Sleep Disorder Prediction

In [2]:
```
#loading the dataset
df = pd.read_csv('C:/Users/zizhe/Desktop/Leo Zhao/Sleep Disorder Prediction/Sleep_health_and_lifestyle_dataset.csv')
df.head()
```

Out[2]:

	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	None
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
2	3	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None
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

Data Preprocessing Part 1

In [3]:
```
#checking for missing values
df.isnull().sum()
```

Out[3]:
```
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
```

In [4]:
```
#replacing the null values with 'None' in the column 'Sleep Disorder'
df['Sleep Disorder'].fill('None', inplace=True)
```

The nan/None value in the 'sleep disorder' column stands for no sleep disorder, so it is not a missing value.

In [5]:
```
#drop column Person ID
df.drop('Person ID', axis=1, inplace=True)
```

In [6]:
```
#checking the number of unique values in each column
print("Unique values in each column are:")
for col in df.columns:
    print(col,df[col].nunique())
```

```
Unique values in each column are:
Gender 2
Age 31
Occupation 11
Sleep Duration 27
Quality of Sleep 6
Physical Activity Level 16
Stress Level 6
BMI Category 4
Blood Pressure 25
Heart Rate 19
Daily Steps 20
Sleep Disorder 3
```

Splitting the blood pressure into systolic and diastolic

In [7]:
```
#spliting the blood pressure into two columns
df['systolic_bp'] = df['Blood Pressure'].apply(lambda x: x.split('/'))
df['diastolic_bp'] = df['Blood Pressure'].apply(lambda x: x.split('/'))
#droping the blood pressure column
df.drop('Blood Pressure', axis=1, inplace=True)
```

In [8]:
```
#replacing normal weight with normal in BMI column
df['BMI Category'] = df['BMI Category'].replace('Normal Weight', 'Normal')
```

In [9]:
```
df.head()
```

Out[9]:

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate	Daily Steps	Sleep Disorder	systolic_bp	diastolic_bp
0	Male	27	Software Engineer	6.1	6	42	6	Overweight	77	4200	None	126	83
1	Male	28	Doctor	6.2	6	60	8	Normal	75	10000	None	125	80
2	Male	28	Doctor	6.2	6	60	8	Normal	75	10000	None	125	80
3	Male	28	Sales Representative	5.9	4	30	8	Obese	85	3000	Sleep Apnea	140	90
4	Male	28	Sales Representative	5.9	4	30	8	Obese	85	3000	Sleep Apnea	140	90

Checking the unique values from each categorical column

In [10]:
```
#unique values from categorical columns
print(df.Occupation.unique())
print(df['BMI Category'].unique())
print(df['Sleep Disorder'].unique())
```

