

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Load the data
data = pd.read_csv("/content/Raw_final_sleep_data.csv")
```

```
data
```

Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Cat
0	1	Male	27	Software Engineer	6.1	6	42	6 Over
1	2	Male	28	Doctor	6.2	6	60	8 Over
2	3	Male	28	Doctor	6.2	6	60	8 Over
3	4	Male	28	Sales Representative	5.9	4	30	8 Over
4	5	Male	28	Sales Representative	5.9	4	30	8 Over
...
554	555	Female	43	Teacher	6.7	7	45	4 Over
555	556	Male	43	Salesperson	6.5	6	45	7 Over
556	557	Female	43	Teacher	6.7	7	45	4 Over
557	558	Male	43	Salesperson	6.4	6	45	7 Over
558	559	Male	43	Salesperson	6.5	6	45	7 Over

550 rows × 9 columns

```
# Display the first few rows of the dataframe  
print(data.head())
```

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

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

	Blood Pressure	Heart Rate	Daily Steps	Sleep Disorder	
0	126/83	77	4200	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	

```
# Check for incorrect outliers and incorrect values  
print(data.describe())
```

	Person ID	Age	Sleep Duration	Quality of Sleep	\\
count	559.000000	559.000000	559.000000	559.000000	
mean	280.000000	39.987478	7.114132	7.271914	
std	161.513673	8.099616	0.742149	1.133293	
min	1.000000	27.000000	5.800000	4.000000	
25%	140.500000	33.000000	6.500000	6.000000	
50%	280.000000	38.000000	7.200000	7.000000	
75%	419.500000	44.000000	7.700000	8.000000	
max	559.000000	59.000000	8.500000	9.000000	

	Physical Activity Level	Stress Level	Heart Rate	Daily Steps	
count	559.000000	559.000000	559.000000	559.000000	
mean	58.758497	5.463327	70.182469	6820.751342	
std	19.961450	1.674711	3.875617	1528.814803	
min	30.000000	3.000000	65.000000	3000.000000	
25%	45.000000	4.000000	68.000000	5500.000000	
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	

For age, the participants range from 27 to 59 years old, with a mean age of approximately 40 years.

Sleep duration ranges from 5.8 to 8.5 hours, with a mean duration of around 7.1 hours.

Quality of sleep ranges from 4 to 9, with a mean rating of approximately 7.3.

Physical activity level ranges from 30 to 90 minutes per day, with a mean of around 58.8 minutes.

Stress level ranges from 3 to 8, with a mean rating of approximately 5.5.

Heart rate ranges from 65 to 86 beats per minute, with a mean of around 70.2 beats per minute.

Daily steps range from 3000 to 10000 steps, with a mean of approximately 6820.8 steps.

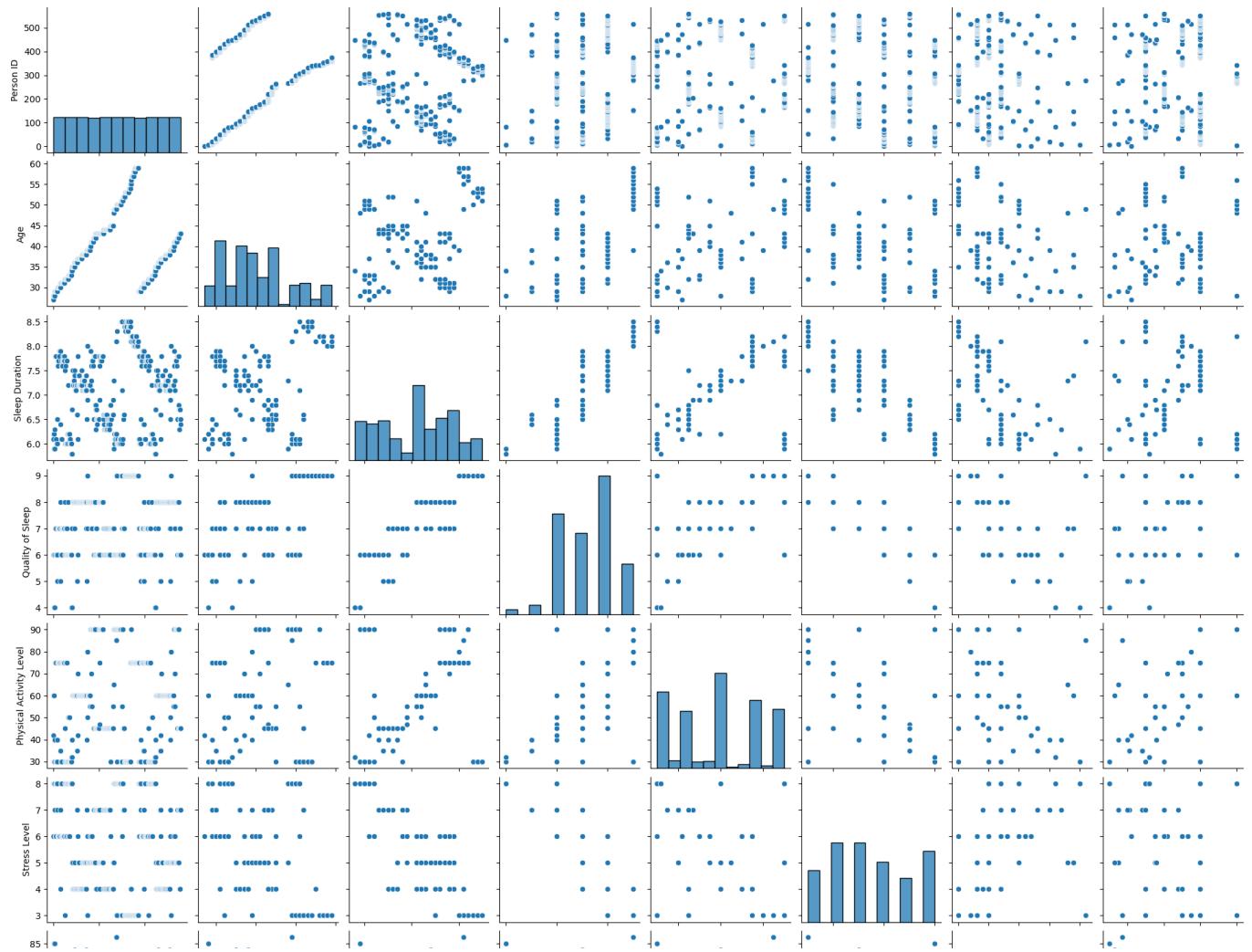
```
# Handle missing values
# For numeric columns, we can replace missing values with the mean or median
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Sleep Duration'].fillna(data['Sleep Duration'].median(), inplace=True)
data['Quality of Sleep'].fillna(data['Quality of Sleep'].median(), inplace=True)
data['Physical Activity Level'].fillna(data['Physical Activity Level'].median(),
data['Stress Level'].fillna(data['Stress Level'].median(), inplace=True)
data['Heart Rate'].fillna(data['Heart Rate'].median(), inplace=True)
data['Daily Steps'].fillna(data['Daily Steps'].median(), inplace=True)
```

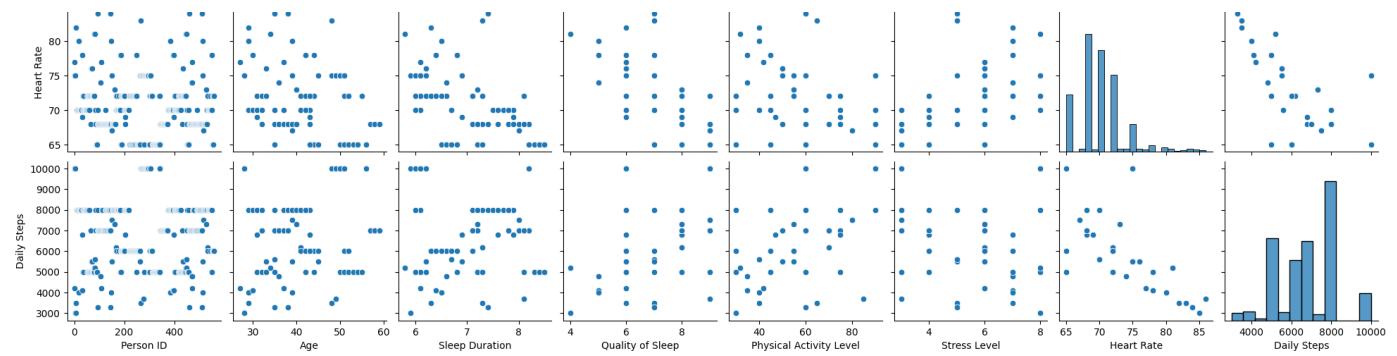
```
# For categorical columns, we can replace missing values with the mode
data['Gender'].fillna(data['Gender'].mode()[0], inplace=True)
data['Occupation'].fillna(data['Occupation'].mode()[0], inplace=True)
data['BMI Category'].fillna(data['BMI Category'].mode()[0], inplace=True)
data['Sleep Disorder'].fillna(data['Sleep Disorder'].mode()[0], inplace=True)
```

