In [1]:
```
# 10/27/2025
# BSAN 360 Lab 4
# Lucas Doyle
```

In [2]:
```
# 1. Download and read your data into a data frame. Check that the process does not
import pandas as pd
```

In [3]:
```
# 1
import os
os.chdir("C:/Users/lucas/Programming")
print(os.getcwd())
```

```
C:\Users\lucas\Programming
```

In [4]:
```
# 1
sleep = pd.read_csv('./Sleep_health_and_lifestyle_dataset.csv')
sleep.head()
```

Out[4]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Pr
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	

In [5]:
```
# 2. How many rows and columns does your data frame have? Check that this is the co
print("Shape of dataset:", sleep.shape)
print("Number of rows:", sleep.shape[0])
print("Number of columns:", sleep.shape[1])
```

```
Shape of dataset: (374, 13)
Number of rows: 374
Number of columns: 13
```

In [6]:
```
# 2
# 372 rows
# 13 columns
```

In [7]:
```
# 3 Check the first and last few rows of data. Does everything look correct?
sleep.head()
```

Out[7]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Pr
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	

In [8]:
```
# 3
sleep.tail()
```

Out[8]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	BMI Category	Pr
369	370	Female	59	Nurse	8.1	9	75	3	Overweight	
370	371	Female	59	Nurse	8.0	9	75	3	Overweight	
371	372	Female	59	Nurse	8.1	9	75	3	Overweight	
372	373	Female	59	Nurse	8.1	9	75	3	Overweight	
373	374	Female	59	Nurse	8.1	9	75	3	Overweight	

In [9]:
```
# 4. Make sure that column and row labels got processed correctly. Re-label columns
sleep.columns
```

Out[9]:
```
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')
```

In [10]:
```
# 4
sleep.rename(columns={'Physical Activity Level': 'Activity'}, inplace=True)
sleep.columns
```

Out[10]:
```
Index(['Person ID', 'Gender', 'Age', 'Occupation', 'Sleep Duration',
       'Quality of Sleep', 'Activity', 'Stress Level', 'BMI Category',
       'Blood Pressure', 'Heart Rate', 'Daily Steps', 'Sleep Disorder'],
      dtype='object')
```

In [11]:
```
# 5. Look at the descriptive statistics and perform the following checks and any ne
sleep.describe()
```

Out[11]:

	Person ID	Age	Sleep Duration	Quality of Sleep	Activity	Stress Level	Heart Rate
count	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
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000

In [12]:
```
# 5
sleep.describe(include='object')
```

Out[12]:

	Gender	Occupation	BMI Category	Blood Pressure	Sleep Disorder
count	374	374	374	374	155
unique	2	11	4	25	2
top	Male	Nurse	Normal	130/85	Sleep Apnea
freq	189	73	195	99	78

In [13]:
```
# 5
# a What information is provided in this column?
# Person ID - An identifier for each person
# Age - The participants range from 27 to 59. (mean is 42)
# Sleep Duration - Hours of sleep per night (min 5.8, max 8.5, mean ≈ 7.1). Reflect
# Quality of sleep - Self-rated quality on a 1–10 scale (range 4–9, mean ≈ 7.3). In
# Activity - Daily physical-activity score (30–90, mean ≈ 59). Suggests moderate to
# Stress Level - Self rated stress on a 1–10 scale (3–8, mean ≈ 5.4). Indicates mod
# Heart rate - Average resting heart rate in bpm (65–86, mean ≈ 70). All values wit
# Daily Steps - Steps taken per day (3 000–10 000, mean ≈ 6 817). Represents mostly
# gender - Two categories – Male (189) and Female (185). Balanced sample; no missin
# occupation - Eleven job types  most frequent = Nurse (73). Reflects varied profes
# BMI Category - Four categories – Normal (most common 195), Overweight, Obese, Und
# Blood Pressure - 25 unique readings (most common 130/85). Indicates variety in ca
# Sleep Disorder - Two diagnoses (Sleep Apnea = 78, Insomnia = 77) with 155 non-nul
```

In [14]:
```
# 5 b What type of data is given in this column? The types we've seen most are Nume
# (integer or real number), Character Strings, Boolean, Datetime. If the data type
# correct, correct it as best you can. (A full resetting of data types may require
# some handling of missing values, which we will cover on Wednesday, so no worries
sleep.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   Activity          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    155 non-null    object 
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

In [15]:
```
# 5 C What is the range of values in this column? Check the min and max values to m
sleep.describe()
```

Out[15]:

	Person ID	Age	Sleep Duration	Quality of Sleep	Activity	Stress Level	Heart Rate
count	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
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000

In [16]:
```
# 5 d Are there any missing values in this column? If so, how do you plan to handle
sleep.isnull().sum()
#  Only the Sleep Disorder column has missing values (219).
```

Out[16]:
```
Person ID              0
Gender                 0
Age                    0
Occupation             0
Sleep Duration         0
Quality of Sleep       0
Activity               0
Stress Level           0
BMI Category           0
Blood Pressure         0
Heart Rate             0
Daily Steps            0
Sleep Disorder       219
dtype: int64
```

In [17]:
```
# 6. Next, we will focus on the project questions and figure out a roadmap for your
# your project questions from your assignment here, and for each question, determin

# Project Question:
# How do lifestyle factors like stress and activity relate to sleep duration and sl

# Columns needed:
# - Sleep Duration
# - Quality of Sleep
# - Stress Level
# - Activity

# Plan:
# Compare sleep duration and sleep quality against stress and activity levels
# to see if healthier daytime behaviors lead to better sleep.
```

In [18]:
```
# 7. Create a new data frame that contains all the columns you'll need for your pro
# computer's memory allows, create individual data frames for each of your research

sleep_project = sleep[['Sleep Duration', 'Quality of Sleep', 'Stress Level', 'Activ

sleep_project.head()
```

Out[18]:

	Sleep Duration	Quality of Sleep	Stress Level	Activity
0	6.1	6	6	42
1	6.2	6	8	60
2	6.2	6	8	60
3	5.9	4	8	30
4	5.9	4	8	30

In [19]:
```
# 7
stress_sleep = sleep[['Stress Level', 'Sleep Duration', 'Quality of Sleep']]
activity_sleep = sleep[['Activity', 'Sleep Duration', 'Quality of Sleep']]
```

In [20]:
```
# 8 Going back to your research questions themselves, determine what type of statis
# you'll need to answer each one. Some examples could be ANOVA to determine if ther
# differences among groups, regression for prediction, time series analysis for for
# classification or clustering methods

# Analyses I Will Use:
# - Correlation analysis  to see if there is a relationship between stress, activit
# - Simple linear regression  to predict sleep duration or sleep quality from one f
# - Multiple linear regression to test how both stress and activity together affect
```

In [21]:
```
# Project assignment 2
# PROJECT DATA CLEANING AND PREPARATION
sleep_project = sleep[['Sleep Duration', 'Quality of Sleep', 'Stress Level', 'Activ
sleep_project.head()
```

Out[21]:

	Sleep Duration	Quality of Sleep	Stress Level	Activity
0	6.1	6	6	42
1	6.2	6	8	60
2	6.2	6	8	60
3	5.9	4	8	30
4	5.9	4	8	30

In [22]:
```
# looking for missing values
sleep_project.isnull().sum()
```

Out[22]:
```
Sleep Duration      0
Quality of Sleep    0
Stress Level        0
Activity            0
dtype: int64
```

In [23]:
```
# All four columns (Sleep Duration, Quality of Sleep, Stress Level, Activity) have
# No further handling of missing data is required.
# This confirms the dataset is complete and ready for analysis.
```

In [24]:
```
# Data transformations
# making sure everything is numeric
sleep_project.dtypes
```

Out[24]:
```
Sleep Duration      float64
Quality of Sleep      int64
Stress Level          int64
Activity              int64
dtype: object
```

In [25]:
```
# All columns are numeric (float or int).
# No type conversions are needed.
```

In [26]:
```
# Final verification of cleaned data
#isplay descriptive statistics to confirm all values look reasonable
```

