# Health Monitoring and Analysis

Health Monitoring and Analysis refers to the systematic use of health data to track and evaluate the health status of individuals or populations over time. It contains a range of activities from real-time physiological data collection (like heart rate, blood pressure, and temperature) to the analysis of more complex health records (including patient history, lifestyle choices, and genetic information).

# Health Monitoring and Analysis: Getting Started

The dataset we are working with contains the following columns:

1. PatientID: Numerical identifier for the patient.
2. Age: Age of the patient in years.
3. Gender: Gender of the patient.
4. HeartRate: Heart rate in beats per minute.
5. BloodPressure: Blood pressure readings, formatted inconsistently.
6. RespiratoryRate: Respiratory rate in breaths per minute.
7. BodyTemperature: Body temperature in Fahrenheit.
8. ActivityLevel: Activity level at the time of the measurement.
9. OxygenSaturation: Oxygen saturation percentage.
10. SleepQuality: Quality of sleep reported by the patient.
11. StressLevel: Reported level of stress.
12. Timestamp: Date and time of the measurement.

For the task of Health Monitoring and Analysis, we will aim to monitor the health of the patients in the data, analyze the patterns found in different types of patients and group them based on their health standards.

Now, let’s get started with the task of Health Monitoring and Analysis by importing the necessary Python libraries and the dataset:

In [14]:
```python
import warnings
warnings.filterwarnings("ignore")
```

In [1]:
```python
import pandas as pd
health_data= pd.read_csv('healthmonitoring.csv')
print(health_data.head())
```

```
   PatientID  Age  Gender  HeartRate BloodPressure  RespiratoryRate  \
0          1   69    Male  60.993428        130/85               15   
1          2   32    Male  98.723471        120/80               23   
2          3   78  Female  82.295377        130/85               13   
3          4   38  Female  80.000000        111/78               19   
4          5   41    Male  87.531693        120/80               14   

   BodyTemperature ActivityLevel  OxygenSaturation SleepQuality StressLevel  \
0        98.885236       resting              95.0    excellent         low   
1        98.281883       walking              97.0         good        high   
2        98.820286       resting              98.0         fair        high   
3        98.412594       running              98.0         poor    moderate   
4        99.369871       resting              98.0         good         low   

                    Timestamp  
0  2024-04-26 17:28:55.286711  
1  2024-04-26 17:23:55.286722  
2  2024-04-26 17:18:55.286726  
3  2024-04-26 17:13:55.286728  
4  2024-04-26 17:08:55.286731  
```

In [2]:
```python
#Let's have a look at whether the data contains any null values or not:
health_data.isnull().sum()
```

Out[2]:
```
PatientID             0
Age                   0
Gender                0
HeartRate             0
BloodPressure         0
RespiratoryRate       0
BodyTemperature      18
ActivityLevel         0
OxygenSaturation    163
SleepQuality          0
StressLevel           0
Timestamp             0
dtype: int64
```

So, the data contains null values in body temperature and oxygen saturation columns. for simplicity, I'll fill the values using the median value:

In [3]:
```python
# calculate medians
median_body_temp= health_data['BodyTemperature'].median()
median_oxygen_sat= health_data['OxygenSaturation'].median()

#fill missing values
health_data['BodyTemperature'].fillna(median_body_temp, inplace= True)
health_data['OxygenSaturation'].fillna(median_oxygen_sat, inplace= True)
```

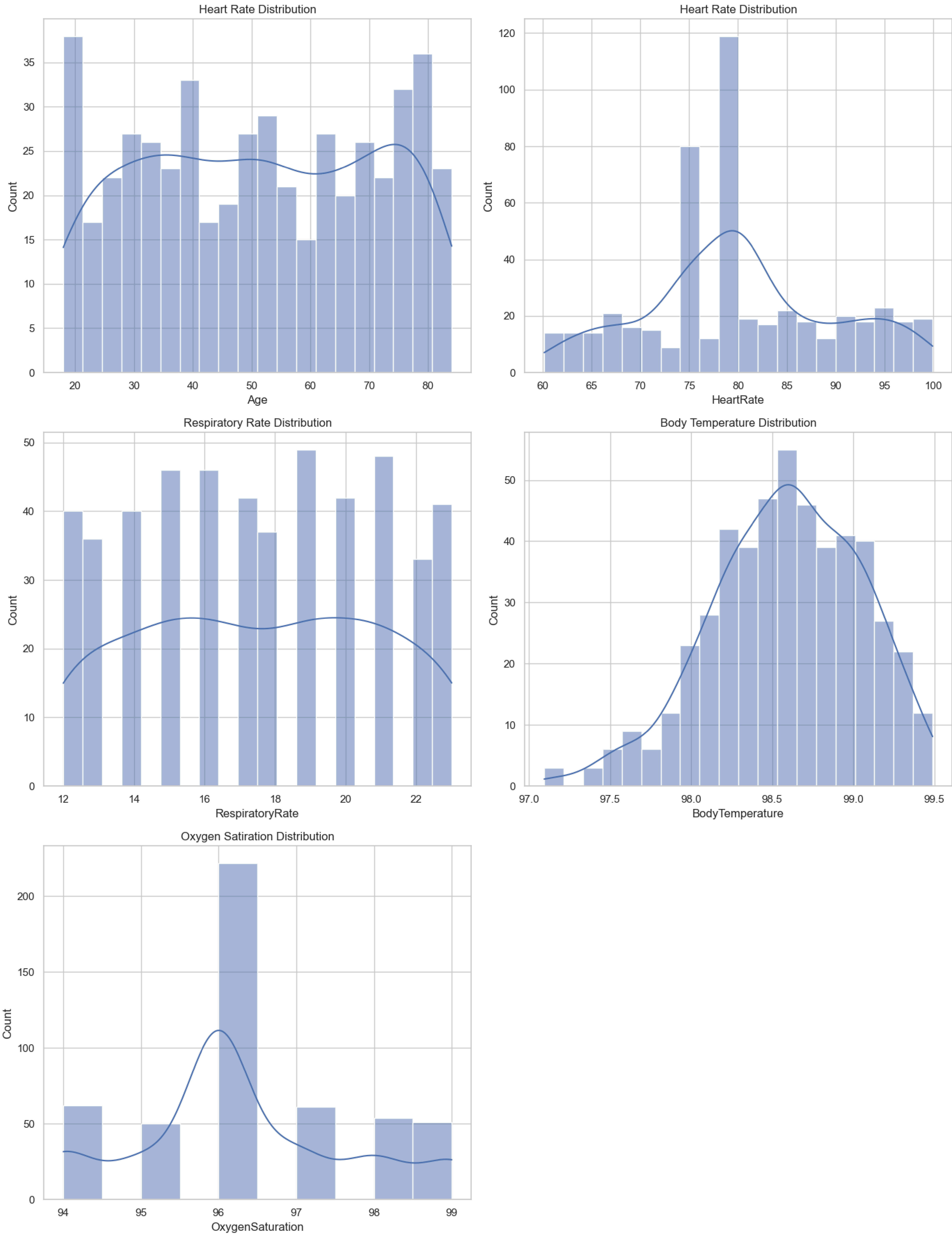
In [4]:
```python
# to check again the missing values
health_data.isnull().sum()
```