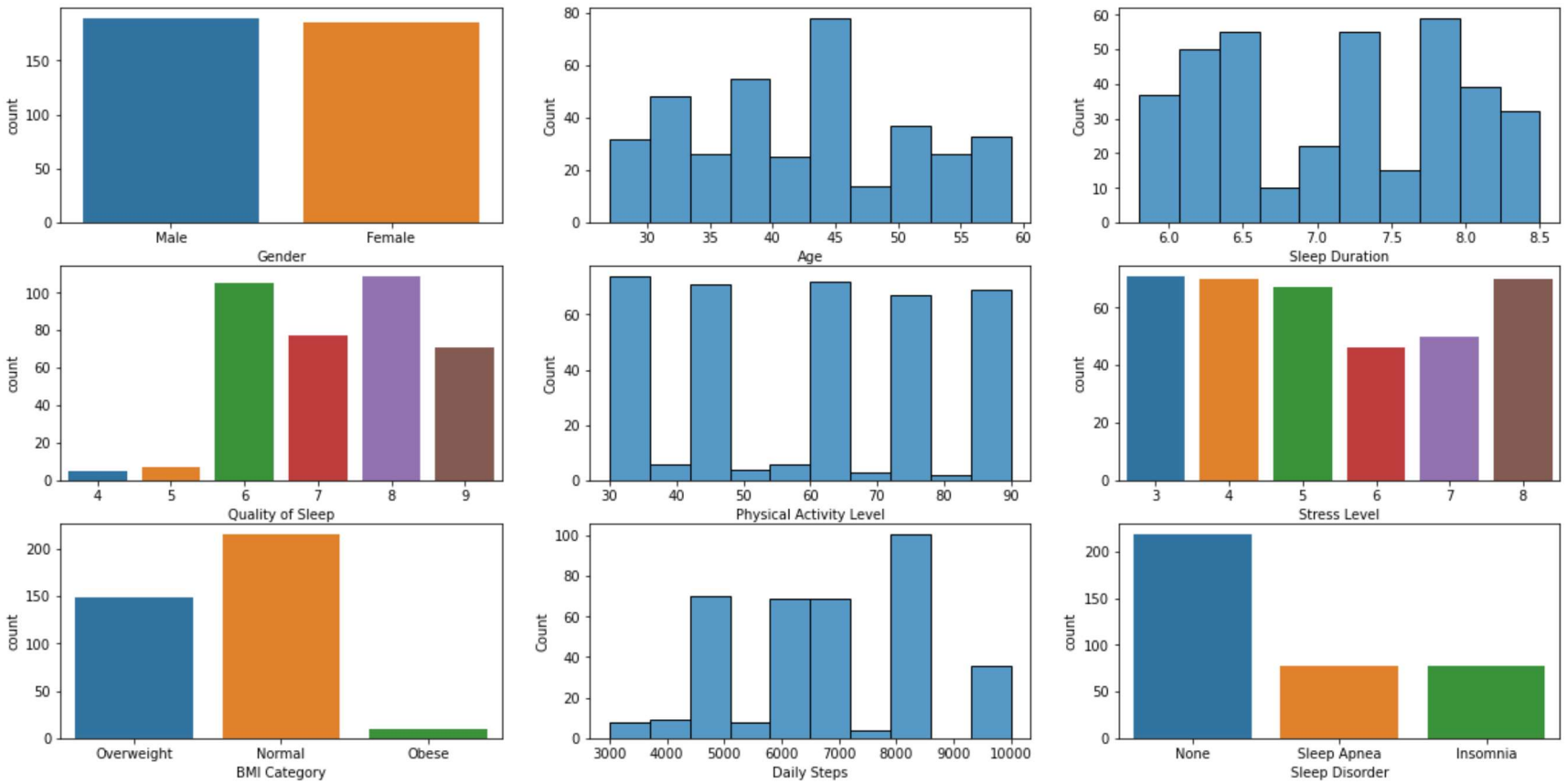
```
['Software Engineer' 'Doctor' 'Sales Representative' 'Teacher' 'Nurse'
 'Engineer' 'Accountant' 'Scientist' 'Lawyer' 'Salesperson' 'Manager']


['Overweight' 'Normal' 'Obese']


['None' 'Sleep Apnea' 'Insomnia']
```

Exploratory Data Analysis

Understanding the data by plotting its variables

In [11]:
```
fig,ax = plt.subplots(3,3,figsize=(20,10))
sns.countplot(x = 'Gender', data = df, ax = ax[0,0])
sns.histplot(x = 'Age', data = df, ax = ax[0,1])
sns.histplot(x = 'Sleep Duration', data = df, ax = ax[0,2])
sns.countplot(x = 'Quality of Sleep', data = df, ax = ax[1,0])
sns.histplot(x = 'Physical Activity Level', data = df, ax = ax[1,1])
sns.countplot(x = 'Stress Level', data = df, ax = ax[1,2])
sns.countplot(x = 'BMI Category', data = df, ax = ax[2,0])
sns.histplot(x = 'Daily Steps', data = df, ax = ax[2,1])
sns.countplot(x = 'Sleep Disorder', data = df, ax = ax[2,2])
```

