Users/redarerizani/Desktop/ironhack/Final project /Sleep_and_health /csv file /Sleep_health_and_lifestyle_dataset_part_2 2.csv")
#display data
df1.head(5)
```

✓ 0.0s

	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
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	NaN
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	NaN
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea

```
df2.drop(0, inplace=True)
df2
```

✓ 0.0s

	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
1	375	Male	29	Doctor	7.8	7	75	6	Normal	120/80	70	8000	NaN
2	376	Male	29	Doctor	7.8	7	75	6	Normal	120/80	70	8000	NaN
3	377	Male	29	Doctor	6.1	6	30	8	Normal	120/80	70	8000	NaN
4	378	Male	29	Doctor	7.8	7	75	6	Normal	120/80	70	8000	NaN
5	379	Male	29	Doctor	6.1	6	30	8	Normal	120/80	70	8000	NaN

2. Web scraping

The first table I got from the website is about body mass index (BMI) categories. It's from this link: https://en.wikipedia.org/wiki/Body_mass_index. I'm using it to add more info about BMI in my main dataset.

	Category	BMI (kg/m2)[c]	BMI Prime[c]
0	Severe thinness	< 16.0	< 0.64
1	Moderate thinness	16.0 – 16.9	0.64 – 0.67
2	Underweight	17.0 – 18.4	0.68 – 0.73
3	Normal	18.5 – 24.9	0.74 – 0.99
4	Overweight	25.0 – 29.9	1.00 – 1.19
5	Obese I	30.0 – 34.9	1.20 – 1.39
6	Obese II	35.0 – 39.9	1.40 – 1.59
7	Obese III	≥ 40.0	≥ 1.60

The second table, from https://en.wikipedia.org/wiki/List_of_sovereign_states_by_body_mass_index, lists countries by their average BMI. I want to check how BMI relates to the richest and poorest countries based on GDP.

	Country	Both	Male	Female
183	Afghanistan*	21.6	21.5	21.8
88	Albania*	26.1	26.6	25.6
87	Algeria*	26.2	25.5	27.0
33	Andorra*	27.5	27.8	27.1
138	Angola*	24.1	23.5	24.6
...	...	...	...	...
46	Venezuela*	27.2	27.4	27.1
182	Vietnam*	21.6	21.5	21.6
102	Yemen*	25.8	25.2	26.5
165	Zambia*	22.6	21.5	23.8
148	Zimbabwe*	23.4	21.8	25.0

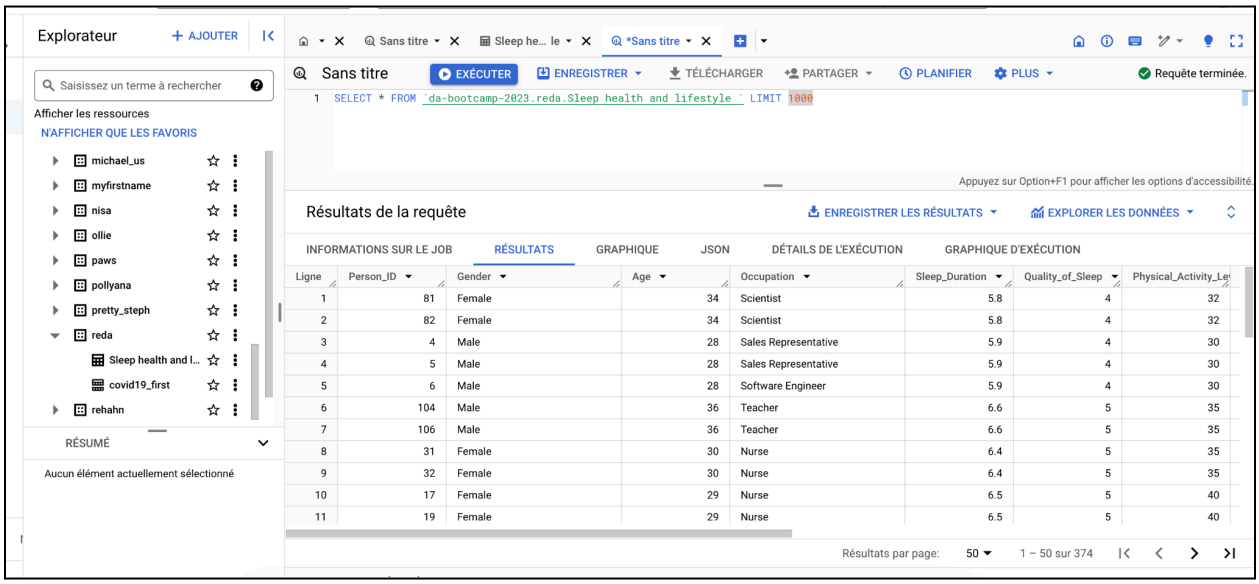
3. API

World Bank Data API was used as an additional resource for my analysis. The dataset 'GDP by country' was created using this API. Its purpose is to rank the countries from poorest to richest based on their GDP.

	Country	GDP_MEAN
0	Afghanistan	1.776877e+10
1	Africa Eastern and Southern	1.043852e+12
2	Africa Western and Central	8.197695e+11
3	Albania	1.651359e+10
4	Algeria	1.701771e+11
...	...	...
261	West Bank and Gaza	1.723254e+10
262	World	9.152510e+13
263	Yemen, Rep.	2.160616e+10
264	Zambia	2.379820e+10
265	Zimbabwe	2.664717e+10

4. Big Query Google Cloud Platform

The big query is available at https://console.cloud.google.com/bigquery?project=da-bootcamp-2023&ws=!1m5!1m4!4m3!1sda-bootcamp-2023!2sreda!3sSleep%2520health%2520and%2520lifestyle%2520



Data cleaning and Exploratory data analysis

1. Main dataset overview

Key Features of the Dataset:

The Sleep Health and Lifestyle Dataset consists of 3 key features: personal information, comprehensive sleep metrics, and sleep disorders and factors influencing sleep patterns.

Dataset Columns:

- Person ID: An identifier for each individual.
- Gender: The gender of the person (Male/Female).
- Age: The age of the person in years.
- Occupation: The occupation or profession of the person.
- Sleep Duration (hours): The number of hours the person sleeps per day.
- Quality of Sleep (scale: 1-10): A subjective rating of the quality of sleep, ranging from 1 to 10.
- Physical Activity Level (minutes/day): The number of minutes the person engages in physical activity daily.
- Stress Level (scale: 1-10): A subjective rating of the stress level experienced by the person, ranging from 1 to 10.
- BMI Category: The BMI category of the person (e.g., Underweight, Normal, Overweight).
- Blood Pressure (systolic/diastolic): The blood pressure measurement of the person, indicated as systolic pressure over diastolic pressure.
- Heart Rate (bpm): The resting heart rate of the person in beats per minute.
- Daily Steps: The number of steps the person takes per day.
- Sleep Disorder: The presence or absence of a sleep disorder in the person (None, Insomnia, Sleep Apnea).

2. Data Cleaning main data (from kaggle)

I combined my two main tables collected from Kaggle into a single dataframe. Regarding data cleaning, I followed these steps:

- Checked the number of observations.
- Checked data types.
- Renamed and formatted the column names by removing special characters and
- them to lowercase.

