Sleep Health and Lifestyle

November 22, 2023

[1]:

```
## Importing the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
```

[2]:

```
## Loading the data
df = pd.read_csv('/Users/Home/OneDrive/Desktop/Python/
 ↪Sleep_health_and_lifestyle_dataset.csv')
```

[3]:

```
df.head()
```

[3]:

```
   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           None
1         125/80          75        10000           None
2         125/80          75        10000           None
3         140/90          85         3000    Sleep Apnea
4         140/90          85         3000    Sleep Apnea
```

1 Data Cleaning

[4]:
```
## Checking for Missing data
total = df.isnull().sum().sort_values(ascending=False)
percent = (df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['total', 'Percent'])
missing_data.head(10)
```

[4]:
```
                         total  Percent
Person ID                    0      0.0
Gender                       0      0.0
Age                          0      0.0
Occupation                   0      0.0
Sleep Duration               0      0.0
Quality of Sleep             0      0.0
Physical Activity Level      0      0.0
Stress Level                 0      0.0
BMI Category                 0      0.0
Blood Pressure               0      0.0
```

[5]:
```
df.describe()
```

[5]:
```
        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
```

[6]:
```
## Dropping person ID Column
df = df.drop('Person ID', axis=1)
```

[7]:
```
df.head()
```

[7]:

```
  Gender  Age            Occupation  Sleep Duration  Quality of Sleep  \
0   Male   27     Software Engineer             6.1                 6
1   Male   28                Doctor             6.2                 6
2   Male   28                Doctor             6.2                 6
3   Male   28  Sales Representative             5.9                 4
4   Male   28  Sales Representative             5.9                 4

   Physical Activity Level  Stress Level BMI Category Blood Pressure  \
0                       42             6   Overweight         126/83
1                       60             8       Normal         125/80
2                       60             8       Normal         125/80
3                       30             8        Obese         140/90
4                       30             8        Obese         140/90

   Heart Rate  Daily Steps Sleep Disorder
0          77         4200           None
1          75        10000           None
2          75        10000           None
3          85         3000    Sleep Apnea
4          85         3000    Sleep Apnea
```

[8]:

```
df.info()
```

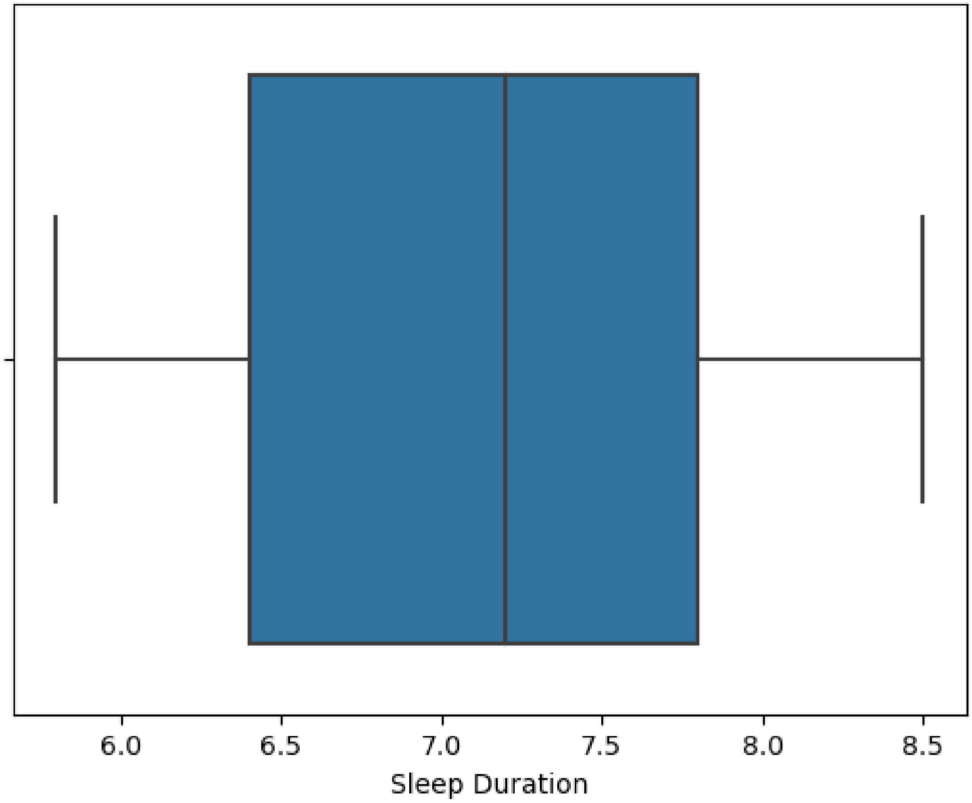
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 12 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Gender                   374 non-null    object
 1   Age                      374 non-null    int64
 2   Occupation               374 non-null    object
 3   Sleep Duration           374 non-null    float64
 4   Quality of Sleep         374 non-null    int64
 5   Physical Activity Level  374 non-null    int64
 6   Stress Level             374 non-null    int64
 7   BMI Category             374 non-null    object
 8   Blood Pressure           374 non-null    object
 9   Heart Rate               374 non-null    int64
 10  Daily Steps              374 non-null    int64
 11  Sleep Disorder           374 non-null    object
dtypes: float64(1), int64(6), object(5)
memory usage: 35.2+ KB
```

[9]:

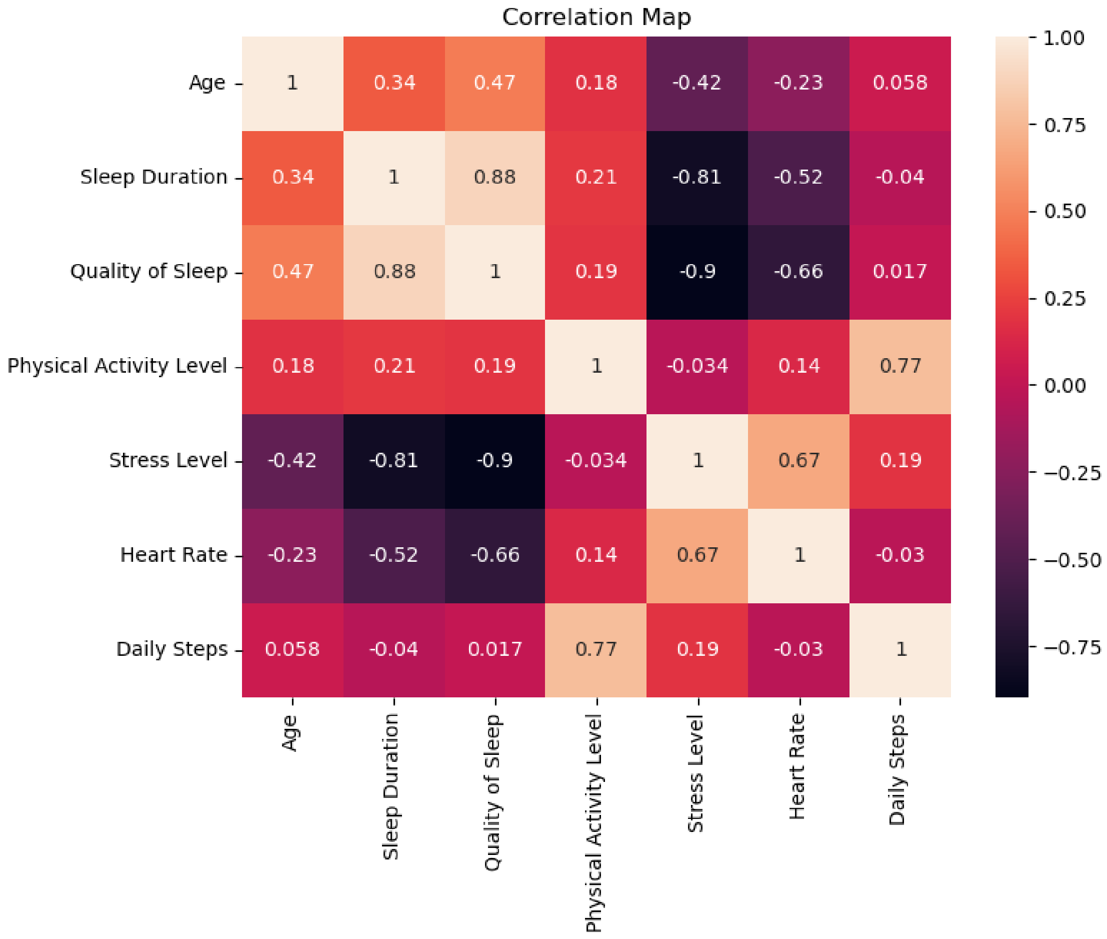
```
## Saving data fie
df.to_csv("Sleep health and Lifestyle", index=False)
```

[10]:

```
## Checking for outliers
sns.boxplot(df['Sleep Duration']);
```



```
[11]: ## Checking for correlation
      plt.figure(figsize=(8,6))
      sns.heatmap(df.corr(), annot=True)
      plt.title("Correlation Map")
      plt.show()
```



2 Explore Data Analysis

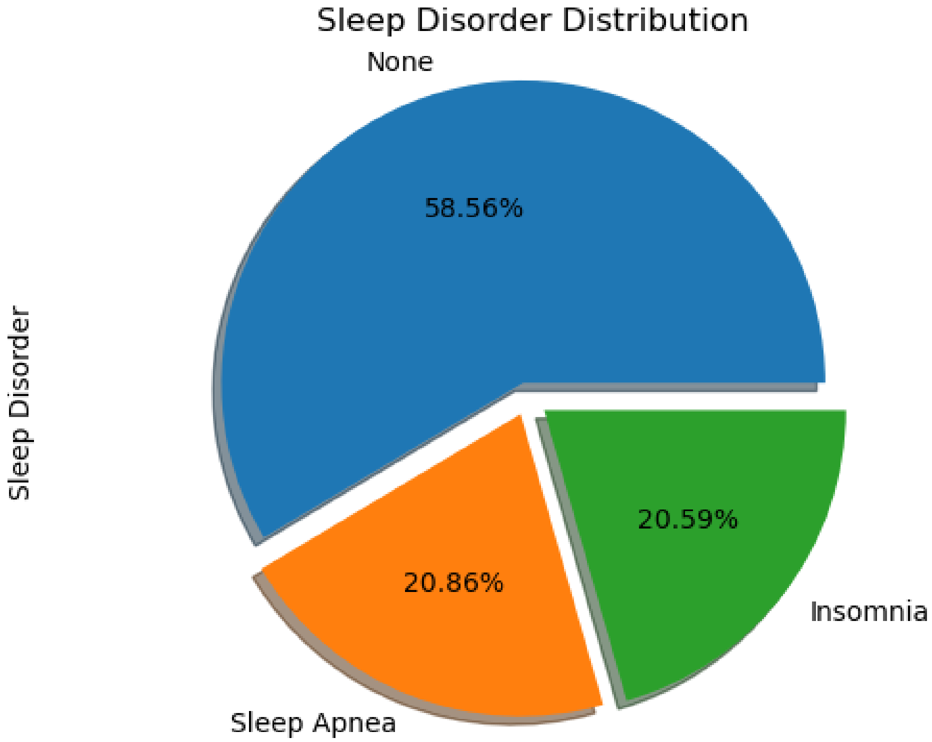
3 Questions Asked for data:

- Sleep Disorder Percentage:
- Gender Percentage in the Data using a pie chart:
- Distribution of Age using a histogram
- Determine the highest occupation in the dataset.
- Analyze the distribution of sleep duration based on gender.
- Visualize the average sleep duration across different occupations using a bar chart.
- Explore the relationship between average sleep duration and BMI category.
- Identify the dominant occupation within the male category.
- Find the Average Heart with Bmi category

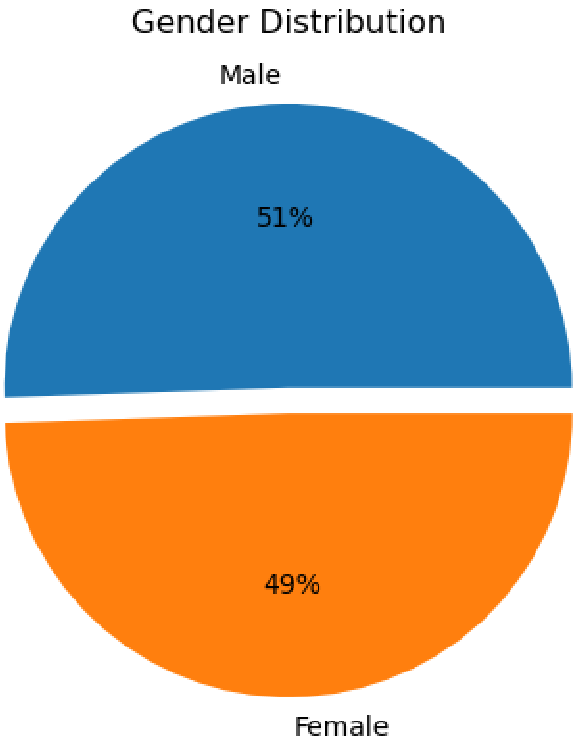
[12]:

```
## Percentage of sleep disorder
sleep_disorder_counts = df['Sleep Disorder'].value_counts(normalize=True)
```

```
sleep_disorder_counts.plot(kind='pie', autopct='%1.2f%%', explode=[0.05, 0.06,␣
↪0.07], shadow=True)
plt.title("Sleep Disorder Distribution")
plt.axis('equal')
plt.show()
```



[13]:
```
## finding the distribution of the ender in the dataset
plt.pie(x=df['Gender'].value_counts(),labels=df['Gender'].unique(), explode=[0.
↪05,0.04], autopct='%.0f%%')
plt.title("Gender Distribution")
plt.show()
```



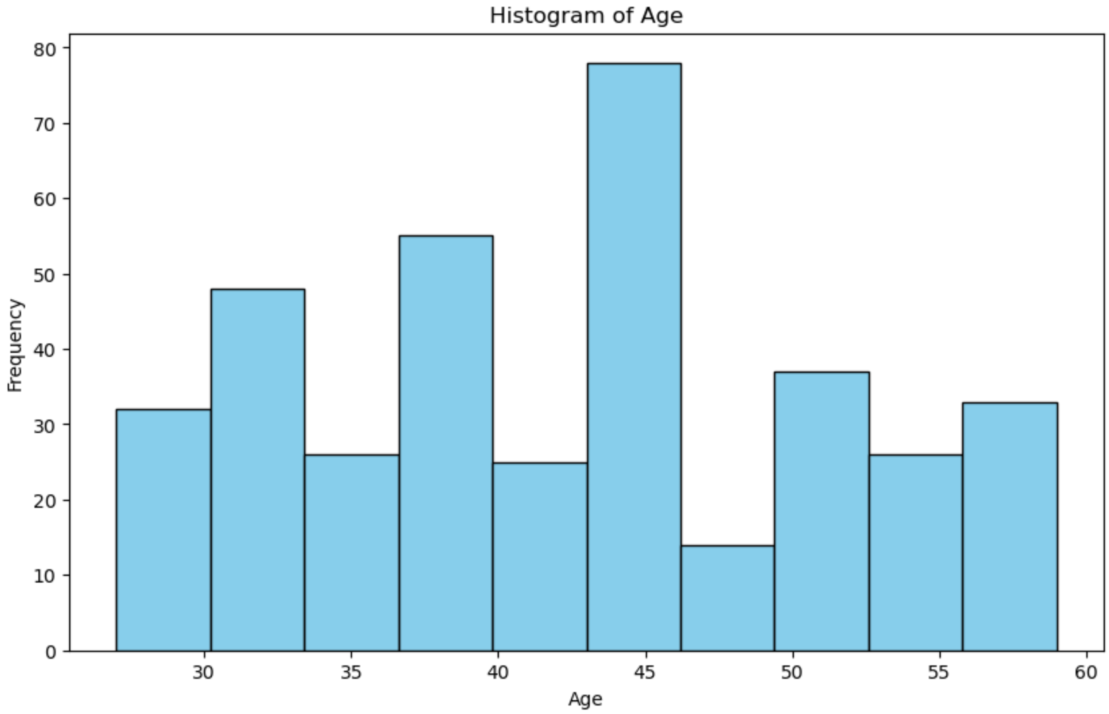
4 Observations

5 From the above two pie charts, we can observe several pieces of information:

- Firstly, the highest percentage in the sleep disorder pie chart is “None,” indicating that a significant portion of the data does not have reported sleep disorders.
- The second-highest sleep disorder category is “Sleep Apnea.”
- In the second pie chart depicting gender percentages, the male percentage is higher compared to the female percentage.

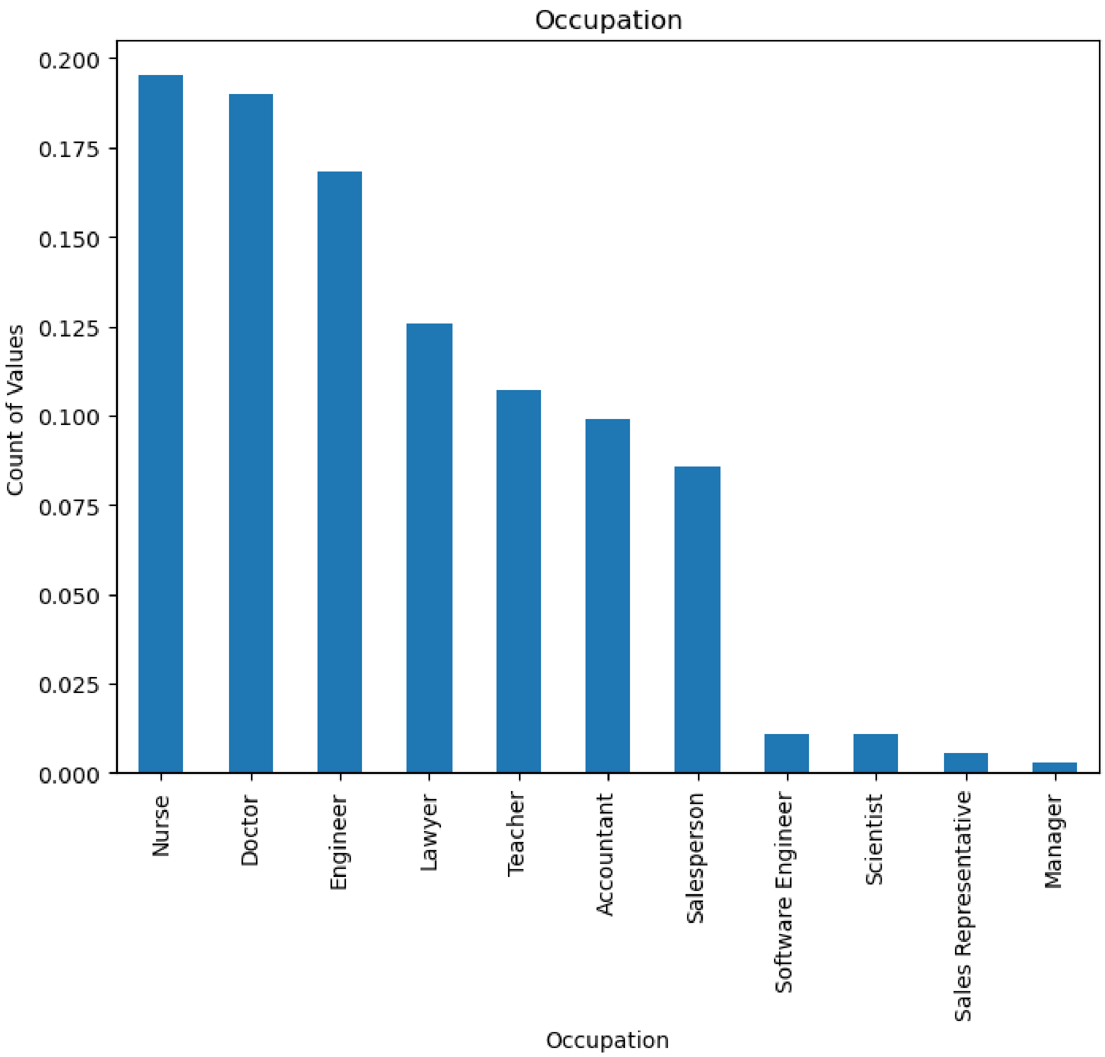
[14]:

```
#Distribution of the age columns
plt.figure(figsize=(10,6))
plt.hist(df['Age'], bins=10, color='skyblue', edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram of Age')
plt.show()
```



[15]:

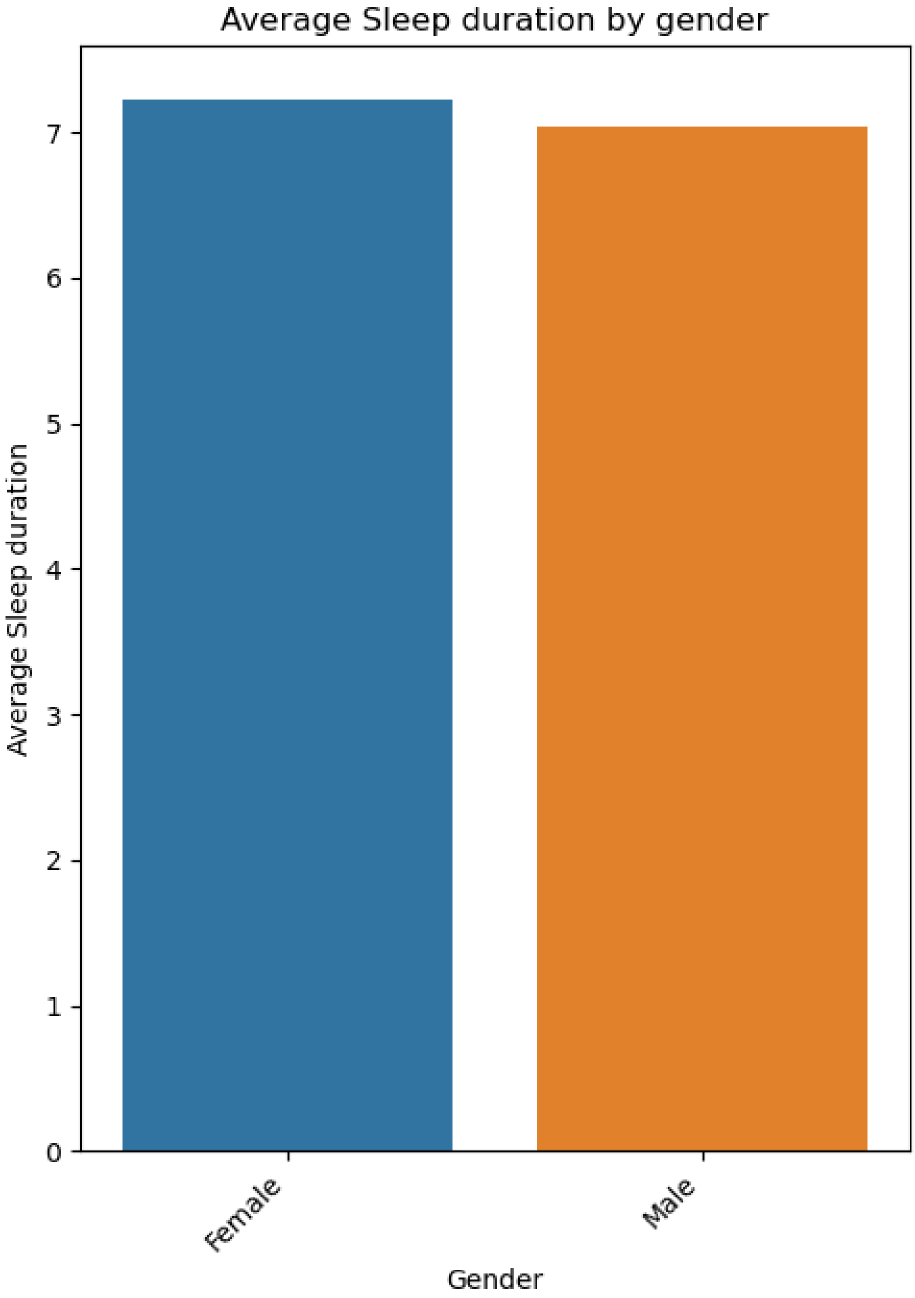
```
## visualizing the occupation distribution in the dataset
occupation_counts = df['Occupation'].value_counts(normalize=True)
occupation_counts.plot(kind='bar', title="Occupation", figsize=(8,6))
plt.xlabel('Occupation')
plt.ylabel('Count of Values')
plt.show()
```



6 Observations:

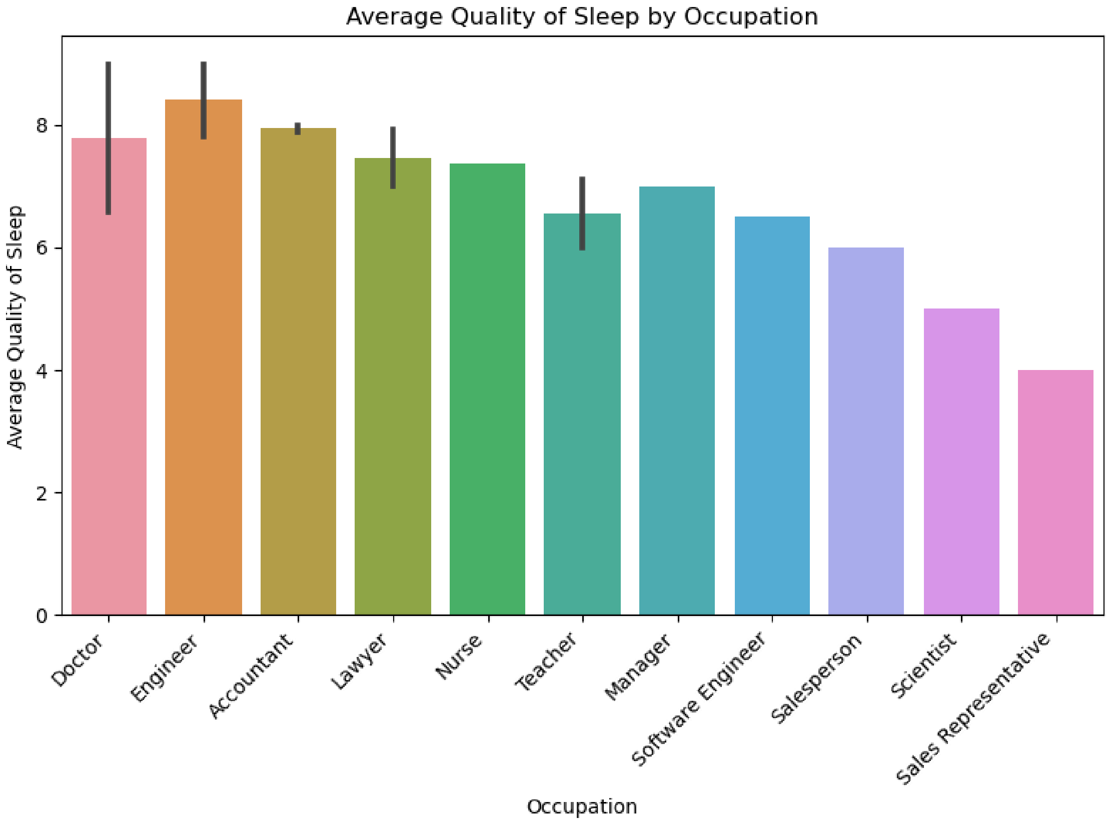
From the above two charts, one being a histogram and the other a bar chart, we can observe the following patterns: * In the age histogram, a significant number of individuals in the dataset fall within the age range 45. * In the bar chart, we can determine that the most job based on the dataset is “Nurse,” while the least job is “Manager.”

[16]:
```
## finding the average sleep time of the different gender
plt.figure(figsize=(5,7))
average_sleep_by_gender = df.groupby(['Gender'])['Sleep Duration'].mean().
  ↪sort_values(ascending=False).reset_index()
ax = sns.barplot(data=average_sleep_by_gender, x='Gender', y='Sleep Duration')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
plt.xlabel('Gender')
plt.ylabel('Average Sleep duration')
plt.title('Average Sleep duration by gender')
plt.tight_layout()
plt.show()
```



[17]:
```
## comparing the sleep time by occupation
plt.figure(figsize=(8,6))
```