Out[4]:
```
PatientID           0
Age                 0
Gender              0
HeartRate           0
BloodPressure       0
RespiratoryRate     0
BodyTemperature     0
ActivityLevel       0
OxygenSaturation    0
SleepQuality        0
StressLevel         0
Timestamp           0
dtype: int64
```

Next, we will examine summary statistics and the distribution of the numerical health metrics (Age, Heart Rate, Respiratory Rate, Body Temperature, and Oxygen Saturation). It will help us to undersatand the typical values and the spread of the data. I'll also include some visualization to better understand these distributions:

In [15]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='whitegrid')

#summery statistics
summary_stats= health_data.describe()

#plotting distribution of numerical features
fig, axes= plt.subplots(3,2 ,figsize= (14, 18))
sns.histplot(health_data['Age'], bins=20, kde= True, ax=axes[0,0])
```

```python
axes[0,0].set_title('Heart Rate Distribution')

sns.histplot(health_data['HeartRate'], bins=20, kde= True, ax=axes[0,1])
axes[0,1].set_title('Heart Rate Distribution')

sns.histplot(health_data['RespiratoryRate'], bins=20, kde=True, ax=axes[1, 0])
axes[1, 0].set_title('Respiratory Rate Distribution')

sns.histplot(health_data['BodyTemperature'], bins=20, kde=True, ax=axes[1, 1])
axes[1, 1].set_title('Body Temperature Distribution')

sns.histplot(health_data['OxygenSaturation'],bins=10, kde= True, ax= axes[2,0])
axes[2,0].set_title('Oxygen Satiration Distribution')

fig.delaxes(axes[2,1]) # removw unused subplot
plt.tight_layout()
plt.show()
```

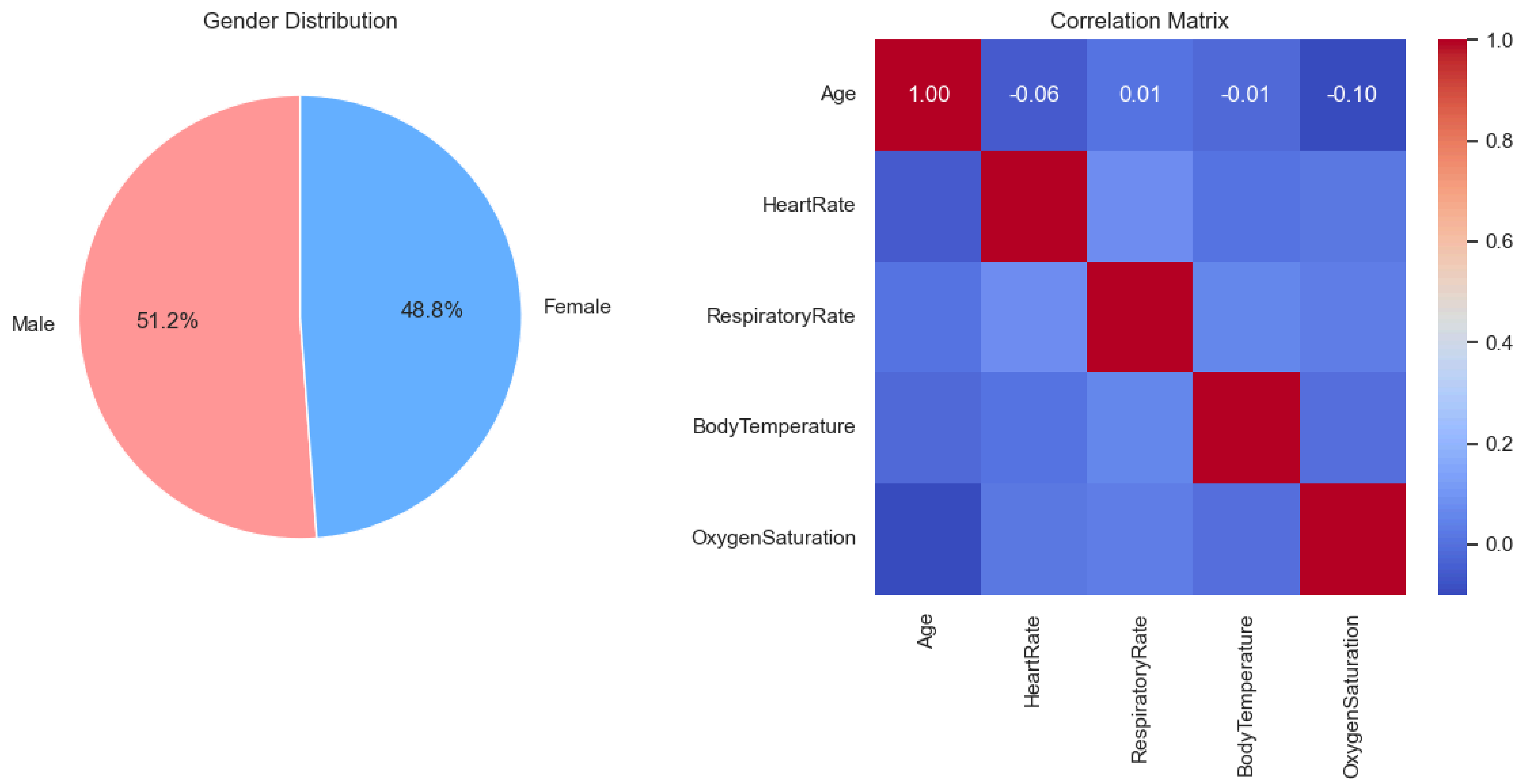



In [16]:
```python
print(summary_stats)
```

