```python
from google.colab import drive
drive.mount('/content/drive')
```

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

```python
data=pd.read_csv('/content/drive/MyDrive/Sleep_health_dataset.csv')
data.head()
```

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight
1	2	Male	28	Doctor	6.2	6	60	8	Normal
2	3	Male	28	Doctor	6.2	6	60	8	Normal
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese

```python
data.info()
```

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 374 entries, 0 to 373
    Data columns (total 13 columns):
     #   Column   Non-Null Count Dtype
    --- ------   -------------- -----
     0   Person ID   374 non-null   int64
     1   Gender   374 non-null   object
     2   Age   374 non-null   int64
     3   Occupation   374 non-null   object
     4   Sleep Duration   374 non-null   float64
     5   Quality of Sleep   374 non-null   int64
     6   Physical Activity Level 374 non-null   int64
     7   Stress Level   374 non-null   int64
     8   BMI Category   374 non-null   object
     9   Blood Pressure   374 non-null   object
     10 Heart Rate   374 non-null   int64
     11 Daily Steps   374 non-null   int64
     12 Sleep Disorder   374 non-null   object
    dtypes: float64(1), int64(7), object(5)
    memory usage: 38.1+ KB

```python
df=data.drop('Person ID',axis=1)
df.head()
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressure
0	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/8
1	Male	28	Doctor	6.2	6	60	8	Normal	125/8
2	Male	28	Doctor	6.2	6	60	8	Normal	125/8
3	Male	28	Sales Representative	5.9	4	30	8	Obese	140/9
	Male								

```python
df['Gender'].unique()
```

    array(['Male', 'Female'], dtype=object)

```python
df['Occupation'].unique()
```

```
array(['Software Engineer', 'Doctor', 'Sales Representative', 'Teacher',
       'Nurse', 'Engineer', 'Accountant', 'Scientist', 'Lawyer',
       'Salesperson', 'Manager'], dtype=object)
```

```
df['BMI Category'].unique()
```

```
array(['Overweight', 'Normal', 'Obese', 'Normal Weight'], dtype=object)
```

```
to_combine=['Normal','Normal Weight']
df['BMI Category']=df['BMI Category'].replace(to_combine,'Normal')
```

```
df['BMI Category'].unique()
```

```
array(['Overweight', 'Normal', 'Obese'], dtype=object)
```

```
df=pd.concat([df,df['Blood Pressure'].str.split('/',expand=True)],axis=1)
df.head()
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Blood Pressur
0	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/8
1	Male	28	Doctor	6.2	6	60	8	Normal	125/8
2	Male	28	Doctor	6.2	6	60	8	Normal	125/8
3	Male	28	Sales Representative	5.9	4	30	8	Obese	140/9
4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/9

```
df=df.drop('Blood Pressure',axis=1)
df.head()
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate
0	Male	27	Software Engineer	6.1	6	42	6	Overweight	77
1	Male	28	Doctor	6.2	6	60	8	Normal	75
2	Male	28	Doctor	6.2	6	60	8	Normal	75
3	Male	28	Sales Representative	5.9	4	30	8	Obese	85
	Male		Sales						

rename

```
df=df.rename(columns={0: 'bp_upper',1: 'bp_lower'})
df.head()
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate
0	Male	27	Software Engineer	6.1	6	42	6	Overweight	77
1	Male	28	Doctor	6.2	6	60	8	Normal	75
2	Male	28	Doctor	6.2	6	60	8	Normal	75
3	Male	28	Sales Representative	5.9	4	30	8	Obese	85
4	Male	28	Sales Representative	5.9	4	30	8	Obese	85

```
df['bp_upper']= df['bp_upper'].astype('int')
df['bp_lower']=df['bp_lower'].astype('int')
```

```
df['Sleep Disorder'].unique()
```

```
array(['None', 'Sleep Apnea', 'Insomnia'], dtype=object)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 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   Heart Rate               374 non-null    int64
 9   Daily Steps              374 non-null    int64
 10  Sleep Disorder           374 non-null    object
 11  bp_upper                 374 non-null    int64
 12  bp_lower                 374 non-null    int64
dtypes: float64(1), int64(8), object(4)
memory usage: 38.1+ KB
```
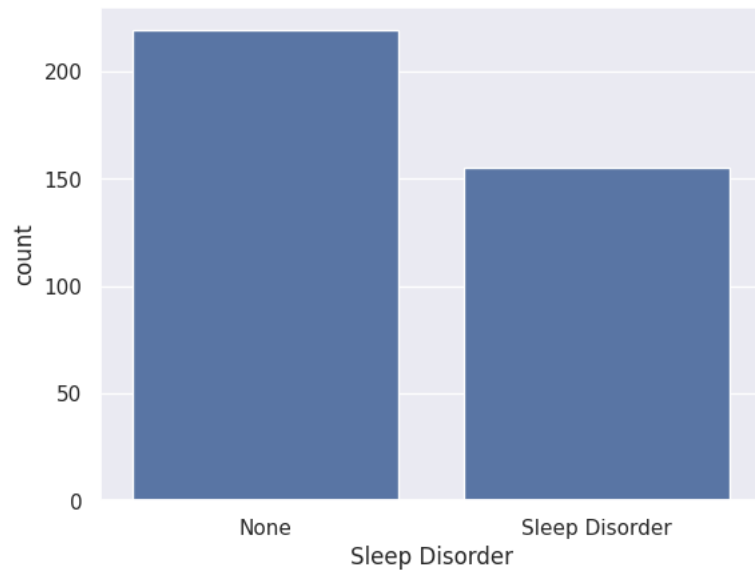
EXPLORATORY

```
sns.countplot(x='Sleep Disorder',data=df)
plt.show()
```
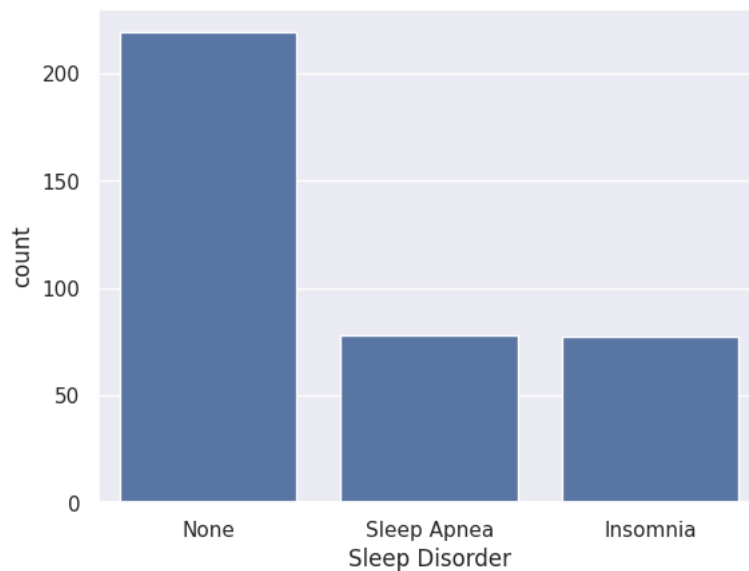


```
to_combine= ['Sleep Apnea','Insomnia']
df['BMI Category']=df['BMI Category'].replace(to_combine,'Normal')
```
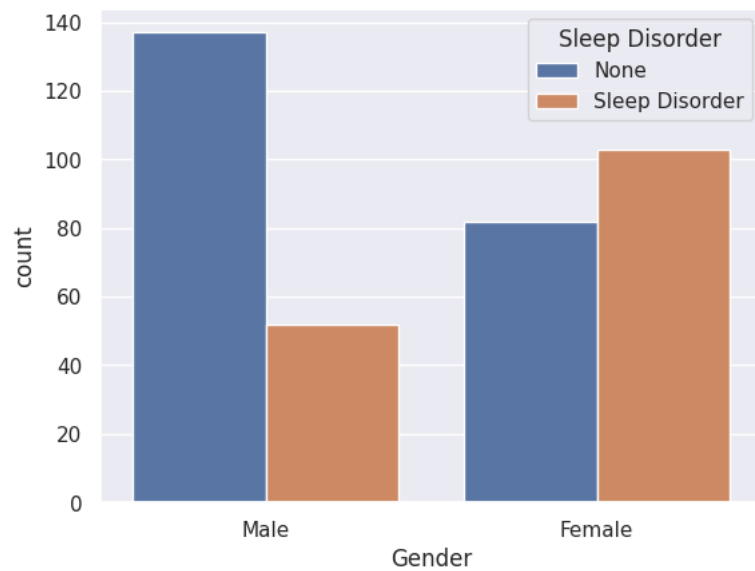
```
to_combine=['Sleep Apnea','Insomnia']
df['Sleep Disorder']=df['Sleep Disorder'].replace(to_combine,'Sleep Disorder')
```

```
sns.countplot(x='Sleep Disorder',data=df)
plt.show()
```
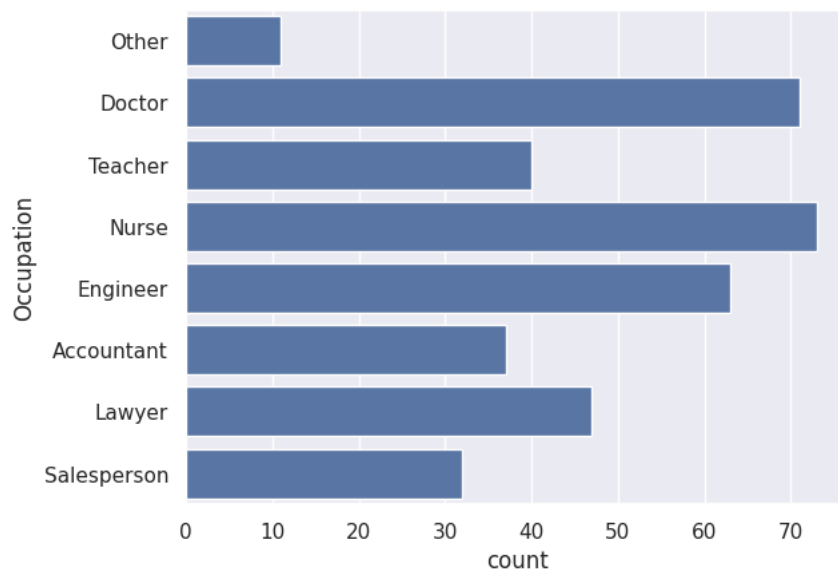
```
sns.countplot(x='Gender',data=df,hue='Sleep Disorder')
plt.show
```