```
# Check if there are any remaining missing values  
print(data.isnull().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     0  
dtype: int64
```

```
# Visualize distributions and relationships between variables  
sns.pairplot(data)  
plt.show()
```





```
def fix_BMI (x):
    if x == 'Normal Weight':
        return 'Normal'
    return x

data['BMI Category'] = data['BMI Category'].apply(lambda x : fix_BMI(x))
```

```
# Data collection
data.insert(loc = 2, column = 'Age Group', value = pd.cut(x = data['Age'], bins=[
```

```
def blood_pressure_targets(bp):
    result = None
    sys, dias = map(int, bp.split('/'))
    if sys < 90 or dias < 60:
        result = 'Low'
    elif sys < 120 and dias < 80:
        result = 'Normal'
    elif sys in range(120, 130) and dias < 80:
        result = 'Elevated'
    elif sys in range(130, 140) or dias in range(80,90):
        result = 'Hypertension Stage 1'
    elif sys >= 140 or dias >= 90:
        result = 'Hypertension Stage 2'
    elif sys > 180 or dias > 120:
        result = 'Hypertensive Crisis'
    return result
```

```
def heart_rate_targets(hr):
    result = None
    if hr < 60:
        result = 'Bradycardia'
    elif hr <= 100:
        result = 'Normal'
    else:
        result = 'Tachycardia'
    return result
```

```

# Feature engineering
data_with_categories = data.assign(
    Blood_Pressure_Category=data['Blood Pressure'].apply(blood_pressure_targets),
    Heart_Rate_Category=data['Heart Rate'].apply(heart_rate_targets)
)

# Displaying the updated DataFrame
print(data_with_categories.head())

```

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

	Blood_Pressure_Category	Heart_Rate_Category	
0	Hypertension Stage 1	Normal	
1	Hypertension Stage 1	Normal	
2	Hypertension Stage 1	Normal	
3	Hypertension Stage 2	Normal	
4	Hypertension Stage 2	Normal	