```
        PatientID         Age   HeartRate  RespiratoryRate  BodyTemperature  \
count  500.000000  500.000000  500.000000       500.000000       500.000000   
mean   250.500000   51.146000   80.131613        17.524000        98.584383   
std    144.481833   19.821566    9.606273         3.382352         0.461502   
min      1.000000   18.000000   60.169259        12.000000        97.094895   
25%    125.750000   34.000000   75.000000        15.000000        98.281793   
50%    250.500000   51.000000   80.000000        17.500000        98.609167   
75%    375.250000   69.000000   86.276413        20.000000        98.930497   
max    500.000000   84.000000   99.925508        23.000000        99.489150   

       OxygenSaturation  
count        500.000000  
mean          96.296000  
std            1.408671  
min           94.000000  
25%           96.000000  
50%           96.000000  
75%           97.000000  
max           99.000000  
```

Now, let's have a look at the gender distribution in the data and the correlaton between the numerical columns in the dataset:

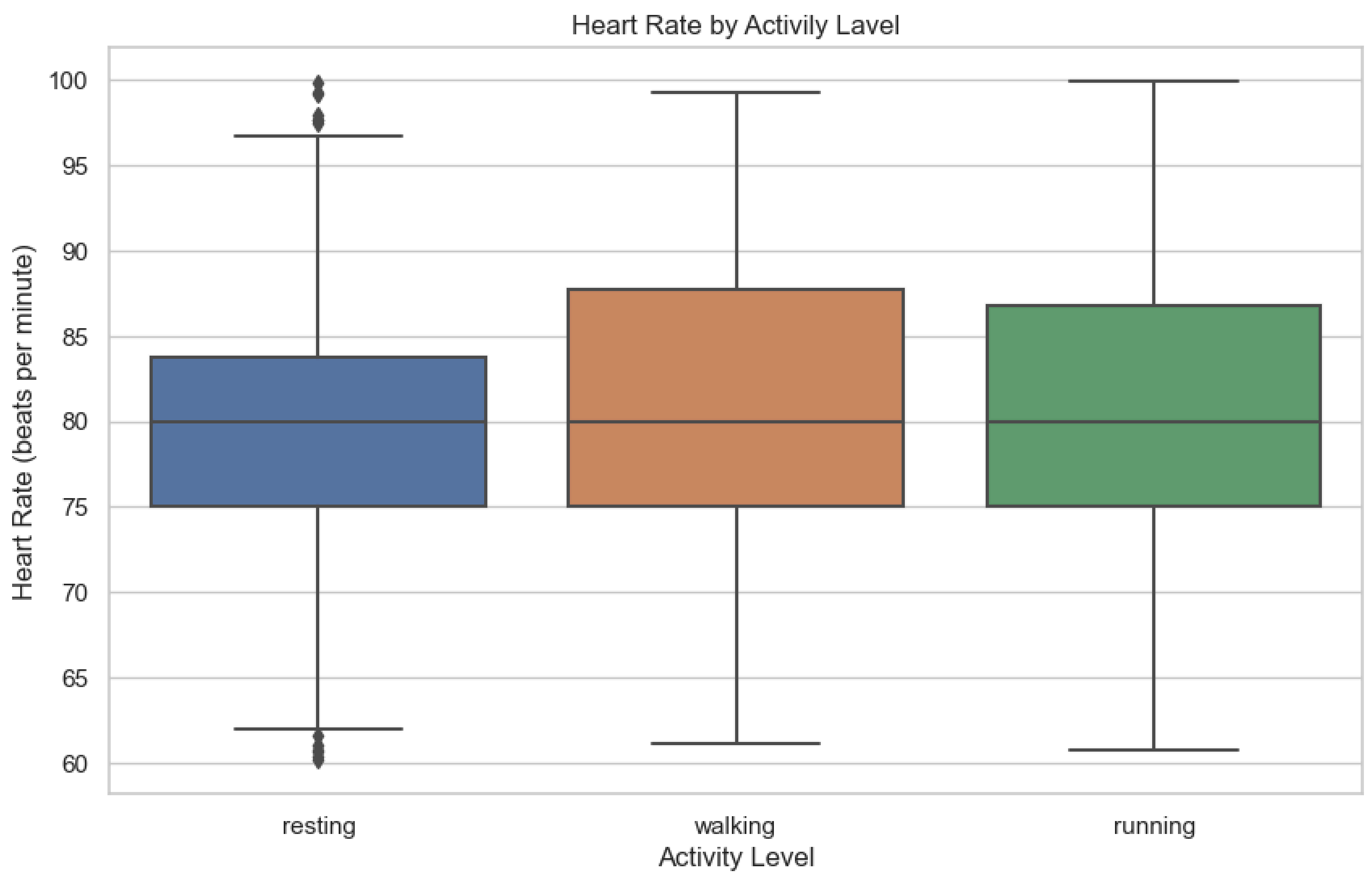
In [17]:
```
# Gender Distribution
gender_counts= health_data['Gender'].value_counts()

# correlation Matrix for numerical health metrics
correlation_matrix= health_data[['Age', 'HeartRate', 'RespiratoryRate', 'BodyTemperature', 'OxygenSaturation']].corr()

#plotting the findings
fig, axes= plt.subplots(1, 2, figsize= (12, 6))