    <function matplotlib.pyplot.show(close=None, block=None)>


```
sns.countplot(y='Occupation',data=df)
plt.show
```

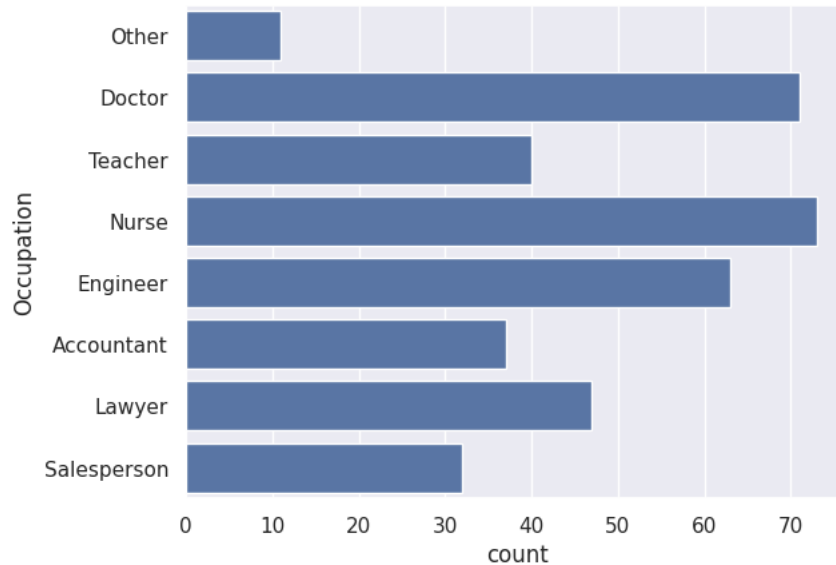   <function matplotlib.pyplot.show(close=None, block=None)>

```
to_combine=['Software Engineer','Sales Representative','Scientist','Manager']
df['Occupation']=df['Occupation'].replace(to_combine,'Other')
```

```
df['Occupation'].unique()
```
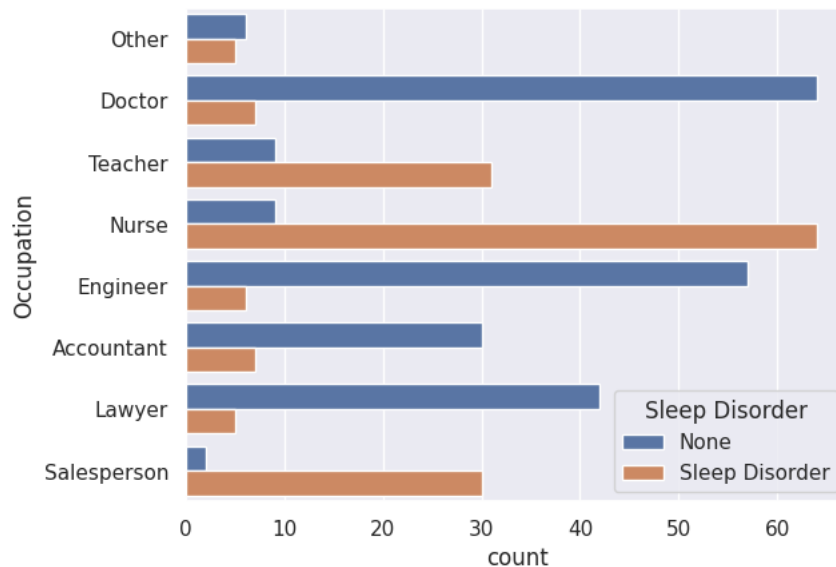
```
array(['Other', 'Doctor', 'Teacher', 'Nurse', 'Engineer', 'Accountant',
       'Lawyer', 'Salesperson'], dtype=object)
```

```
sns.countplot(y='Occupation',data=df)
plt.show()
```
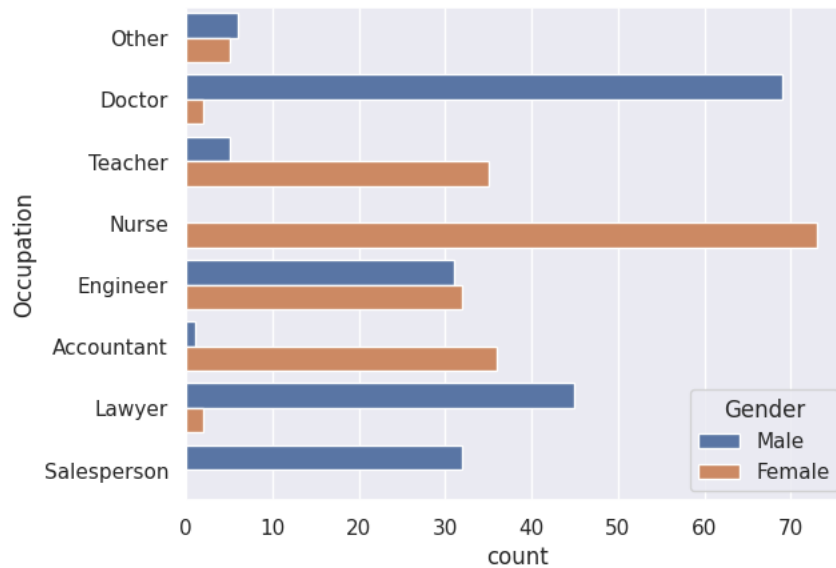


```
sns.countplot(y='Occupation',data=df,hue='Sleep Disorder')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```
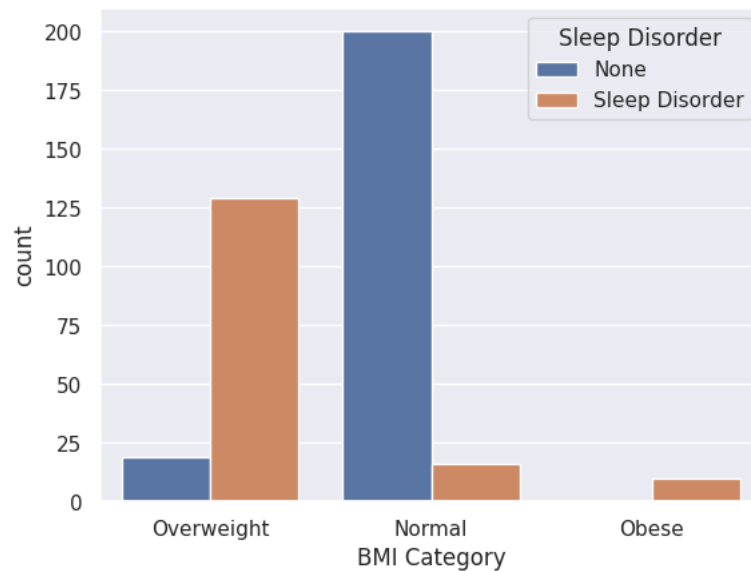


```
sns.countplot(y='Occupation',data=df,hue='Gender')
plt.showsns.countplot(y='Occupation',data=df,hue='Gender')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



```
sns.countplot(x='BMI Category',data=df,hue='Sleep Disorder')
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```


ALL obese and over weight people having sleep disorder

Label Encoder

```
from sklearn.preprocessing import LabelEncoder
```

```
enc=LabelEncoder()
```

```
df_transformed=df.copy()
df_transformed.head()
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate	Da St
0	Male	27	Other	6.1	6	42	6	Overweight	77	4
1	Male	28	Doctor	6.2	6	60	8	Normal	75	10
2	Male	28	Doctor	6.2	6	60	8	Normal	75	10
3	Male	28	Other	5.9	4	30	8	Obese	85	3
4	Male	28	Other	5.9	4	30	8	Obese	85	3

```
df_transformed['Gender']= enc.fit_transform(df['Gender'])
df_transformed
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate
0	1	27	Other	6.1	6	42	6	Overweight	77
1	1	28	Doctor	6.2	6	60	8	Normal	75
2	1	28	Doctor	6.2	6	60	8	Normal	75
3	1	28	Other	5.9	4	30	8	Obese	85
4	1	28	Other	5.9	4	30	8	Obese	85
...	...	...	...	...	...	...	...	...	...
369	0	59	Nurse	8.1	9	75	3	Overweight	68
370	0	59	Nurse	8.0	9	75	3	Overweight	68
371	0	59	Nurse	8.1	9	75	3	Overweight	68
372	0	59	Nurse	8.1	9	75	3	Overweight	68
373	0	59	Nurse	8.1	9	75	3	Overweight	68

374 rows × 13 columns

```
df_transformed['Occupation']= enc.fit_transform(df['Occupation'])
df_transformed['BMI Category']= enc.fit_transform(df['BMI Category'])
df_transformed['Sleep Disorder']= enc.fit_transform(df['Sleep Disorder'])
df_transformed
```

	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate	D S
0	1	27	5	6.1	6	42	6	2	77	
1	1	28	1	6.2	6	60	8	0	75	1
2	1	28	1	6.2	6	60	8	0	75	1
3	1	28	5	5.9	4	30	8	1	85	
4	1	28	5	5.9	4	30	8	1	85	
...	...	...	...	...	...	...	...	...	...	
369	0	59	4	8.1	9	75	3	2	68	
370	0	59	4	8.0	9	75	3	2	68	
371	0	59	4	8.1	9	75	3	2	68	
372	0	59	4	8.1	9	75	3	2	68	
373	0	59	4	8.1	9	75	3	2	68	

374 rows × 13 columns

model

```
df_transformed.columns
```

```
Index(['Gender', 'Age', 'Occupation', 'Sleep Duration', 'Quality of Sleep',
       'Physical Activity Level', 'Stress Level', 'BMI Category', 'Heart Rate',
       'Daily Steps', 'Sleep Disorder', 'bp_upper', 'bp_lower'],
      dtype='object')
```

```
inputs=df_transformed[['Gender', 'Age', 'Occupation', 'Sleep Duration', 'Quality of Sleep',
       'Physical Activity Level', 'Stress Level', 'BMI Category', 'Heart Rate',
       'Daily Steps', 'Sleep Disorder', 'bp_upper', 'bp_lower']]
```

```
target=df_transformed['Sleep Disorder']
```

TRAIN AND TEST

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(inputs,target,test_size=0.3,random_state=10)
```

```
from sklearn.metrics import accuracy_score,confusion_matrix
```

LOGISTIC REGRESSION

```
from sklearn.linear_model import LogisticRegression
reg=LogisticRegression()
reg.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Conver
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
  ▼ LogisticRegression
  LogisticRegression()
```

```
log_pred=reg.predict(x_test)
```

```
confusion_matrix(y_test,log_pred)
```

```
array([[66,  5],
       [ 2, 40]])
```

```
accuracy_score(y_test,log_pred)
```

```
0.9380530973451328
```

svm