```python
average_sleep_by_occupation = df.groupby(['Occupation', 'Gender'])['Quality of Sleep'].mean().sort_values(ascending=False).reset_index()
ax = sns.barplot(data=average_sleep_by_occupation, x='Occupation', y='Quality of Sleep')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
plt.xlabel('Occupation')
plt.ylabel('Average Quality of Sleep')
plt.title('Average Quality of Sleep by Occupation')
plt.tight_layout()
plt.show()
```



7 Observations:

From the two charts, we can analyze the average sleep duration based on gender and the quality of sleep based on different occupations.

- According to the bar chart, the average sleep duration for males is approximately 6.8 hours, while for females, it is around 7.5 hours.
- In the bar chart, it is evident that the occupation “Engineer” has the highest sleep quality among the different roles, while the occupation “Sales Representative” has the lowest sleep quality.

[18]:

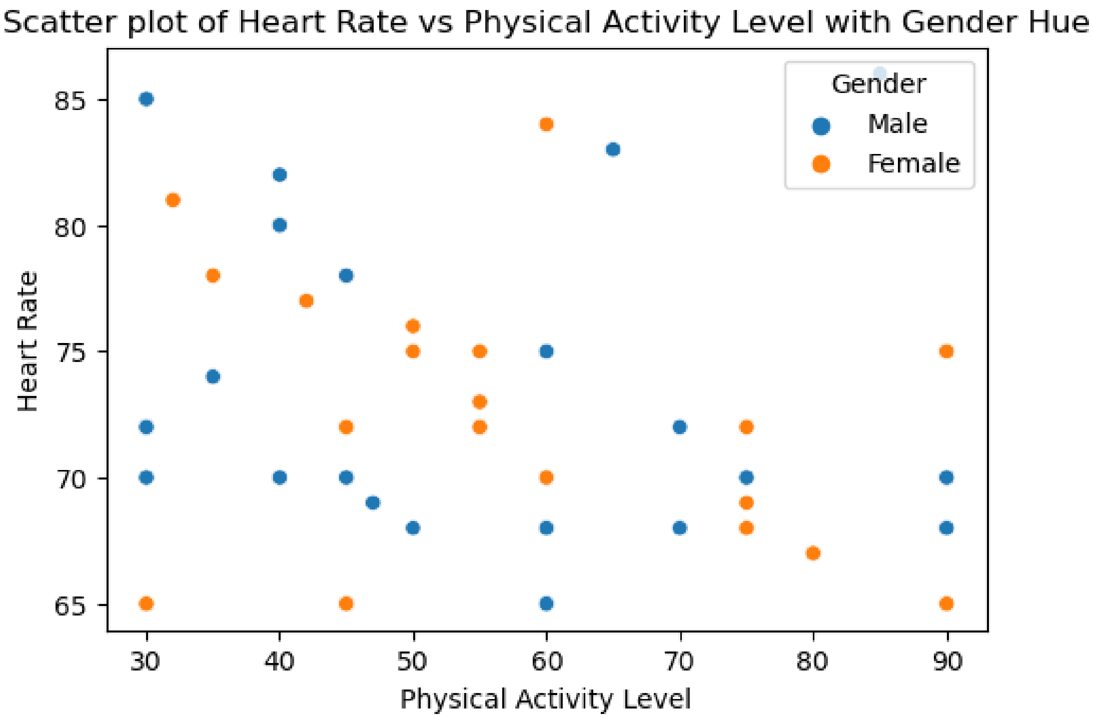
```
#Some intresting questions asked in the data
#Gender imbalance
gender_counts = df['Gender'].value_counts()
imbalanced_data = gender_counts['Male'] / gender_counts['Female']
print('\nGender Imbalance',imbalanced_data)
#find the which is the dominate_occupation in the data
dominate_occupation=df['Occupation'].value_counts().idxmax()
print('\nDominate_occupation',dominate_occupation)
#find the least demanding job
least_demanding_job=df['Occupation'].value_counts().idxmin()
print('\nLeast_demanding_job',least_demanding_job)
# find the top 5 strees level
top_5_stress_level=df['Stress Level'].value_counts().nlargest(5)
print('\nTop_5_stress_level',top_5_stress_level)
#find the age range in the data
age_range=(df['Age'].min(),df['Age'].max())
print('\nAge_range',age_range)
# find the daily steps range in the data
daily_steps=(df['Daily Steps'].min(),df['Daily Steps'].max())
print('\nDaily_steps',daily_steps)
# find the skewss of the sleep durations
sleep_quality_skewness=df['Sleep Duration'].value_counts().skew()
print('\nSleep_quality_skewness',sleep_quality_skewness)
```

```
Gender Imbalance 1.0216216216216216

Dominate_occupation Nurse

Least_demanding_job Manager

Top_5_stress_level 3    71
8    70
4    70
5    67
7    50
Name: Stress Level, dtype: int64

Age_range (27, 59)

Daily_steps (3000, 10000)

Sleep_quality_skewness 0.7855254005718885
```

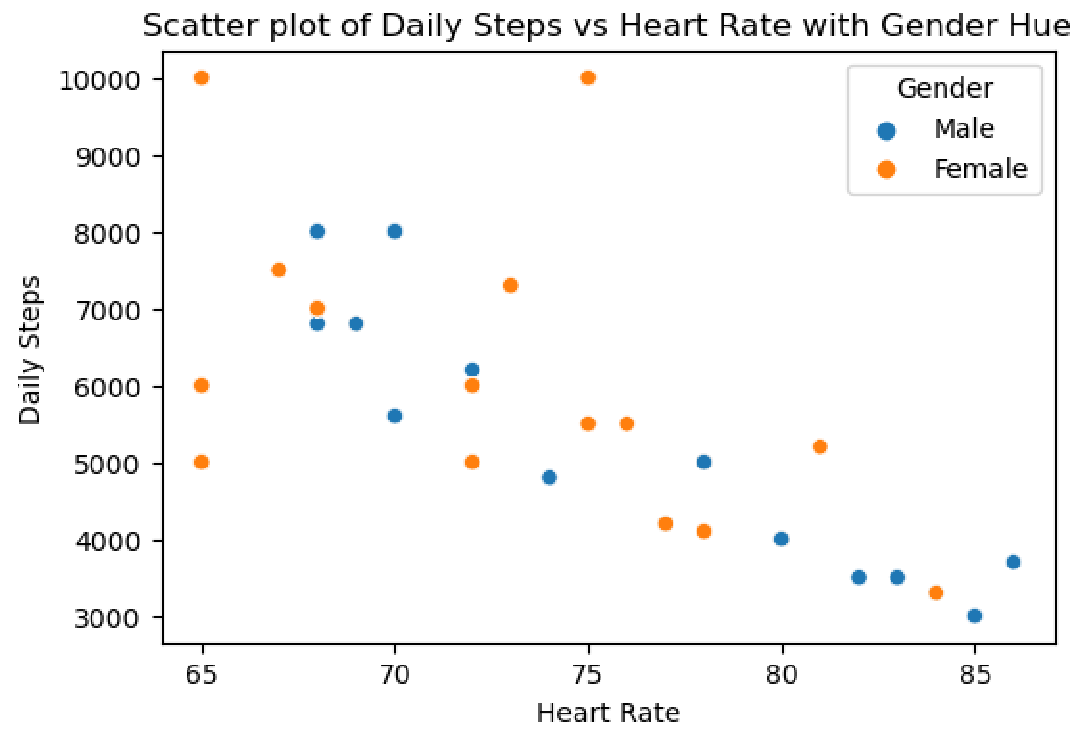
[19]:

```
##Visualize the Physical Activity with Heart Rate by Gender
var = 'Physical Activity Level'
## Plotting with Seaborn scatter plot
plt.figure(figsize=(6, 4))
```

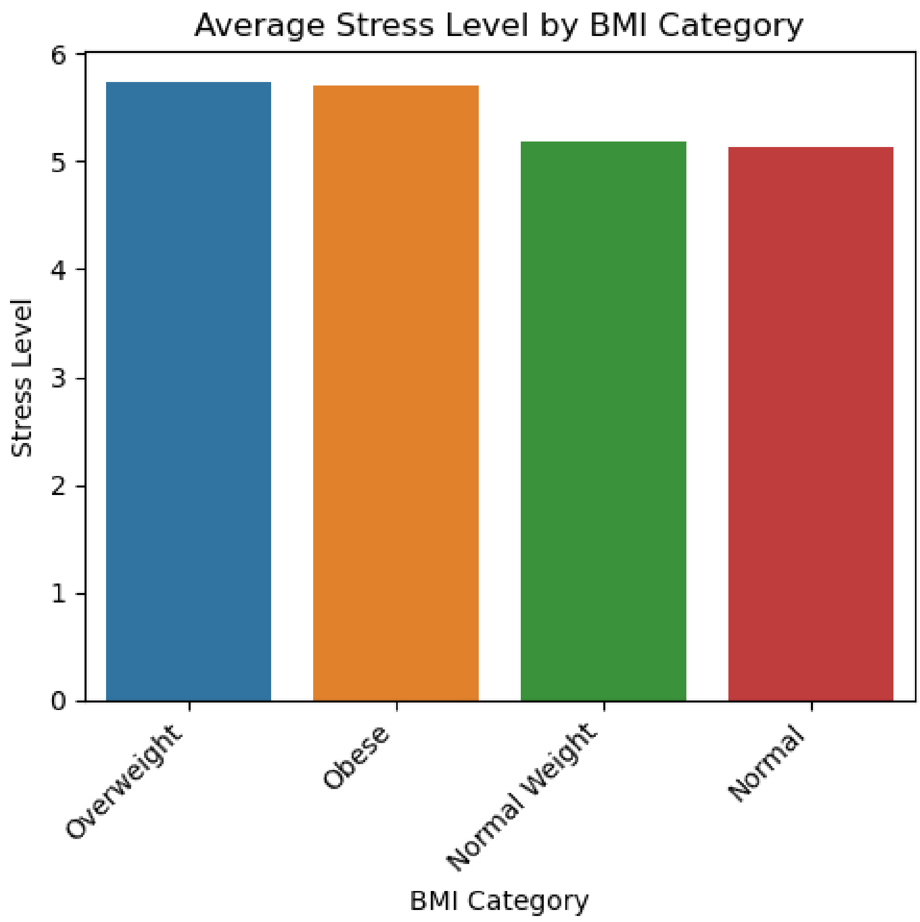
```python
sns.scatterplot(data=df, x=var, y='Heart Rate', hue='Gender')
plt.title(f'Scatter plot of Heart Rate vs {var} with Gender Hue')
plt.show()
```



[20]:
```python
## Find the Relationship between Heart Rate with Daily steps by gender
var = 'Heart Rate'
## Plotting with Seaborn scatter plot
plt.figure(figsize=(6, 4))
sns.scatterplot(data=df, x=var, y='Daily Steps', hue='Gender')
plt.title(f'Scatter plot of Daily Steps vs {var} with Gender Hue')
plt.show()
```



[21]:
```
##Finding the average Stress level by BMI Category
plt.figure(figsize=(5,5))
average_stress_by_BMI = df.groupby(['BMI Category'])['Stress Level'].mean().
 ↪sort_values(ascending=False).reset_index()
ax = sns.barplot(data=average_stress_by_BMI, x='BMI Category', y='Stress Level')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
plt.title('Average Stress Level by BMI Category')
plt.tight_layout()
plt.show()
```

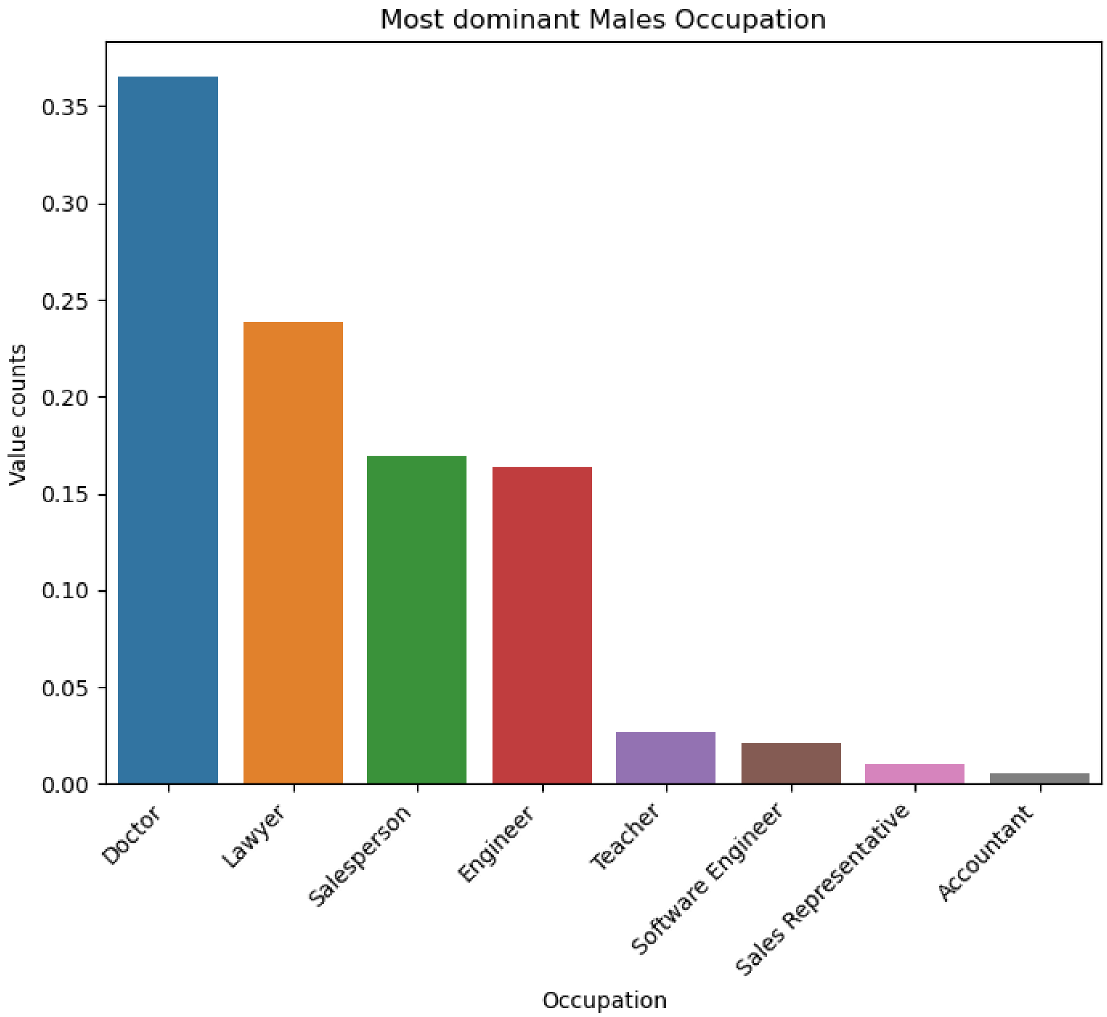


[22]:
```
##Finding the averages daily steps with BMI Category on Gender using pivot_table
pivot_table = pd.pivot_table(df, values='Daily Steps', index=['BMI␣
 ↪Category','Gender'], aggfunc='mean').style.
 ↪background_gradient(cmap='viridis')

# Display the pivot table
pivot_table
```

[22]:
```
<pandas.io.formats.style.Styler at 0x1f1c8af28e0>
```

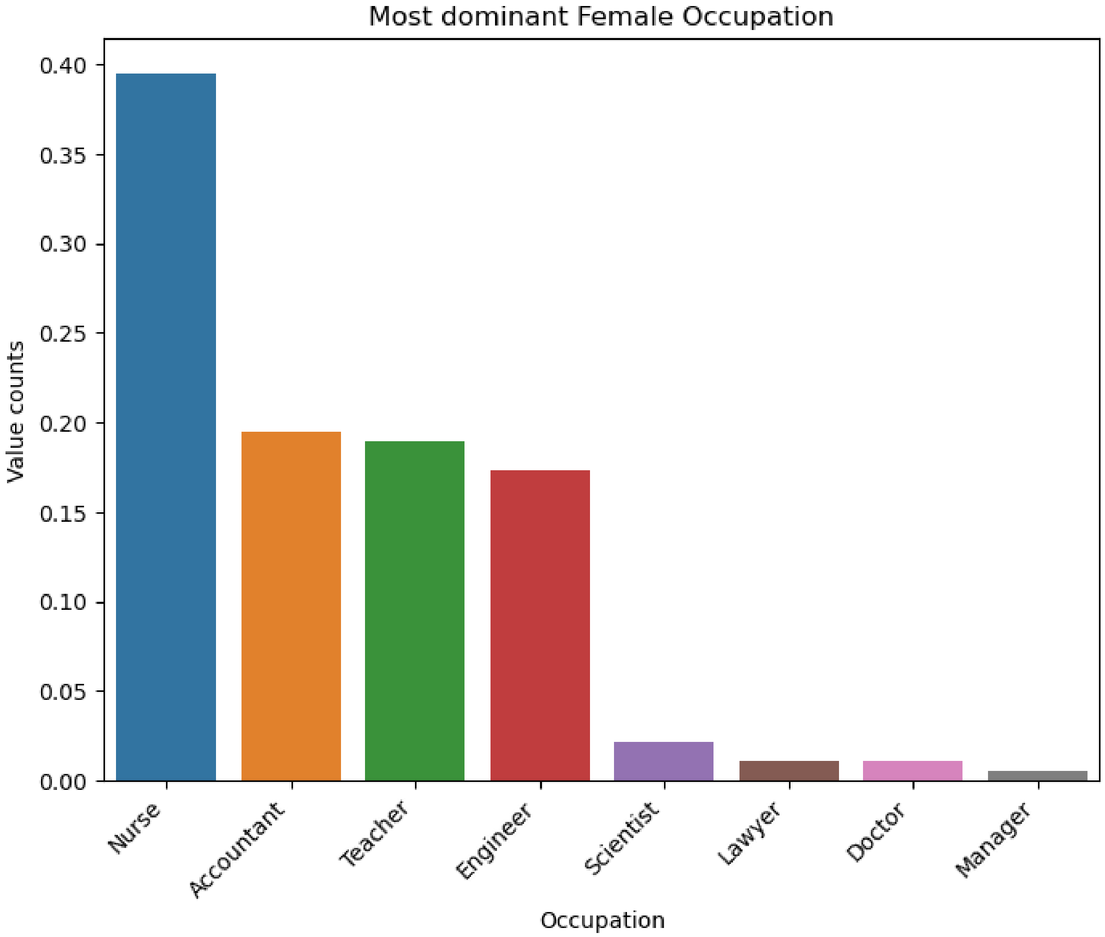
[23]:
```
## Finding the most dominant occupation in the male category fro the dataset

male_occupation = df[df['Gender'] == 'Male']['Occupation'].
 ↪value_counts(normalize=True)

# Reset the index and get the result as a DataFrame
male_occupation_df = male_occupation.reset_index()
```

```
# Plotting with Seaborn bar plot
plt.figure(figsize=(8, 6))
ax = sns.barplot(data=male_occupation_df, x='index', y='Occupation',␣
 ↪dodge=False)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
plt.title("Most dominant Males Occupation")
plt.xlabel('Occupation')
plt.ylabel('Value counts')
plt.show()
```



[24]:
```
## Finding the most dominant occupation in the female category from the dataset

female_occupation = df[df['Gender'] == 'Female']['Occupation'].
 ↪value_counts(normalize=True)
```

```
# Reset the index and get the result as a DataFrame
female_occupation_df = female_occupation.reset_index()

# Plotting with Seaborn bar plot
plt.figure(figsize=(8, 6))
ax = sns.barplot(data=female_occupation_df, x='index', y='Occupation',␣
 ↪dodge=False)
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right')
plt.title("Most dominant Female Occupation")
plt.xlabel('Occupation')
plt.ylabel('Value counts')
plt.show()
```

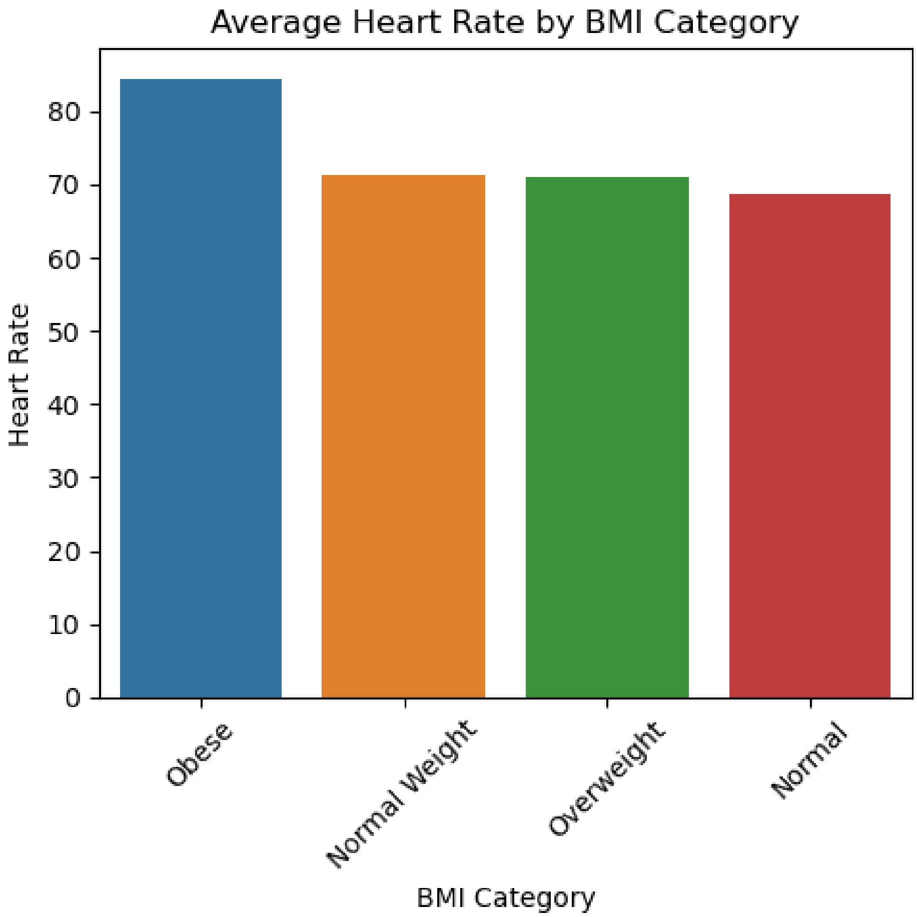


8 Observations:

- From the above chart, we can observe an interesting pattern where overweight people tend to have higher stress levels, while normal-weight individuals have lower stress levels.

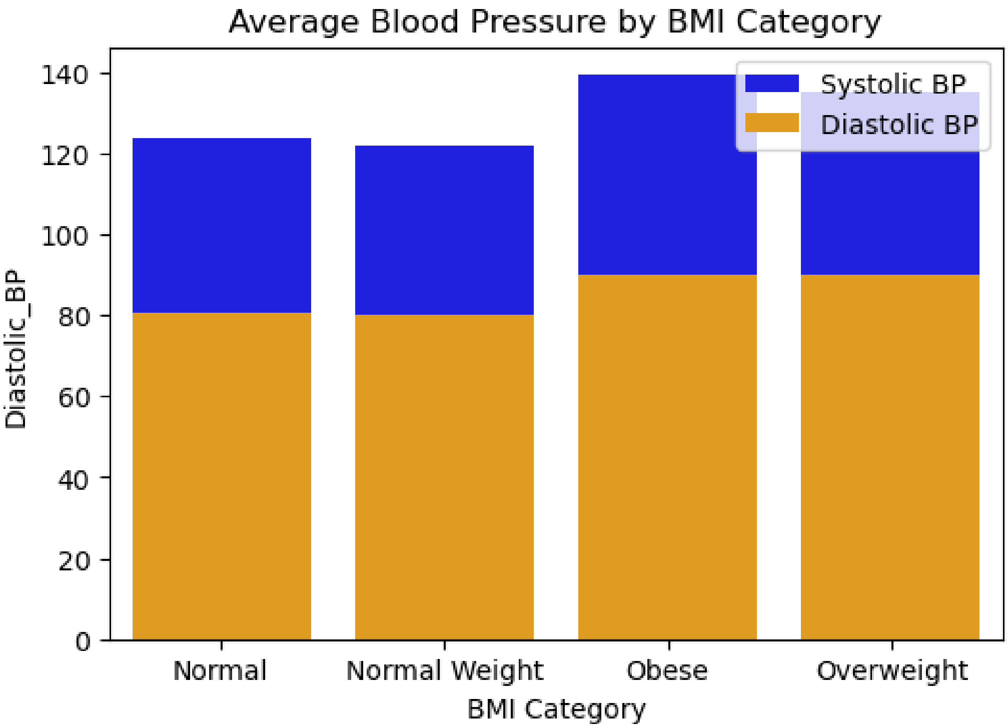
- It can be seen that the "Doctor" profession is the most dominating male job where as, "Nurse" profession is most dominating in the female category, while the "Accountant" and "Manager" are the least dominant within the genders respectively.

[25]:
```
## Comparing the average heart rate by the BMI category
plt.figure(figsize=(5,5))
average_HeartRate_by_BMI = df.groupby(['BMI Category'])['Heart Rate'].mean().
  ↪sort_values(ascending=False).reset_index()
ax = sns.barplot(data=average_HeartRate_by_BMI, x='BMI Category', y='Heart␣
  ↪Rate')
plt.xticks(rotation=45)
plt.title('Average Heart Rate by BMI Category')
plt.tight_layout()
plt.show()
```



[26]:
```
## Comparing the blood pressure by BMI Category

# Assuming 'Blood Pressure' column contains values like '125/80'
```

```python
# Extract systolic and diastolic values and convert them to numeric
df[['Systolic_BP', 'Diastolic_BP']] = df['Blood Pressure'].str.split('/', expand=True)
df['Systolic_BP'] = pd.to_numeric(df['Systolic_BP'], errors='coerce')
df['Diastolic_BP'] = pd.to_numeric(df['Diastolic_BP'], errors='coerce')

# Calculate average blood pressure by BMI category
average_BloodPressure_by_BMI = df.groupby(['BMI Category'])[['Systolic_BP', 'Diastolic_BP']].mean().reset_index()

# Plotting with Seaborn bar plot
plt.figure(figsize=(6, 4))
sns.barplot(data=average_BloodPressure_by_BMI, x='BMI Category', y='Systolic_BP', label='Systolic BP', color='blue')
sns.barplot(data=average_BloodPressure_by_BMI, x='BMI Category', y='Diastolic_BP', label='Diastolic BP', color='orange')
plt.title('Average Blood Pressure by BMI Category')
plt.legend()
plt.show()
```



9 Conclusion:

- It can be seen that the “Obese” have the highest heart rate while the lowest is “Normal”.
- The “Obese” tends to have a higher blood pressure than the other categories, followed by the “Overweight”.

This is another project in data science where data is obtained from Kaggle. The usual data preprocessing steps are performed, including data cleaning. Exploratory data analysis (EDA) techniques are applied to gain insights from the data, and questions are formulated based on the analysis. Visualizations such as bar charts, pie charts, and other types of charts are created to present the findings.