#gender distribution plot
gender_counts.plot(kind= 'pie', ax= axes[0], autopct= '%1.1f%%', startangle= 90, colors= ['#ff9999', '#66b3ff'])
axes[0].set_ylabel('')
axes[0].set_title('Gender Distribution')

# correlation matrix plot

sns.heatmap(correlation_matrix, annot= True, fmt= ".2f", cmap= 'coolwarm', ax= axes[1])
axes[1].set_title('Correlation Matrix')

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




The pie chart indicates a nearly even split between male and female subjects in the dataset, with males comprising a slight majority at 51.2%. The correlation matrix shows no strong correlations between the variables, as all the values are close to zero. Specifically, none of the health metrics (Age, Heart Rate, Respiratory Rate, Body Temperature, and Oxygen Saturation) display a strong positive or negative linear relationship with one another in this particular dataset. It suggests that, for this group of individuals, changes in one metric are not strongly associated with changes in the others

Now, let's have a look at the heart rate by activily level:

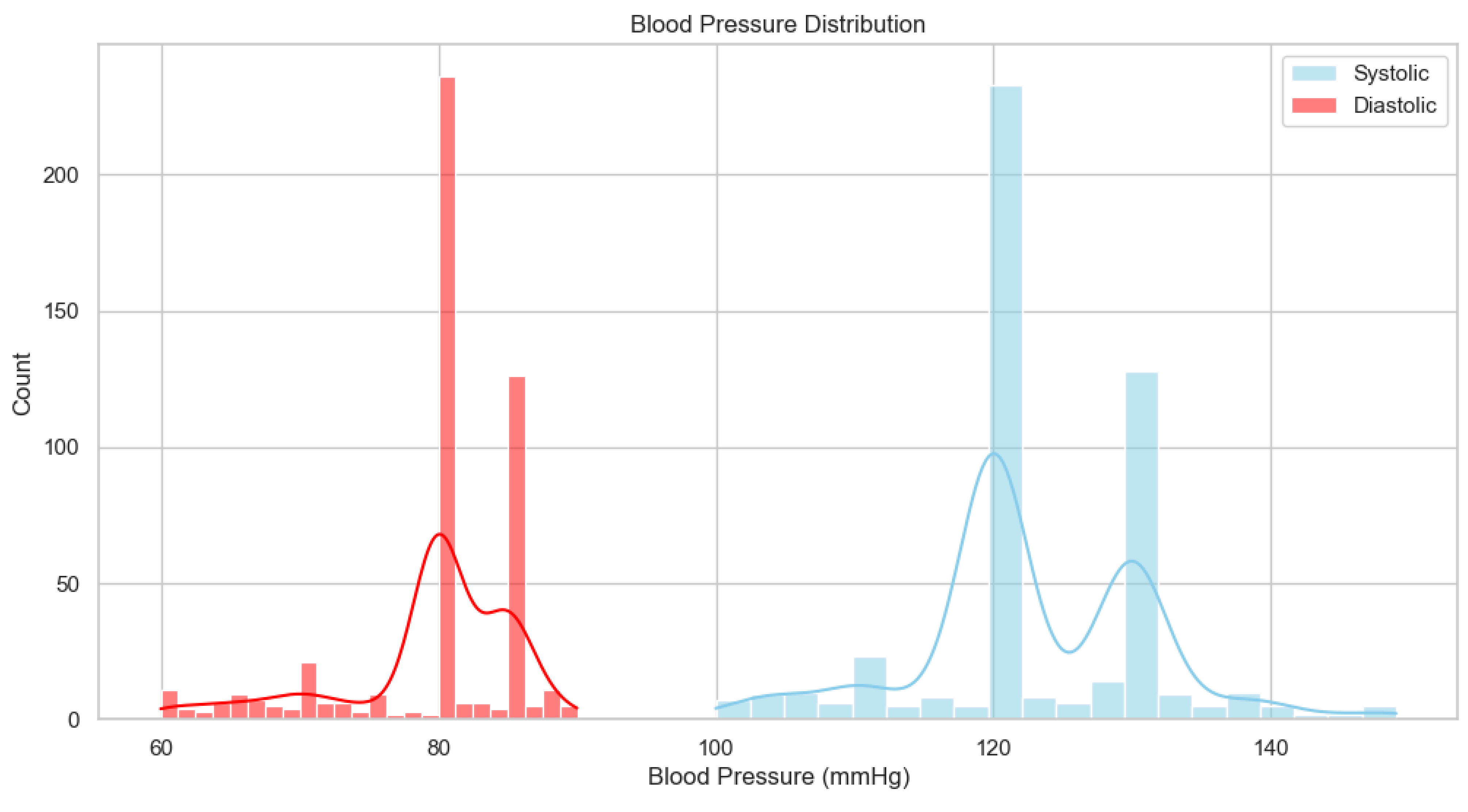
In [19]:
```
# heart Rate by activily level
plt.figure(figsize=(10,6))
sns.boxplot(x= 'ActivityLevel', y='HeartRate', data= health_data)
plt.title('Heart Rate by Activily Lavel')
plt.ylabel('Heart Rate (beats per minute)')
plt.xlabel('Activity Level')
plt.show()
```