```
data_with_categories
```

	Person ID	Gender	Age Group	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Lev
0	1	Male	Young Adults	27	Software Engineer	6.1	6	42	
1	2	Male	Young Adults	28	Doctor	6.2	6	60	
2	3	Male	Young Adults	28	Doctor	6.2	6	60	
3	4	Male	Young Adults	28	Sales Representative	5.9	4	30	
4	5	Male	Young Adults	28	Sales Representative	5.9	4	30	
...
554	555	Female	Middle-aged Adults	43	Teacher	6.7	7	45	
555	556	Male	Middle-aged Adults	43	Salesperson	6.5	6	45	
556	557	Female	Middle-aged Adults	43	Teacher	6.7	7	45	
557	558	Male	Middle-aged Adults	43	Salesperson	6.4	6	45	
558	559	Male	Middle-aged Adults	43	Salesperson	6.5	6	45	

559 rows × 16 columns

```
data_with_categories.to_csv('Cleaned_data.csv', index=False)
```

```
data_with_categories.describe()
```

	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate
count	559.000000	559.000000	559.000000	559.000000	559.000000	559.000000	559.000000
mean	280.000000	39.987478	7.114132	7.271914	58.758497	5.463327	70.18246
std	161.513673	8.099616	0.742149	1.133293	19.961450	1.674711	3.8756
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.00000
25%	140.500000	33.000000	6.500000	6.000000	45.000000	4.000000	68.00000
50%	280.000000	38.000000	7.200000	7.000000	60.000000	5.000000	70.00000
75%	419.500000	44.000000	7.700000	8.000000	75.000000	7.000000	72.00000

```
# Create individual box plots for each numeric variable
```

```
plt.figure(figsize=(12, 8))
```

```
plt.subplot(3, 3, 1)
```

```
sns.boxplot(x=data['Age'])
```

```
plt.subplot(3, 3, 2)
```

```
sns.boxplot(x=data['Sleep Duration'])
```

```
plt.subplot(3, 3, 3)
```

```
sns.boxplot(x=data['Quality of Sleep'])
```

```
plt.subplot(3, 3, 4)
```

```
sns.boxplot(x=data['Physical Activity Level'])
```

```
plt.subplot(3, 3, 5)
```

```
sns.boxplot(x=data['Stress Level'])
```

```
plt.subplot(3, 3, 6)
```

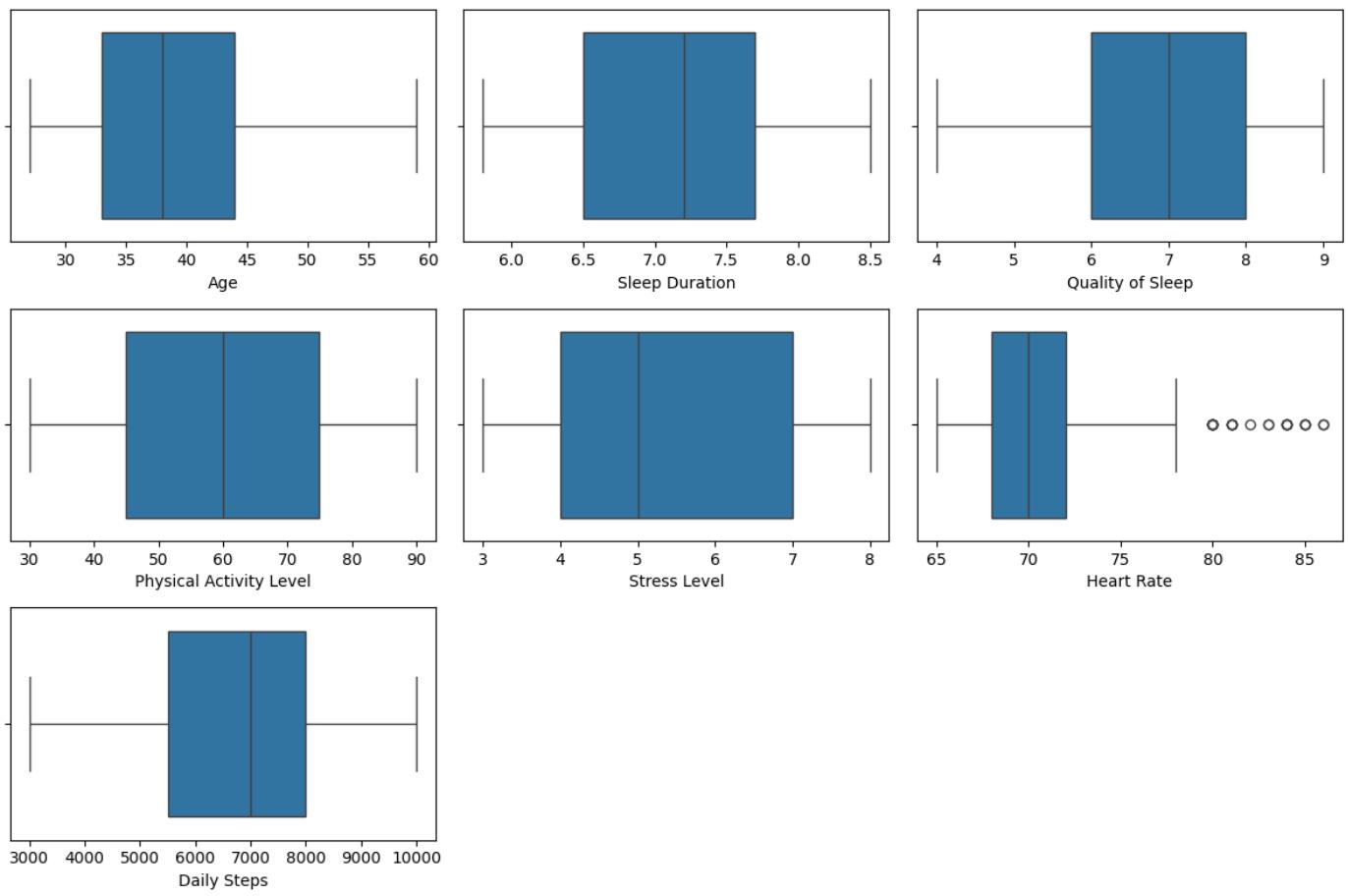
```
sns.boxplot(x=data['Heart Rate'])
```