Out[11]:
```
<AxesSubplot:xlabel='Sleep Disorder', ylabel='count'>
```



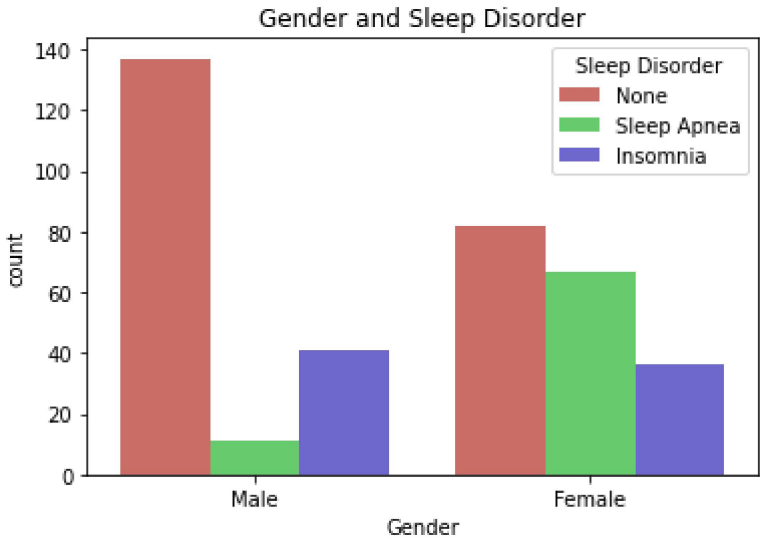
The number of males and females is almost equal, with the majority of people aged between 30 and 45 years. Most individuals have a sleep quality rating greater than 5, indicating that they are getting sufficient sleep. Furthermore, the majority of people have a normal BMI, which directly relates to the distribution of sleep disorders, showing an equal number of people with and without sleep disorders.

Understanding the correlation between the variables

Gender and Sleep Disorder

In [12]:
```
#Gender count plot
sns.countplot(x = 'Gender', data = df, palette = 'hls', hue = 'Sleep Disorder').set_title('Gender and Sleep Disorder')
```

Out[12]:
```
Text(0.5, 1.0, 'Gender and Sleep Disorder')
```

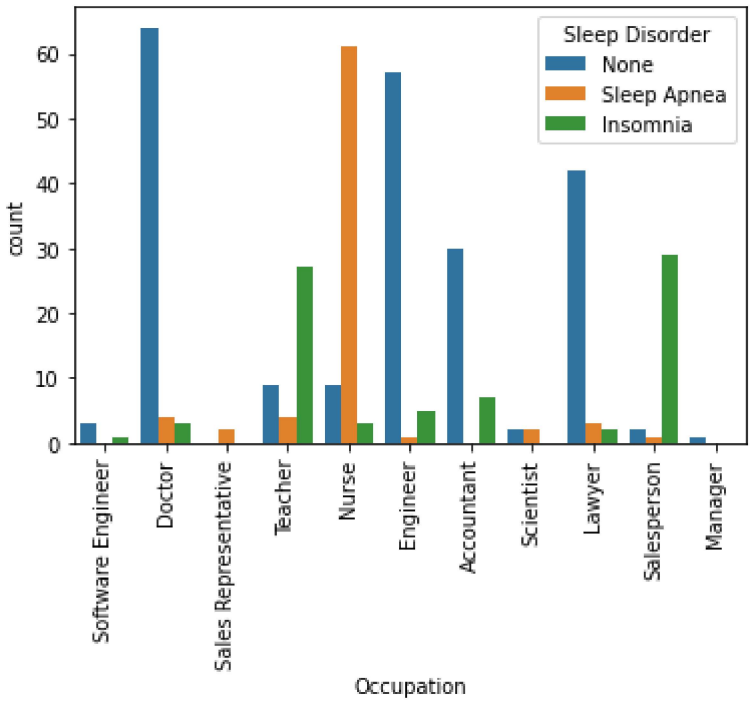


Most males and females do not suffer from any sleep disorder. However, females tend to experience sleep disorders more frequently compared to males. The number of females with Sleep Apnea is notably higher than that of males. In contrast, a greater number of males suffer from Insomnia compared to females.