```
sleep_project.describe()
```

Out[26]:

	Sleep Duration	Quality of Sleep	Stress Level	Activity
count	374.000000	374.000000	374.000000	374.000000
mean	7.132086	7.312834	5.385027	59.171123
std	0.795657	1.196956	1.774526	20.830804
min	5.800000	4.000000	3.000000	30.000000
25%	6.400000	6.000000	4.000000	45.000000
50%	7.200000	7.000000	5.000000	60.000000
75%	7.800000	8.000000	7.000000	75.000000
max	8.500000	9.000000	8.000000	90.000000

In [27]:
```
# The dataset has been fully cleaned and verified.
# All numeric columns (Sleep Duration, Quality of Sleep, Stress Level, Activity)
# contain valid values within expected ranges.
```

In [28]:
```
# BSAN 360 - Week 07 Project
# Student Lucas Doyle
# Topics String Processing, Data Wrangling, Reshaping & Pivoting
```

In [29]:
```
# String Processing
sleep.dtypes
```

Out[29]:
```
Person ID              int64
Gender                object
Age                    int64
Occupation            object
Sleep Duration       float64
Quality of Sleep       int64
Activity               int64
Stress Level           int64
BMI Category          object
Blood Pressure        object
Heart Rate             int64
Daily Steps            int64
Sleep Disorder        object
dtype: object
```

In [30]:
```
# String Processing
string_cols = ['Gender', 'Occupation', 'BMI Category', 'Blood Pressure', 'Sleep Dis

for col in string_cols:
    print(f"\nColumn: {col}")
    print(sleep[col].unique()[:10])
```

```
Column: Gender
['Male' 'Female']

Column: Occupation
['Software Engineer' 'Doctor' 'Sales Representative' 'Teacher' 'Nurse'
 'Engineer' 'Accountant' 'Scientist' 'Lawyer' 'Salesperson']

Column: BMI Category
['Overweight' 'Normal' 'Obese' 'Normal Weight']

Column: Blood Pressure
['126/83' '125/80' '140/90' '120/80' '132/87' '130/86' '117/76' '118/76'
 '128/85' '131/86']

Column: Sleep Disorder
[nan 'Sleep Apnea' 'Insomnia']
```

In [31]:
```
# Processing Strings

string_cols = ['Gender', 'Occupation', 'BMI Category', 'Blood Pressure', 'Sleep Dis

sleep[string_cols] = (sleep[string_cols]
                      .apply(lambda s: s.str.strip().str.replace(r'\s+', ' ', regex
                      .apply(lambda s: s.str.title()))

sleep[string_cols].head()
```

Out[31]:

	Gender	Occupation	BMI Category	Blood Pressure	Sleep Disorder
0	Male	Software Engineer	Overweight	126/83	NaN
1	Male	Doctor	Normal	125/80	NaN
2	Male	Doctor	Normal	125/80	NaN
3	Male	Sales Representative	Obese	140/90	Sleep Apnea
4	Male	Sales Representative	Obese	140/90	Sleep Apnea

In [32]:
```
# Processing Strings

sleep['BMI Category'] = sleep['BMI Category'].replace({'Normal Weight': 'Normal'})

sleep['BMI Category'].value_counts()
```

Out[32]:
```
BMI Category
Normal        216
Overweight    148
Obese          10
Name: count, dtype: int64
```

In [33]:
```
# Processing Strings

bp = sleep['Blood Pressure'].str.strip()
sleep[['Systolic', 'Diastolic']] = bp.str.split('/', expand=True).astype('int64')
```

```
sleep[['Blood Pressure', 'Systolic', 'Diastolic']].head()
```

Out[33]:

	Blood Pressure	Systolic	Diastolic
0	126/83	126	83
1	125/80	125	80
2	125/80	125	80
3	140/90	140	90
4	140/90	140	90

In [34]:
```
# Processing Strings

sleep[['Gender', 'Occupation', 'BMI Category', 'Sleep Disorder']].head()
```

Out[34]:

	Gender	Occupation	BMI Category	Sleep Disorder
0	Male	Software Engineer	Overweight	NaN
1	Male	Doctor	Normal	NaN
2	Male	Doctor	Normal	NaN
3	Male	Sales Representative	Obese	Sleep Apnea
4	Male	Sales Representative	Obese	Sleep Apnea

In [35]:
```
# Part 2: Data Wrangling – Combining and Merging Datasets
import pandas as pd

datasets = [name for name in globals() if isinstance(globals()[name], pd.DataFrame)
print("Available DataFrames:", datasets)
```

```
Available DataFrames: ['__', '_4', 'sleep', '_7', '_8', '_11', '_12', '_15', '_18',
'sleep_project', 'stress_sleep', 'activity_sleep', '_21', '_26', '_31', '_33', '_3
4']
```

In [36]:
```
# Part 2: Data Wrangling – Combining and Merging Datasets
# Since only one primary dataset ('sleep') is used in this project, merging or conc
# The smaller DataFrames ('sleep_project', 'stress_sleep', 'activity_sleep') are su
# created for specific analyses, not separate files. Therefore, no merge operations
```

In [37]:
```
# Part 3 - Data Wrangling: Reshaping and Pivoting

sleep.info()
sleep.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 15 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   Activity          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    155 non-null    object 
 13  Systolic          374 non-null    int64  
 14  Diastolic         374 non-null    int64  
dtypes: float64(1), int64(9), object(5)
memory usage: 44.0+ KB
```

Out[37]:

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Activity	Stress Level	BMI Category	Pr
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	

In [38]:

```
# Part 3: Reshaping & Pivoting

#  Mean sleep metrics by Gender × BMI Category
df_pivot1 = (sleep
             .pivot_table(index='Gender',
                          columns='BMI Category',
                          values=['Sleep Duration','Quality of Sleep'],
                          aggfunc='mean')
             .round(2))
df_pivot1
```

Out[38]:

	Quality of Sleep			Sleep Duration		
BMI Category	**Normal**	**Obese**	**Overweight**	**Normal**	**Obese**	**Overweight**
Gender						
Female	8.28	7.00	7.22	7.70	7.40	6.88
Male	7.28	6.33	6.10	7.21	6.91	6.49

In [39]:
```
# Part 3: Reshaping & Pivoting

# Mean sleep/stress/activity by Occupation
df_pivot2 = (sleep
             .pivot_table(index='Occupation',
                          values=['Sleep Duration','Quality of Sleep','Stress Level
                          aggfunc='mean')
             .round(2))
df_pivot2
```

Out[39]:

	Activity	Quality of Sleep	Sleep Duration	Stress Level
Occupation				
Accountant	58.11	7.89	7.11	4.59
Doctor	55.35	6.65	6.97	6.73
Engineer	51.86	8.41	7.99	3.89
Lawyer	70.43	7.89	7.41	5.06
Manager	55.00	7.00	6.90	5.00
Nurse	78.59	7.37	7.06	5.55
Sales Representative	30.00	4.00	5.90	8.00
Salesperson	45.00	6.00	6.40	7.00
Scientist	41.00	5.00	6.00	7.00
Software Engineer	48.00	6.50	6.75	6.00
Teacher	45.62	6.98	6.69	4.53

In [40]:
```
# Part 3: Reshaping & Pivoting
# Melt to long format for plotting/analysis

