

✓ Introduction

This is an 'end-of-phase' project that I completed during my time at Flatiron School. In this phase, we took a look at machine learning and modeling to project and predict outcomes. In combination with what I have learned in previous phases, I created this project

Notebook 1

This notebook is one of two used for this particular project. In this notebook the goal is to perform a descriptive and inferential analysis. I am required to answer 5-10 descriptive analysis question and 2-4 inferential analysis questions

✓ Table of Contents

- [Introduction](#)
 - [Notebook 1](#)
 - [Table of Contents](#)
- [Imports, Data Cleaning and Analysis Prep](#)
- [Demographics of Dataset](#)
- [Descriptive Questions](#)
 - [Question 1: Which Profession in my dataset receives the most amount of sleep?](#)
 - [Question 2: Which Profession has the most recorded cases of a sleeping disorder?](#)
 - [Question 3: On average, how much sleep does a person in each BMI category receive?](#)
 - [Question 4: On average, how much sleep does someone with a sleeping disorder receive compared to someone with no sleeping disorder?](#)
 - [Question 5: On average, what is the quality of sleep that someone with a sleeping disorder receives compared to those without a sleeping disorder?](#)
 - [Question 6: What distribution of Males and Females in my dataset are diagnosed with sleeping disorders?](#)
 - [Question 7: What percentage of my dataset has insomnia?](#)
 - [Question 8: What is the distribution of male to female for patients with insomnia?](#)
 - [Question 9: What is the age distribution for patients with insomnia?](#)
 - [Insomnia vs Everyone Else](#)
- [Inferential Analysis](#)

- [Sleep Duration](#)
 - [Hypothesis 1.1: Sleep Duration ANOVA](#)
 - [Hypothesis 1.2: Sleep Duration - Sleeping Disorder vs NO Sleeping Disorder](#)
 - [Hypothesis 1.3: Sleep Duration - Sleep Apnea vs NO Sleep Apnea](#)
 - [Hypothesis 1.4: Sleep Duration - Insomnia vs NO Insomnia](#)
- [Quality of Sleep](#)
 - [Hypothesis 2.1: Quality of Sleep ANOVA](#)
 - [Hypothesis 2.2: Quality of Sleep - Sleeping Disorder vs NO Sleeping Disorder](#)
 - [Hypothesis 2.3: Quality of Sleep - Sleep Apnea vs NO Sleep Apnea](#)
 - [Hypothesis 2.4: Quality of Sleep - Insomnia vs NO Insomnia](#)
- [Physical Activity Level](#)
 - [Hypothesis 3.1: Physical Activity Level ANOVA](#)
 - [Hypothesis 3.2: Physical Activity Level - Sleeping Disorder vs NO Sleeping Disorder](#)
 - [Hypothesis 3.3: Physical Activity Level - Sleep Apnea vs NO Sleep Apnea](#)
 - [Hypothesis 3.4: Physical Activity Level - Insomnia vs NO Insomnia](#)
- [Stress Level](#)
 - [Hypothesis 4.1: Stress Level ANOVA](#)
 - [Hypothesis 4.2: Stress Level - Sleeping Disorder vs NO Sleeping Disorder](#)
 - [Hypothesis 4.3: Stress Level - Sleep Apnea vs NO Sleep Apnea](#)
 - [Hypothesis 4.4: Stress Level - Insomnia vs NO Insomnia](#)
- [Heart Rate](#)
 - [Hypothesis 5.1: Heart Rate ANOVA](#)
 - [Hypothesis 5.2: Heart Rate - Sleeping Disorder vs NO Sleeping Disorder](#)
 - [Hypothesis 5.3: Heart Rate - Sleep Apnea vs NO Sleep Apnea](#)
 - [Hypothesis 5.4: Heart Rate - Insomnia vs NO Insomnia](#)
- [Daily Steps](#)
 - [Hypothesis 6.1: Daily Steps ANOVA](#)
 - [Hypothesis 6.2: Daily Steps - Sleeping Disorder vs NO Sleeping Disorder](#)
 - [Hypothesis 6.3: Daily Steps - Sleep Apnea vs NO Sleep Apnea](#)
 - [Hypothesis 6.4: Daily Steps - Insomnia vs NO Insomnia](#)
- [Summary](#)
 - [Links](#)

```
import nbformat
```

```

def generate_toc(notebook_path):
    with open(notebook_path) as f:
        nb = nbformat.read(f, as_version=4)

    toc = []
    for cell in nb.cells:
        if cell.cell_type == 'markdown':
            lines = cell.source.split('\n')
            for line in lines:
                if line.startswith('#'):
                    header_level = line.count('#')
                    header_text = line.replace('#', '').strip()
                    toc.append((header_level, header_text))

    toc_md = ['## Table of Contents']
    for level, text in toc:
        toc_md.append(f"{' ' * (level - 1)}- [{text}](#{text.replace(' ', '-')})")

    return '\n'.join(toc_md)

notebook_path = 'Insomnia Analysis.ipynb'
toc_md = generate_toc(notebook_path)

# Print the generated TOC
# print(toc_md)

```

✓ Imports, Data Cleaning and Analysis Prep

```

# imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats import shapiro, normaltest, anderson

# loading dataframe
df = pd.read_csv('sleep_data.csv')
df.head()

```



The first thing I noticed was the Blood Pressure is registered as a string because it has a / in each entry. I decided to split those up and create two separate columns one for the Systolic reading (first number) and one for the Diastolic reading (second number)

```
# splitting the blood pressure column
df[['Systolic', 'Diastolic']] = df['Blood Pressure'].str.split('/', expand=True)
df['Systolic'] = df['Systolic'].astype(int)
df['Diastolic'] = df['Diastolic'].astype(int)
```

After splitting the data into two separate columns I wanted to categorize blood pressure based on if patients had "high blood pressure" "low blood pressure" or "normal blood pressure" but after some research and consulting a medical professional, I learned that you wouldn't diagnose blood pressure strictly based on the reading. There are a lot of symptoms that need to be present before treating a patient for high blood pressure. Therefore, I will be keeping the Sytolic and Diastolic readings as continuous variables for now.

```
df.info()
```



```
<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   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                       155 non-null    object
```

```

13   Systolic              374 non-null    int32
14   Diastolic             374 non-null    int32
dtypes: float64(1), int32(2), int64(7), object(5)
memory usage: 41.0+ KB

```

Next, I saw that the 'Sleep Disorder' column had null values in place for patients that were not diagnosed with sleeping disorders.

Lets fill those up!

```
df['Sleep Disorder'].fillna('No', inplace=True)
```



C:\Users\Shank\AppData\Local\Temp\ipykernel_15820\4279392291.py:1: FutureWarning: A value is no longer supported in pandas 3.0. This inplace method will never work because the

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col

```
df['Sleep Disorder'].fillna('No', inplace=True)
```

✓ Demographics of Dataset

In this section, I want to get a better idea of who is included in my dataset

✓ Gender

```
gender = df['Gender'].value_counts()
```

```
# Setting Seaborn style
sns.set_style(style="whitegrid")
```

```
# Setting up the figure size
plt.figure(figsize=(10, 6))
```

```
# Creating the bar plot with a blues colormap
sns.barplot(x=gender.index, y=gender.values, palette='Blues_r', hue=gender.index)
```

```
# Adding labels and title
plt.xlabel('Gender', fontsize=14, labelpad=10)
plt.ylabel('Count', fontsize=14, labelpad=10)
plt.title('Gender Demographics', fontsize=16, pad=15)
```

```
# Adding value labels on the bars
for i, value in enumerate(gender.values):
```

```
plt.text(i, value + 0.1, f'{value:.0f}', ha='center', va='bottom', fontsize=12, color='t

# Customizing the axes and grid
plt.ylim(0, gender.max() + 1)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('sample_gender.png')
plt.show()
```



✓ Age

```
df["Age"].describe()
```



```
count    374.000000
mean     42.184492
std       8.673133
min      27.000000
25%      35.250000
50%      43.000000
75%      50.000000
max      59.000000
Name: Age, dtype: float64
```

```
# Define the bin edges manually
bin_edges = [25, 30, 35, 40, 45, 50, 55, 60] # Adjust the edges as needed

plt.figure(figsize=(10, 6))

# Use the Blues color palette
colors = sns.color_palette('Blues', n_colors=len(bin_edges) - 1)

# Plot the histogram with manual bin edges and colors
sns.histplot(df['Age'], bins=bin_edges, kde=False, color=colors[0])

# For more control, we can use a loop to manually color each bin:
for i in range(len(bin_edges) - 1):
    sns.histplot(df['Age'], bins=[bin_edges[i], bin_edges[i + 1]], kde=False, color=colors[i])

# Set the x-ticks at the bin edges
plt.xticks(bin_edges)

plt.title('Age Distribution', fontsize=16)
plt.xlabel('Age', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
```

```
plt.savefig('sample_age.png')  
plt.show()
```



✓ Occupation

```
occupation = df['Occupation'].value_counts()  
  
# Setting Seaborn style  
sns.set_style(style="whitegrid")  
  
# Setting up the figure size  
plt.figure(figsize=(10, 6))  
  
# Creating the barplot with a blues color palette  
sns.barplot(y=occupation.index, x=occupation.values, palette="Blues_r", hue=occupation.index)  
  
# Adding labels  
plt.xlabel('Count', fontsize=14, labelpad=10)  
plt.ylabel('Occupation', fontsize=14, labelpad=10)  
plt.title('Occupation Demographics', fontsize=16, pad=15)  
  
# Adding value labels on the bars  
for index, value in enumerate(occupation.values):  
    plt.text(value, index, f'{value:.0f}', color='black', ha='left', va='center', fontsize=14)  
  
# Customizing the axes and grid  
plt.xlim(0, occupation.max() + 0.5)  
plt.xticks(visible=False)  
plt.grid(True, axis='x', linestyle='--', alpha=0.6)  
sns.despine(left=True, bottom=True)  
  
# Display the plot  
plt.savefig('sample_occupation.png')  
plt.show()
```



✓ BMI

```
bmi = df['BMI Category'].value_counts()
```

```
# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
sns.barplot(x=bmi.index, y=bmi.values, palette='Blues_r', hue=bmi.index)

# Adding labels and title
plt.xlabel('BMI Category', fontsize=14, labelpad=10)
plt.ylabel('Count', fontsize=14, labelpad=10)
plt.title('BMI Demographics', fontsize=16, pad=15)

# Adding value labels on the bars
for i, value in enumerate(bmi.values):
    plt.text(i, value + 0.1, f'{value:.0f}', ha='center', va='bottom', fontsize=12, color='t

# Customizing the axes and grid
plt.ylim(0, bmi.max() + 1)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('sample_BMI.png')
plt.show()
```



✓ Sleep Disorder

```
sd = df['Sleep Disorder'].value_counts()

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
sns.barplot(x=sd.index, y=sd.values, palette='Blues_r', hue=sd.index)

# Adding labels and title
plt.xlabel('Sleep Disorder', fontsize=14, labelpad=10)
plt.ylabel('Count', fontsize=14, labelpad=10)
plt.title('Sleep Disorder Demographics', fontsize=16, pad=15)
```



```
# Adding value labels on the bars
for i, value in enumerate(sd.values):
    plt.text(i, value + 0.1, f'{value:.0f}', ha='center', va='bottom', fontsize=12, color='k')

# Customizing the axes and grid
plt.ylim(0, sd.max() + 1)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('sample_sd.png')
plt.show()
```



✓ Descriptive Questions

✓ Question 1: Which Profession in my dataset receives the most amount of sleep?

For this question, I grouped my dataset by occupation calculated the mean sleep duration for each group.

```
Q1 = df.groupby('Occupation')['Sleep Duration'].mean().sort_values(ascending=False).head(10)

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the barplot with a blues color palette
sns.barplot(y=Q1.index, x=Q1.values, palette="Blues_r", hue=Q1.index)

# Adding labels
plt.xlabel('Average Sleep Duration (Hours)', fontsize=14, labelpad=10)
plt.ylabel('Occupation', fontsize=14, labelpad=10)
plt.title('Top 10 Occupations by Average Sleep Duration', fontsize=16, pad=15)

# Adding value labels on the bars
for index, value in enumerate(Q1.values):
    plt.text(value, index, f'{value:.2f}', color='black', ha='left', va='center', fontsize=12)
```

```
# Customizing the axes and grid
plt.xlim(0, Q1.max() + 0.5)
plt.xticks(visible=False)
plt.grid(True, axis='x', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.show()
```



✓ Question 2: Which Profession has the most recorded cases of a sleeping disorder?

For this question, I removed all cases where there was no sleeping disorder recorded, and then of the rows remaining I did a value count of each profession and the type of disorder recorded

```
Q2 = df.groupby('Occupation')['Sleep Disorder'].value_counts().drop('No', level=1).sort_valu

# Unstack the Series to create a DataFrame with Sleep Disorders as columns
Q2_unstacked = Q2.unstack().fillna(0)

# Setting Seaborn style
sns.set_style(style="whitegrid")
```

```
# Setting up the figure size
plt.figure(figsize=(12, 8))

# Creating the stacked bar plot
plt.bar(Q2_unstacked.index, Q2_unstacked['Sleep Apnea'], color="#1f77b4", label='Sleep Apnea')
plt.bar(Q2_unstacked.index, Q2_unstacked['Insomnia'], bottom=Q2_unstacked['Sleep Apnea'], color='black', label='Insomnia')

# Adding labels
plt.xlabel('Occupation', fontsize=14, labelpad=10)
plt.ylabel('Number of Cases', fontsize=14, labelpad=10)
plt.title('Distribution of Sleep Disorders by Occupation', fontsize=16, pad=15)

# Adding value labels on the bars
for i, occupation in enumerate(Q2_unstacked.index):
    if Q2_unstacked['Sleep Apnea'].iloc[i] > 0:
        plt.text(i, Q2_unstacked['Sleep Apnea'].iloc[i] / 2, f'{int(Q2_unstacked["Sleep Apnea"].iloc[i])}',
                 color='Black', ha='center', va='center', fontsize=14)
    if Q2_unstacked['Insomnia'].iloc[i] > 0:
        plt.text(i, Q2_unstacked['Sleep Apnea'].iloc[i] + Q2_unstacked['Insomnia'].iloc[i] / 2, f'{int(Q2_unstacked["Insomnia"].iloc[i])}',
                 color='Black', ha='center', va='center', fontsize=14)

# Customizing the axes and grid
plt.ylim(0, Q2_unstacked.sum(axis=1).max() + 10)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the legend
plt.legend(title='Sleep Disorder')

# Display the plot
plt.show()
```



✓ Question 3: On average, how much sleep does a person in each BMI category receive?

Similar to Question 1, for this question I grouped my data based on BMI category and found the mean sleep duration.

```
Q3 = df.groupby('BMI Category')['Sleep Duration'].mean()

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
sns.barplot(x=Q3.index, y=Q3.values, palette='Blues_r', hue=Q3.index)

# Adding labels and title
plt.xlabel('BMI Category', fontsize=14, labelpad=10)
plt.ylabel('Average Sleep Duration (hours)', fontsize=14, labelpad=10)
plt.title('Average Sleep Duration by BMI Category', fontsize=16, pad=15)
```

```
# Adding value labels on the bars
for i, value in enumerate(Q3.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=12, color='k')

# Customizing the axes and grid
plt.ylim(0, Q3.max() + 1)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.show()
```



Question 4: On average, how much sleep does someone with a
✓ sleeping disorder receive compared someone with no sleeping
disorder?

For this question, similar to the previous question, I grouped my data based on sleeping disorder (or lack thereof) and calculated the mean sleeping duration for each group.

```
Q4 = df.groupby('Sleep Disorder')['Sleep Duration'].mean()

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
sns.barplot(x=Q4.index, y=Q4.values, palette='Blues_r', hue=Q4.index)

# Adding labels and title
plt.xlabel('Sleep Disorder', fontsize=14, labelpad=10)
plt.ylabel('Average Sleep Duration (hours)', fontsize=14, labelpad=10)
plt.title('Average Sleep Duration by Sleep Disorder', fontsize=16, pad=15)

# Adding value labels on the bars
for i, value in enumerate(Q4.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=12, color='t

# Customizing the axes and grid
plt.ylim(0, Q4.max() + 1)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.show()
```



- Question 5: On average, what is the quality of sleep that someone
- ✓ with a sleeping disorder receives compared to those without a sleeping disorder?

For this question, similar to the previous, I grouped my data based on sleeping disorder (or lack thereof) and calculated the mean quality of sleep for each group.

```
Q5 = df.groupby('Sleep Disorder')['Quality of Sleep'].mean()

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
sns.barplot(x=Q5.index, y=Q5.values, palette='Blues_r', hue=Q5.index)

# Adding labels and title
plt.xlabel('Sleep Disorder', fontsize=14, labelpad=10)
plt.ylabel('Average Quality of Sleep', fontsize=14, labelpad=10)
```

```
plt.title('Average Quality of Sleep by Sleep Disorder', fontsize=16, pad=15)

# Adding value labels on the bars
for i, value in enumerate(Q5.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=12, color='t

# Customizing the axes and grid
plt.ylim(0, Q5.max() + 1)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.show()
```



✓ Question 6: What distribution of Males and Females in my dataset are diagnosed with sleeping disorders?

```
Q6 = df.groupby('Gender')['Sleep Disorder'].value_counts()
```

```
# Unstack the Series to create a DataFrame with Sleep Disorders as columns
Q6_unstacked = Q6.unstack().fillna(0)
```



```

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the stacked bar plot
plt.bar(Q6_unstacked.index, Q6_unstacked['Sleep Apnea'], color="#1f77b4", label='Sleep Apnea')
plt.bar(Q6_unstacked.index, Q6_unstacked['Insomnia'], bottom=Q6_unstacked['Sleep Apnea'], color='black', label='Insomnia')

# Adding labels and title
plt.xlabel('Gender', fontsize=14, labelpad=10)
plt.ylabel('Number of Cases', fontsize=14, labelpad=10)
plt.title('Distribution of Sleep Disorders by Gender', fontsize=16, pad=15)

# Adding value labels on the bars
for i, gender in enumerate(Q6_unstacked.index):
    if Q6_unstacked['Sleep Apnea'].iloc[i] > 0:
        plt.text(i, Q6_unstacked['Sleep Apnea'].iloc[i] / 2, f'{int(Q6_unstacked["Sleep Apnea"].iloc[i])}',
                 color='black', ha='center', va='center', fontsize=12)
    if Q6_unstacked['Insomnia'].iloc[i] > 0:
        plt.text(i, Q6_unstacked['Sleep Apnea'].iloc[i] + Q6_unstacked['Insomnia'].iloc[i] / 2, f'{int(Q6_unstacked["Insomnia"].iloc[i])}',
                 color='black', ha='center', va='center', fontsize=12)

# Customizing the axes and grid
plt.ylim(0, Q6_unstacked.sum(axis=1).max() + 10)
plt.yticks(visible=False)
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the legend
plt.legend(title='Sleep Disorder')

# Display the plot
plt.show()

```



✓ Question 7: What percentage of my dataset has insomnia?

```
# Create masks for 'Insomnia' and 'Non-Insomnia'
MASK_I = df['Sleep Disorder'] == 'Insomnia'
MASK_NI = df['Sleep Disorder'] != 'Insomnia'

# Count the number of instances for each mask
insomnia_count = df[MASK_I].shape[0]
no_insomnia_count = df[MASK_NI].shape[0]

# Combine the counts into a DataFrame or Series (if that's what you want)
combined_counts = pd.Series({'Insomnia': insomnia_count, 'No Insomnia': no_insomnia_count})

# Calculating percentages
total_count = combined_counts.sum()
percentages = (combined_counts / total_count) * 100

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))
```

```

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=combined_counts.index, y=combined_counts.values, palette='Blues_r')

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('Insomnia?', fontsize=16, labelpad=10, color='white')
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Insomnia Count', fontsize=16, pad=15, color='white')

# Adding value labels with percentages on the bars
for i, (value, percent) in enumerate(zip(combined_counts.values, percentages.values)):
    plt.text(i, value + 0.1, f'{value} ({percent:.2f}%)', ha='center', va='bottom', fontsize=12)

# Customizing the axes and grid
plt.ylim(0, combined_counts.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_count.png', transparent=True)
plt.show()

```



✓ Question 8: What is the distribution of male to female for patients with insomnia?

```

insomnia = df[MASK_I]

Q8 = insomnia['Gender'].value_counts()

# Calculating percentages
total_count = Q8.sum()
percentages = (Q8 / total_count) * 100

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

```

```

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=Q8.index, y=Q8.values, palette='Blues_r')

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('Gender', fontsize=16, labelpad=10, color='white')
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Gender Demographics', fontsize=16, pad=15, color='white')

# Adding value labels on the bars
for i, (value, percent) in enumerate(zip(Q8.values, percentages.values)):
    plt.text(i, value + 0.1, f'{value} ({percent:.2f}%)', ha='center', va='bottom', fontsize=12)

# Customizing the axes and grid
plt.ylim(0, Q8.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_gender.png', transparent=True)
plt.show()

```



✓ Question 9: What is the age distribution for patients with insomnia?

```

insomnia = df[MASK_I]

# Define the bin edges manually
bin_edges = [25, 30, 35, 40, 45, 50, 55] # Adjust the edges as needed

# Calculate the counts and percentages for the bins
counts, edges = np.histogram(insomnia['Age'], bins=bin_edges)
total_count = counts.sum()
percentages = (counts / total_count) * 100

# Setting Seaborn style
sns.set_style(style="whitegrid")

```

```

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Use the Blues color palette
colors = sns.color_palette('Blues_r', n_colors=len(bin_edges) - 1)

# Plot the histogram with manual bin edges and colors
sns.histplot(insomnia['Age'], bins=bin_edges, kde=False, color=colors[0], edgecolor='black')

# Adding labels on each bin with counts and percentages
for i in range(len(bin_edges) - 1):
    plt.hist(insomnia['Age'], bins=[bin_edges[i], bin_edges[i + 1]], color=colors[i], edgecc

    # Add the count and percentage text on top of each bar with height adjustment
    plt.text(edges[i] + (edges[i+1] - edges[i]) / 2, counts[i] + 1.5, # Adding height with
             f'{counts[i]:.0f} ({percentages[i]:.2f}%)',
             ha='center', va='bottom', fontsize=12, color='white')

# Set the x-ticks at the bin edges
plt.xticks(bin_edges)

# Adding labels and title
plt.xlabel('Age', fontsize=14, labelpad=10, color='white')
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Age Distribution', fontsize=16, pad=15, color='white')

# Customizing the axes and grid
plt.ylim(0, counts.max() + 15) # Adding extra space above the highest bar
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Save the figure with a transparent background
plt.savefig('insomnia_age.png', transparent=True)

# Show the plot
plt.show()

```



✓ Insomnia vs Everyone Else

In this section I will compare means of various statistics between patients with insomnia vs without prior to performing hypothesis tests on the same statistics

First I want to make a column of the dataframe I can group by for the following questions

```
df['insomnia'] = df['Sleep Disorder'].apply(lambda x: 'Insomnia' if x == 'Insomnia' else 'Ev
df.head()
```



✓ Sleep Duration

```
# establishing the dataset
sd = df.groupby('insomnia')['Sleep Duration'].mean().sort_values(ascending=True)

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=sd.index, y=sd.values, palette='Blues_r')

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('', fontsize=14, labelpad=10)
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Average Sleep Duration', fontsize=20, pad=15, color='white')

# Adding value labels on the bars
for i, value in enumerate(sd.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=14, color='w')

# Customizing the axes and grid
plt.ylim(0, sd.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_sd.png', transparent=True)
plt.show()
```



✓ Quality of Sleep

```
# establishing the dataset
qos = df.groupby('insomnia')['Quality of Sleep'].mean().sort_values(ascending=True)

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=qos.index, y=qos.values, palette='Blues_r')

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('', fontsize=14, labelpad=10)
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Average Quality of Sleep', fontsize=20, pad=15, color='white')

# Adding value labels on the bars
for i, value in enumerate(qos.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=14, color='v')

# Customizing the axes and grid
plt.ylim(0, qos.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_qos.png', transparent=True)
plt.show()
```



✓ Physical Activity Level

```
# establishing the dataset
pal = df.groupby('insomnia')['Quality of Sleep'].mean().sort_values(ascending=True)

# Setting Seaborn style
```

```

sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=pal.index, y=pal.values, palette='Blues_r')

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('', fontsize=14, labelpad=10)
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Average Physical Activity Level', fontsize=20, pad=15, color='white')

# Adding value labels on the bars
for i, value in enumerate(pal.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=14, color='v')

# Customizing the axes and grid
plt.ylim(0, pal.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_pal.png', transparent=True)
plt.show()

```



✓ Daily Steps

```

# establishing the dataset
ds = df.groupby('insomnia')['Daily Steps'].mean().sort_values(ascending=True)

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=ds.index, y=ds.values, palette='Blues_r')

```



```

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('', fontsize=14, labelpad=10)
plt.ylabel('', fontsize=14, labelpad=10)
plt.title('Average Daily Steps', fontsize=20, pad=15, color='white')

# Adding value labels on the bars
for i, value in enumerate(ds.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=14, color='w')

# Customizing the axes and grid
plt.ylim(0, ds.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_ds.png', transparent=True)
plt.show()

```



✓ Stress Level

```

# establishing the dataset
sl = df.groupby('insomnia')['Stress Level'].mean().sort_values(ascending=False)

# Setting Seaborn style
sns.set_style(style="whitegrid")

# Setting up the figure size
plt.figure(figsize=(10, 6))

# Creating the bar plot with a blues colormap
barplot = sns.barplot(x=sl.index, y=sl.values, palette='Blues_r')

# Adding a black outline to the bars
for patch in barplot.patches:
    patch.set_edgecolor('black')
    patch.set_linewidth(1.5)

# Adding labels and title
plt.xlabel('', fontsize=14, labelpad=10)
plt.ylabel('', fontsize=14, labelpad=10)

```

```
plt.title('Average Stress Level', fontsize=20, pad=15, color='white')

# Adding value labels on the bars
for i, value in enumerate(sl.values):
    plt.text(i, value + 0.1, f'{value:.2f}', ha='center', va='bottom', fontsize=14, color='v

# Customizing the axes and grid
plt.ylim(0, sl.max() + 1)
plt.yticks(visible=False)
plt.xticks(fontsize=14, color='white')
plt.grid(True, axis='y', linestyle='--', alpha=0.6)
sns.despine(left=True, bottom=True)

# Display the plot
plt.savefig('insomnia_sl.png', transparent=True)
plt.show()
```



✓ Inferential Analysis

In this section of the notebook I will be performing hypothesis tests on various statistics found in my dataset, each statistic has 4 tests that will be performed.

First thing to do is create my masks and functions for my hypothesis testings

```
# Masks for with and without sleep disorder
MASK_SD = df['Sleep Disorder'] != 'No'
MASK_NO = df['Sleep Disorder'] == 'No'

# Mask for each sleep disorder or lack thereof
MASK_SA = df['Sleep Disorder'] == 'Sleep Apnea'
MASK_I = df['Sleep Disorder'] == 'Insomnia'
MASK_NO = df['Sleep Disorder'] == 'No'

# Masks for with and without Sleep Apnea
MASK_SA = df['Sleep Disorder'] == 'Sleep Apnea'
MASK_NSA = df['Sleep Disorder'] != 'Sleep Apnea'

# Masks for with and without Insomnia
MASK_I = df['Sleep Disorder'] == 'Insomnia'
MASK_NI = df['Sleep Disorder'] != 'Insomnia'

# Creating a function to plot the With and Without Sleep Disorder Test
def two_sample_kdeplot(dataset1, dataset2, label1 = 'WITH Sleep Disorder', label2 = 'WITHOUT
```

```

# Calculate the means of both datasets
mean1 = np.mean(dataset1)
mean2 = np.mean(dataset2)

# Perform two sample t-test
_, p_value = stats.ttest_ind(dataset1, dataset2)
if p_value < 0.05:
    print('Reject the Null Hypothesis')
else:
    print('Fail to reject the null hypothesis')

# Use a style template
sns.set_style('whitegrid')

plt.figure(figsize=(12, 8))

# Plot both datasets
sns.kdeplot(dataset1, label=label1, color='red', alpha=1)
sns.kdeplot(dataset2, label=label2, color='blue', alpha=1)

# Compute KDE values
kde_dataset1 = stats.gaussian_kde(dataset1)
kde_dataset2 = stats.gaussian_kde(dataset2)

# Generate x values for KDE
x_vals = np.linspace(min(dataset1.min(), dataset2.min()), max(dataset1.max(), dataset2.max()), 100)

# Fill the entire area under the curves
plt.fill_between(x_vals, kde_dataset1(x_vals), color='red', alpha=0.3)
plt.fill_between(x_vals, kde_dataset2(x_vals), color='blue', alpha=0.3)

# Get KDE values at means
kde_dataset1_at_mean1 = kde_dataset1(mean1)[0]
kde_dataset2_at_mean2 = kde_dataset2(mean2)[0]

# Add vertical lines at the means, stopping at the KDE curves
plt.plot([mean1, mean1], [0, kde_dataset1_at_mean1], 'r--', label=f'{label1} Mean: {mean1:.2f}')
plt.plot([mean2, mean2], [0, kde_dataset2_at_mean2], 'b--', label=f'{label2} Mean: {mean2:.2f}')

# Add labels and title
plt.title('Test Between With Sleeping Disorder vs Without Sleeping Disorder', fontsize=20)
plt.xlabel('', fontsize=16)
plt.ylabel('')
plt.yticks(visible=False)

# Add annotations
plt.annotate(f'{label1} | Mean: {mean1:.2f}', xy=(0.75, 0.85), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='red', boxstyle='round,pad=0.5'))
plt.annotate(f'{label2} | Mean: {mean2:.2f}', xy=(0.75, 0.75), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='blue', boxstyle='round,pad=0.5'))

```

```

if save == True:
    plt.savefig(f'{label1} v {label2} TTest.png', transparent=True)
plt.show()

# Creating a function to plot the test comparing each sleeping disorder or lack thereof
def three_way_ANOVA(dataset1, dataset2, dataset3, label1 = 'Sleep Apnea', label2 = 'No Disor
    # Calculate the means of both datasets
    mean1 = np.mean(dataset1)
    mean2 = np.mean(dataset2)
    mean3 = np.mean(dataset3)

    # Perform two sample t-test
    _, p_value = stats.f_oneway(dataset1, dataset2, dataset3)
    if p_value < 0.05:
        print('Reject the Null Hypothesis')
    else:
        print('Fail to reject the null hypothesis')

    # Use a style template
    sns.set_style('whitegrid')

    plt.figure(figsize=(12, 8))

    # Plot both datasets
    sns.kdeplot(dataset1, label=label1, color='red', alpha=1)
    sns.kdeplot(dataset2, label=label2, color='blue', alpha=1)
    sns.kdeplot(dataset3, label=label3, color='green', alpha=1)

    # Compute KDE values
    kde_dataset1 = stats.gaussian_kde(dataset1)
    kde_dataset2 = stats.gaussian_kde(dataset2)
    kde_dataset3 = stats.gaussian_kde(dataset3)

    # Generate x values for KDE
    x_vals = np.linspace(min(dataset1.min(), dataset2.min()), max(dataset1.max(), dataset2.n

    # Fill the entire area under the curves
    plt.fill_between(x_vals, kde_dataset1(x_vals), color='red', alpha=0.3)
    plt.fill_between(x_vals, kde_dataset2(x_vals), color='blue', alpha=0.3)
    plt.fill_between(x_vals, kde_dataset3(x_vals), color='green', alpha=0.3)

    # Get KDE values at means
    kde_dataset1_at_mean1 = kde_dataset1(mean1)[0]
    kde_dataset2_at_mean2 = kde_dataset2(mean2)[0]
    kde_dataset3_at_mean3 = kde_dataset3(mean3)[0]

    # Add vertical lines at the means, stopping at the KDE curves
    plt.plot([mean1, mean1], [0, kde_dataset1_at_mean1], 'r--', label=f'{label1} Mean: {mear
    plt.plot([mean2, mean2], [0, kde_dataset2_at_mean2], 'b--', label=f'{label2} Mean: {mear

```

```

plt.plot([mean3, mean3], [0, kde_dataset3_at_mean3], 'g--', label=f'{label3} Mean: {mear

# Add labels and title
plt.title('Sleep Apnea vs Insomnia vs No Disorder', fontsize=20, fontweight='bold')
plt.xlabel('', fontsize=16)
plt.ylabel('')
plt.yticks(visible=False)

# Add annotations
plt.annotate(f'{label1} | Mean: {mean1:.2f}', xy=(0.75, 0.85), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='red', boxstyle='round,pad=0.5'))
plt.annotate(f'{label2} | Mean: {mean2:.2f}', xy=(0.75, 0.75), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='blue', boxstyle='round,pad=0.5'))
plt.annotate(f'{label3} | Mean: {mean3:.2f}', xy=(0.75, 0.65), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='green', boxstyle='round,pad=0.5'))

if save == True:
    plt.savefig(f'{label1} v {label2} TTest.png', transparent=True)
plt.show()

# Creating a function to plot the With and Without Sleep Apnea Test
def SA_sample_kdeplot(dataset1, dataset2, label1 = 'WITH Sleep Apnea', label2 = 'WITHOUT Sle
    # Calculate the means of both datasets
    mean1 = np.mean(dataset1)
    mean2 = np.mean(dataset2)

    # Perform two sample t-test
    _, p_value = stats.ttest_ind(dataset1, dataset2)
    if p_value < 0.05:
        print('Reject the Null Hypothesis')
    else:
        print('Fail to reject the null hypothesis')

    # Use a style template
    sns.set_style('whitegrid')

    plt.figure(figsize=(12, 8))

    # Plot both datasets
    sns.kdeplot(dataset1, label=label1, color='red', alpha=1)
    sns.kdeplot(dataset2, label=label2, color='blue', alpha=1)

    # Compute KDE values
    kde_dataset1 = stats.gaussian_kde(dataset1)
    kde_dataset2 = stats.gaussian_kde(dataset2)

    # Generate x values for KDE
    x_vals = np.linspace(min(dataset1.min(), dataset2.min()), max(dataset1.max(), dataset2.n

```

```

# Fill the entire area under the curves
plt.fill_between(x_vals, kde_dataset1(x_vals), color='red', alpha=0.3)
plt.fill_between(x_vals, kde_dataset2(x_vals), color='blue', alpha=0.3)

# Get KDE values at means
kde_dataset1_at_mean1 = kde_dataset1(mean1)[0]
kde_dataset2_at_mean2 = kde_dataset2(mean2)[0]

# Add vertical lines at the means, stopping at the KDE curves
plt.plot([mean1, mean1], [0, kde_dataset1_at_mean1], 'r--', label=f'{label1} Mean: {mean1:.2f}')
plt.plot([mean2, mean2], [0, kde_dataset2_at_mean2], 'b--', label=f'{label2} Mean: {mean2:.2f}')

# Add labels and title
plt.title('Test Between With Sleeping Apnea vs Without Sleeping Apnea', fontsize=20, fontweight='bold')
plt.xlabel('Sleeping Apnea', fontsize=16)
plt.ylabel('Probability Density', fontsize=16)
plt.yticks(visible=False)

# Add annotations
plt.annotate(f'{label1} | Mean: {mean1:.2f}', xy=(0.75, 0.85), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='red', boxstyle='round,pad=0.5'))
plt.annotate(f'{label2} | Mean: {mean2:.2f}', xy=(0.75, 0.75), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='blue', boxstyle='round,pad=0.5'))

if save == True:
    plt.savefig(f'{label1} v {label2} TTest.png', transparent=True)
plt.show()

# Creating a function to plot the With and Without Insomnia Test
def I_sample_kdeplot(dataset1, dataset2, label1 = 'WITH Insomnia', label2 = 'WITHOUT Insomnia'):
    # Calculate the means of both datasets
    mean1 = np.mean(dataset1)
    mean2 = np.mean(dataset2)

    # Perform two sample t-test
    _, p_value = stats.ttest_ind(dataset1, dataset2)
    if p_value < 0.05:
        print('Reject the Null Hypothesis')
    else:
        print('Fail to reject the null hypothesis')

    # Use a style template
    sns.set_style('whitegrid')

    plt.figure(figsize=(12, 8))

    # Plot both datasets
    sns.kdeplot(dataset1, label=label1, color='red', alpha=1)
    sns.kdeplot(dataset2, label=label2, color='blue', alpha=1)

```

```

# Compute KDE values
kde_dataset1 = stats.gaussian_kde(dataset1)
kde_dataset2 = stats.gaussian_kde(dataset2)

# Generate x values for KDE
x_vals = np.linspace(min(dataset1.min(), dataset2.min()), max(dataset1.max(), dataset2.max()), 100)

# Fill the entire area under the curves
plt.fill_between(x_vals, kde_dataset1(x_vals), color='red', alpha=0.3)
plt.fill_between(x_vals, kde_dataset2(x_vals), color='blue', alpha=0.3)

# Get KDE values at means
kde_dataset1_at_mean1 = kde_dataset1(mean1)[0]
kde_dataset2_at_mean2 = kde_dataset2(mean2)[0]

# Add vertical lines at the means, stopping at the KDE curves
plt.plot([mean1, mean1], [0, kde_dataset1_at_mean1], 'r--', label=f'{label1} Mean: {mean1:.2f}')
plt.plot([mean2, mean2], [0, kde_dataset2_at_mean2], 'b--', label=f'{label2} Mean: {mean2:.2f}')

# Add labels and title
plt.title('Test Between With Insomnia vs Without Insomnia', fontsize=20, fontweight='bold')
plt.xlabel('', fontsize=16)
plt.ylabel('')
plt.yticks(visible=False)

# Add annotations
plt.annotate(f'{label1} | Mean: {mean1:.2f}', xy=(0.75, 0.85), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='red', boxstyle='round,pad=0.5'))
plt.annotate(f'{label2} | Mean: {mean2:.2f}', xy=(0.75, 0.75), xycoords='axes fraction',
            bbox=dict(facecolor='white', edgecolor='blue', boxstyle='round,pad=0.5'))

if save == True:
    plt.savefig(f'{label1} v {label2} TTest.png', transparent=False)
plt.show()

```

✓ Sleep Duration

This is the first of six different statistics that will be tested on during this section, each statistic will start by applying the central limit theorem to display that it is applied before performing these tests.

```
sns.kdeplot(df['Sleep Duration'])
```



```

# Apply CLT
sample_size = 30 # Usually n >= 30

```

```

num_samples = 1000

# Draw multiple samples and calculate their means
sample_means = []

for _ in range(num_samples):
    sample = df['Sleep Duration'].sample(sample_size, replace=True)
    sample_means.append(sample.mean())

# Plot the distribution of sample means
sns.histplot(sample_means, kde=True)
plt.title('Distribution of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()

```



✓ Hypothesis 1.1: Sleep Duration ANOVA

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO} = \mu_I$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO} \neq \mu_I$

The null hypothesis states that the mean sleep duration for individuals with sleep apnea, individuals with insomnia and those without a sleeping disorder are equal.

The alternate hypothesis states that the null hypothesis is not true

```

H1_SA = df[MASK_SA]['Sleep Duration']
H1_NO = df[MASK_NO]['Sleep Duration']
H1_I = df[MASK_I]['Sleep Duration']

```

```
three_way_ANOVA(H1_SA, H1_NO, H1_I)
```



✓ Hypothesis 1.2: Sleep Duration - Sleeping Disorder vs NO Sleeping Disorder

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

The null hypothesis states that the mean sleep duration for individuals with a sleeping disorder is equal to those without a sleeping disorder.

The alternate hypothesis states the mean sleep duration for individuals with a sleeping disorder is NOT equal to those without a sleeping disorder.

```
H1_SD = df[MASK_SD]['Sleep Duration']
H1_NO = df[MASK_NO]['Sleep Duration']
```

```
two_sample_kdeplot(H1_SD, H1_NO)
```



✓ Hypothesis 1.3: Sleep Duration - Sleep Apnea vs NO Sleep Apnea

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

The null hypothesis states that the mean sleep duration for individuals with sleep apnea is equal to those without sleep apnea.

The alternate hypothesis states the mean sleep duration for individuals with sleep apnea is NOT equal to those without sleep apnea.

```
H1_SA = df[MASK_SA]['Sleep Duration']
H1_NSA = df[MASK_NSA]['Sleep Duration']
```

```
SA_sample_kdeplot(H1_SA, H1_NSA)
```



✓ Hypothesis 1.4: Sleep Duration - Insomnia vs NO Insomnia

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

The null hypothesis states that the mean sleep duration for individuals with insomnia is equal to those without insomnia

The alternate hypothesis states the mean sleep duration for individuals with insomnia is NOT equal to those without insomnia.

```
H1_I = df[MASK_I]['Sleep Duration']
H1_NI = df[MASK_NI]['Sleep Duration']
```

```
I_sample_kdeplot(H1_I, H1_NI, save=True)
```



✓ Quality of Sleep

Similar to the last statistic and all of the next statistics, this one will start with applying CLT before beginning the hypothesis tests

```
sns.kdeplot(df['Quality of Sleep'])
```



```
# Apply CLT
sample_size = 30 # Usually n >= 30
num_samples = 1000

# Draw multiple samples and calculate their means
sample_means = []

for _ in range(num_samples):
    sample = df['Quality of Sleep'].sample(sample_size, replace=True)
    sample_means.append(sample.mean())

# Plot the distribution of sample means
sns.histplot(sample_means, kde=True)
plt.title('Distribution of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()
```



✓ Hypothesis 2.1: Quality of Sleep ANOVA

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO} = \mu_I$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO} \neq \mu_I$

The null hypothesis states that the mean quality of sleep for individuals with sleep apnea, individuals with insomnia and those without a sleeping disorder are equal.

The alternate hypothesis states that the null hypothesis is not true

```
H2_SA = df[MASK_SA]['Quality of Sleep']
H2_NO = df[MASK_NO]['Quality of Sleep']
```

```
H2_I = df[MASK_I]['Quality of Sleep']
```

```
three_way_ANOVA(H2_SA,H2_NO,H2_I)
```



✓ Hypothesis 2.2: Quality of Sleep - Sleeping Disorder vs NO Sleeping Disorder

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

The null hypothesis states that the mean sleep quality for individuals with a sleeping disorder is equal to those without a sleeping disorder.

The alternate hypothesis states the mean sleep quality for individuals with a sleeping disorder is NOT equal to those without a sleeping disorder.

```
H2_SD = df[MASK_SD]['Quality of Sleep']
```

```
H2_NO = df[MASK_NO]['Quality of Sleep']
```

```
two_sample_kdeplot(H2_SD, H2_NO)
```



✓ Hypothesis 2.3: Quality of Sleep - Sleep Apnea vs NO Sleep Apnea

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

The null hypothesis states that the mean sleep quality for individuals with sleep apnea is equal to those without sleep apnea.

The alternate hypothesis states the mean sleep quality for individuals with sleep apnea is NOT equal to those without sleep apnea.

```
H2_SA = df[MASK_SA]['Quality of Sleep']
```

```
H2_NSA = df[MASK_NSA]['Quality of Sleep']
```

```
SA_sample_kdeplot(H2_SA,H2_NSA)
```



✓ Hypothesis 2.4: Quality of Sleep - Insomnia vs NO Insomnia

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

The null hypothesis states that the mean sleep quality for individuals with insomnia is equal to those without insomnia

The alternate hypothesis states the mean sleep quality for individuals with insomnia is NOT equal to those without insomnia.

```
H2_I = df[MASK_I]['Quality of Sleep']
H2_NI = df[MASK_NI]['Quality of Sleep']
```

```
I_sample_kdeplot(H2_I, H2_NI, save=True)
```



✓ Physical Activity Level

Next is physical activity level... I'll be the first to tell you, its the same process as the previous two. Central limit theorem and then hypothesis tests.

Here we go!

```
sns.kdeplot(df['Physical Activity Level'])
```



```
# Applying the Central Limit Theorem
sample_size = 30 # Usually n >= 30
num_samples = 1000
```

```
# Draw multiple samples and calculate their means
sample_means = []
```

```
for _ in range(num_samples):
    sample = df['Physical Activity Level'].sample(sample_size, replace=True)
    sample_means.append(sample.mean())
```

```
# Plot the distribution of sample means
sns.histplot(sample_means, kde=True)
plt.title('Distribution of Sample Means')
plt.xlabel('Sample Mean')
```

```
plt.ylabel('Frequency')
plt.show()
```



✓ Hypothesis 3.1: Physical Activity Level ANOVA

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO} = \mu_I$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO} \neq \mu_I$

The null hypothesis states that the mean physical activity level for individuals with sleep apnea, individuals with insomnia and those without a physical activity level are equal.

The alternate hypothesis states that the null hypothesis is not true

```
H3_SA = df[MASK_SA]['Physical Activity Level']
H3_NO = df[MASK_NO]['Physical Activity Level']
H3_I = df[MASK_I]['Physical Activity Level']
```

```
three_way_ANOVA(H3_SA, H3_NO, H3_I)
```



✓ Hypothesis 3.2: Physical Activity Level - Sleeping Disorder vs NO Sleeping Disorder

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

The null hypothesis states that the mean physical activity level for individuals with a sleeping disorder is equal to those without a sleeping disorder.

The alternate hypothesis states the mean physical activity level for individuals with a sleeping disorder is NOT equal to those without a sleeping disorder.

```
H3_SD = df[MASK_SD]['Physical Activity Level']
H3_NO = df[MASK_NO]['Physical Activity Level']
```

```
two_sample_kdeplot(H3_SD, H3_NO)
```



✓ Hypothesis 3.3: Physical Activity Level - Sleep Apnea vs NO Sleep Apnea

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

The null hypothesis states that the mean physical activity level for individuals with sleep apnea is equal to those without sleep apnea.

The alternate hypothesis states the mean physical activity level for individuals with sleep apnea is NOT equal to those without sleep apnea.

```
H3_SA = df[MASK_SA]['Physical Activity Level']
H3_NSA = df[MASK_NSA]['Physical Activity Level']
```

```
SA_sample_kdeplot(H3_SA, H3_NSA)
```



✓ Hypothesis 3.4: Physical Activity Level - Insomnia vs NO Insomnia

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

The null hypothesis states that the mean physical activity level for individuals with insomnia is equal to those without insomnia

The alternate hypothesis states the mean physical activity level for individuals with insomnia is NOT equal to those without insomnia.

```
H3_I = df[MASK_I]['Physical Activity Level']
H3_NI = df[MASK_NI]['Physical Activity Level']
```

```
I_sample_kdeplot(H3_I, H3_NI)
```



✓ Stress Level

I hope you know whats going to happen next by now because it hasnt changed yet.

```
sns.kdeplot(df['Stress Level'])
```



```
# Apply CLT
sample_size = 30 # Usually n >= 30
num_samples = 1000

# Draw multiple samples and calculate their means
sample_means = []

for _ in range(num_samples):
    sample = df['Stress Level'].sample(sample_size, replace=True)
    sample_means.append(sample.mean())

# Plot the distribution of sample means
sns.histplot(sample_means, kde=True)
plt.title('Distribution of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()
```



✓ Hypothesis 4.1: Stress Level ANOVA

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO} = \mu_I$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO} \neq \mu_I$

The null hypothesis states that the mean stress level for individuals with sleep apnea, individuals with insomnia and those without a sleeping disorder are equal.

The alternate hypothesis states that the null hypothesis is not true

```
H4_SA = df[MASK_SA]['Stress Level']
H4_NO = df[MASK_NO]['Stress Level']
H4_I = df[MASK_I]['Stress Level']
```

```
three_way_ANOVA(H4_SA, H4_NO, H4_I)
```



✓ Hypothesis 4.2: Stress Level - Sleeping Disorder vs NO Sleeping Disorder

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

The null hypothesis states that the mean stress level for individuals with a sleeping disorder is equal to those without a sleeping disorder.

The alternate hypothesis states the mean stress level for individuals with a sleeping disorder is NOT equal to those without a sleeping disorder.

```
H4_SD = df[MASK_SD]['Stress Level']
H4_NO = df[MASK_NO]['Stress Level']
```

```
two_sample_kdeplot(H4_SD, H4_NO)
```



✓ Hypothesis 4.3: Stress Level - Sleep Apnea vs NO Sleep Apnea

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

The null hypothesis states that the mean stress level for individuals with sleep apnea is equal to those without sleep apnea.

The alternate hypothesis states the mean stress level for individuals with sleep apnea is NOT equal to those without sleep apnea.

```
H4_SA = df[MASK_SA]['Stress Level']
H4_NSA = df[MASK_NSA]['Stress Level']
```

```
SA_sample_kdeplot(H4_SA, H4_NSA)
```



✓ Hypothesis 4.4: Stress Level - Insomnia vs NO Insomnia

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

The null hypothesis states that the mean stress level for individuals with insomnia is equal to those without insomnia

The alternate hypothesis states the mean stress level for individuals with insomnia is NOT equal to those without insomnia.

```
H4_I = df[MASK_I]['Stress Level']
H4_NI = df[MASK_NI]['Stress Level']
```

```
I_sample_kdeplot(H4_I, H4_NI, save=True)
```



✓ Heart Rate

We are on the second last statistic now, almost finished.

```
sns.kdeplot(df['Heart Rate'])
```



```
# Apply CLT
sample_size = 30 # Usually n >= 30
num_samples = 1000

# Draw multiple samples and calculate their means
sample_means = []

for _ in range(num_samples):
    sample = df['Heart Rate'].sample(sample_size, replace=True)
    sample_means.append(sample.mean())

# Plot the distribution of sample means
sns.histplot(sample_means, kde=True)
plt.title('Distribution of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()
```



✓ Hypothesis 5.1: Heart Rate ANOVA

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO}$

The null hypothesis states that the mean heart rate for individuals with sleep apnea, individuals with insomnia and those without a sleeping disorder are equal.

The alternate hypothesis states that the null hypothesis is not true

```
H5_SA = df[MASK_SA]['Heart Rate']
H5_NO = df[MASK_NO]['Heart Rate']
H5_I = df[MASK_I]['Heart Rate']
```

```
three_way_ANOVA(H5_SA, H5_NO, H5_I)
```



✓ Hypothesis 5.2: Heart Rate - Sleeping Disorder vs NO Sleeping Disorder

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

The null hypothesis states that the mean heart rate for individuals with a sleeping disorder is equal to those without a sleeping disorder.

The alternate hypothesis states the mean heart rate for individuals with a sleeping disorder is NOT equal to those without a sleeping disorder.

```
H5_SD = df[MASK_SD]['Heart Rate']
H5_NO = df[MASK_NO]['Heart Rate']
```

```
two_sample_kdeplot(H5_SD, H5_NO)
```



✓ Hypothesis 5.3: Heart Rate - Sleep Apnea vs NO Sleep Apnea

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

The null hypothesis states that the mean heart rate for individuals with sleep apnea is equal to those without sleep apnea.

The alternate hypothesis states the mean heart rate for individuals with sleep apnea is NOT equal to those without sleep apnea.

```
H5_SA = df[MASK_SA]['Heart Rate']
H5_NSA = df[MASK_NSA]['Heart Rate']
```

```
SA_sample_kdeplot(H5_SA, H5_NSA)
```



✓ Hypothesis 5.4: Heart Rate - Insomnia vs NO Insomnia

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

The null hypothesis states that the mean heart rate for individuals with insomnia is equal to those without insomnia

The alternate hypothesis states the mean heart rate for individuals with insomnia is NOT equal to those without insomnia.

```
H5_I = df[MASK_I]['Heart Rate']
H5_NI = df[MASK_NI]['Heart Rate']
```

```
I_sample_kdeplot(H5_I, H5_NI)
```



✓ Daily Steps

Finally we've made it to the final statistic of this inferential analysis. Nothing will change though, CLT then hypothesis tests.

Let's do this!

```
sns.kdeplot(df['Daily Steps'])
```



```
# Apply CLT
sample_size = 30 # Usually n >= 30
num_samples = 1000

# Draw multiple samples and calculate their means
sample_means = []

for _ in range(num_samples):
    sample = df['Daily Steps'].sample(sample_size, replace=True)
    sample_means.append(sample.mean())

# Plot the distribution of sample means
sns.histplot(sample_means, kde=True)
plt.title('Distribution of Sample Means')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()
```



✓ Hypothesis 6.1: Daily Steps ANOVA

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO} = \mu_I$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO} \neq \mu_I$

The null hypothesis states that the mean daily steps for individuals with sleep apnea, individuals with insomnia and those without a sleeping disorder are equal.

The alternate hypothesis states that the null hypothesis is not true

```
H6_SA = df[MASK_SA]['Daily Steps']
H6_NO = df[MASK_NO]['Daily Steps']
H6_I = df[MASK_I]['Daily Steps']
```

```
three_way_ANOVA(H6_SA, H6_NO, H6_I)
```



✓ Hypothesis 6.2: Daily Steps - Sleeping Disorder vs NO Sleeping Disorder

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

The null hypothesis states that the mean daily steps for individuals with a sleeping disorder is equal to those without a sleeping disorder.

The alternate hypothesis states the mean daily steps for individuals with a sleeping disorder is NOT equal to those without a sleeping disorder.

```
H6_SD = df[MASK_SD]['Daily Steps']
H6_NO = df[MASK_NO]['Daily Steps']
```

```
two_sample_kdeplot(H6_SD, H6_NO)
```



✓ Hypothesis 6.3: Daily Steps - Sleep Apnea vs NO Sleep Apnea

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

The null hypothesis states that the mean daily steps for individuals with sleep apnea is equal to those without sleep apnea.

The alternate hypothesis states the mean daily steps for individuals with sleep apnea is NOT equal to those without sleep apnea.

```
H6_SA = df[MASK_SA]['Daily Steps']
H6_NSA = df[MASK_NSA]['Daily Steps']
```

```
SA_sample_kdeplot(H6_SA, H6_NSA)
```



✓ Hypothesis 6.4: Daily Steps - Insomnia vs NO Insomnia

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

The null hypothesis states that the mean daily steps for individuals with insomnia is equal to those without insomnia

The alternate hypothesis states the mean daily steps for individuals with insomnia is NOT equal to those without insomnia.

```
H6_I = df[MASK_I]['Daily Steps']
H6_NI = df[MASK_NI]['Daily Steps']
```

```
I_sample_kdeplot(H6_I, H6_NI, save=True)
```



✓ Summary

The main focus of my inferential analysis was to focus on how different the lives of individuals with sleeping disorders is in comparison to those without. I asked a number of relatively basic and descriptive questions to get a better idea of my dataset and then I went onto my inferential analysis where i performed many hypothesis test which gave me a better idea of which statistics have a relation with sleeping disorders.

Demographic Questions

I started my analysis by asking a handful of demographic related questions to get an idea of who was included in this sample size. My question were related to gender, age, occupation, BMI index and sleeping disorder.

After gaining a base level of understanding for those demogrpahics, I began my descriptive analysis

✓ Descriptive Analysis

In my descriptive analysis, I asked and answered six descriptive question about my data. Those questions were the following:

- Which-Profession-in-my-dataset-receives-the-most-amount-of-sleep?
- Which Profession has the most recorded cases of a sleeping disorder?
- On average, how much sleep does a person in each BMI category recieve?
- On average, how much sleep does someone with a sleeping disorder recieve compared someone with no sleeping disorder?
- On average, what is the quality of sleep that someone with a sleeping disorder recieves compared to those without a sleeping disorder?
- What distribution of Males and Females in my dataset are diagnosed with sleeping disorders?
- What percentage of my dataset has Insomnia?
- Of those with Insomnia, What percentage is male vs female?
- What is the distribution of age for patients with insomnia in my dataset?

Once I answered these questions, I felt that I had a good enough grasp on the data to continue my data exploration

Insomnia vs Everyone Else

After completing the original descriptive analysis, I wanted to get a preliminary idea of the difference in means for various statistics in patients that have insomnia vs patients that do not have insomnia. To do so, I measured the means of sleep duration, quality of sleep, phsysical activity level, daily steps and stress level.

✓ Inferential Analysis

During my inferential analysis, I performed a number of hypothesis tests on various statistics in my dataset. I performed 4 hypothesis tests for each statistic and my findings are as follows

✓ ANOVA Test

These tests were three-way ANOVA tests in which I grouped my data by patients with Sleep Apnea (SA), Insomina (I) and no sleeping disorder (NO) and found the mean (μ) for each test variable and tested for a significant difference with a significance level (α) of 0.05.

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NO}$

Sleep Duration

In this test, I calculated the means and performed the ANOVA test. In this test, the mean (μ) refers to the mean sleep duration for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in sleep duration to reject the null hypothesis, this does not mean that the reason for the significant difference in sleep duration is due to having a sleep apnea, insomnia, or no sleeping disorder. We simply observed that the difference exists.

Quality of Sleep

In this test, I calculated the means and performed the ANOVA test. In this test, the mean (μ) refers to the mean quality of sleep for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in quality of sleep to reject the null hypothesis, this does not mean that the reason for the significant difference in quality of sleep is due to having a sleep apnea, insomnia, or no sleeping disorder. We simply observed that the difference exists.

Physical Activity Level

In this test, I calculated the means and performed the ANOVA test. In this test, the mean (μ) refers to the mean physical activity level for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in physical activity level to reject the null hypothesis, this does not mean that the reason for the significant difference in physical activity level is due to having a sleep apnea, insomnia, or no sleeping disorder. We simply observed that the difference exists.

Stress Level

In this test, I calculated the means and performed the ANOVA test. In this test, the mean (μ) refers to the mean stress level for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in stress level to reject the null hypothesis, this does not mean that the reason for the significant difference in stress level is due to having a sleep apnea, insomnia, or no sleeping disorder. We simply observed that the difference exists.

Heart Rate

In this test, I calculated the means and performed the ANOVA test. In this test, the mean (μ) refers to the mean heart rate for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in heart rate to reject the null hypothesis, this does not mean that the reason for the significant difference in heart rate is due to having a sleep apnea, insomnia, or no sleeping disorder. We simply observed that the difference exists.

✓ Daily Steps

In this test, I calculated the means and performed the ANOVA test. In this test, the mean (μ) refers to the mean daily steps for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in daily steps to reject the null hypothesis, this does not mean that the reason for the significant difference in daily steps is due to having a sleep apnea, insomnia, or no sleeping disorder. We simply observed that the difference exists.

✓ With vs Without Sleeping Disorder Two Sample T-Tests

These tests were two sample t-tests in which I grouped my data by patients with a sleeping disorder (SD) and no sleeping disorder (NO) and found the mean (μ) for each test variable and tested for a significant difference with a significance level (α) of 0.05.

Null Hypothesis (H_0): $\mu_{SD} = \mu_{NO}$

Alternate Hypothesis (H_A): $\mu_{SD} \neq \mu_{NO}$

Sleep Duration

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean sleep duration for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in sleep duration to reject the null hypothesis, this does not mean that the reason for the significant difference in sleep duration is due to having a sleeping disorder or not having a sleeping disorder. We simply observed that the difference exists.

Quality of Sleep

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean quality of sleep for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in quality of sleep to reject the null hypothesis, this does not mean that the reason for the significant difference in quality of sleep is due to having a sleeping disorder or not having a sleeping disorder. We simply observed that the difference exists.

Physical Activity Level

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean physical activity level for each group.

Result: **Fail to Reject the Null Hypothesis.** In this test, we did not observe enough of a difference to differentiate the change in means that would naturally occur from randomness

Stress Level

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean stress level for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in stress level to reject the null hypothesis, this does not mean that the reason for the significant difference in stress level is due to having a sleeping disorder or not having a sleeping disorder. We simply observed that the difference exists.

Heart Rate

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean heart rate for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in heart rate to reject the null hypothesis, this does not mean that the reason for the significant difference in heart rate is due to having a sleeping disorder or not having a sleeping disorder. We simply observed that the difference exists.

✓ Daily Steps

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean sleep duration for each group.

Result: **Fail to Reject the Null Hypothesis.** In this test, we did not observe enough of a difference to differentiate the change in means that would naturally occur from randomness

✓ With vs Without Sleep Apnea Two Sample T-Tests

These tests were two sample t-tests in which I grouped my data by patients with sleep apnea (SA) and without sleep apnea (NSA) and found the mean (μ) for each test variable and tested for a significant difference with a significance level (α) of 0.05.

Null Hypothesis (H_0): $\mu_{SA} = \mu_{NSA}$

Alternate Hypothesis (H_A): $\mu_{SA} \neq \mu_{NSA}$

Sleep Duration

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean sleep duration for each group.

Result: **Fail to Reject the Null Hypothesis.** In this test, we did not observe enough of a difference to differentiate the change in means that would naturally occur from randomness

Quality of Sleep

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean quality of sleep for each group.

Result: **Fail to Reject the Null Hypothesis.** In this test, we did not observe enough of a difference to differentiate the change in means that would naturally occur from randomness

Physical Activity Level

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean physical activity level for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in physical activity level to reject the null hypothesis, this does not mean that the reason for the significant difference in physical activity level is due to having sleep apnea or not having sleep apnea. We simply observed that the difference exists.

Stress Level

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean stress level for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in stress level to reject the null hypothesis, this does not mean that the reason for the significant difference in stress level is due to having sleep apnea or not having sleep apnea. We simply observed that the difference exists.

Heart Rate

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean heart rate for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in heart rate to reject the null hypothesis, this does not mean that the reason for the significant difference in heart rate is due to having sleep apnea or not having sleep apnea. We simply observed that the difference exists.

✓ Daily Steps

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean daily steps for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in daily steps to reject the null hypothesis, this does not mean that the reason for the significant difference in daily steps is due to having sleep apnea or not having sleep apnea. We simply observed that the difference exists.

✓ With vs Without Insomnia Two Sample T-Tests

These tests were two sample t-tests in which I grouped my data by patients with insomnia (I) and without insomnia (NI) and found the mean (μ) for each test variable and tested for a significant difference with a significance level (alpha) of 0.05.

Null Hypothesis (H_0): $\mu_I = \mu_{NI}$

Alternate Hypothesis (H_A): $\mu_I \neq \mu_{NI}$

Sleep Duration

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean sleep duration for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in sleep duration to reject the null hypothesis, this does not mean that the reason for the significant difference in sleep duration is due to having insomnia or not having insomnia. We simply observed that the difference exists.

Quality of Sleep

In this test, I calculated the means and performed a two sample t-test. In this test, the mean (μ) refers to the mean quality of sleep for each group.

Result: **Reject the Null Hypothesis.** Although we observed a significant enough difference in quality of sleep to reject the null hypothesis, this does not mean that the reason for the significant difference in quality of sleep is due to having insomnia or not having insomnia. We simply observed that the difference exists.

Physical Activity Level