Effect of Occupation on Sleep Disorder

In [13]:
```
ax = sns.countplot(x = 'Occupation', data = df, hue = 'Sleep Disorder')
ax.set_xticklabels(ax.get_xticklabels(), rotation = 90)
```

Out[13]:
```
[Text(0, 0, 'Software Engineer'),
 Text(1, 0, 'Doctor'),
 Text(2, 0, 'Sales Representative'),
 Text(3, 0, 'Teacher'),
 Text(4, 0, 'Nurse'),
 Text(5, 0, 'Engineer'),
 Text(6, 0, 'Accountant'),
 Text(7, 0, 'Scientist'),
 Text(8, 0, 'Lawyer'),
 Text(9, 0, 'Salesperson'),
 Text(10, 0, 'Manager')]
```

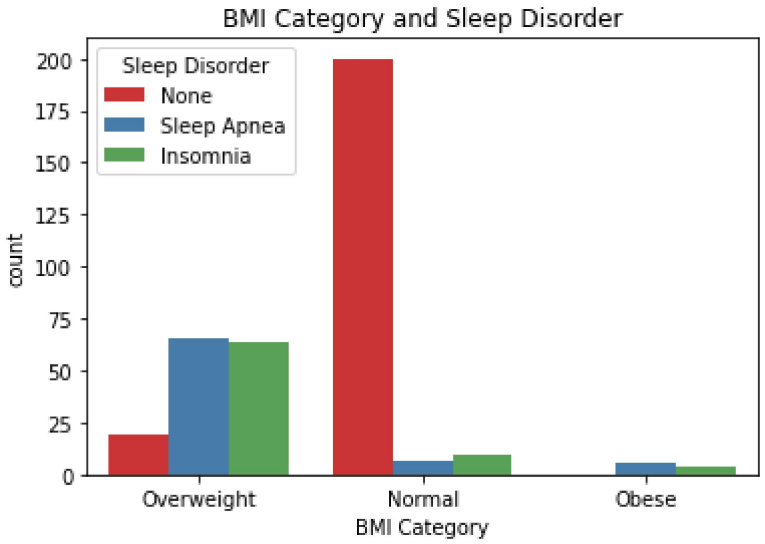


From the graph, it is clear that occupation has a significant impact on sleep disorders. Nurses are more susceptible to Sleep Apnea compared to other occupations, and very few of them have no sleep disorder. After nurses, the next most affected occupation is Salespeople, with the highest incidence of Insomnia, followed by teachers. However, there are some occupations where most people have very few instances of Sleep Apnea and Insomnia, such as Engineers, Doctors, Accountants, and Lawyers. Software Engineers and Managers are in such low numbers that I cannot provide much insight. However, Sales Representatives primarily experience Sleep Apnea without Insomnia or any other sleep disorder.

BMI and Sleep Disorder

In [14]:
```
sns.countplot(x = 'BMI Category', hue = 'Sleep Disorder', data = df, palette = 'Set1').set_title('BMI Category and Slee
```

Out[14]:
```
Text(0.5, 1.0, 'BMI Category and Sleep Disorder')
```



People with a normal BMI are less likely to suffer from any sleep disorder. However, the opposite is true for overweight and obese individuals. Overweight individuals are more likely to experience sleep disorders than obese people.