```
plt.subplot(3, 3, 7)
```

```
sns.boxplot(x=data['Daily Steps'])
```

```
plt.tight_layout()
```

```
plt.show()
```



The series of individual box plots visualizes the distribution of each numeric variable in the dataset. Each box plot displays the median (line inside the box), quartiles (edges of the box), and potential outliers (points outside the whiskers) for the corresponding variable

```
data_with_categories.nunique()
```

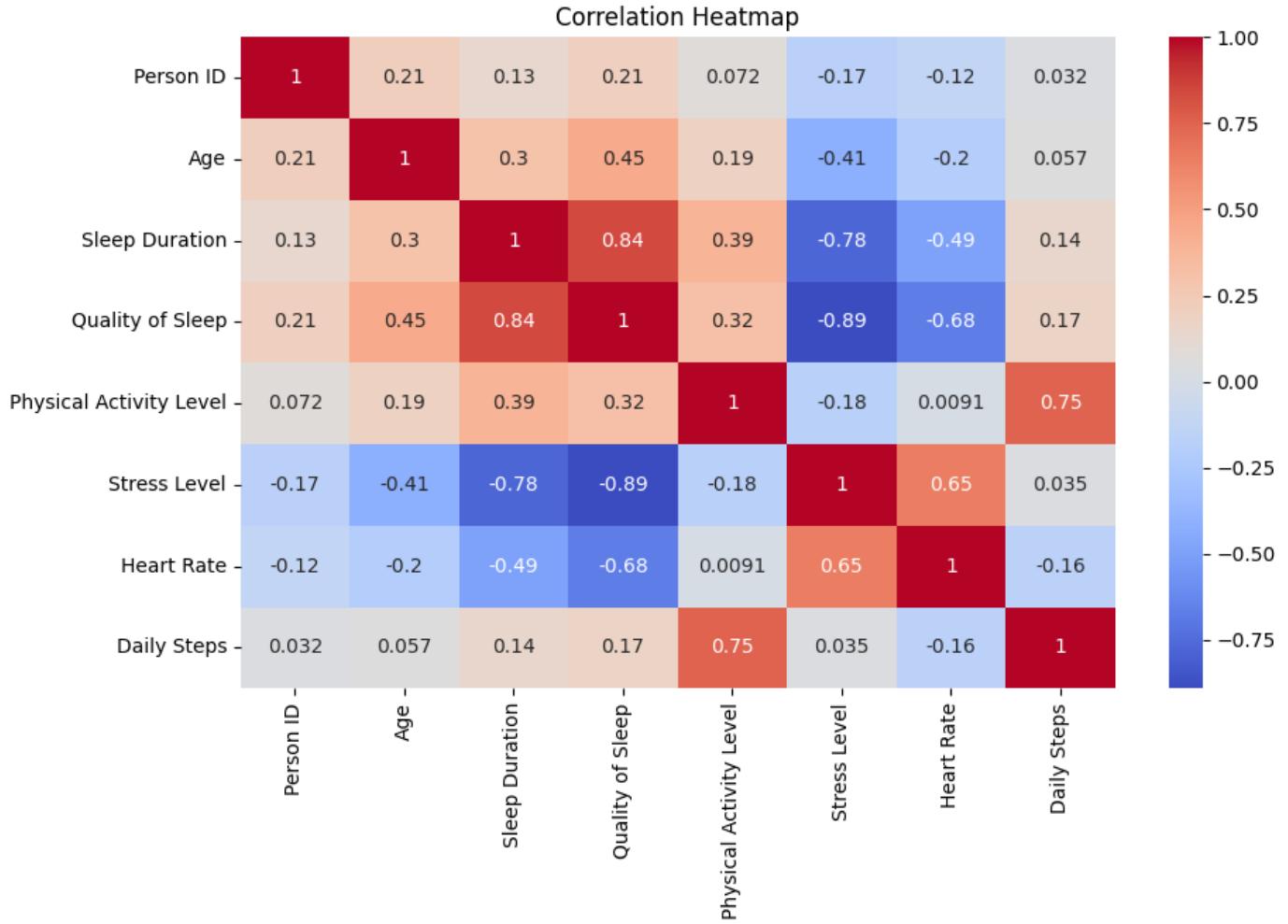
```
Person ID           559
Gender              2
Age Group           3
Age                 31
Occupation          11
Sleep Duration      27
Quality of Sleep     6
Physical Activity Level 16
Stress Level         6
BMI Category         3
Blood Pressure        25
Heart Rate            19
Daily Steps           20
Sleep Disorder        3
Blood_Pressure_Category 4
Heart_Rate_Category    1
dtype: int64
```

▼ EDA

```
# Viz1_Heatmap showing the correlation between numeric variables
plt.figure(figsize=(10, 6))
sns.heatmap(data_with_categories.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

<ipython-input-20-cc212b28536a>:3: FutureWarning: The default value of numerical

```
sns.heatmap(data_with_categories.corr(), annot=True, cmap='coolwarm')
```