```
# Formatting column
# Convert column names to lowercase
sleep_df.columns = sleep_df.columns.str.lower()

# Replace spaces in column names with underscores and remove extra spaces
sleep_df.columns = sleep_df.columns.map(lambda word: word.strip().replace(" ", "_"))
sleep_df
```

- Handled null values using "isna.sum()".

Most of the time, there was no need to clean the data because there were no nulls or duplicates. However, I made some modifications such : as

- Replacing null values with "no disorder" in the "sleep_disorder" column using the 'fillna' function.

```
sleep_df.isna().sum()
```

```
person_id                  0
gender                     0
age                        0
occupation                 0
sleep_duration             0
quality_of_sleep           0
physical_activity_level    0
stress_level               0
bmi_category               0
blood_pressure             0
heart_rate                 0
daily_steps                0
sleep_disorder           375
dtype: int64
```

```
sleep_df["sleep_disorder"].fillna("no disorder", inplace=True)
```

- Modifying the values in the "blood_pressure" column, such as changing "126/83" to a categorical value like "hypertension"...

```python
# Extract Systolic and Diastolic Pressure
sleep_df['systolic_pressure'] = sleep_df['blood_pressure'].apply(lambda x: int(x.split("/")[0]))
sleep_df['diastolic_pressure'] = sleep_df['blood_pressure'].apply(lambda x: int(x.split("/")[1]))

# Define thresholds and categories for blood pressure classes
list_high = [0, 90, 120, 130, max(sleep_df.systolic_pressure)]
classif = [0, 1, 10, 100]  # 0: Hypotension, 1: Normal, 10: Elevated, 100: Hypertension
list_low = [0, 60, 80, 90, max(sleep_df.diastolic_pressure)]

# Categorize Systolic and Diastolic Pressure into Classes
sleep_df['systolic_classes'] = pd.cut(sleep_df['systolic_pressure'], 4, labels=classif).astype(int)
sleep_df['diastolic_classes'] = pd.cut(sleep_df['diastolic_pressure'], 4, labels=classif).astype(int)

# Assign descriptive labels to each row based on combined blood pressure classes
def check_pressure(x):
    if x >= 100:
        return "Hypertension"
    elif x >= 10:
        return "Elevated"
    elif x >= 1:
        return "Normal"
    elif x == 0:
        return "Hypotension"

sleep_df['blood_pressure_class'] = sleep_df['systolic_classes'] + sleep_df['diastolic_classes']
sleep_df['blood_pressure_class'] = sleep_df['blood_pressure_class'].apply(check_pressure)

# Drop unnecessary columns
sleep_df.drop(columns=['systolic_pressure', 'diastolic_pressure', 'systolic_classes', 'diastolic_classes', 'blood_pressure'], inplace=True)
```

3. Clean and wrangling data between BMI and GDP of each country

I tried to find the correlation between BMI and GDP of the richest and poorest countries. To do this, here are the steps:

- Clean and standardize country names.
- Perform a left join using standardized country names between the BMI and GDP DataFrames.