Data Preprocessing Part 2

Label Encoding for categorical variables

In [15]:
```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
```

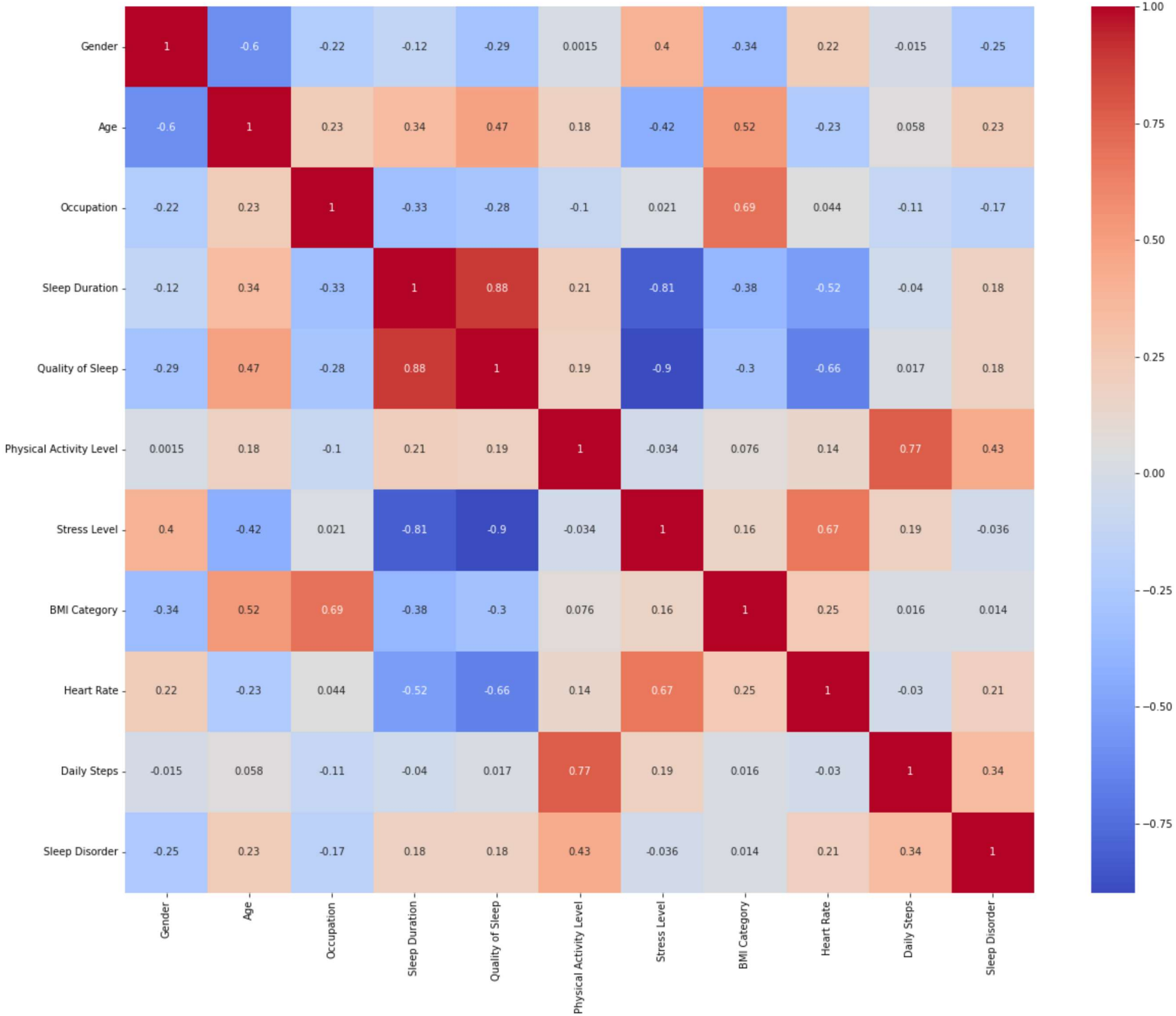
In [16]:
```
vars = ['Gender', 'Occupation','BMI Category','Sleep Disorder']
for i in vars:
    label_encoder.fit(unique())
    df[i] = label_encoder.transform(df[i])
    print(i, df[i].unique())
```

```
Gender : [1 0]
Occupation : [ 9  1  6 10  5  2  0  8  3  7  4]
BMI Category : [2 0 1]
Sleep Disorder : [1 2 0]
```

Correlation Matrix Heatmap

In [17]:
```
#Correlation Matrix Heatmap
plt.figure(figsize=(20, 16))
sns.heatmap(df.cor(), annot = True, map = 'coolwarm')
```