The correlation heatmap visualizes the pairwise correlations between numeric variables in the dataset. Each cell in the heatmap represents the correlation coefficient between two variables, ranging from -1 to 1. A value close to 1 indicates a strong positive correlation, meaning that as one variable increases, the other variable tends to increase as well. Conversely, a value close to -1 indicates a strong negative correlation, meaning that as one variable increases, the other variable tends to decrease. A correlation coefficient close to 0 suggests a weak or no linear relationship between the variables.

```
#Violin plots
plt.figure(figsize=(12, 8))

plt.subplot(3, 3, 1)
sns.violinplot(x='Gender', y='Age', data=data)
plt.title('Age Distribution by Gender')

plt.subplot(3, 3, 2)
sns.violinplot(x='Gender', y='Sleep Duration', data=data)
plt.title('Sleep Duration Distribution by Gender')

plt.subplot(3, 3, 3)
sns.violinplot(x='Gender', y='Quality of Sleep', data=data)
plt.title('Quality of Sleep Distribution by Gender')

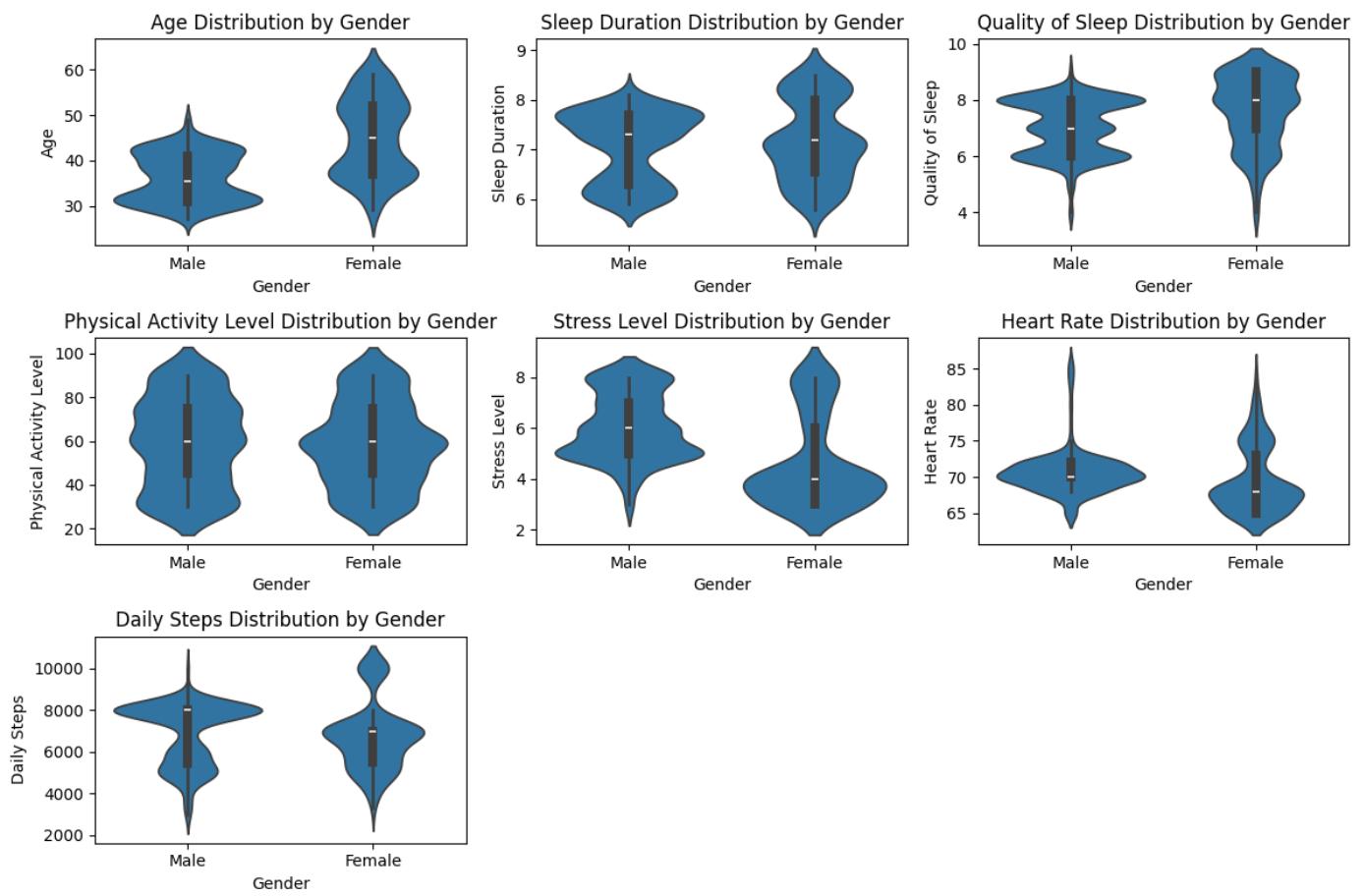
plt.subplot(3, 3, 4)
sns.violinplot(x='Gender', y='Physical Activity Level', data=data)
plt.title('Physical Activity Level Distribution by Gender')

plt.subplot(3, 3, 5)
sns.violinplot(x='Gender', y='Stress Level', data=data)
plt.title('Stress Level Distribution by Gender')

plt.subplot(3, 3, 6)
sns.violinplot(x='Gender', y='Heart Rate', data=data)
plt.title('Heart Rate Distribution by Gender')

plt.subplot(3, 3, 7)
sns.violinplot(x='Gender', y='Daily Steps', data=data)
plt.title('Daily Steps Distribution by Gender')

plt.tight_layout()
plt.show()
```



```
#Violin plots
plt.figure(figsize=(12, 8))

plt.subplot(3, 3, 1)
```

```
sns.violinplot(x='BMI Category', y='Age', data=data)
plt.title('Age Distribution by BMI Category')

plt.subplot(3, 3, 2)
sns.violinplot(x='BMI Category', y='Sleep Duration', data=data)
plt.title('Sleep Duration Distribution by BMI Category')

plt.subplot(3, 3, 3)
sns.violinplot(x='BMI Category', y='Quality of Sleep', data=data)
plt.title('Quality of Sleep Distribution by BMI Category')

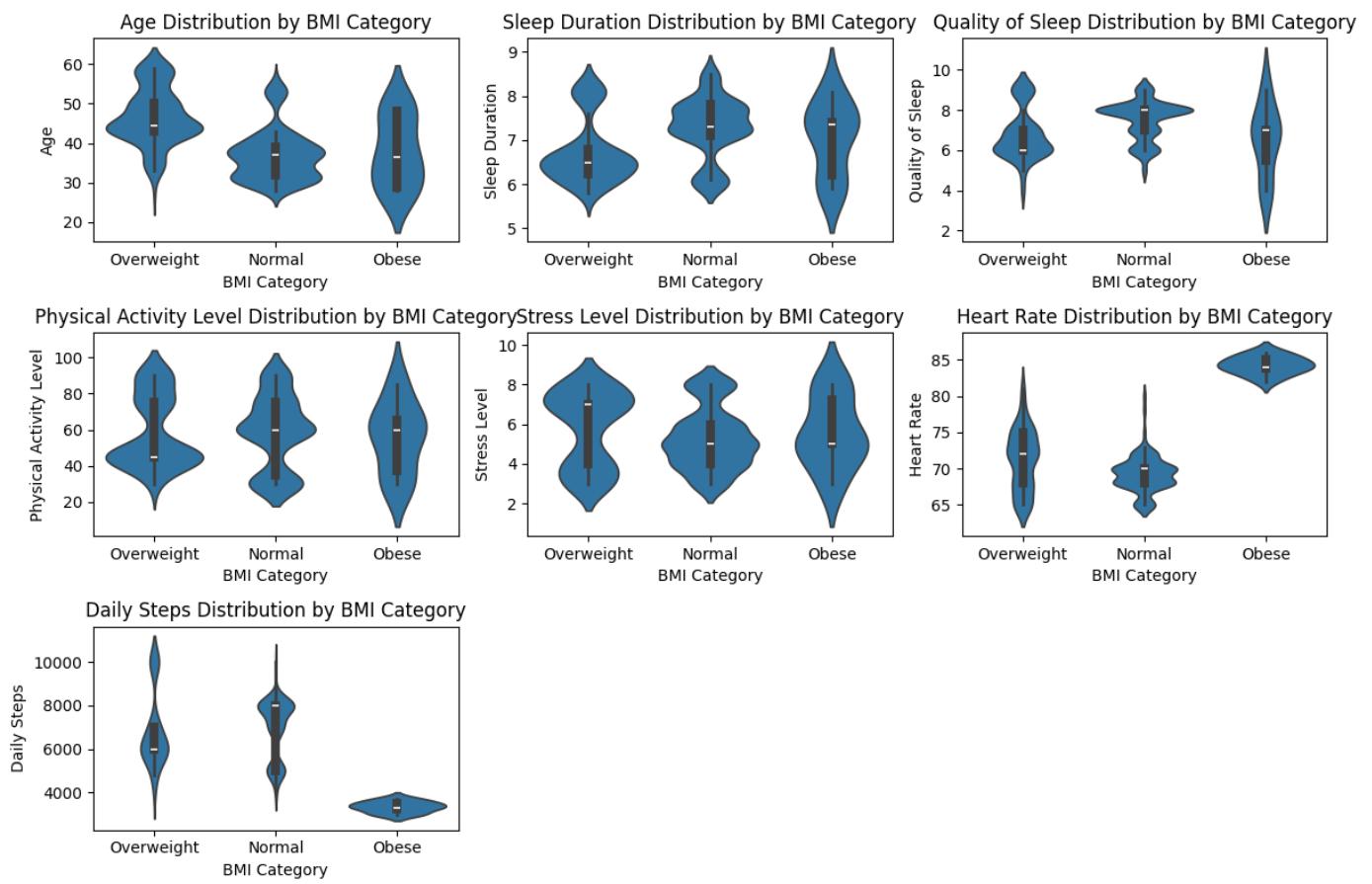
plt.subplot(3, 3, 4)
sns.violinplot(x='BMI Category', y='Physical Activity Level', data=data)
plt.title('Physical Activity Level Distribution by BMI Category')

plt.subplot(3, 3, 5)
sns.violinplot(x='BMI Category', y='Stress Level', data=data)
plt.title('Stress Level Distribution by BMI Category')

plt.subplot(3, 3, 6)
sns.violinplot(x='BMI Category', y='Heart Rate', data=data)
plt.title('Heart Rate Distribution by BMI Category')

plt.subplot(3, 3, 7)
sns.violinplot(x='BMI Category', y='Daily Steps', data=data)
plt.title('Daily Steps Distribution by BMI Category')

plt.tight_layout()
plt.show()
```



```
# Violin plots
plt.figure(figsize=(12, 8))

plt.subplot(3, 3, 1)
```

```
sns.violinplot(x='Sleep Disorder', y='Age', data=data)
plt.title('Age Distribution by Sleep Disorder')

plt.subplot(3, 3, 2)
sns.violinplot(x='Sleep Disorder', y='Sleep Duration', data=data)
plt.title('Sleep Duration Distribution by Sleep Disorder')

plt.subplot(3, 3, 3)
sns.violinplot(x='Sleep Disorder', y='Quality of Sleep', data=data)
plt.title('Quality of Sleep Distribution by Sleep Disorder')

plt.subplot(3, 3, 4)
sns.violinplot(x='Sleep Disorder', y='Physical Activity Level', data=data)
plt.title('Physical Activity Level Distribution by Sleep Disorder')

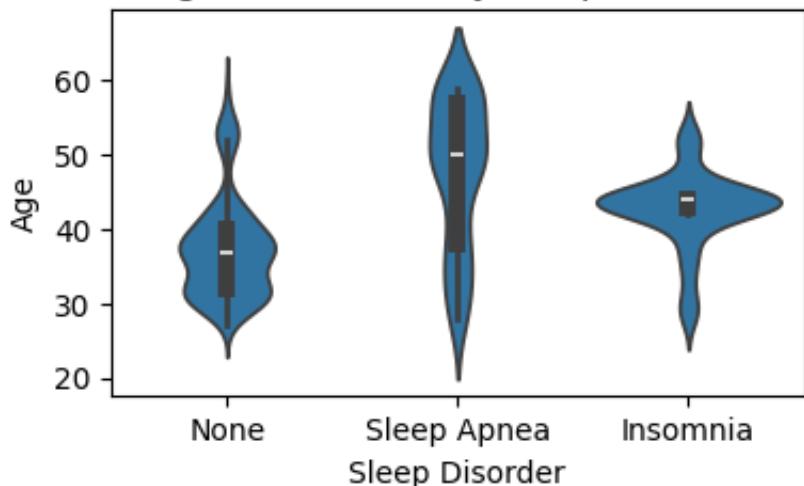
plt.subplot(3, 3, 5)
sns.violinplot(x='Sleep Disorder', y='Stress Level', data=data)
plt.title('Stress Level Distribution by Sleep Disorder')

plt.subplot(3, 3, 6)
sns.violinplot(x='Sleep Disorder', y='Heart Rate', data=data)
plt.title('Heart Rate Distribution by Sleep Disorder')

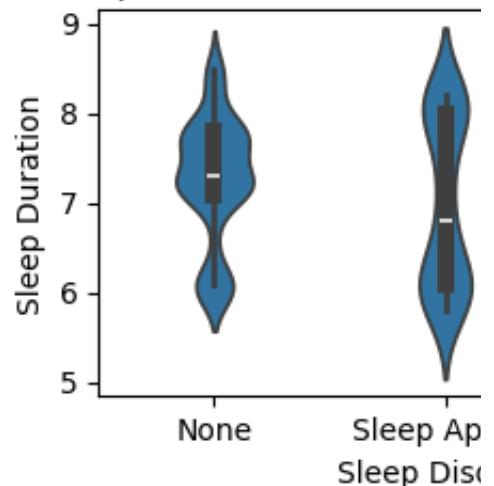
plt.subplot(3, 3, 7)
sns.violinplot(x='Sleep Disorder', y='Daily Steps', data=data)
plt.title('Daily Steps Distribution by Sleep Disorder')

plt.tight_layout()
plt.show()
```

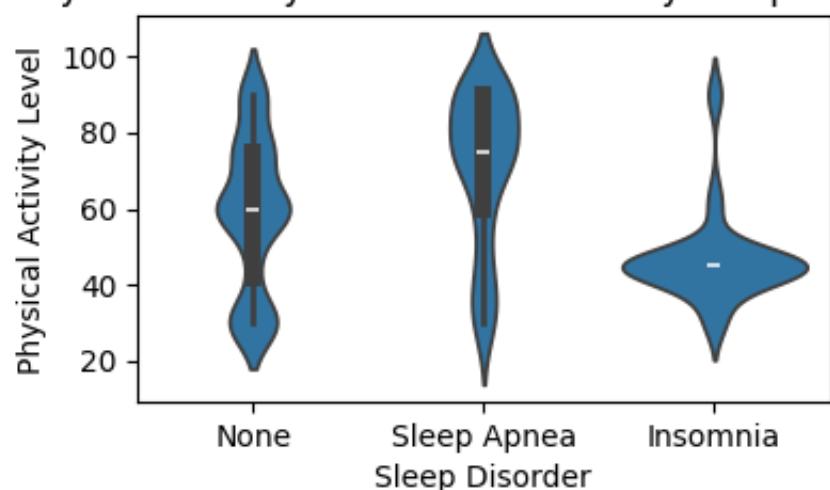
Age Distribution by Sleep Disorder



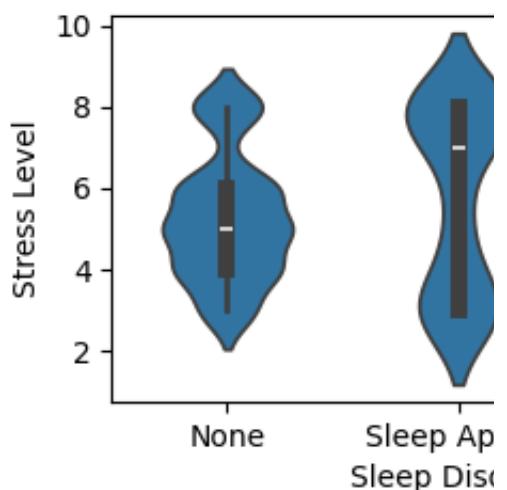
Sleep Duration Distribution



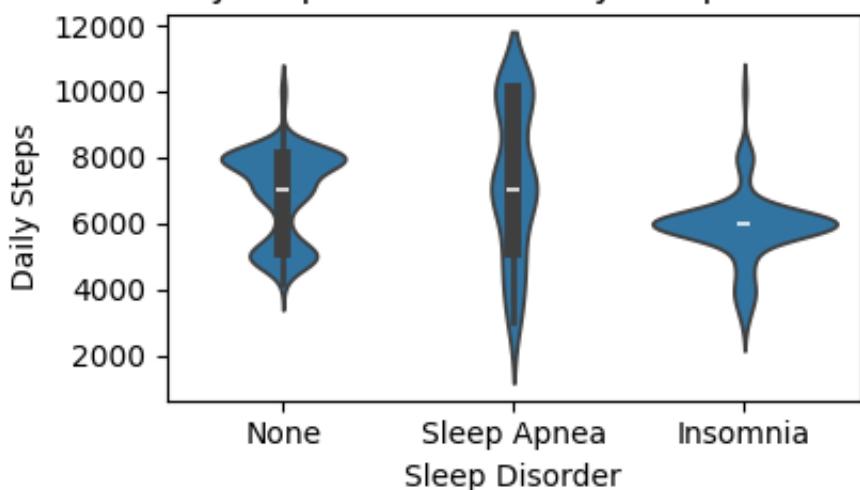
Physical Activity Level Distribution by Sleep Disorder



Stress Level Distribution

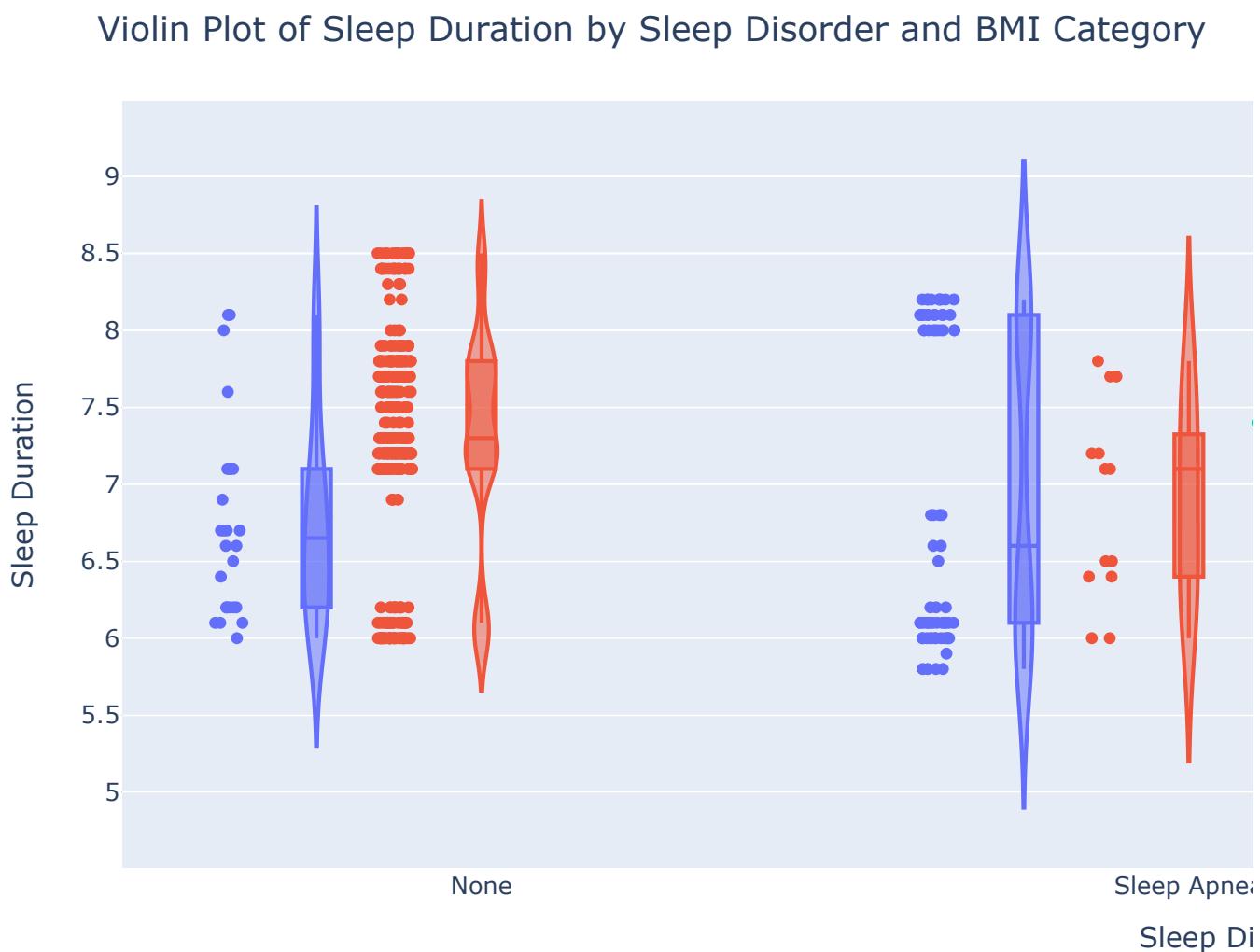


Daily Steps Distribution by Sleep Disorder



The series of violin plots visualize the distribution of each numeric variable within different categories of three categorical variables: 'Gender', 'BMI Category', and 'Sleep Disorder'. Each violin plot displays the kernel density estimation of the distribution, providing insights into the central tendency, spread, and shape of the data distribution within each category. In the 'Sleep Disorder' category, we observe variations in age, sleep duration, and quality of sleep distributions among participants with different sleep disorders, highlighting the importance of understanding sleep patterns and disorders in overall well-being.

```
import plotly.express as px
# Violin plot using Plotly
fig = px.violin(data, x='Sleep Disorder', y='Sleep Duration', color='BMI Category'
fig.update_layout(title='Violin Plot of Sleep Duration by Sleep Disorder and BMI
                     xaxis_title='Sleep Disorder',
                     yaxis_title='Sleep Duration')
fig.show()
```



The violin plot effectively visualizes the relationship between sleep duration, sleep disorder, and BMI category, highlighting potential trends and differences in sleep patterns across different subgroups of the population

```
import plotly.graph_objs as go

# Creating 3D pie chart
fig = go.Figure(data=[go.Pie(labels=data_with_categories['Blood_Pressure_Category'],
                               values=data_with_categories['Percentage'],
                               hole_size=0.5,
                               title='Blood Pressure Category Distribution',
                               scene=dict(aspectratio=[1, 1, 1]))])

# Updating layout
fig.update_layout(title='Blood Pressure Category Distribution', scene=dict(aspectratio=[1, 1, 1]))
fig.show()
```

Blood Pressure Category Distribution