metrics = ['Sleep Duration','Quality of Sleep','Stress Level','Activity']
df_long = (sleep[['Person ID'] + metrics]
           .melt(id_vars='Person ID', var_name='Metric', value_name='Value'))
df_long.head()
```

Out[40]:

	Person ID	Metric	Value
0	1	Sleep Duration	6.1
1	2	Sleep Duration	6.2
2	3	Sleep Duration	6.2
3	4	Sleep Duration	5.9
4	5	Sleep Duration	5.9

In [41]:
```
# Part 3: Reshaping & Pivoting
# Pivot back to wide
df_wide = df_long.pivot(index='Person ID', columns='Metric', values='Value')
df_wide.head()
```

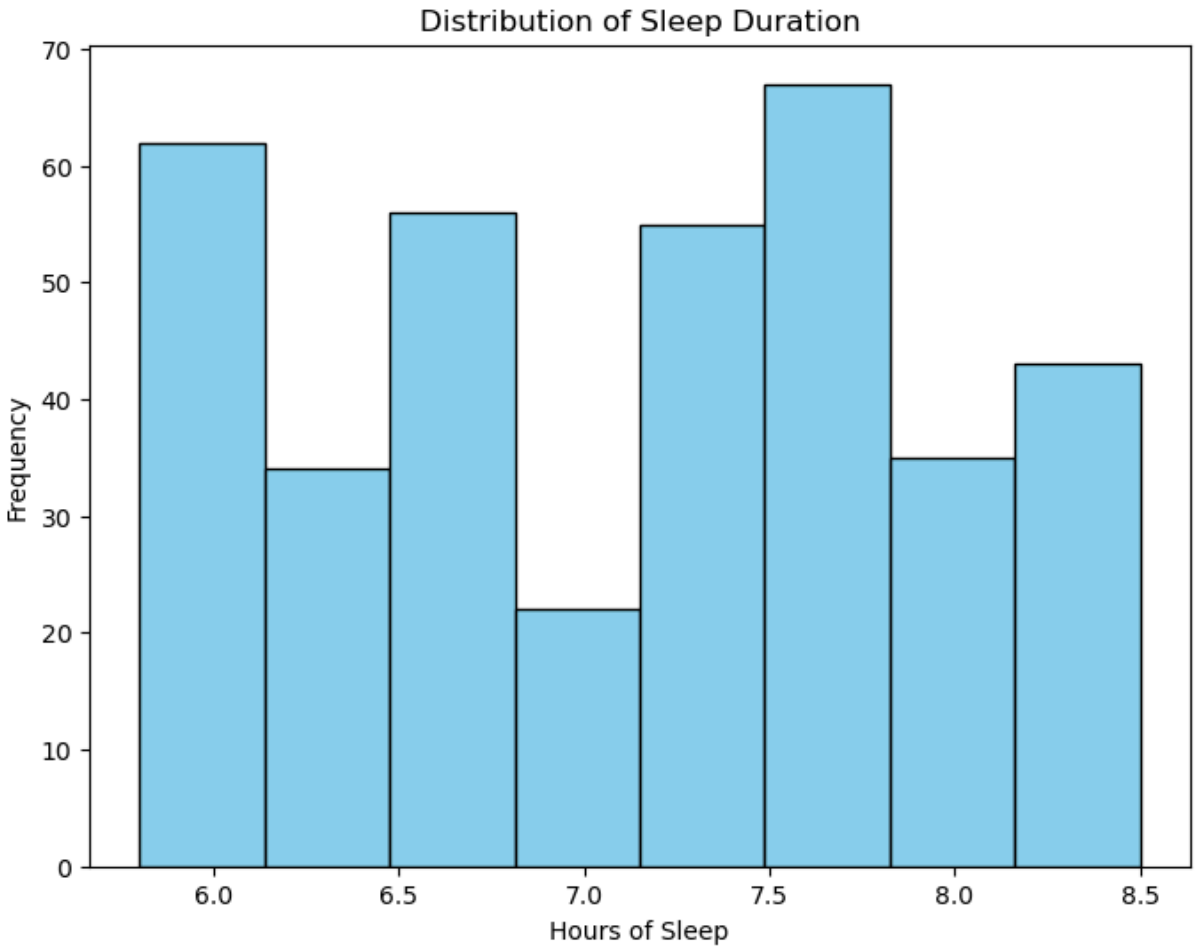
Out[41]:

Metric	Activity	Quality of Sleep	Sleep Duration	Stress Level
Person ID				
1	42.0	6.0	6.1	6.0
2	60.0	6.0	6.2	8.0
3	60.0	6.0	6.2	8.0
4	30.0	4.0	5.9	8.0
5	30.0	4.0	5.9	8.0

In [42]:
```
# Project 4
# For the data you have collected for your project, perform the following task(s):

# Plotting and Visualization
# A few things to watch out for:

# Any visualization you do for this week is expected to be geared towards explorato
# You can use the lecture notebook and/or the hands-on exercises to guide you with
```

In [43]:
```
#4
import matplotlib.pyplot as plt
```

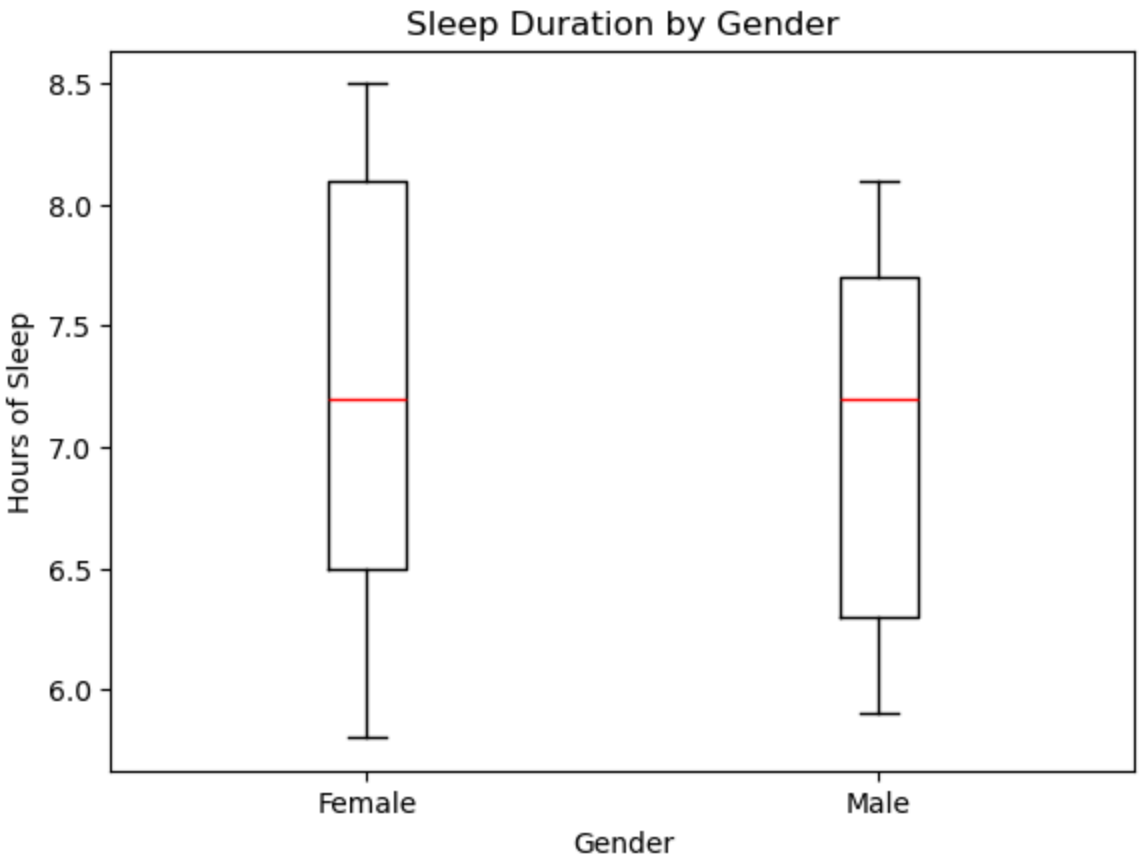
In [44]:
```
# Plotting and Visualization - histogram

plt.figure(figsize=(8,6))
plt.hist(sleep['Sleep Duration'], bins=8, color='skyblue', edgecolor='black')

plt.xlabel("Hours of Sleep")
plt.ylabel("Frequency")
plt.title("Distribution of Sleep Duration")

plt.show()
```



In [45]:
```
# The histogram shows the distribution of sleep duration in my dataset.
# It lets me see the most common sleep ranges and if the datga is skewed or spread.
# fairly even with no strong skew
```

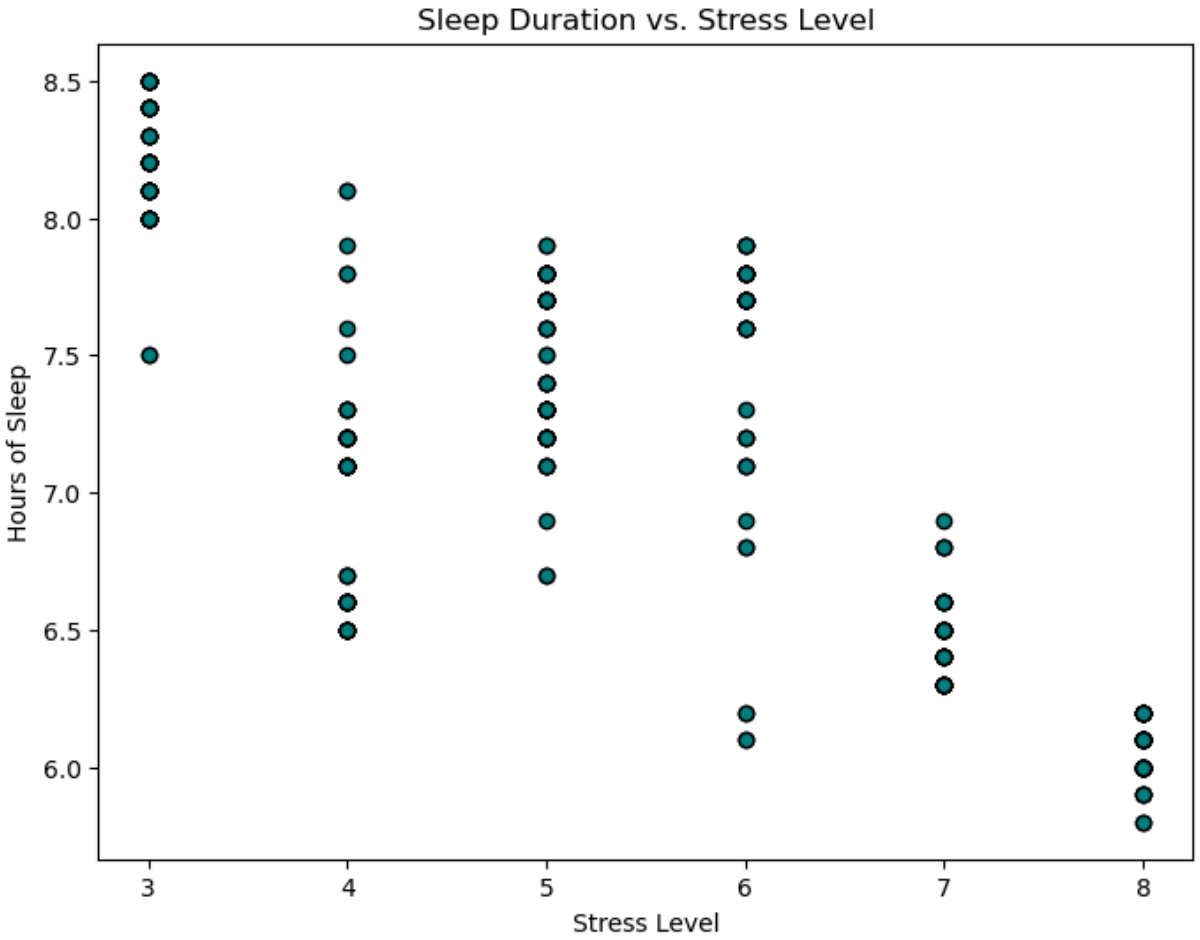
In [46]:
```
#  4 boxplot of sleep duration by gender

plt.figure(figsize=(8,6))
sleep.boxplot(column='Sleep Duration', by='Gender', grid=False,
              boxprops=dict(color='black'),
              medianprops=dict(color='red'),
              whiskerprops=dict(color='black'),
              capprops=dict(color='black'))

plt.xlabel("Gender")
plt.ylabel("Hours of Sleep")
plt.title("Sleep Duration by Gender")
plt.suptitle("")  # remove default title

plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



In [47]:
```
# Sleep duration looks very similar between males and females, with both groups cen
```

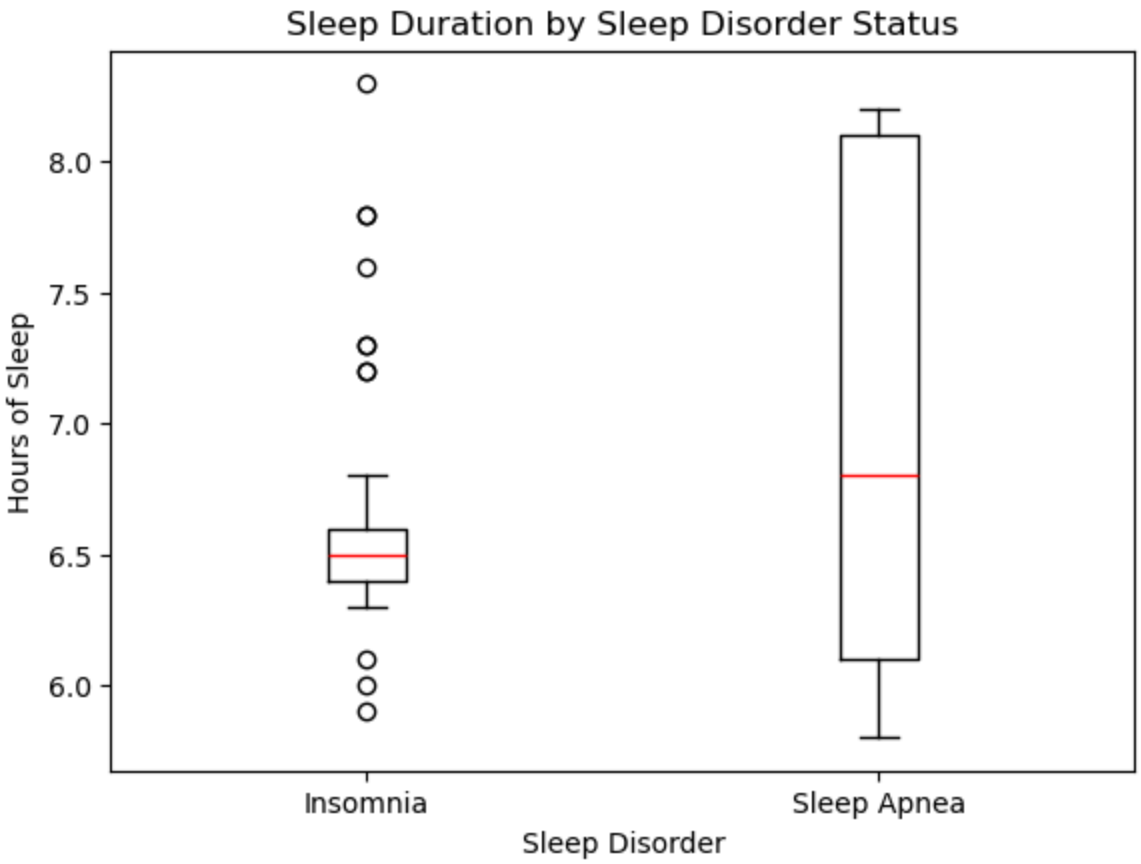
In [48]:
```
# scatterplot of sleep duration vs stress level
plt.figure(figsize=(8,6))
plt.scatter(sleep['Stress Level'], sleep['Sleep Duration'],
            color='teal', edgecolors='black')

plt.xlabel("Stress Level")
plt.ylabel("Hours of Sleep")
plt.title("Sleep Duration vs. Stress Level")

plt.show()
```



In [49]:
```
# As stress level increases, sleep duration tends to decrease slightly, showing a l
```

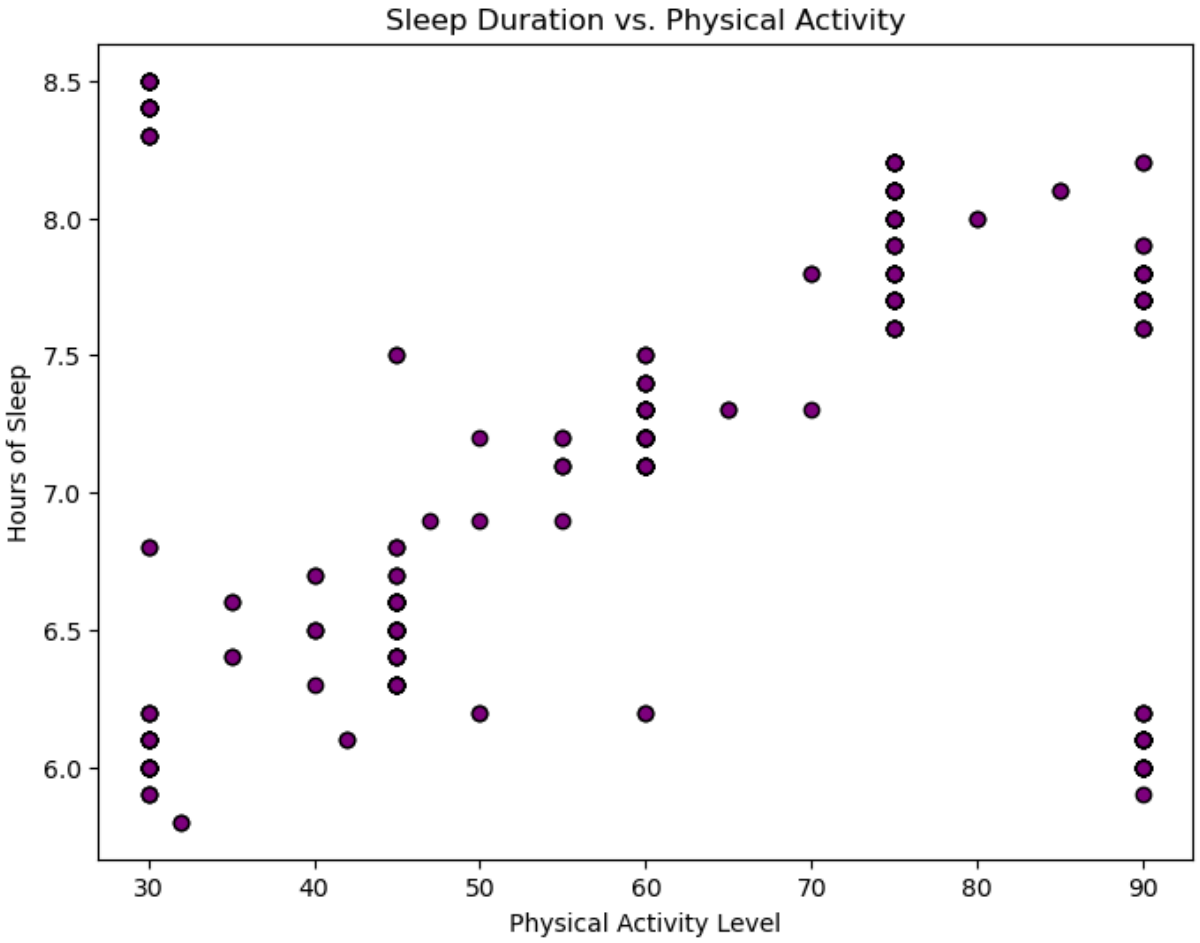
In [50]:
```
# 4  boxplot of sleep duration by sleep disorder

plt.figure(figsize=(8,6))
sleep.boxplot(column='Sleep Duration', by='Sleep Disorder', grid=False,
              boxprops=dict(color='black'),
              medianprops=dict(color='red'),
              whiskerprops=dict(color='black'),
              capprops=dict(color='black'))

plt.xlabel("Sleep Disorder")
plt.ylabel("Hours of Sleep")
plt.title("Sleep Duration by Sleep Disorder Status")
plt.suptitle("")  # remove default title

plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



In [51]:
```
# People with insomnia tend to sleep less on average than those with sleep apnea, s
```

In [52]:
```
# 4 scatterplot of sleep duration vs physical activity

plt.figure(figsize=(8,6))
plt.scatter(sleep['Activity'], sleep['Sleep Duration'],
            color='purple', edgecolors='black')

plt.xlabel("Physical Activity Level")
plt.ylabel("Hours of Sleep")
plt.title("Sleep Duration vs. Physical Activity")

plt.show()
```



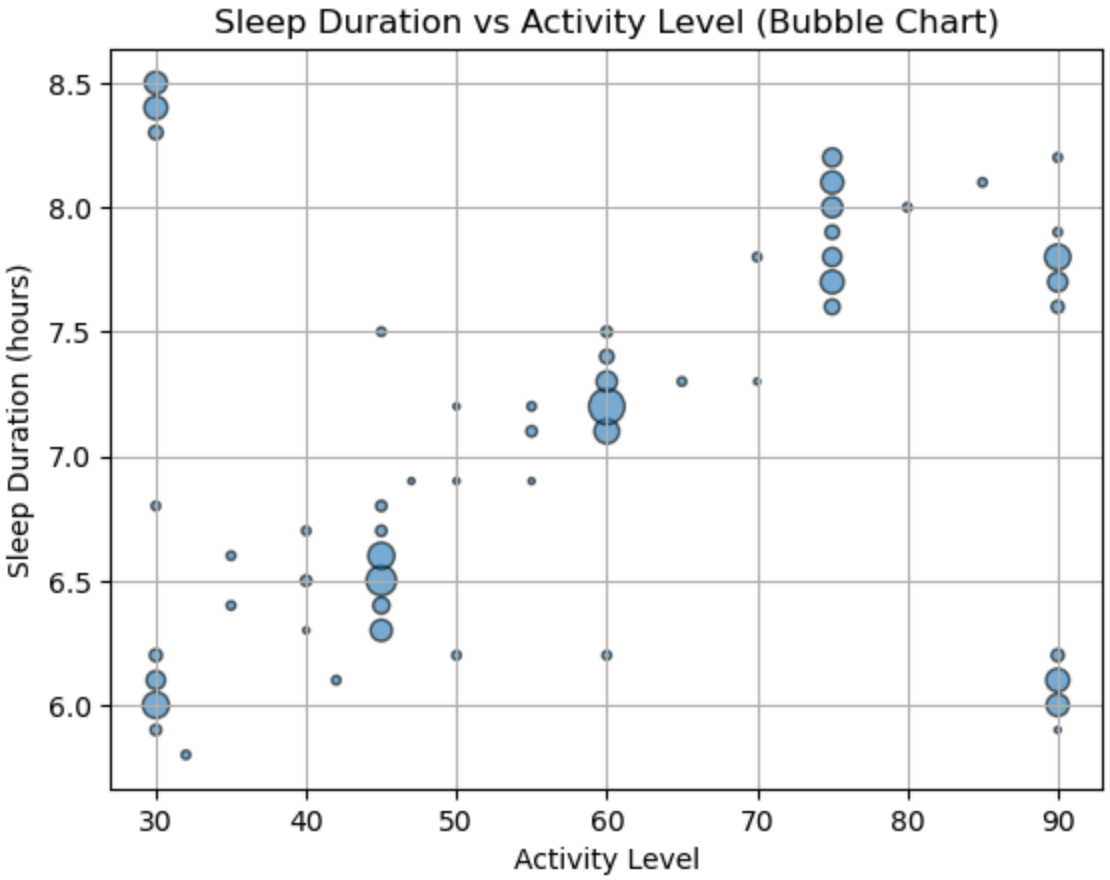
In [53]:
```
# Higher physical activity levels are generally associated with longer sleep durati
```

In [54]:
```
# Note
# Based on your feedback, the original scatterplots for Sleep Duration vs Stress Le
# and Sleep Duration vs Physical Activity Level created vertical streaks due to the
# x-axis containing only a few integer values.
#
# Instead of using the scatterplots above, I am now using bubble charts to reveal t
# true density at each combination of values. Each bubble represents a unique pairi
# of the two variables, with the bubble size proportional to the number of overlapp
# data points. This removes the overplotting.
```

In [55]:
```
# Note
sleep['Sleep_round'] = sleep['Sleep Duration'].round(1)

combo = (
    sleep.groupby(['Activity', 'Sleep_round'])
         .size()
         .reset_index(name='count')
)
```

In [56]:
```
# Note
plt.scatter(
    combo['Activity'],
    combo['Sleep_round'],
    s=combo['count'] * 5,   # Bubble size scaled
```

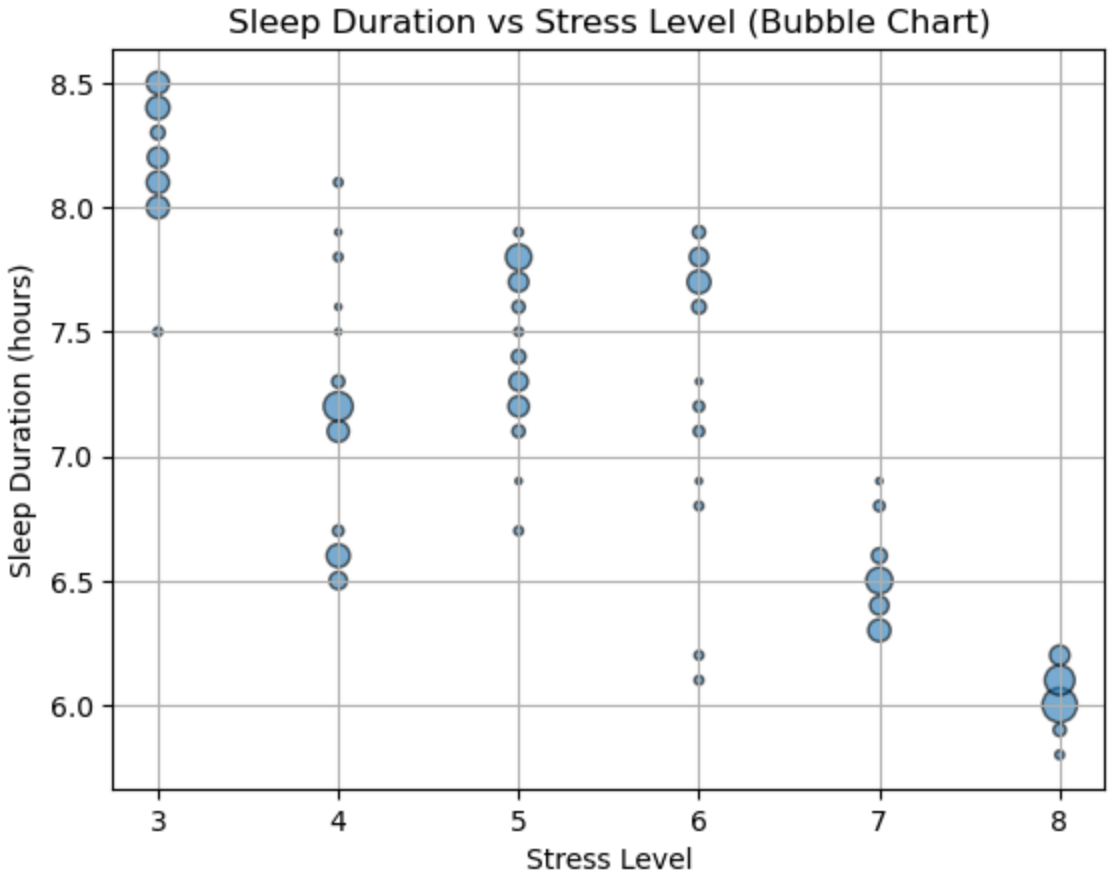
```
    alpha=0.6,
    edgecolors='black'
)

plt.title('Sleep Duration vs Activity Level (Bubble Chart)')
plt.xlabel('Activity Level')
plt.ylabel('Sleep Duration (hours)')
plt.grid(True)
plt.show()
```



In [57]:
```
# Note
sleep['Sleep_round'] = sleep['Sleep Duration'].round(1)

combo2 = (
    sleep.groupby(['Stress Level', 'Sleep_round'])
         .size()
         .reset_index(name='count')
)
```

In [58]:
```
# Note
plt.scatter(
    combo2['Stress Level'],
    combo2['Sleep_round'],
    s=combo2['count'] * 5,
    alpha=0.6,
    edgecolors='black'
)
```

```
plt.title('Sleep Duration vs Stress Level (Bubble Chart)')
plt.xlabel('Stress Level')
plt.ylabel('Sleep Duration (hours)')
plt.grid(True)
plt.show()
```



In [59]:
```
# Project Assignment 5
# For the data you have collected for your project, perform the following task(s):

# Data Aggregation and Group Operations
# Grouping Data
# Pivot Tables and Cross-Tabulation
```

In [60]:
```
# 1 Data Aggregation and Group Operations

sleep.describe()
```

Out[60]:

	Person ID	Age	Sleep Duration	Quality of Sleep	Activity	Stress Level	Heart Rate
count	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
std	108.108742	8.673133	0.795657	1.196956	20.830804	1.774526	4.135676
min	1.000000	27.000000	5.800000	4.000000	30.000000	3.000000	65.000000
25%	94.250000	35.250000	6.400000	6.000000	45.000000	4.000000	68.000000
50%	187.500000	43.000000	7.200000	7.000000	60.000000	5.000000	70.000000
75%	280.750000	50.000000	7.800000	8.000000	75.000000	7.000000	72.000000
max	374.000000	59.000000	8.500000	9.000000	90.000000	8.000000	86.000000

In [61]:
```
# 1
sleep['Age_Group'] = pd.cut(
    sleep['Age'],
    bins=[25, 35, 45, 55, 65],
    labels=['26-35', '36-45', '46-55', '56-65']
).round(2)

sleep.groupby('Age_Group')[['Sleep Duration', 'Quality of Sleep', 'Stress Level']].
```

```
C:\Users\lucas\AppData\Local\Temp\ipykernel_26952\839748768.py:8: FutureWarning: The
default of observed=False is deprecated and will be changed to True in a future vers
ion of pandas. Pass observed=False to retain current behavior or observed=True to ad
opt the future default and silence this warning.
  sleep.groupby('Age_Group')[['Sleep Duration', 'Quality of Sleep', 'Stress Leve
l']].mean()
```

Out[61]:

Age_Group	Sleep Duration	Quality of Sleep	Stress Level
26–35	6.854255	6.500000	6.574468
36–45	7.050588	7.341176	5.135294
46–55	7.236364	7.519481	5.506494
56–65	8.100000	9.000000	3.000000

In [62]:
```
# 2 Grouping Data

sleep.groupby('Stress Level')[['Sleep Duration', 'Activity']].mean()
```

Out[62]:

Stress Level	Sleep Duration	Activity
3	8.226761	54.718310
4	7.030000	55.785714
5	7.483582	74.253731
6	7.454348	67.152174
7	6.468000	43.840000
8	6.050000	58.342857

In [63]:
```
# 2
sleep['Activity_Group'] = pd.cut(
    sleep['Activity'],
    bins=[0, 40, 70, 100],
    labels=['Low', 'Medium', 'High']
).round(2)

# Group by Activity_Group
sleep.groupby('Activity_Group')[['Sleep Duration', 'Quality of Sleep', 'Stress Leve
```

```
C:\Users\lucas\AppData\Local\Temp\ipykernel_26952\1441615967.py:9: FutureWarning: Th
e default of observed=False is deprecated and will be changed to True in a future ve
rsion of pandas. Pass observed=False to retain current behavior or observed=True to
adopt the future default and silence this warning.
  sleep.groupby('Activity_Group')[['Sleep Duration', 'Quality of Sleep', 'Stress Lev
el']].mean()
```

Out[63]:

Activity_Group	Sleep Duration	Quality of Sleep	Stress Level
Low	7.067500	7.037500	5.775000
Medium	6.880128	7.211538	5.205128
High	7.454348	7.586957	5.362319

In [64]:
```
# 3 Pivot Tables and Cross-Tabulation

sleep.pivot_table(
    values='Sleep Duration',
    index='Age_Group',
    columns='Activity_Group',
    aggfunc='mean'
).round(2)
```

```
C:\Users\lucas\AppData\Local\Temp\ipykernel_26952\379069570.py:3: FutureWarning: The
default value of observed=False is deprecated and will change to observed=True in a
future version of pandas. Specify observed=False to silence this warning and retain
the current behavior
  sleep.pivot_table(
```

Out[64]:

Activity_Group	Low	Medium	High
Age_Group			
26–35	6.10	7.02	7.74
36–45	6.66	6.87	7.76
46–55	8.43	6.79	6.29
56–65	NaN	NaN	8.10

In [65]:
```
# 3
pd.crosstab(
    sleep['Stress Level'],
    sleep['Activity_Group']
)
```

Out[65]:

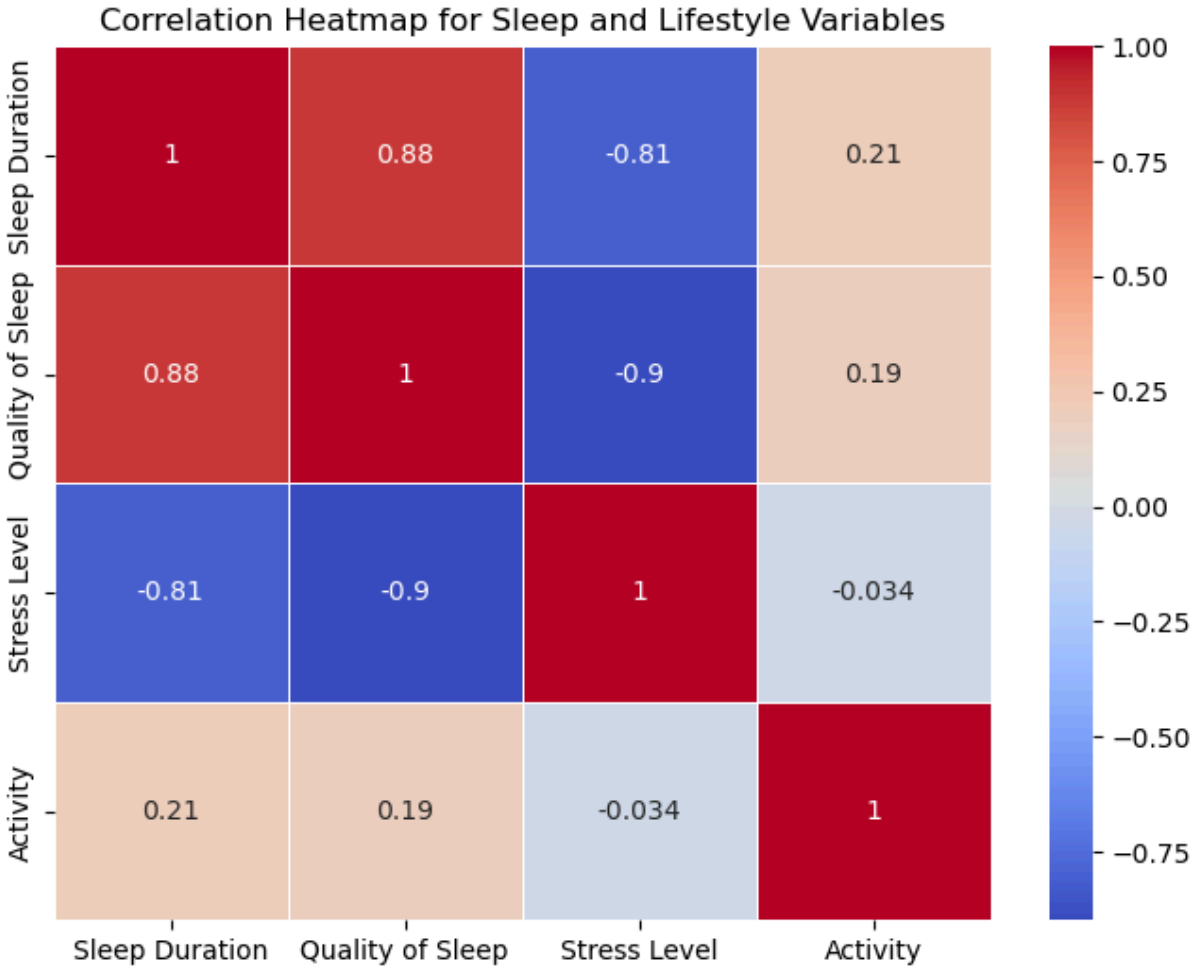
Activity_Group	Low	Medium	High
Stress Level			
3	32	2	37
4	0	66	4
5	2	32	33
6	2	12	32
7	8	42	0
8	36	2	32

In [66]:
```
# Lab 6
# This week, we covered some advanced topics (Time Series Data and JSON Data) which
# What is relevant, however, is that you are now ready to learn and code advanced t
# For this week, you will need to finish any remaining analyses you need to conduct
# This may require some research on your part.
# For example, if you are looking to compare three groups to each other, you will n
# The statistics packages make most analysis tasks fairly straightforward, and plea
```

In [67]:
```
# Remaining analysis
# Correlation Heatmap
# Regression Model
# Regression Visualizations
# ANOVA
```

In [68]:
```
# Correlation Heatmap

import seaborn as sns

corr_vars = sleep[['Sleep Duration', 'Quality of Sleep', 'Stress Level', 'Activity'
corr_matrix = corr_vars.corr()

plt.figure(figsize=(8, 6))
```

```
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap for Sleep and Lifestyle Variables')
plt.show()
```



In [69]:
```
# Regression Model 1
# Whether lifestyle factors (stress level and physical activity) predict total slee

import statsmodels.api as sm

X1 = sleep[['Stress Level', 'Activity']]
y1 = sleep['Sleep Duration']

X1 = sm.add_constant(X1)

model1 = sm.OLS(y1, X1).fit()

print("Model 1: Sleep Duration ~ Stress Level + Activity")
print(model1.summary())
```

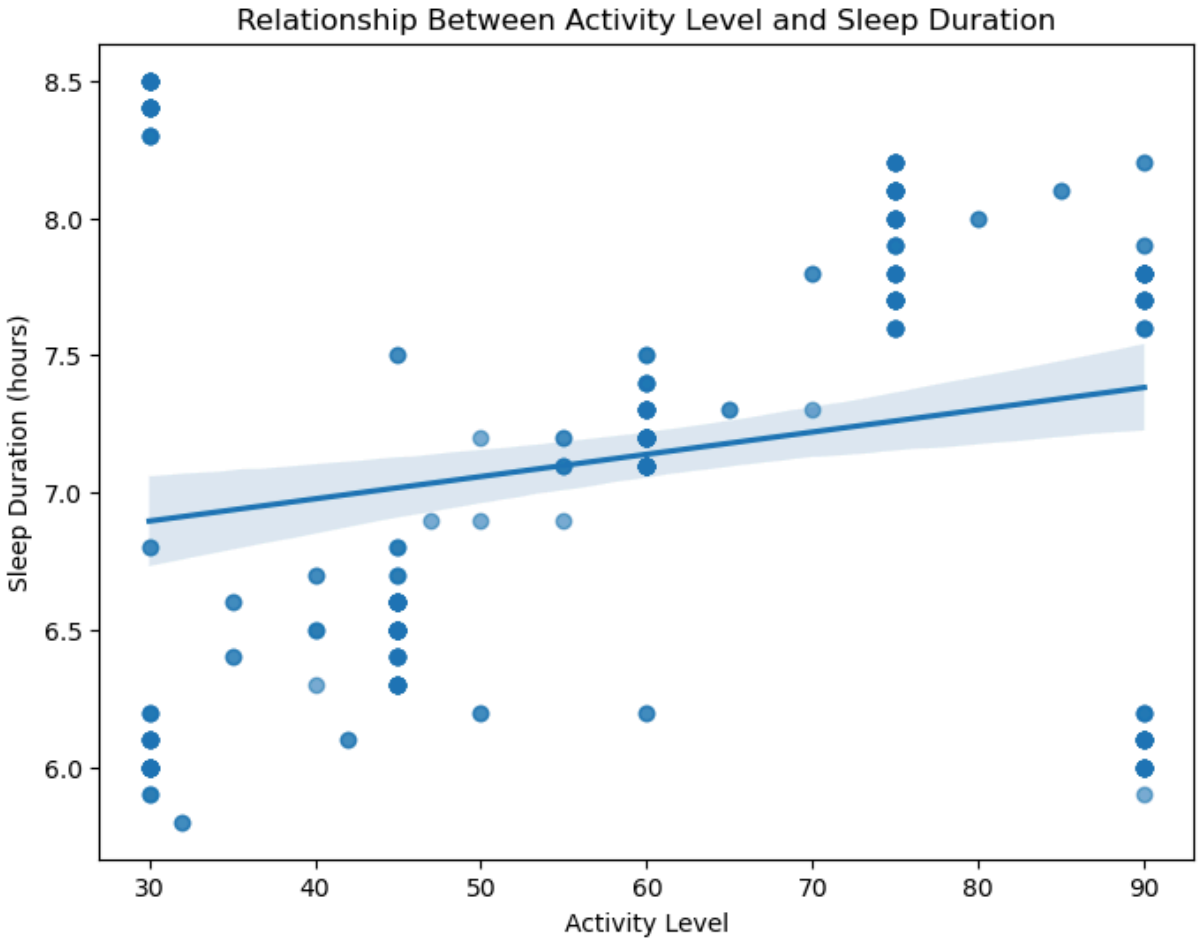
```
Model 1: Sleep Duration ~ Stress Level + Activity
                            OLS Regression Results
==============================================================================
Dep. Variable:         Sleep Duration   R-squared:                       0.692
Model:                            OLS   Adj. R-squared:                  0.690
Method:                 Least Squares   F-statistic:                     416.6
Date:                Wed, 10 Dec 2025   Prob (F-statistic):           1.42e-95
Time:                        20:36:17   Log-Likelihood:                -224.53
No. Observations:                 374   AIC:                             455.1
Df Residuals:                     371   BIC:                             466.8
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          8.6572      0.100     86.885      0.000       8.461       8.853
Stress Level  -0.3608      0.013    -27.908      0.000      -0.386      -0.335
Activity       0.0071      0.001      6.412      0.000       0.005       0.009
==============================================================================
Omnibus:                        3.064   Durbin-Watson:                   0.656
Prob(Omnibus):                  0.216   Jarque-Bera (JB):                3.134
Skew:                          -0.202   Prob(JB):                        0.209
Kurtosis:                       2.807   Cond. No.                         275.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

In [70]:

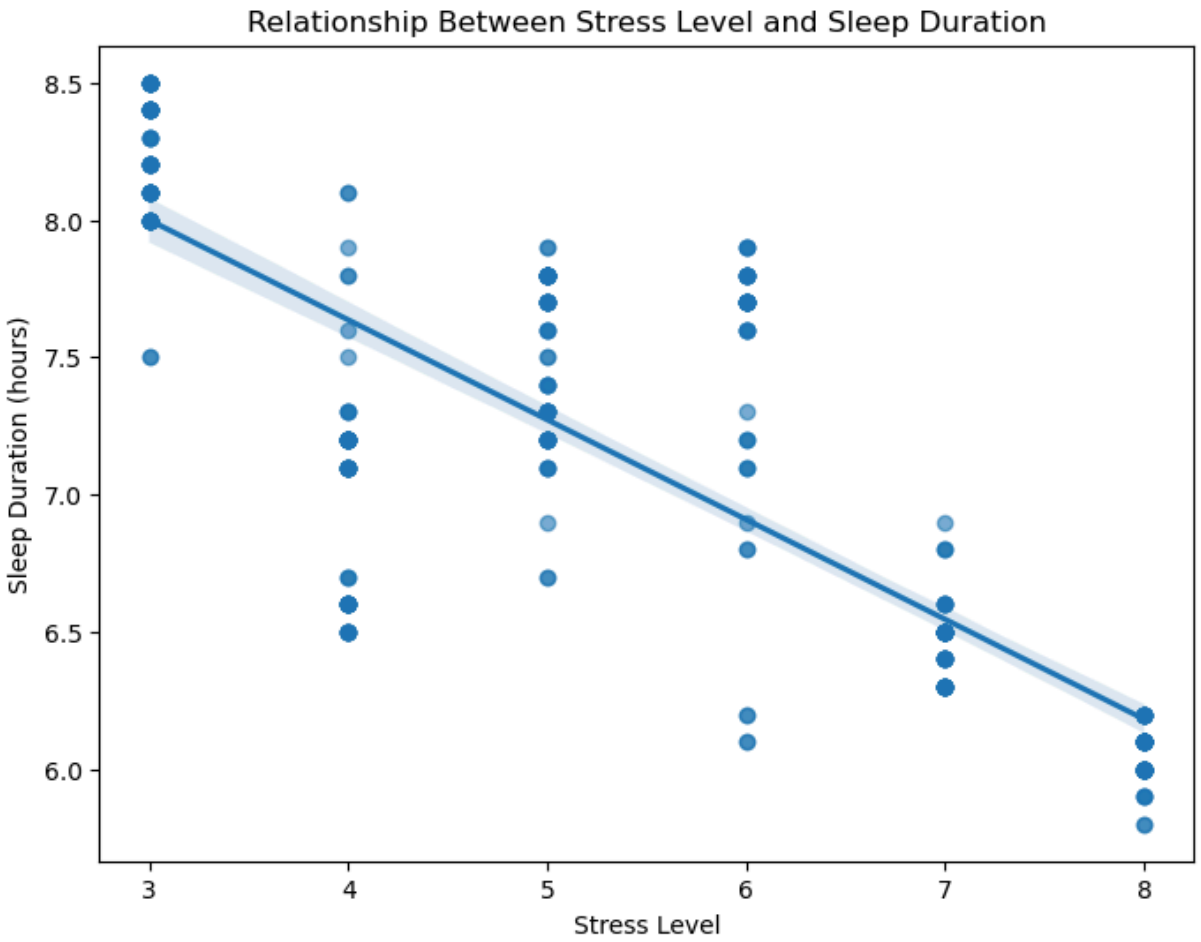
```python
# Reggesion 2
# Predicting Quality of Sleep
X2 = sleep[['Sleep Duration', 'Stress Level', 'Activity']]
y2 = sleep['Quality of Sleep']


X2 = sm.add_constant(X2)


model2 = sm.OLS(y2, X2).fit()

print("\nModel 2: Quality of Sleep ~ Sleep Duration + Stress Level + Activity")
print(model2.summary())
```

```
Model 2: Quality of Sleep ~ Sleep Duration + Stress Level + Activity
                            OLS Regression Results
==============================================================================
Dep. Variable:       Quality of Sleep   R-squared:                       0.884
Model:                           OLS   Adj. R-squared:                  0.883
Method:                Least Squares   F-statistic:                     942.3
Date:               Wed, 10 Dec 2025   Prob (F-statistic):          8.05e-173
Time:                       20:36:17   Log-Likelihood:                -194.18
No. Observations:                374   AIC:                             396.4
Df Residuals:                    370   BIC:                             412.0
Df Model:                          3
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const              4.7503      0.425     11.176      0.000       3.914       5.586
Sleep Duration     0.6070      0.048     12.663      0.000       0.513       0.701
Stress Level      -0.3835      0.021    -18.248      0.000      -0.425      -0.342
Activity           0.0050      0.001      4.708      0.000       0.003       0.007
==============================================================================
Omnibus:                       52.471   Durbin-Watson:                   1.060
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               81.235
Skew:                          -0.876   Prob(JB):                     2.29e-18
Kurtosis:                       4.463   Cond. No.                     1.28e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 1.28e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [71]:
```python
# Regression Visualizations
# how Activity and Stress Level relate to Sleep Duration.

plt.figure(figsize=(8, 6))
sns.regplot(data=sleep, x='Activity', y='Sleep Duration', scatter_kws={'alpha': 0.6
plt.title('Relationship Between Activity Level and Sleep Duration')
plt.xlabel('Activity Level')
plt.ylabel('Sleep Duration (hours)')
plt.show()
```



In [72]:
```
#  Regression Visualizations 2
# Sleep Duration vs Stress Level
plt.figure(figsize=(8, 6))
sns.regplot(data=sleep, x='Stress Level', y='Sleep Duration', scatter_kws={'alpha':
plt.title('Relationship Between Stress Level and Sleep Duration')
plt.xlabel('Stress Level')
plt.ylabel('Sleep Duration (hours)')
plt.show()
```



In [73]:
```
# ANOVA - Sleep Duration across Activity Groups
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.anova import anova_lm
```

In [74]:
```
# ANOVA
anova_model = smf.ols('Q("Sleep Duration") ~ C(Activity_Group)', data=sleep).fit()
anova_results = anova_lm(anova_model, typ=2)

anova_results
```

Out[74]:

	sum_sq	df	F	PR(>F)
C(Activity_Group)	24.568685	2.0	21.541669	1.409553e-09
Residual	211.566289	371.0	NaN	NaN

In [75]:
```
# Tukey test recommendation
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey_results = pairwise_tukeyhsd(
    endog=sleep['Sleep Duration'],
    groups=sleep['Activity_Group'],
    alpha=0.05
)
```

```
print(tukey_results)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
===================================================
group1 group2 meandiff p-adj   lower   upper  reject
---------------------------------------------------
  High    Low  -0.3868 0.0009 -0.6366 -0.1371   True
  High Medium  -0.5742    0.0 -0.7819 -0.3666   True
   Low Medium  -0.1874 0.1695 -0.4317   0.057  False
---------------------------------------------------
```