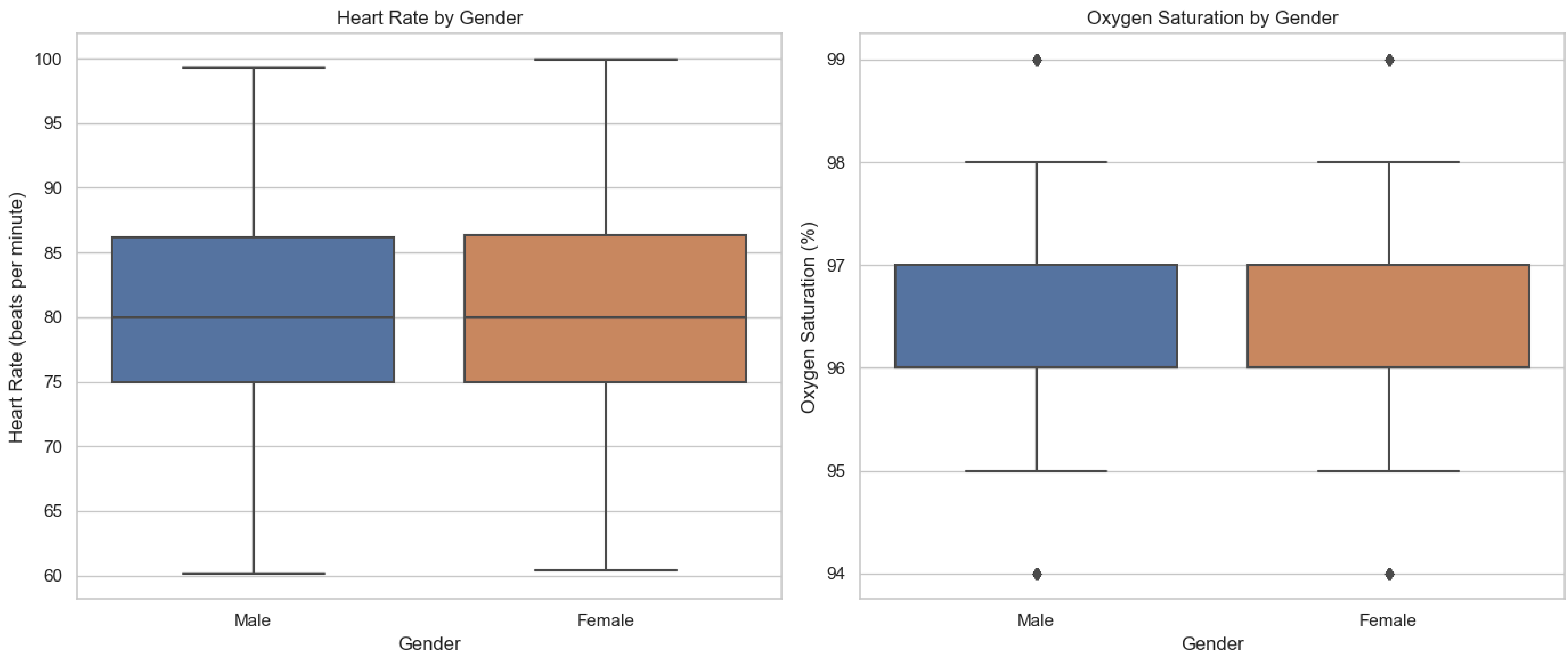



It shows that the median heart rate increases from resting to walking, which is expected as physical activity increases. However, the median heart rate does not significantly increase further during running compared to walking, which is unusual since we would expect a higher median heart rate for a more strenuous activity. Additionally, there is considerable overlap in the interquartile ranges between walking and running, suggesting similar variability in heart rates for these activities within the sampled population. The presence of outliers in the resting category indicates that some individuals have resting heart rates that are much higher than the typical range for the rest of the group.

Now, let's have a look at the distribution of blood pressure levels and some health metrics by gender:

In [22]:
```
# extracting systolic and diastolic blood pressure for analysis
health_data[['SystolicBP', 'DiastolicBP']] = health_data['BloodPressure'].str.split('/', expand=True).astype(int)

# blood pressure distribution
plt.figure(figsize=(12, 6))
sns.histplot(health_data['SystolicBP'], color="skyblue", label="Systolic", kde=True)
sns.histplot(health_data['DiastolicBP'], color="red", label="Diastolic", kde=True)
plt.title('Blood Pressure Distribution')
plt.xlabel('Blood Pressure (mmHg)')
plt.legend()

#health metrics by gender
fig, axes= plt.subplots(1, 2, figsize= (14,6))
sns.boxplot(x= 'Gender', y='HeartRate', data= health_data, ax= axes[0])
axes[0].set_title('Heart Rate by Gender')
axes[0].set_xlabel('Gender')
axes[0].set_ylabel('Heart Rate (beats per minute)')

sns.boxplot(x='Gender', y='OxygenSaturation', data=health_data, ax=axes[1])
axes[1].set_title('Oxygen Saturation by Gender')
axes[1].set_xlabel('Gender')
axes[1].set_ylabel('Oxygen Saturation (%)')

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





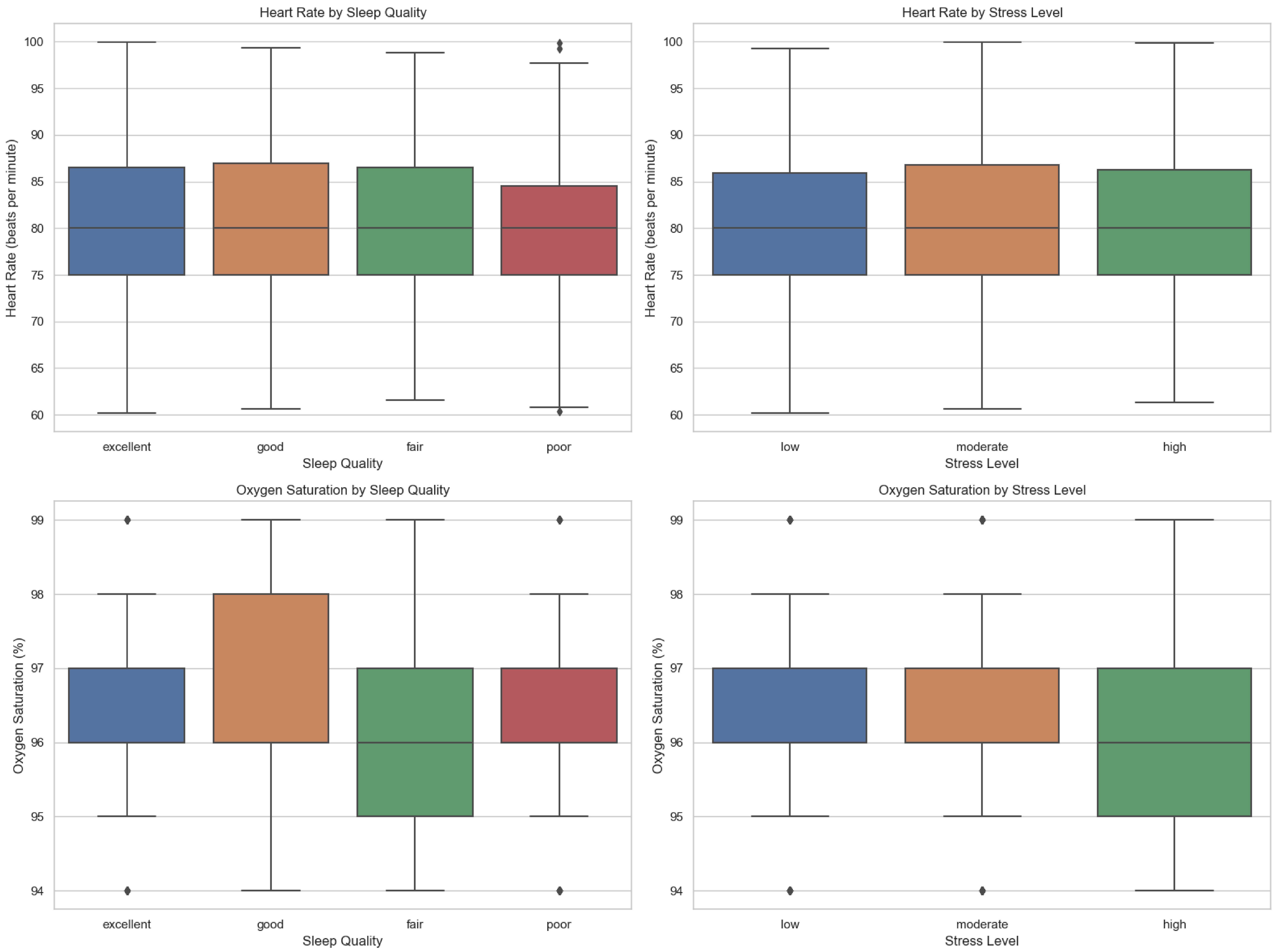

The systolic blood pressure, represented in blue, shows a more spread-out distribution with peaks suggesting common readings around 120 mmHg and 140 mmHg. The diastolic blood pressure, in red, appears to have a narrower distribution, with a significant peak around 80 mmHg. The spread of systolic values is broader than the diastolic ones, which is typical as systolic pressure tends to vary more with factors like activity level and stress. This distribution is consistent with general population trends where a systolic reading of around 120 mmHg and a diastolic reading of around 80 mmHg are considered normal.

For heart rate, both males and females show similar median values with a relatively similar interquartile range, indicating no significant difference in heart rate between genders within this dataset. In terms of oxygen saturation, again, both genders exhibit nearly identical medians and interquartile ranges, suggesting that oxygen saturation does not differ notably between males and females in this sample. There are a few outliers in oxygen saturation for both genders, indicating a few individuals with lower than typical values, but these do not seem to affect the overall distribution significantly.

Now, let’s analyze heart rate and oxygen saturation by sleep quality and stress levels:

In [23]:

```
# categorizing sleep quality and stress level for better analysis
sleep_quality_order = ['excellent', 'good', 'fair', 'poor']
stress_level_order = ['low', 'moderate', 'high']

# creating plots to examine relationships
fig, axes = plt.subplots(2, 2, figsize=(16, 12))

# heart rate by sleep quality
sns.boxplot(x='SleepQuality', y='HeartRate', data=health_data, order=sleep_quality_order, ax=axes[0, 0])
axes[0, 0].set_title('Heart Rate by Sleep Quality')
axes[0, 0].set_xlabel('Sleep Quality')
axes[0, 0].set_ylabel('Heart Rate (beats per minute)')

# heart rate by stress level
sns.boxplot(x='StressLevel', y='HeartRate', data=health_data, order=stress_level_order, ax=axes[0, 1])
axes[0, 1].set_title('Heart Rate by Stress Level')
axes[0, 1].set_xlabel('Stress Level')
axes[0, 1].set_ylabel('Heart Rate (beats per minute)')

# oxygen saturation by sleep quality
sns.boxplot(x='SleepQuality', y='OxygenSaturation', data=health_data, order=sleep_quality_order, ax=axes[1, 0])
axes[1, 0].set_title('Oxygen Saturation by Sleep Quality')
axes[1, 0].set_xlabel('Sleep Quality')
axes[1, 0].set_ylabel('Oxygen Saturation (%)')

# oxygen saturation by stress level
sns.boxplot(x='StressLevel', y='OxygenSaturation', data=health_data, order=stress_level_order, ax=axes[1, 1])
axes[1, 1].set_title('Oxygen Saturation by Stress Level')
axes[1, 1].set_xlabel('Stress Level')
axes[1, 1].set_ylabel('Oxygen Saturation (%)')