```
# Clean and standardize country names in the BMI DataFrame
bmi['Country'] = bmi['Country'].str.strip()  # Remove leading and trailing spaces
bmi['Country'] = bmi['Country'].str.replace('.', '')  # Remove periods
bmi['Country'] = bmi['Country'].str.replace(',', '')  # Remove commas
bmi['Country'] = bmi['Country'].str.replace('–', ' ')  # Replace dashes with spaces
bmi['Country'] = bmi['Country'].str.upper()  # Convert to uppercase

# Clean and standardize country names in the GDP DataFrame
gdp['Country'] = gdp['Country'].str.strip()  # Remove leading and trailing spaces
gdp['Country'] = gdp['Country'].str.replace('.', '')  # Remove periods
gdp['Country'] = gdp['Country'].str.replace(',', '')  # Remove commas
gdp['Country'] = gdp['Country'].str.replace('–', ' ')  # Replace dashes with spaces
gdp['Country'] = gdp['Country'].str.upper()  # Convert to uppercase

# Perform the join using standardized country names
df_bmi_gdp = bmi.merge(gdp, on='Country', how='left')
```

- Define a function to convert GDP values from scientific notation to "billion".
- Drop rows with missing values.
- Retrieve the top 5 richest countries based on GDP_MEAN.
- Retrieve the top 5 poorest countries based on GDP_MEAN.

We obtain the following two tables:

```
top_10_rich_countries = df_bmi_gdp.nlargest(5, 'GDP_MEAN')
top_10_rich_countries
```

	Country	BMI	GDP_MEAN
186	UNITED STATES	28.8	22345.86
35	CHINA	23.9	15729.25
84	JAPAN	22.6	4901.10
63	GERMANY	26.3	4022.46
75	INDIA	21.9	2955.42

```
top_10_poor_countries = df_bmi_gdp.nsmallest(5, 'GDP_MEAN')
top_10_poor_countries
```

	Country	BMI	GDP_MEAN
181	TUVALU	29.3	0.05
120	NAURU	32.5	0.14
88	KIRIBATI	29.6	0.20
107	MARSHALL ISLANDS	29.2	0.24
132	PALAU	29.4	0.27

Note: We cannot observe any significant relationship between the GDP of rich and poor countries and BMI (Body Mass Index). Therefore, for the remainder of my data analysis, I will not take them into consideration.

4. Exploratory data analysis

To begin with, I identified the numerical and categorical variables.

```
sleep_df.dtypes
✓ 0.0s
```

```
person_id                 int64
gender                   object
age                       int64
occupation               object
sleep_duration          float64
quality_of_sleep          int64
physical_activity_level   int64
stress_level              int64
id_bmi                    int64
heart_rate                int64
daily_steps               int64
sleep_disorder           object
blood_pressure_class     object
dtype: object
```

Then, I separated numerical variables into discrete and continuous variables, considering that discrete variables could potentially be treated as categorical.

```
# Separating between discrete and continuous variables, as discrete ones could potentially be treated as categorical.
potential_categorical_from_numerical = sleep_df.select_dtypes("number").loc[:, sleep_df.select_dtypes("number").nunique()