Out[17]:
```
<AxesSubplot:>
```


Gender 1 -0.6 -0.22 -0.12 -0.29 0.0015 0.4 -0.34 0.22 -0.015 -0.25
Age -0.6 1 0.23 0.34 0.47 0.18 -0.42 0.52 -0.23 0.058 0.23
Occupation -0.22 0.23 1 -0.33 -0.28 -0.1 0.021 0.69 0.044 -0.11 -0.17
Sleep Duration -0.12 0.34 -0.33 1 0.88 0.21 -0.81 -0.38 -0.52 -0.04 0.18
Quality of Sleep -0.29 0.47 -0.28 0.88 1 0.19 -0.9 -0.3 -0.66 0.017 0.18
Physical Activity Level 0.0015 0.18 -0.1 0.21 0.19 1 -0.034 0.076 0.14 0.77 0.43
Stress Level 0.4 -0.42 0.021 -0.81 -0.9 -0.034 1 0.16 0.67 0.19 -0.036
BMI Category -0.34 0.52 0.69 -0.38 -0.3 0.076 0.16 1 0.25 0.016 0.014
Heart Rate 0.22 -0.23 0.044 -0.52 -0.66 0.14 0.67 0.25 1 -0.03 0.21
Daily Steps -0.015 0.058 -0.11 -0.04 0.017 0.77 0.19 0.016 -0.03 1 0.34
Sleep Disorder -0.25 0.23 -0.17 0.18 0.18 0.43 -0.036 0.014 0.21 0.34 1
Gender
Age
Occupation
Sleep Duration
Quality of Sleep
Physical Activity Level
Stress Level
BMI Category
Heart Rate
Daily Steps
Sleep Disorder
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

Train Test Split

In [18]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df.drop('Sleep Disorder',axis=1), df['Sleep Disorder'], test_size=0
```

Model Building

For predicting sleep disorders through classification algorithms, I will use the following algorithms:

1. Decision Tree Classifier
2. Random Forest Classifier

Decision Tree Classifier

In [19]:
```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree
```

Out[19]:
```
DecisionTreeClassifier()
```

Training the model with train dataset

In [20]:
```
dtree.fit(X_train, y_train)
```

Out[20]:
```
DecisionTreeClassifier()
```

In [21]:
```
#training accuracy
print("Training Accuracy:",dtree.score(X_train,y_train))
```

```
Training Accuracy: 0.9348659003831418
```

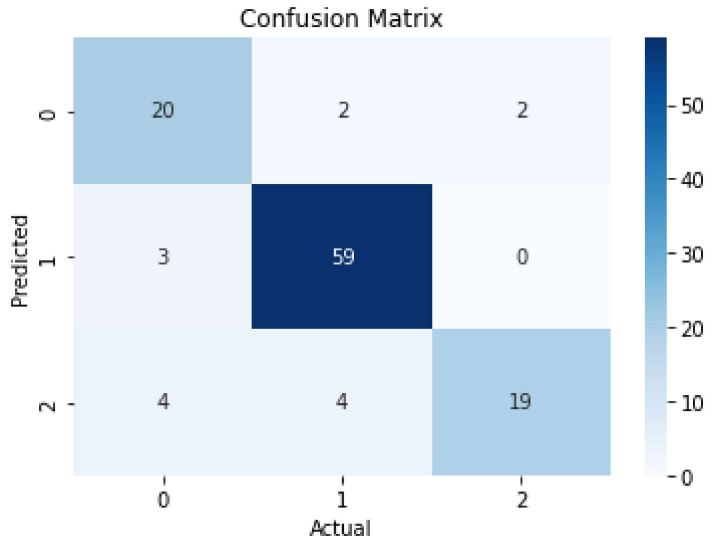
Decision Tree Model Evalution

In [22]:
```
d_pred = dtree.predict(X_test)
d_pred
```

Out[22]:
```
array([1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 2, 1, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 0, 2, 0, 0,
       1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 0, 2, 0, 2, 2, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 0, 1, 2, 2, 0, 1, 1, 2, 0, 1, 2, 1, 1, 1, 2, 0, 2,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 2, 0, 1, 1, 0, 2, 1, 1,
       2, 1, 0])
```

Using Confusion matrix heatmap to visualize the model accuracy

In [23]:
```
from sklearn.metrics import confusion_matrix
sns.heatmap(confusion_matrix(y_test, d_pred), annot=True, map='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.show()
```



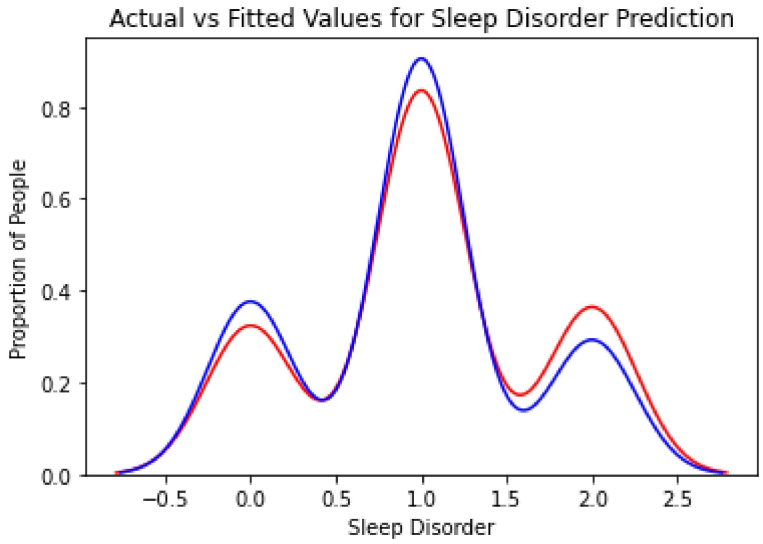
The diagonal boxes show the count of true positive results, i.e., correct predictions made by the model. The off-diagonal boxes show the count of false positive results, i.e., incorrect predictions made by the model.

Dsitribution plot for predicted and actual values

In [24]:
```
ax = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
sns.distplot(d_pred, hist=False, color="b", label="Fitted Values")
plt.title('Actual vs Fitted Values for Sleep Disorder Prediction')
plt.xlabel('Sleep Disorder')
```