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

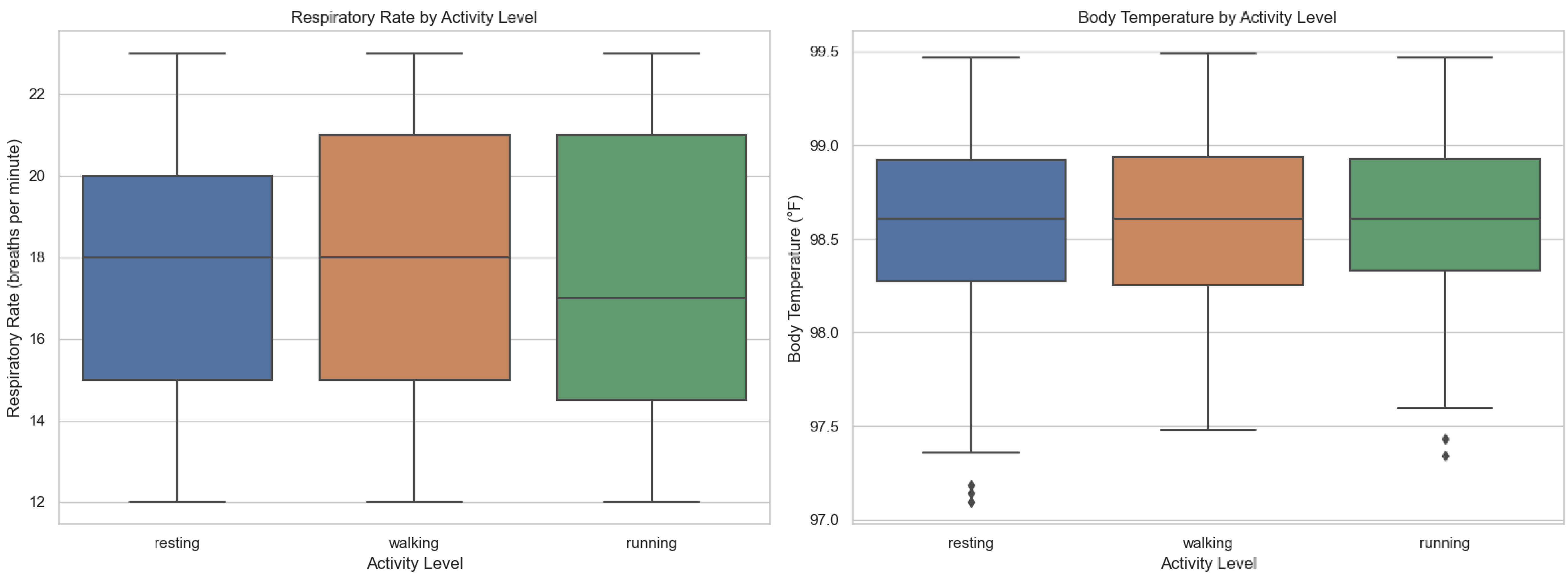



Heart rate appears relatively consistent across different levels of sleep quality and stress, with a slight increase in variation for those reporting poor sleep. Oxygen saturation shows a minimal decrease in median values from excellent to poor sleep quality, with some outliers indicating lower saturation for excellent and good sleep. When correlated with stress levels, oxygen saturation remains largely unchanged. Overall, while there are outliers, the central tendencies suggest that neither heart rate nor oxygen saturation is greatly affected by sleep quality or stress level within this dataset.

Now, let’s analyze the respiratory rate and body temperature by activity levels:

In [24]:

```
# creating plots to examine relationships between activity level and other health metrics
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# respiratory rate by activity level
sns.boxplot(x='ActivityLevel', y='RespiratoryRate', data=health_data, ax=axes[0])
axes[0].set_title('Respiratory Rate by Activity Level')
axes[0].set_xlabel('Activity Level')
axes[0].set_ylabel('Respiratory Rate (breaths per minute)')

# body temperature by activity level
sns.boxplot(x='ActivityLevel', y='BodyTemperature', data=health_data, ax=axes[1])
axes[1].set_title('Body Temperature by Activity Level')
axes[1].set_xlabel('Activity Level')
axes[1].set_ylabel('Body Temperature (°F)')

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




The respiratory rate tends to increase with activity level, as indicated by higher median rates for walking and running compared to resting. It aligns with physiological responses to exercise, where the breathing rate increases to meet oxygen demands. For body temperature, there is a slight upward trend from resting to running, which is consistent with the body heating up during physical exertion. There are outliers in body temperature at the resting and running levels, suggesting some individuals have temperatures outside the typical range for these activities. Overall, the trends observed are in line with expected physiological responses to varying levels of activity.

# Grouping Patients

The data is not so complicated enough that we need to use a clustering algorithm to group patients. So, let's group patients based on:

1. Age Group: Young, Middle-aged, Senior
2. Blood Pressure Category: Normal, Elevated, Hypertension Stage 1, Hypertension Stage 2
3. Heart Rate Category: Low, Normal, High
4. Oxygen Saturation Category: Normal, Low

In [27]:

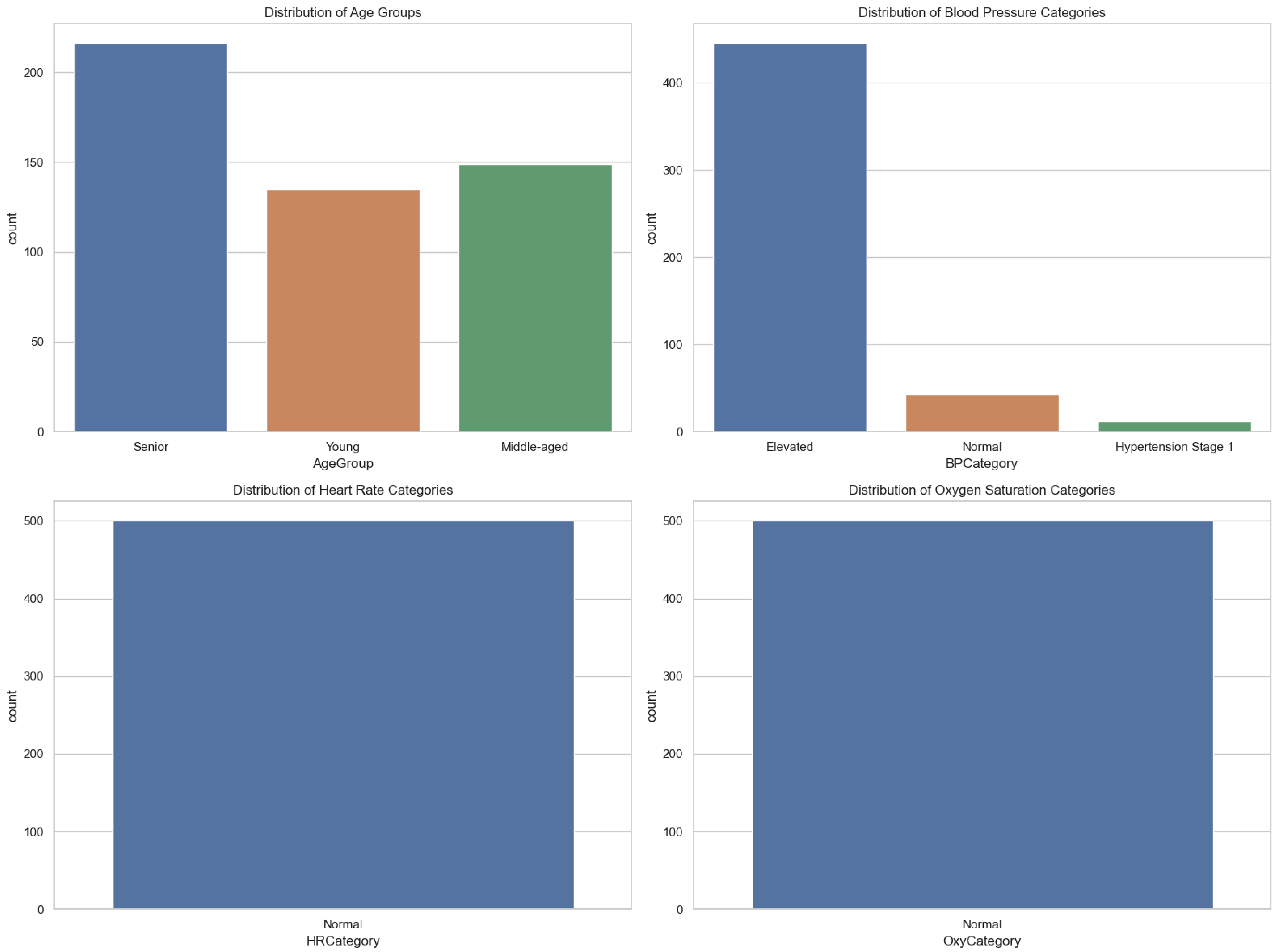
```
# function to categorize Age
def age_group(age):
    if age <= 35:
        return 'Young'
    elif age <=55:
        return 'Middle-aged'
    else:
        return 'Senior'