# Separate data in categorical and numerical dataframe
df_categorical = pd.concat([sleep_df.select_dtypes("object"), potential_categorical_from_numerical], axis=1)
df_numerical = sleep_df.select_dtypes("number").drop(columns=potential_categorical_from_numerical.columns)
```

Finally we find :

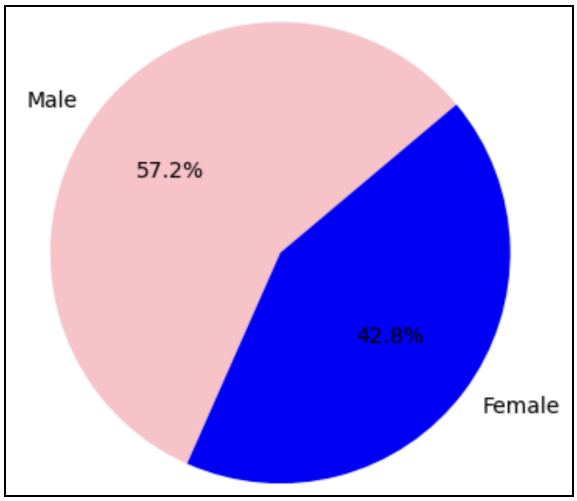
- Categorical values : 'gender', 'occupation', 'sleep_disorder', 'blood_pressure_class', 'id_bmi'
- Numérical variable: 'person_id', 'age', 'sleep_duration', 'quality_of_sleep', 'physical_activity_level', 'stress_level', 'heart_rate', 'daily_steps'

Summary statistics for the dataset (Numerical variable)

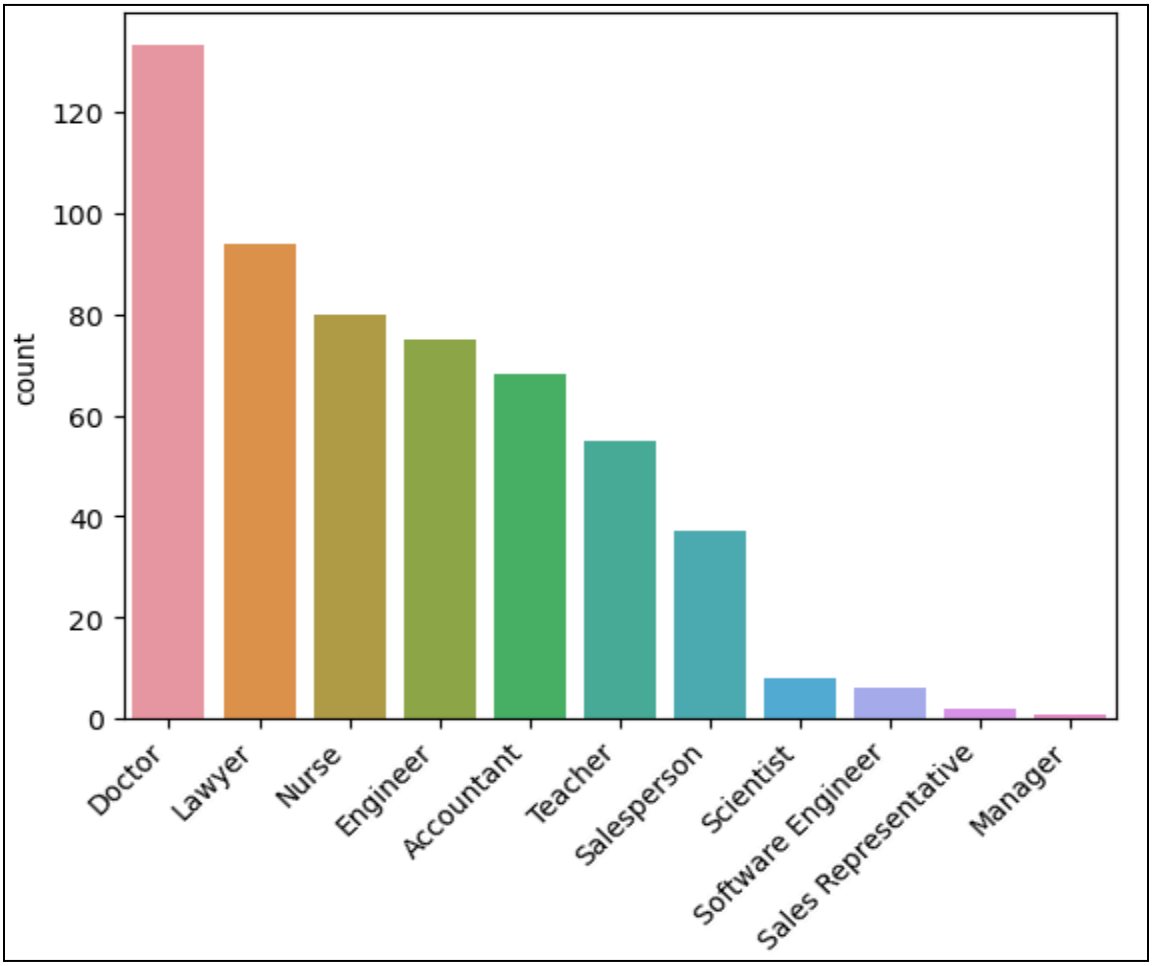
	person_id	age	sleep_duration	quality_of_sleep	physical_activity_level	stress_level	heart_rate
count	559.00	559.00	559.00	559.00	559.00	559.00	559.00
mean	280.00	39.99	7.11	7.27	58.76	5.46	70.18
std	161.51	8.10	0.74	1.13	19.96	1.67	3.88
min	1.00	27.00	5.80	4.00	30.00	3.00	65.00
25%	140.50	33.00	6.50	6.00	45.00	4.00	68.00
50%	280.00	38.00	7.20	7.00	60.00	5.00	70.00
75%	419.50	44.00	7.70	8.00	75.00	7.00	72.00
max	559.00	59.00	8.50	9.00	90.00	8.00	86.00

Visualisations

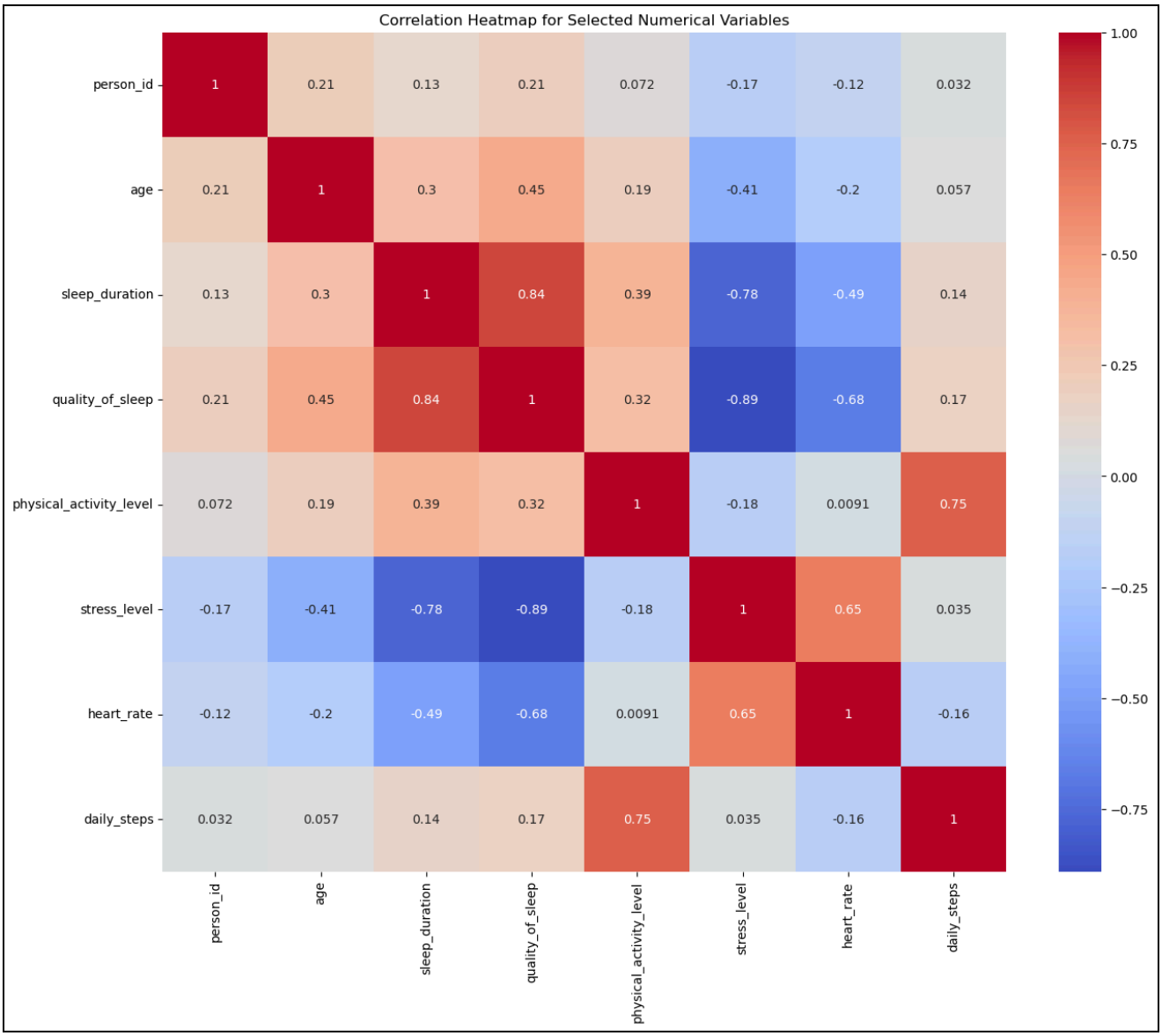
- **Gender distribution:**



- **Occupation distribution**



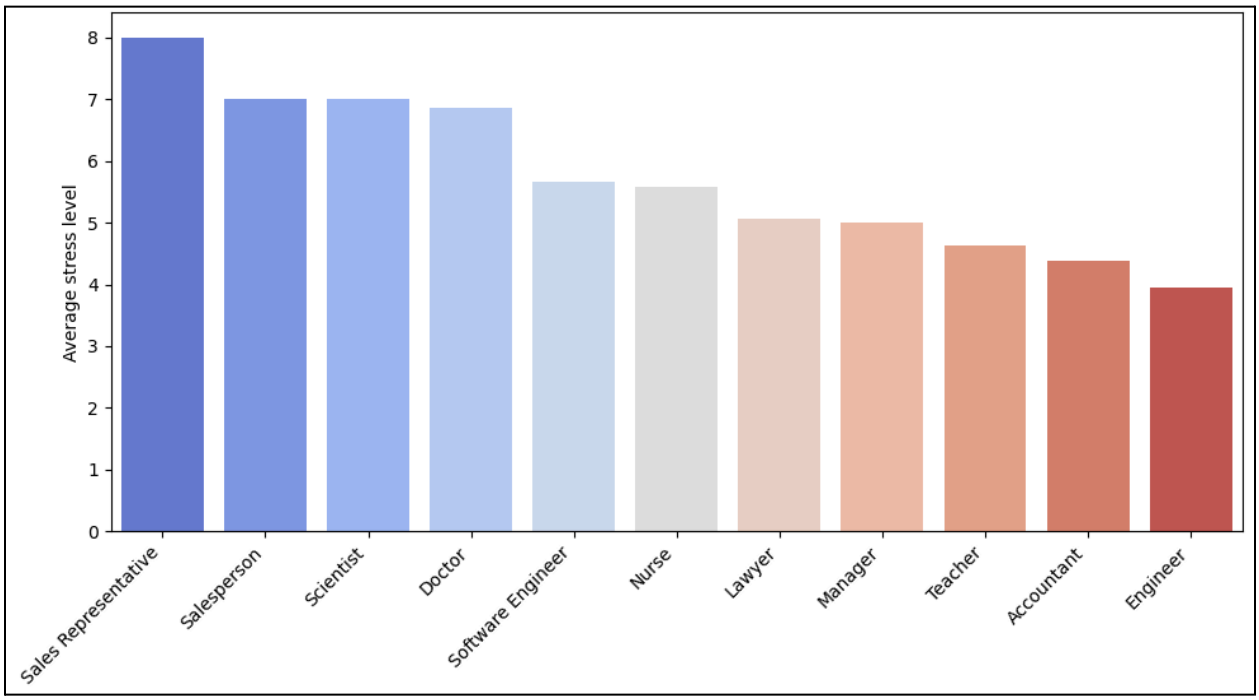
- **Correlation between quality of sleep and the factors / lifestyle**



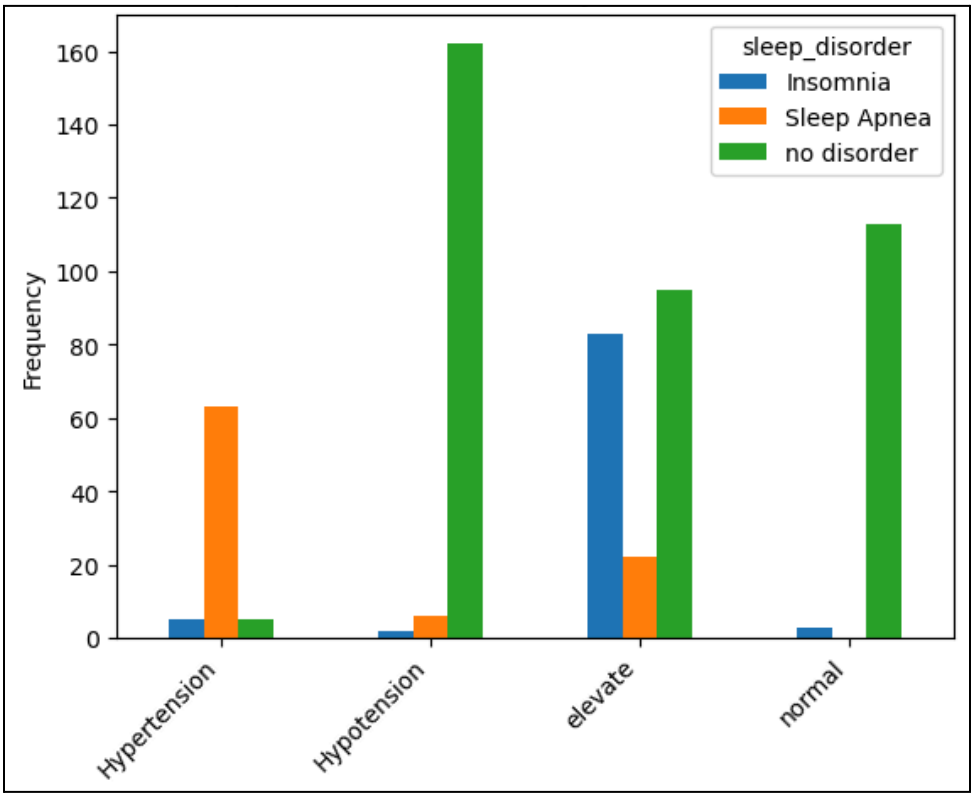
Insight :

- Strong positive correlation between quality of sleep and sleep duration.
- Positive correlation between heart rate and quality sleep
- Strong negative correlation between stress level and both quality of sleep and sleep duration.
- There is a slight positive correlation between physical activity and quality of sleep.

- **Average stress level by profession**



- **Sleep Disorder Distribution by Blood Pressure Class**



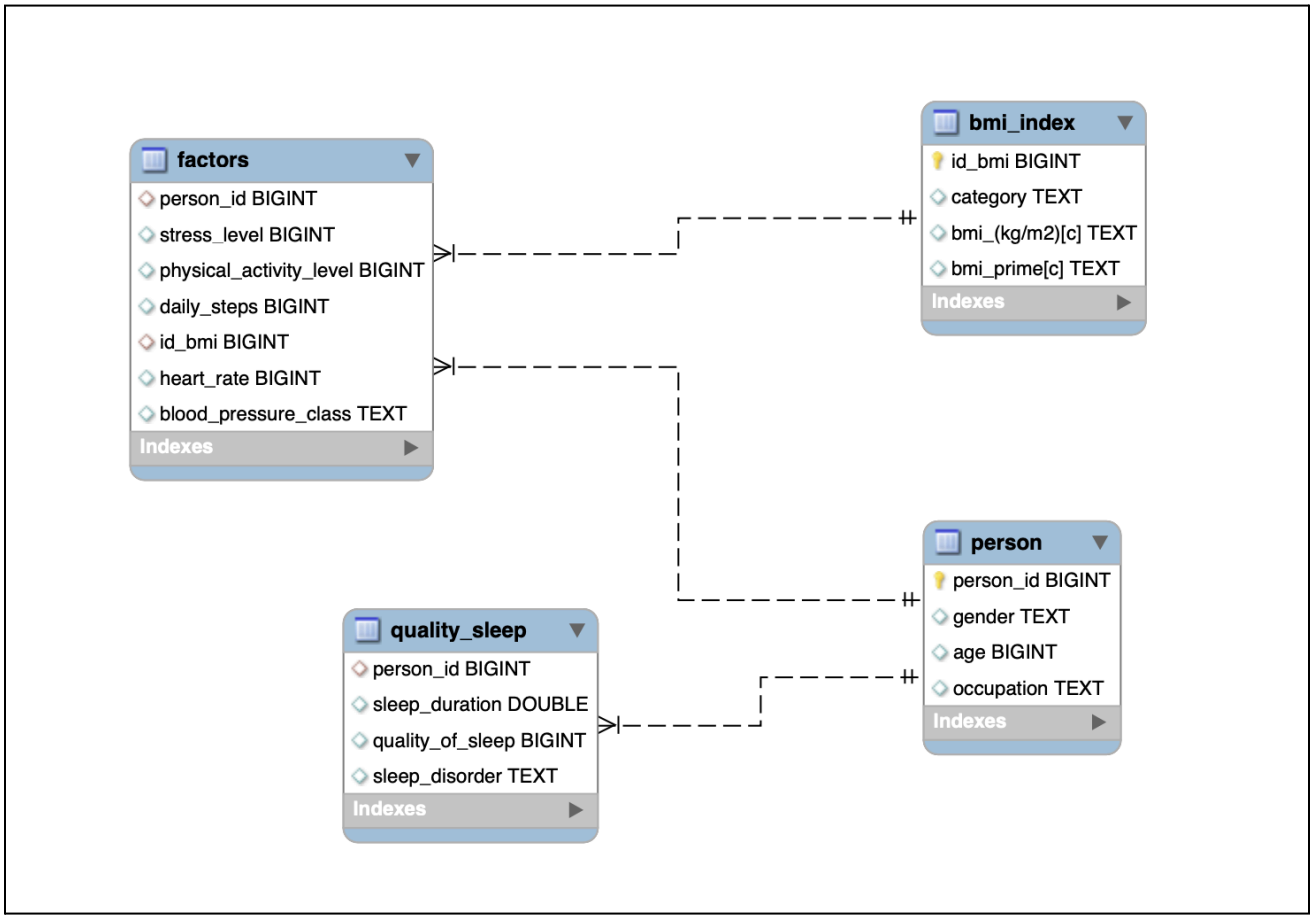
Database type selection

I chose MySQL for my project because it's excellent at managing the type of data I have. The organization of my data involved defining my tables along with their relationships and appropriate primary and foreign keys.

"After cleaning the data in Python, I connected to SQL to transfer my tables to my 'sleep' database that I created."

```
from sqlalchemy import create_engine, text
import pymysql.cursors
pw_raw = getpass.getpass(prompt="Enter your password: ")
pw = urllib.parse.quote_plus(pw_raw)
connection_string = 'mysql+pymysql://root:' + pw + '@localhost:3306/sleep'
engine = create_engine(connection_string)
```

Entities ERD



SQL queries

Examples of queries in MySQL (all queries are available in repository):

_ Average quality and hours of sleep according to BMI

```
33 •  SELECT bi.category, ROUND(AVG(qs.quality_of_sleep),2) AS quality_sleep, ROUND(AVG(qs.sleep_duration), 2)
      AS sleep_duration
34    FROM quality_sleep qs
35    INNER JOIN factors f ON f.person_id = qs.person_id
36    INNER JOIN bmi_index bi ON bi.id_bmi = f.id_bmi
37    GROUP BY bi.category
38    ORDER BY quality_sleep DESC;
39
```

100% 18:29

Result Grid Filter Rows: Search Export:

category	quality_sleep	sleep_duration
Normal	7.52	7.29
Overweight	6.78	6.74
Obese I	6.50	7.03

_ Create view quality and hour of sleep according to occupation and stress level

```
41 •  CREATE VIEW quality_sleep_by_occupation_and_stress_level AS
42    SELECT p.occupation, ROUND(AVG(f.stress_level),2) as avg_stress_level, ROUND(AVG(qs.quality_of_sleep),2)
      AS quality_sleep, ROUND(AVG(qs.sleep_duration), 2) AS sleep_duration
43    FROM person p
44    JOIN quality_sleep qs ON p.person_id = qs.person_id
45    JOIN factors f ON p.person_id = f.person_id
46    GROUP BY p.occupation;
47
```

100% 44:45

Result Grid Filter Rows: Search Export:

occupation	avg_stress_le...	quality_sleep	sleep_duration
Software Engineer	5.67	7.00	7
Doctor	6.86	6.58	6.94
Sales Representative	8.00	4.00	5.9
Teacher	4.64	6.98	6.74
Nurse	5.59	7.25	7.04
Engineer	3.95	8.31	7.88
Accountant	4.38	7.97	7.16
Scientist	7.00	5.00	6
Lawyer	5.06	7.89	7.41
Salesperson	7.00	6.00	6.41
Manager	5.00	7.00	6.9

_ Create view average quality and hours of sleep according to BMI

```
49 •  CREATE VIEW quality_sleep_by_bmi AS
50    SELECT bi.category, ROUND(AVG(qs.quality_of_sleep),2) AS quality_sleep, ROUND(AVG(qs.sleep_duration), 2)
      AS sleep_duration
51    FROM quality_sleep qs
52    INNER JOIN factors f ON f.person_id = qs.person_id
53    INNER JOIN bmi_index bi ON bi.id_bmi = f.id_bmi
54    GROUP BY bi.category;
55
```

100% 14:43

Result Grid | Filter Rows: Search | Export:

category	quality_sleep	sleep_duration
Overweight	6.78	6.74
Normal	7.52	7.29
Obese I	6.50	7.03

Result Grid | Form

_ Quality of sleep according to disorder

```
57 •  SELECT sleep_disorder, ROUND(AVG(quality_of_sleep), 2) AS quality_sleep, ROUND(AVG(sleep_duration), 2) AS
      sleep_duration
58    FROM quality_sleep
59    GROUP BY sleep_disorder
60    ORDER BY quality_sleep DESC;
61
```

100% 19:68

Result Grid | Filter Rows: Search | Export:

sleep_disorder	quality_sleep	sleep_duration
no disorder	7.52	7.27
Sleep Apnea	7.03	6.99
Insomnia	6.49	6.6

Result Grid | Form

_ The average quality of sleep for female who gave a physical activity level greater than 40 minutes

```
63 •  SELECT p.gender, ROUND(AVG(qs.quality_of_sleep), 2) AS avg_quality_of_sleep , ROUND(AVG(sleep_duration), 2
      ) AS sleep_duration
64    FROM person p
65    JOIN quality_sleep qs ON p.person_id = qs.person_id
66    JOIN factors f ON p.person_id = f.person_id
67    WHERE p.gender = 'female' AND f.physical_activity_level > 40
68    GROUP BY p.gender;
69
```

100% 14:72

Result Grid | Filter Rows: Search | Export:

gender	avg_quality_of_sleep	sleep_duration
Female	7.56	7.04

Result Grid

_ The average quality of sleep and sleep duration for doctors who have a stress level less than 5

```
71 •  SELECT p.occupation, ROUND(AVG(qs.quality_of_sleep), 2) AS avg_quality_of_sleep, ROUND(AVG(qs.
      sleep_duration), 2) AS sleep_duration
72    FROM person p
73    JOIN quality_sleep qs ON p.person_id = qs.person_id
74    JOIN factors f ON p.person_id = f.person_id
75    WHERE p.occupation = 'Doctor' AND f.stress_level < 5
76    GROUP BY p.occupation;
77
78
```

100% 1:77

Result Grid Filter Rows: Search Export:

occupation	avg_quality_of_sleep	sleep_duration
Doctor	9.00	8.15

Create API

To expose a portion of the data from the database, I have created an API with two main roots:

Root 1: " http://192.168.0.21:8080/person "
Provides information about individuals from our database (age, gender, occupation). By including the option include_details = 1, we can obtain information about sleep quality in addition to the basic information. "
http://192.168.0.21:8080/person?include_details=1 "


```
{
  "last_page": "/person?page=19&page_size=30&include_details=1",
  "next_page": "/person?page=1&page_size=30&include_details=1",
  "person": [
    {
      "age": 27,
      "gender": "Male",
      "occupation": "Software Engineer",
      "person_id": 1,
      "quality_sleep": [
        {
          "quality_of_sleep": 6,
          "sleep_disorder": "no disorder",
          "sleep_duration": 6.1
        }
      ]
    },
    {
      "age": 28,
      "gender": "Male",
      "occupation": "Doctor",
      "person_id": 2,
      "quality_sleep": [
        {
          "quality_of_sleep": 6,
          "sleep_disorder": "no disorder",
          "sleep_duration": 6.2
        }
```


We can also retrieve all information about an individual by adding their ID number, including personal information, sleep quality, and lifestyle details.
http://192.168.0.21:8080/person/<int:person_id>"

Root 2: "http://192.168.0.21:8080/quality_sleep"
Offers details on sleep quality for each person, including sleep duration, sleep rate, and sleep disorders.


```
[
  {
    "person_id": 1,
    "quality_of_sleep": 6,
    "sleep_disorder": "no disorder",
    "sleep_duration": 6.1
  },
  {
    "person_id": 2,
    "quality_of_sleep": 6,
    "sleep_disorder": "no disorder",
    "sleep_duration": 6.2
  },
  {
    "person_id": 3,
    "quality_of_sleep": 6,
    "sleep_disorder": "no disorder",
    "sleep_duration": 6.2
  },
  {
```


http://192.168.0.21:8080/quality_sleep/bmi provides data on sleep quality based on BMI.


```
[
  {
    "category": "Overweight",
    "quality_sleep": "6.78",
    "sleep_duration": 6.74
  },
  {
    "category": "Normal",
    "quality_sleep": "7.52",
    "sleep_duration": 7.29
  },
  {
    "category": "Obese I",
    "quality_sleep": "6.50",
    "sleep_duration": 7.03
  }
]
```

http://192.168.0.21:8080/quality_sleep/occupation gives insights into sleep quality based on occupation and stress levels.


```
[
  {
    "avg_stress_level": "5.67",
    "occupation": "Software Engineer",
    "quality_sleep": "7.00",
    "sleep_duration": 7.0
  },
  {
    "avg_stress_level": "6.86",
    "occupation": "Doctor",
    "quality_sleep": "6.58",
    "sleep_duration": 6.94
  },
  {
    "avg_stress_level": "8.00",
    "occupation": "Sales Representative",
    "quality_sleep": "4.00",
    "sleep_duration": 5.9
  },
```

Conclusions

Insights from the project reveal significant correlations between sleep quality and various lifestyle factors. There's a strong positive correlation between sleep quality and duration, indicating the importance of getting enough sleep for overall well-being.

Additionally, a positive correlation was observed between heart rate and sleep quality, suggesting that maintaining a healthy heart may contribute to better sleep.

Conversely, stress level showed a strong negative correlation with both sleep quality and duration, emphasizing the detrimental effects of stress on sleep.

These insights underscore the importance of considering lifestyle factors, such as stress management and physical activity, in promoting healthy sleep habits.

For companies in the nutritional supplement industry, using this information can help them create products that improve sleep and overall health for customers. Concentrating on reducing stress could be especially helpful, as being stressed out can worsen sleep and affect overall health.

GDPR

Upon thorough examination of the data collected for this project, I confirm that no personal data was utilized throughout the project. All data sources used are publicly available at a country level, ensuring transparency and compliance with General Data Protection Regulation (GDPR) guidelines.