```
plt.ylabel('Proportion of People')
plt.show()
```

```
C:\Users\zizhe\New folder\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\zizhe\New folder\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



The actual values are represented in red, and the predicted ones are in blue. As shown in the graph, the model's predictions are able to follow the curve of the actual values, but the predicted values still differ from the actual ones. Therefore, the model is unable to predict the values accurately.

Classification Report

In [25]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, d_pred))
```

```
              precision    recall  f1-score   support

           0       0.74      0.83      0.78        24
           1       0.91      0.95      0.93        62
           2       0.90      0.70      0.79        27

    accuracy                           0.87       113
   macro avg       0.85      0.83      0.84       113
weighted avg       0.87      0.87      0.87       113
```

The model yields pretty decent results with an accuracy of 87% and an average F1 score of 0.83. It effectively predicts sleep disorders with good accuracy.

Random Forest Classifier

In [26]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=100, random_state=42)
```

Training the model with train dataset

In [27]:
```
rfc.fit(X_train, y_train)
```

Out[27]:
```
RandomForestClassifier(random_state=42)
```

In [28]:
```
#Training accuracy
print("Training accuracy: ",rfc.score(X_train,y_train))
```

```
Training accuracy:  0.9348659003831418
```

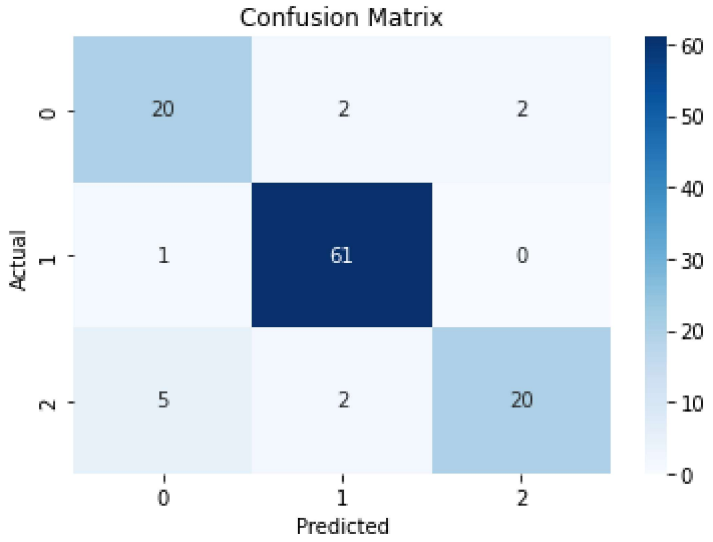
Random Forest Classifier Evaluation

In [29]:
```
rfc_pred = rfc.predict(X_test)
rfc_pred
```

Out[29]:
```
array([1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 2, 1, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 0, 0, 0, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 0, 2, 0, 0,
       1, 1, 1, 1, 2, 1, 2, 0, 2, 1, 0, 2, 0, 2, 2, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 1, 1, 1, 0, 1, 2, 2, 0, 1, 1, 2, 0, 1, 2, 1, 1, 1, 2, 1, 2,
       1, 1, 2, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 2, 0, 1, 2, 0, 2, 1, 1,
       2, 1, 0])
```

Using confusion matrix heatmap to visualize the model accuracy

In [30]:
```
#confusion matrix heatmap
sns.heatmap(confusion_matrix(y_test, rfc_pred), annot=True, map='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

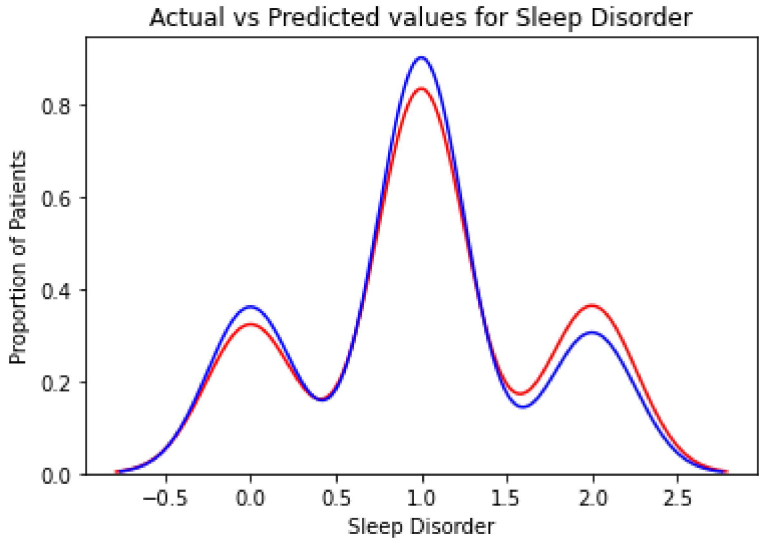


The Random Forest Classifier model has a higher accuracy than the Decision Tree Classifier model. The diagonal boxes represent True Positives, i.e., correct predictions, whereas the off-diagonal boxes show the count of false positive results, i.e., incorrect predictions made by the model. Since the number of false positive values is low, it indicates that the model is proficient at making accurate predictions.

Distribution plot for predicted and acutal values

In [31]:
```
ax = sns.distplot(y_test, hist=False, color="r", label="Actual Value")
sns.distplot(rfc_pred, hist=False, color="b", label="Predicted Values")
plt.title('Actual vs Predicted values for Sleep Disorder')
plt.xlabel('Sleep Disorder')
plt.ylabel('Proportion of Patients')
plt.show()
```

```
C:\Users\zizhe\New folder\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\zizhe\New folder\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function
with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```



The Random Forest classifier has improved accuracy compared to the Decision Tree, which is evident from the wider gap between the actual and predicted values in the case of the Decision Tree Classifier.

Classification Report

In [32]:
```
print(classification_report(y_test, rfc_pred))
```

```
              precision    recall  f1-score   support

           0       0.77      0.83      0.80        24
           1       0.94      0.98      0.96        62
           2       0.91      0.74      0.82        27

    accuracy                           0.89       113
   macro avg       0.87      0.85      0.86       113
weighted avg       0.90      0.89      0.89       113
```

The Random Forest Classifier model has an accuracy of 89% and an average F1 score of 0.86. From these metrics, it is evident that the model can effectively predict sleep disorders with higher accuracy compared to the Decision Tree Classifier.

Conclusion

From the exploratory data analysis, I have concluded that sleep disorders depend on three main factors: gender, occupation, and the patient's BMI. Males have a higher incidence of Insomnia, while females have a higher incidence of Sleep Apnea. Additionally, individuals with occupations such as nursing are more prone to sleep disorders. The patient's BMI also plays a vital role in predicting sleep disorders, with patients who are either obese or overweight being more prone to such disorders.

Regarding the classification models, both models performed well; however, the Random Forest Classifier delivered excellent results with an accuracy of 89%.

In [ ]: