LOADING THE DATA SETS

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

sleep_df = pd.read_csv("/Sleep_health_and_lifestyle_dataset.csv")
sleep_df.shape
sleep_df.to_csv("/Sleep_health_and_lifestyle_cleaned.csv")
```

```
sleep_df.head()
```

	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	8	30	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

```
sleep_df.tail()
```

	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
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	140/95	68	7000	Sleep Apnea

```
sleep_df.info()
```

```
<class 'DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13):
 #   Person ID                374 non-null    int64
 #   Gender                   374 non-null    object
 #   Age                      374 non-null    int64
 #   Occupation               374 non-null    object
 #   Sleep Duration           374 non-null    float64
 #   Quality of Sleep         374 non-null    int64
 #   Physical Activity Level  374 non-null    int64
 #   Stress Level             374 non-null    int64
 #   BMI Category             374 non-null    object
 #   Blood Pressure           374 non-null    object
 #   Heart Rate               374 non-null    int64
 #   Daily Steps              374 non-null    int64
 #   Sleep Disorder           374 non-null    object
dtype: float64(1), int64(9), object(3)
memory usage: 38.0+ KB
```

Finding Null Values

```
sleep_df.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: float64
```

```
sleep_df.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: float64
```

```
sleep_df.describe()
```

	Person ID	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Heart Rate	Daily Steps
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000
mean	187.500000	42.184492	7.132086	7.312834	59.171123	5.385027	70.165775	6816.844920
std	108.119735	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676	1617.915679
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000	3000.000000
25%	94.250000	35.250000	6.400000	6.000000	32.250000	3.000000	68.000000	5000.000000
50%	187.500000	43.000000	7.200000	7.500000	45.000000	5.000000	70.000000	7000.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000	8000.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000	10000.000000

```
sleep_df.shape
```

```
(374, 13)
```

```
sleep_df.dtypes
```

```
Person ID                  int64
Gender                    object
Age                        int64
Occupation                object
Sleep Duration           float64
Quality of Sleep           int64
Physical Activity Level    int64
Stress Level               int64
BMI Category              object
Blood Pressure            object
Heart Rate                 int64
Daily Steps                int64
Sleep Disorder            object
dtype: object
```

```
sleep_df.columns
```

```
Index(['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'],
      dtype: object
```

```
# Separating Textual Features from Numerics and Categories
sleep_df['Systolic_BP'] = sleep_df['Blood Pressure'].str.split('/').astype(int)
sleep_df['Diastolic_BP'] = sleep_df['Blood Pressure'].str.split('/').astype(int)
sleep_df['Occupation_Dummy'] = pd.Categorical(sleep_df['Occupation'])
sleep_df['Occupation_Dummy'].head()
```

```
0    Software Engineer
1               Doctor
2               Doctor
3    Sales Representative
4    Sales Representative
Name: Occupation_Dummy, dtype: object
```

```
# Assigning "Gender" to "Gender" in Sleep_df for cross-Mapping
sleep_df['Gender'] = sleep_df['Gender'].map({'Male': 1, 'Female': 0})
```

```
# Displaying the updated DataFrame
sleep_df.head()
```

```
Person ID                  1
Gender                     1
Age                       27
Occupation                Software Engineer
Sleep Duration             6.1
Quality of Sleep           6
Physical Activity Level    42
Stress Level               6
BMI Category               Overweight
Blood Pressure             126/83
Heart Rate                 77
Daily Steps                4200
Sleep Disorder             None
Systolic_BP                126
Diastolic_BP               126
Occupation_Dummy           Software Engineer
dtype: object
```

```
sleep_df.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
Systolic_BP                0
Diastolic_BP               0
Occupation_Dummy           0
dtype: int64
```

```
sleep_df.dtypes
```

```
Person ID                  int64
Gender                     int64
Age                        int64
Occupation                object
Sleep Duration           float64
Quality of Sleep           int64
Physical Activity Level    int64
Stress Level               int64
BMI Category              object
Blood Pressure            object
Heart Rate                 int64
Daily Steps                int64
Sleep Disorder            object
Systolic_BP                int64
Diastolic_BP               int64
Occupation_Dummy          object
dtype: object
```

```
# Displaying Textual Features and Numerics
sleep_df[['Occupation', 'Blood Pressure', 'Sleep Disorder', 'Systolic_BP', 'Diastolic_BP']].head()
```

	Occupation	Blood Pressure	Sleep Disorder	Systolic_BP	Diastolic_BP
0	Software Engineer	126/83	None	126	126
1	Doctor	125/80	None	125	125
2	Doctor	125/80	None	125	125
3	Sales Representative	140/90	Sleep Apnea	140	140
4	Sales Representative	140/90	Sleep Apnea	140	140

OUTLIER DETECTION:

```
def Remove_Outliers(df, columns):
    for col in columns:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df[col] = df[col].clip(lower_bound, upper_bound)
    return df[columns]

Sleep_Outliers = Remove_Outliers(sleep_df, ['Sleep Duration', 'Heart Rate', 'Stress Level'])
print("Sleep Outliers Shape:", Sleep_Outliers.shape)
Sleep_Outliers = Remove_Outliers(sleep_df, ['Systolic_BP', 'Diastolic_BP'])

print("Outliers Removed Sleep Dataset")
print("Sleep Outliers Shape:", Sleep_Outliers.shape)
print("Systolic_BP Outliers Shape:", Systolic_BP_Outliers.shape)
print("Diastolic_BP Outliers Shape:", Diastolic_BP_Outliers.shape)

print("Sleep Outliers Head:")
Sleep_Outliers.head()
print("Systolic_BP Outliers Head:")
Systolic_BP_Outliers.head()
print("Diastolic_BP Outliers Head:")
Diastolic_BP_Outliers.head()

Outliers in Sleep Dataset:
Sleep Duration: 0 outliers
Heart Rate: 0 outliers
Stress Level: 0 outliers

Outliers in Blood Pressure:
Systolic_BP: 0 outliers
Diastolic_BP: 0 outliers

Outliers in Sex Dataset:
Gender: 0 outliers
Age: 0 outliers
```

Fixing Outliers by Capping Method

```
def cap_outliers(df, columns):
    for col in columns:
        q1 = df[col].quantile(0.25)
        q3 = df[col].quantile(0.75)
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        df[col] = df[col].clip(lower_bound, upper_bound)
    return df[columns]

cleaned_sleep_df = cap_outliers(sleep_df, ['Sleep Duration', 'Heart Rate', 'Stress Level'])
print("Cleaned Sleep Dataset Shape:", cleaned_sleep_df.shape)
print("Cleaned Sleep Dataset Head:")
cleaned_sleep_df.head()

Cleaned Sleep Dataset Shape: (374, 3)
Cleaned Sleep Dataset Head:
```

	Sleep Duration	Heart Rate	Stress Level
0	6.1	77	6
1	6.2	75	8
2	6	75	8
3	5.9	85	8
4	5.9	85	8

```
# Displaying the updated sleep dataset
sleep_df.head()
```

	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	Systolic_BP	Diastolic_BP
0	1	Male	27	Software Engineer	6.1	6	42	6	Overweight	126/83	77	4200	None	126	126
1	2	Male	28	Doctor	6.2	6	60	8	Normal	125/80	75	10000	None	125	125
2	3	Male	28	Doctor	6	8	30	8	Normal	125/80	75	10000	None	125	125
3	4	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea	140	140
4	5	Male	28	Sales Representative	5.9	4	30	8	Obese	140/90	85	3000	Sleep Apnea	140	140

```
# Displaying the updated sleep dataset
sleep_df.to_csv("/Sleep_Health_and_Lifestyle_Cleaned_Updated.csv")
sleep_df.shape
```

```
(374, 15)
```

```
# Exploratory Data Analysis (EDA)
sleep_df.describe()
```

	Person ID	Gender	Age	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Heart Rate	Daily Steps	Sleep Disorder	Systolic_BP	Diastolic_BP
count	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000	374.000000
mean	187.500000	0.510695	42.184492	7.132086	7.312834	59.171123	5.385027	1.000000	70.165775	6816.844920	0.000000	128.553476	128.553476
std	108.119735	0.500668	8.673133	0.795657	1.196956	20.830804	1.774526	0.000000	4.135676	1617.915679	0.000000	18.807427	18.807427
min	1.000000	0.000000	27.000000	5.800000	4.000000	30.000000	3.000000	1.000000	65.000000	3000.000000	0.000000	115.000000	115.000000
25%	94.250000	0.000000	35.250000	6.400000	6.000000	32.250000	3.000000	1.000000	68.000000	5000.000000	0.000000	125.000000	125.000000
50%	187.500000	1.000000	43.000000	7.200000	7.500000	45.000000	5.000000	1.000000	70.000000	7000.000000	0.000000	128.000000	128.000000
75%	280.750000	1.000000	50.000000	7.800000	8.000000	75.000000	7.000000	1.000000	72.000000	8000.000000	0.000000	140.000000	140.000000
max	374.000000	1.000000	59.000000	8.500000	9.000000	90.000000	8.000000	1.000000	86.000000	10000.000000	1.000000	142.000000	142.000000

```
# Count of Heart Disease Cases
plt.figure(figsize=(6,4))
sns.countplot(data=sleep_df, x='Heart Disease', palette='Pastel1')
plt.title('Count of Heart Disease Cases')
plt.xlabel('Heart Disease')
plt.ylabel('Count')
plt.xticks([0, 1], ['No Heart Disease', 'Heart Disease'])
plt.show()
```

```
correlation_columns = ['Sleep Duration', 'Quality of Sleep', 'Physical Activity Level',
                       'Stress Level', 'Heart Rate', 'Daily Steps', 'Systolic_BP',
                       'Diastolic_BP', 'Heart Disease']
correlation_df = sleep_df[correlation_columns].dropna()
```

```
# Correlation Heatmap
plt.figure(figsize=(12,8))
sns.heatmap(correlation_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

```
# Separating BP Categories and Sleep Quality into Labelled Groups
sleep_df['BP_Category'] = pd.cut(sleep_df['Systolic_BP'], bins=[0, 120, 130, 140, 180], labels=['Normal', 'Elevated', 'Stage 1 Hypertension', 'Stage 2 Hypertension'])
sleep_df['Sleep_Quality_Group'] = pd.cut(sleep_df['Quality of Sleep'], bins=[0, 5, 7, 9], labels=['Poor', 'Average', 'Good'])
sleep_df[['Systolic_BP', 'BP_Category', 'Quality of Sleep', 'Sleep_Quality_Group']].head()
```

	Systolic_BP	BP_Category	Quality of Sleep	Sleep_Quality_Group
0	126	Elevated	6	Average
1	125	Elevated	6	Average
2	125	Elevated	8	Good
3	140	Stage 1 Hypertension	4	Poor
4	140	Stage 1 Hypertension	4	Poor

```
# Displaying BP Category and Sleep Quality Groups
print("BP Category:", sleep_df['BP_Category'].value_counts())
print("Sleep Quality Group:", sleep_df['Sleep_Quality_Group'].value_counts())
```

```
BP Category: BP_Category
Elevated                 150
Stage 1 Hypertension     148
Normal                    76
Stage 2 Hypertension      0
Name: count, dtype: int64
Sleep Quality Group: Sleep_Quality_Group
Average    [6, 7)
Good       [7, 9]
Poor       [0, 5]
Name: count, dtype: int64
```

```
# Assigning BP Category and Sleep Quality Groups to Dummies Columns
sleep_df = pd.get_dummies(sleep_df, columns=['BP_Category', 'Sleep_Quality_Group'])
print("Dummies Columns:", sleep_df.columns[-6:])
```

```
Dummies Columns: Index(['BP_Category_Normal', 'BP_Category_Elevated',
       'BP_Category_Stage 1 Hypertension', 'BP_Category_Stage 2 Hypertension',
       'Sleep_Quality_Group_Poor', 'Sleep_Quality_Group_Average'],
      dtype: object)
```

```
# Assigning Dummies Columns to Sleep Dataset
sleep_df['BP_Normal'] = sleep_df['BP_Category_Normal']
sleep_df['BP_Elevated'] = sleep_df['BP_Category_Elevated']
sleep_df['BP_Stage_1_Hypertension'] = sleep_df['BP_Category_Stage 1 Hypertension']
sleep_df['BP_Stage_2_Hypertension'] = sleep_df['BP_Category_Stage 2 Hypertension']
sleep_df['Sleep_Quality_Poor'] = sleep_df['Sleep_Quality_Group_Poor']
sleep_df['Sleep_Quality_Average'] = sleep_df['Sleep_Quality_Group_Average']
sleep_df['Sleep_Quality_Good'] = sleep_df['Sleep_Quality_Group_Good']
sleep_df.head()
```

```
Dummies Columns: Index(['BP_Normal', 'BP_Elevated', 'BP_Stage_1_Hypertension',
       'BP_Stage_2_Hypertension', 'Sleep_Quality_Poor',
       'Sleep_Quality_Average', 'Sleep_Quality_Good'],
      dtype: object)
```

```
# Assigning Dummies Columns to Sleep Dataset
sleep_df['BP_Normal'] = sleep_df['BP_Category_Normal']
sleep_df['BP_Elevated'] = sleep_df['BP_Category_Elevated']
sleep_df['BP_Stage_1_Hypertension'] = sleep_df['BP_Category_Stage 1 Hypertension']
sleep_df['BP_Stage_2_Hypertension'] = sleep_df['BP_Category_Stage 2 Hypertension']
sleep_df['Sleep_Quality_Poor'] = sleep_df['Sleep_Quality_Group_Poor']
sleep_df['Sleep_Quality_Average'] = sleep_df['Sleep_Quality_Group_Average']
sleep_df['Sleep_Quality_Good'] = sleep_df['Sleep_Quality_Group_Good']
sleep_df.head()
```

```
Person ID                  int64
Gender                     int64
Age                        int64
Occupation                object
Sleep Duration           float64
Quality of Sleep           int64
Physical Activity Level    int64
Stress Level               int64
BMI Category              object
Blood Pressure            object
Heart Rate                 int64
Daily Steps                int64
Sleep Disorder            object
Systolic_BP                int64
Diastolic_BP               int64
Occupation_Dummy          object
BP_Normal                 bool
BP_Elevated               bool
BP_Stage_1_Hypertension   bool
BP_Stage_2_Hypertension   bool
Sleep_Quality_Poor        bool
Sleep_Quality_Average     bool
Sleep_Quality_Good        bool
dtype: object
```

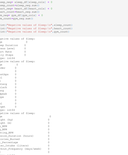
```
# Exploring Sleep Duration Distribution
plt.figure(figsize=(6,4))
sns.histplot(data=sleep_df, x='Sleep Duration', kde=True, color='Pink')
plt.title('Distribution of Sleep Duration')
plt.xlabel('Sleep Duration')
plt.ylabel('Count')
plt.show()
```



```
# Exploring Sleep Duration Distribution by Gender
sns.boxplot(data=sleep_df, x='Sleep Duration', y='Gender', palette='Pastel1')
plt.title('Sleep Duration Distribution by Gender')
plt.xlabel('Sleep Duration')
plt.ylabel('Gender')
plt.yticks([0, 1], ['Female', 'Male'])
plt.show()
```

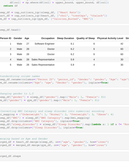
```
# Exploring Sleep Duration Distribution by Occupation
plt.figure(figsize=(12,6))
sns.boxplot(data=sleep_df, x='Sleep Duration', y='Occupation', palette='Pastel1')
plt.title('Sleep Duration Distribution by Occupation')
plt.xlabel('Sleep Duration')
plt.ylabel('Occupation')
plt.show()
```

```
# Exploring Sleep Duration Distribution by Stress Level
plt.figure(figsize=(6,4))
sns.boxplot(data=sleep_df, x='Stress Level', y='Sleep Duration', palette='Pastel1')
plt.title('Sleep Duration Distribution by Stress Level')
plt.xlabel('Stress Level')
plt.ylabel('Sleep Duration')
plt.show()
```




```
# Exploring Sleep Duration Distribution by Heart Rate
plt.figure(figsize=(6,4))
sns.boxplot(data=sleep_df, x='Heart Rate', y='Sleep Duration', palette='Pastel1')
plt.title('Sleep Duration Distribution by Heart Rate')
plt.xlabel('Heart Rate')
plt.ylabel('Sleep Duration')
plt.show()
```

```
# Exploring Sleep Duration Distribution by Blood Pressure
plt.figure(figsize=(12,6))
sns.boxplot(data=sleep_df, x='Sleep Duration', y='Blood Pressure', palette='Pastel1')
plt.title('Sleep Duration Distribution by Blood Pressure')
plt.xlabel('Sleep Duration')
plt.ylabel('Blood Pressure')
plt.show()
```




```
# Exploring Correlation between Sleep Duration,Heart Rate and Stress Level
correlation_columns = ['Sleep Duration', 'Heart Rate', 'Stress Level']
correlation_matrix = sleep_df[correlation_columns].corr()
```

```
# Correlation Heatmap
plt.figure(figsize=(6,4))
sns.heatmap(data=correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap for Sleep Duration')
plt.show()
```

```
# Exploring Correlation between Sleep Duration,Heart Rate and Stress Level
correlation_columns = ['Sleep Duration', 'Heart Rate', 'Stress Level']
correlation_matrix = sleep_df[correlation_columns].corr()
```

```
# Correlation Heatmap
plt.figure(figsize=(6,4))
sns.heatmap(data=correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap for Sleep Duration')
plt.show()
```