# function to categorize Blood Pressure
def bp_category(systolic, diastolic):
    if systolic <120 and diastolic < 80:
        return 'Normal'
    elif 120 <= systolic <140 or 80 <= diastolic <90:
        return 'Elevated'
    elif 140 <= systolic <160 or 90 <= diastolic <100:
        return 'Hypertension Stage 1'
    else:
        return 'Hypertension Stage 2'

# function to categorize Heart Rate
def hr_category(hr):
    if hr < 60:
        return 'Low'
    elif hr <= 100:
        return 'Normal'
    else:
        return 'High'

# function to categorize Oxygen Saturation
def oxy_category(oxy):
    if oxy < 94:
        return 'Low'
    else:
        return 'Normal'


# applying categorizations
health_data['AgeGroup'] = health_data['Age'].apply(age_group)
health_data['BPCategory'] = health_data.apply(lambda x: bp_category(x['SystolicBP'], x['DiastolicBP']), axis=1)
health_data['HRCategory'] = health_data['HeartRate'].apply(hr_category)
health_data['OxyCategory'] = health_data['OxygenSaturation'].apply(oxy_category)

print(health_data[['Age', 'AgeGroup', 'SystolicBP', 'DiastolicBP', 'BPCategory', 'HeartRate', 'HRCategory', 'OxygenSaturation', 'OxyCategory']].head())
```

```
   Age     AgeGroup  SystolicBP  DiastolicBP BPCategory  HeartRate HRCategory  \
0   69       Senior         130           85   Elevated  60.993428     Normal
1   32        Young         120           80   Elevated  98.723471     Normal
2   78       Senior         130           85   Elevated  82.295377     Normal
3   38  Middle-aged         111           78     Normal  80.000000     Normal
4   41  Middle-aged         120           80   Elevated  87.531693     Normal

   OxygenSaturation OxyCategory
0              95.0      Normal
1              97.0      Normal
2              98.0      Normal
3              98.0      Normal
4              98.0      Normal
```

# Now, let's visualize the groups:

In [28]:

```
fig, axes = plt.subplots(2, 2, figsize=(16, 12))

# Age Group count plot
sns.countplot(x='AgeGroup', data=health_data, ax=axes[0, 0])
axes[0, 0].set_title('Distribution of Age Groups')

# Blood Pressure Category count plot
sns.countplot(x='BPCategory', data=health_data, ax=axes[0, 1])
axes[0, 1].set_title('Distribution of Blood Pressure Categories')

# Heart Rate Category count plot
sns.countplot(x='HRCategory', data=health_data, ax=axes[1, 0])
axes[1, 0].set_title('Distribution of Heart Rate Categories')

# Oxygen Saturation Category count plot
sns.countplot(x='OxyCategory', data=health_data, ax=axes[1, 1])
axes[1, 1].set_title('Distribution of Oxygen Saturation Categories')

# Show the plots
plt.tight_layout()
plt.show()
```




# Observation:

1. Distribution of Age Groups: The count plot shows that the ‘Senior’ category has the highest count, followed by the ‘Young’ and ‘Middle-aged’ categories. It suggests that seniors are the largest age group in this dataset.
2. Distribution of Blood Pressure Categories: The majority of the dataset falls under ‘Normal’ blood pressure, with fewer instances in the ‘Elevated’ and ‘Hypertension Stage 1’. ‘Hypertension Stage 2’ has the lowest count, indicating that severe hypertension is less common among the participants.
3. Distribution of Heart Rate Categories: Most individuals have a ‘Normal’ heart rate, with very few falling into the ‘Low’ or ‘High’ categories. It indicates that most individuals in this dataset have a heart rate that falls within the expected range.
4. Distribution of Oxygen Saturation Categories: Almost everyone has ‘Normal’ oxygen saturation levels, with very few instances of ‘Low’ saturation. It suggests that oxygen deprivation is not a common issue in this group.

# Summary

So, Health Monitoring and Analysis contains a range of activities from real-time physiological data collection (like heart rate, blood pressure, and temperature) to the analysis of more complex health records (including patient history, lifestyle choices, and genetic information).

In [ ]: