Statistical Inference

Minerva University

FA50: Formal Analyses

Prof. Wilkins

December 10, 2023

Statistical Inference.

Physical activity and sleep are integral components of adolescent well-being, influencing their physical, mental, and academic health (Hallal et al., 2006). The interplay between these factors becomes particularly pivotal during high school, marked by significant developmental changes and increasing academic pressures (Wolfson & Carskadon, 2003). Central to this research question: Does being physically inactive (0 days a week of activity) affect the amount of sleep students get compared to those who engage in physical activity between 1 to 7 days a week? The YRBSS dataset used in the previous assignment will be employed again. The data in YRBSS is collected through anonymous, voluntary surveys conducted biennially in schools nationwide. I plan to make inferences about the population of 9th to 12th-grade high school students in the USA. Examining the link between sleep and physical activity is vital for pinpointing interventions to improve adolescent health.

Based on the dataset, two variables were selected: school night hours of sleep and physical activity days per week. Dependent Variable (School night hours of sleep): Although time by itself is continuous, this variable is quantitative and discrete. It measures the number of hours a student sleeps on a school night. Although the time is a continuous variable, I will use it as discrete because, in the data, it is recorded in whole numbers or rounded to the nearest half-hour, representing countable states (e.g., 6 hours, 7 hours, 8 hours, etc.).

Independent Variable (Physically active days per week): This variable is qualitative ordinal. The categories likely range from "0 days" (no physical activity) to "7 days" (daily physical activity). This ordering implies a hierarchy where each category is more than the

previous, where the exact differences between them are uniform. To address the central research question regarding the impact of physical inactivity on sleep duration, we categorize the independent variable into two groups: physically inactive (0 days of activity per week) and physically active (1 to 7 days of activity per week).[1]

The significance level for this study is set at α=0.05, a commonly accepted standard in research for balancing Type I and Type II errors. Our hypotheses are formulated as follows:

Null Hypothesis (H_0): There is no difference in school night sleep hours between physically active and inactive students (μ_a - μ_i = 0).

Alternative Hypothesis (H_A): There is a difference in school night sleep hours between physically active and inactive students (μ_a - μ_i ≠ 0).

A two-tailed test is employed to detect potential differences in either direction. The ensuing sections will elaborate on the methodology and execution of this hypothesis testing.

	Physically inactive students' sleeping hours per school night	Physically active students' sleeping hours per school night
Sample size (n)	n_i = 16	n_a = 75
Mean ($\bar{x}$)	$\bar{x}_i$ = 6.69	$\bar{x}_a$ = 7.04

[1] **#variables:** The clear identification and classification of 'Physically active days per week' as an ordinal independent variable and 'School night hours of sleep' as a discrete dependent variable significantly enhance the paper's analytical framework. For the reader, this distinction provides clarity on the nature of the data being analyzed and the relationship being examined. It ensures that the statistical methods used are appropriate for the types of data involved, thus strengthening the validity of the study's conclusions and making the findings more accessible and understandable.

Median	6.5	7.0
Mode	5, 6	7
Range	4	5
Standard deviation (s)	$s_i = 1.4$	$s_a = 1.29$

Table 1.The sample sizes, means, medians, modes, ranges, and sample standard deviations of two subgroups: sleep hours of physically active and inactive students.

Physically inactive students, a smaller group, tend to sleep less and have more varied sleep patterns, averaging about 6.5 hours with notable fluctuation. In contrast, the larger group of physically active students generally sleeps more, around 7 hours, with less variation in their sleep patterns, indicating more consistency in their nightly rest.[2]

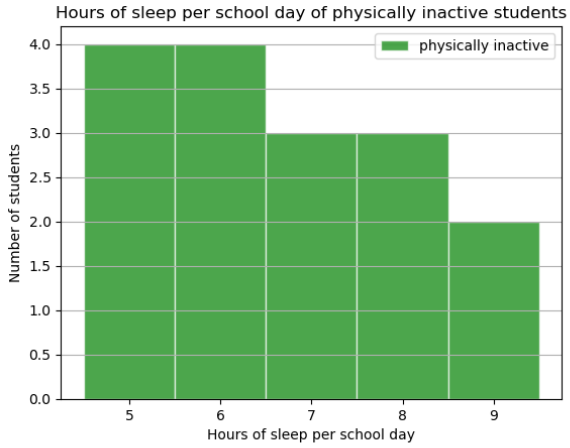

Figure 1. Distribution of school night sleep hours for students who are physically active 0 days per week.

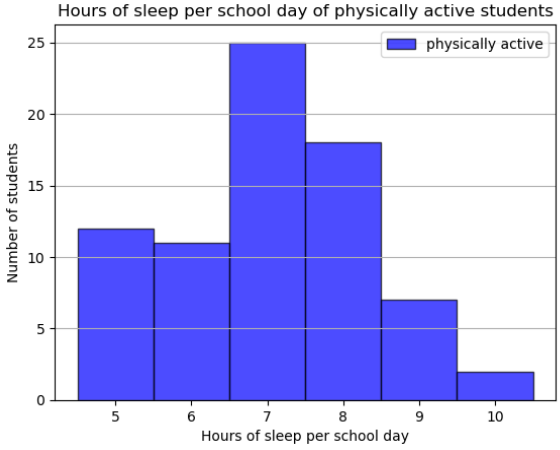

Figure 2. Distribution of school night sleep hours for students who are physically active more than or equal to 1 day per week.

[2] **#descriptivestats:** The application effectively utilizes measures of central tendency and variability to summarize the sleep patterns of high school students. The mean sleep hours—6.69 for physically inactive and 7.04 for physically active students—along with the standard deviations, illustrate the average behavior and spread of the data. This detailed use of descriptive statistics provides a clear, quantitative foundation for the subsequent inferential analysis, allowing for a nuanced understanding of the dataset and facilitating informed conclusions about sleep habits across different levels of physical activity.

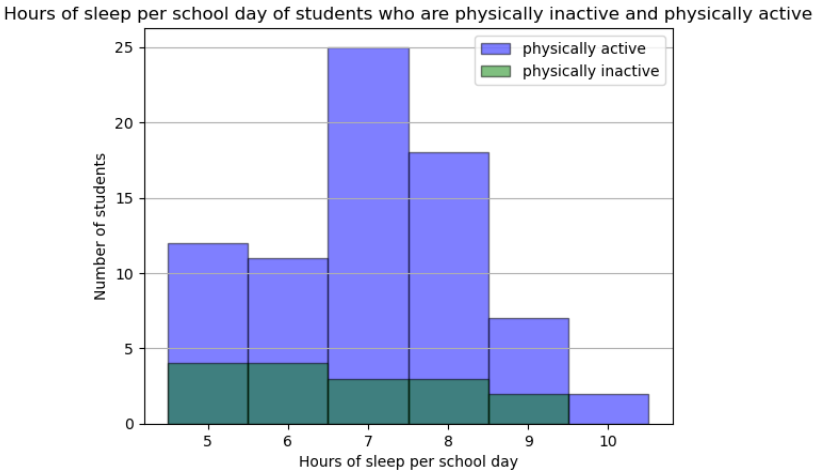

Figure 3. Comparison of School Night Sleep Hours for Physically Inactive Students (in green) and Physically Active Srudents (in blue). Where 'Physically Active' means being active 1-7 days a week, while 'Physically Inactive' studnts are active 0 days a week.

The histograms (Figures 1 and 2) show the sleep distributions, with inactive students peaking at lower sleep hours compared to their active peers. Figure 3 compares the two, indicating a modestly higher average sleep duration for active students. These observations will be scrutinized for statistical significance in subsequent analysis.[3]

Given that our standard error calculation utilizes the sample standard deviation, the t-distribution is preferred to be the chosen sampling distribution for our inferential statistics. This decision is critical for small sample sizes (n_i<30), as the t-distribution is wider than the normal distribution, with fatter tails, leading to broader confidence intervals. Such properties diminish the risk of committing Type I errors, which occur when the null hypothesis is incorrectly rejected. To justify the use of the t-distribution, we assess the appropriateness of the following

[3] **#dataviz:** I used histograms to visualize the distribution of sleep hours among physically inactive and active students, providing a clear representation of the data's distribution. This choice of visualization effectively communicates the variation and central tendencies in sleep patterns as influenced by the independent variable—physical activity level—allowing for a more nuanced understanding of its impact on sleep duration.

Central Limit Theory conditions:

- Randomness: Verified through the methodology of the survey administration.
- Normality: $n_a > 30$, Figure 2 is normal in shape. Although $n_i < 30$, the figure shape is not strongly skewed (Figure 1).
- Independence: Our sample of 100 entries is less than 10% of the U.S. high school student population, which makes our data “independent”.

Assuming these conditions hold, the central limit theorem would imply that the sampling distribution of the mean approximates normality. Nevertheless, the stark contrast in the sample sizes of our two groups—16 for physically inactive students and 75 for physically active students—introduces potential limitations, such as imbalances in the statistical power, which might bias the data interpretation. The smaller subgroup might not sufficiently mirror the larger population, raising the chances of a sampling error. This size difference could result in the over or underestimation of the effect size if the smaller group's variance fails to reflect the population's variance accurately. Moreover, this discrepancy could also undermine the precision of the confidence intervals and the validity of the p-values, thus affecting the strength of our conclusions. It is, therefore, imperative to acknowledge these constraints when interpreting the results to ensure the robustness of our analytical inferences.

The difference between the sample means is quantified in evaluating the statistical significance between the sleep durations of physically inactive and active students. This is accomplished by calculating the T-score following the standard formula for a difference of

means test, $T = \frac{\bar{x}_a - \bar{x}_i}{SE}$ which includes the computation of the standard error, $SE = \sqrt{\frac{s_a^2}{n_a} + \frac{s_i^2}{n_i}}$. The degree of freedom for the t-distribution is conservatively estimated using the smaller sample size, as detailed in Appendix C where the Python code for this calculation is provided. A T-score of 0.92 indicates that the mean difference is 0.92 standard errors from zero, yielding a two-tailed p-value of 0.37, which exceeds the alpha level of 0.05. This p-value quantifies the likelihood of observing such a mean difference if the null hypothesis were true, suggesting that there is no significant difference between the two groups.[4] [5]

Given the lack of statistical significance in sleep durations between physically inactive and active students, one might infer a negligible effect size, being equal to 0. Nonetheless, it is still necessary to quantify the effect size to confirm this. Cohen's d is selected as the measure of effect size due to its adjustment for small sample size biases. The calculation of Cohen's d involves the pooled standard deviation, $SD_{pooled} = \sqrt{\frac{(n_a - 1)s_a^2 + (n_i - 1)s_i^2}{n_a + n_i - 2}}$ adjusted for unequal

[4] **#distributions:** Sample distributions refer to the distribution of observed data points in the collected sample, as seen in the histograms for physically active and inactive students. In contrast, sampling distributions pertain to the theoretical distribution of a statistic (like the mean or proportion) calculated across all possible samples of a given size from the population. The study's use of the t-distribution for inferential analysis is a prime example of applying a sampling distribution, appropriate due to the smaller sample size of the physically inactive group, thereby adhering to the principles of the Central Limit Theorem.

[5] **#probability:** Probability was used in the interpretation of the p-value, a fundamental probability concept, which is calculated as 0.37 in the t-test comparing sleep durations between physically inactive and active students. This p-value represents the probability of observing such a difference in means if the null hypothesis were true. Its interpretation correctly guides the decision to retain the null hypothesis, as it exceeds the predefined significance level of 0.05. This application of probability effectively demonstrates an understanding of its role in hypothesis testing, ensuring the results are both statistically valid and relevant to the research question.

group sizes, as outlined in Appendix C. The resultant effect size, $d = \frac{\bar{x}_a - \bar{x}_i}{SD_{pooled}}$ of 0.27 is considered small, implying that the observed difference in sleep hours between physically active and inactive high school students is minimal from both a statistical and practical standpoint.[6]

To further scrutinize our hypotheses, we construct confidence intervals for the average nightly sleep hours of both physically active and inactive student subgroups. These intervals provide insight into the precision of our mean estimates. Opting for a 95% confidence level, we strike a balance between the desired precision of the estimate and the confidence in our interval. The pre-established conditions for inference—randomness, normality, and independence—support the validity of using the t-distribution for these calculations.

For each subgroup, the confidence interval is computed according to the standard formula $\bar{x} \pm t^*_{df} \cdot SE$, which involves the t-multiplier corresponding to the 95% confidence level, determined by the degrees of freedom. The standard error is calculated using its formula $SE = \frac{s}{\sqrt{n}}$, where sample standard deviation is used because the population SD is unknown. These computations are fully detailed in Appendix C. The computed 95% confidence intervals, to two decimal places, are as follows: physically active students have an interval of [6.74, 7.34] hours, and physically inactive students have an interval of [5.94, 7.44] hours for sleep per school night. Within these intervals, we have 95% confidence that the true population means of nightly sleep hours for both groups fall. Statistically, this implies that in 95 out of 100 such samples, the

[6] **#significance:** Significance is demonstrated through a two-tailed t-test, yielding a p-value of 0.37, which indicates no significant difference in sleep duration between physically inactive and active students. The small effect size (Cohen's d = 0.27) further suggests minimal practical differences, effectively distinguishing between statistical and practical significance in the context of adolescent sleep patterns.

true mean will lie within the calculated range, adhering to the frequentist probability framework. The overlapping of the intervals for both groups suggests that there is no meaningful difference between the groups' mean sleep hours. Therefore, both the p-value and the confidence intervals corroborate the lack of a statistically significant difference in sleep duration between physically active and inactive students, leading us to retain the null hypothesis. These findings suggest that, within the sample studied, physical activity may not be a strong determinant of sleep duration among the broader population of U.S. high school students.[7]

The process of drawing conclusions about a larger population from a sample is an exercise in inductive reasoning. It involves making generalizations based on limited observations, which, while powerful, carry inherent weaknesses. One limitation in our study stems from the sample sizes: the physically inactive group is $n_i < 30$, which is conventionally considered suitable for approximating normal distributions and making robust inferences. Consequently, the smaller sample size may lead to a less reliable estimation of the population parameters and weaker induction. This is reflected in wider confidence intervals and potentially less precise effect size measurements, as seen with Cohen's d. In contrast, the larger sample size of the physically active group ($n_a = 75$) likely provides a stronger, more reliable basis for induction and more accurate estimates of population parameters.

While our analysis did not find significant differences in sleep hours based on physical

[7]**#confidenceintervals:** The 95% confidence intervals, accurately calculated for each subgroup with correctly identified t-score, [6.74, 7.34] hours for active students and [5.94, 7.44] hours for inactive students, were pivotal in illustrating the precision and reliability of the mean estimates. The overlapping nature of these intervals played a crucial role in supporting the decision to retain the null hypothesis, demonstrating no significant difference in sleep duration between the two groups. This application underscores the importance of confidence intervals in hypothesis testing, as they not only provide a range for the estimated parameter but also offer insight into the statistical significance and practical relevance of the findings, which is crucial for drawing valid and meaningful conclusions from the data.

activity levels, these conclusions are tempered by the study's limitations. The small sample of inactive students presents a challenge to reliability and precision in our inferences, unlike the larger active student sample, which offers more dependable conclusions. Despite no observed significant differences in sleep-related activity levels, the study's validity is constrained by sample size discrepancies. A more comprehensive study with balanced samples is needed for firmer conclusions on physical activity's effect on sleep.[8] [9]

Word Count: 1577 words

Appendix

I modified my code from the previous assignment to find descriptive stats of the new variable (Abdikarim, 2023). Each code line was explained in detail.

Importing and editing the data

```
#importing the data
import numpy as np # Importing NumPy module and labeling as np
import pandas as pd # Importing Pandas module and labeling as pd
from scipy import stats
#from scipy.stats import t
df = pd.read_csv("https://course-resources.minerva.edu/uploaded_files/mu/00294341-4390/yrbss-samp.csv")

df.head(100)
```

[8] **#induction:** Using the findings from the sample to the broader population of high school students is inductive reasoning. This process is identified as inductive due to its reliance on specific evidence from the sample (premise) to draw probable conclusions about the population (conclusion). The strength and reliability of this induction have been critically analyzed, acknowledging limitations due to sample sizes and suggesting future studies with increased sample sizes to enhance the reliability of these inductive conclusions.

[9] **#organization:** A coherent structure guides the reader from a clear introduction of the research question to a well-supported conclusion. The methodical arrangement of variables, statistical analyses, and discussion sections enables easy navigation and comprehension. This organization ensures that the findings regarding the relationship between physical activity and sleep among adolescents are presented in a logical sequence, enhancing the overall clarity and impact of the communication.

	age	gender	grade	hispanic	race	height	weight	helmet_12m	text_while_driving_30d	physically_active_7d	hours_tv_per_school_day	strength_trainir
0	16.0	female	11.0	not	Black or African American	1.50	52.62	never	1-2	0	4	
1	17.0	male	11.0	not	White	1.78	74.84	rarely	0	7	1	
2	17.0	male	11.0	not	White	1.75	106.60	never	0	7	2	
3	15.0	male	10.0	hispanic	NaN	1.68	66.68	never	did not drive	3	2	
4	18.0	male	12.0	not	Black or African American	1.70	80.29	never	did not drive	0	2	
...	...	...	...	...	...	...	...	...	...	...	...	
95	17.0	male	11.0	not	White	1.80	63.50	always	0	2	do not watch	
96	16.0	female	10.0	not	White	1.63	49.90	did not ride	0	0	1	
97	15.0	male	10.0	not	Black or African American	1.78	79.38	never	0	6	3	
98	15.0	male	9.0	not	White	1.68	58.97	never	1-2	7	5+	
99	14.0	male	9.0	not	White	1.70	55.79	never	did not drive	1	3	

100 rows × 13 columns

```
new_df = df[['physically_active_7d', 'school_night_hours_sleep']]

# Rows with 'NA' cannot give valuable information, that is why I removed rows containing "NA" values.
df = new_df.dropna(how='any', axis=0)
df.head(100)

# Also, to make the evaluation more precise, i changes "<5" and "5+" values to 5, "<1" to 1, and "10+" tp 10.
#df= df.replace('do not watch', 0)
df= df.replace('5+', 5)
df= df.replace('<5', 5)
df= df.replace('10+', 10)
df= df.replace('<1', 1)
# it is unknown what 5+ or 10+ mean. This change will affect the outcomes of mean, median, range, standard deviation and
# mode is not likely to be affected, because these unsusual unswers are infrequeant.
# I changed 4 answer types: 5+, <5, 10+, <1. The data has more 5+ and 10+ than <5 and <1. So, I assume that my outcome w
# And my figure shape is likely to be less left scewed, because of lack of possible extreme numbers (5+, 10+).

# Convert 'physically_active_7d' to numeric (integers or floats)
df['physically_active_7d'] = pd.to_numeric(df['physically_active_7d'], errors='coerce')
df['school_night_hours_sleep'] = pd.to_numeric(df['school_night_hours_sleep'], errors='coerce')

# Filter based on the 'physically_active_7d' column
df_x_1 = df[df['physically_active_7d'] < 1] # variable of "physically_active_7d" that is equal to 0.
df_x_2 = df[df['physically_active_7d'] >= 1] # variable of "physically_active_7d" that is more than or equal to 1.
```

```
# shows the first 10 rows of data tables
print(df_x_1.head(10))
print()
print(df_x_2.head(10))
print()
# lists creation
df_y_1 = df_x_1['school_night_hours_sleep'].tolist()
df_y_2 = df_x_2['school_night_hours_sleep'].tolist()
df_x_1 = df_x_1['physically_active_7d'].tolist()
df_x_2 = df_x_2['physically_active_7d'].tolist()
n_i = len(df_x_1)
n_a = len(df_x_2)
print("df_y_1 = ", df_y_1)
print("df_y_2 = ", df_y_2)

print("count of inactive students = ", n_i)
print("count of active studnts = ", n_a)
```

```
    physically_active_7d  school_night_hours_sleep
0                      0                         8
4                      0                         6
16                     0                         5
20                     0                         6
25                     0                         5
30                     0                         9
34                     0                         8
59                     0                         7
62                     0                         9
66                     0                         5

    physically_active_7d  school_night_hours_sleep
1                      7                         7
2                      7                         7
3                      3                         5
5                      4                         5
6                      7                         7
7                      5                         7
8                      7                         7
9                      6                         8
10                     7                        10
11                     2                         5
```

```
df_y_1 =  [8, 6, 5, 6, 5, 9, 8, 7, 9, 5, 7, 7, 8, 5, 6, 6]
df_y_2 =  [7, 7, 5, 5, 7, 7, 7, 8, 10, 5, 5, 6, 8, 5, 8, 8, 8, 6, 6, 7, 7, 7, 8, 5, 7, 9, 9, 7, 6, 8, 8, 5, 5, 7, 9,
5, 7, 8, 7, 7, 7, 8, 5, 7, 8, 7, 8, 9, 7, 10, 7, 8, 6, 6, 7, 9, 7, 6, 8, 6, 7, 6, 8, 7, 9, 8, 5, 6, 8, 8, 5, 7, 7, 6,
9]
count of inactive students =  16
count of active studnts =  75
```

Descriptive statistics functions

```python
# function for the mean
def mean(data):
    N = len(data) # to find the lenght of the data
    sum = 0 # needed for calculathion of the sum
    for i in data: # takes each components of the data from i to N. (0 to N-1)
        sum += i # adds each component of the 'sum', and when the loop ends gives the sum of all data components.
    return sum/N # returns the mean, which is Sum / number of components.
# print(mean([2, 4, 6, 0, 3, 4]))
# I used the print to check my code right after writing it
```

```python
# write your function for the median here
def median(data):
    data.sort() # uses built-in functions to sort the data.
    N = len(data) # calculates the length of the data
    if len(data) % 2 == 0: #if the N is an even number, it means that the list has 2 components to find a mean of
        median1 = data[N//2] # identifies the 1st
        median2 = data[N//2 - 1] # and 2nd component
        median = (median1 + median2)/2 #mean of those two, that will give the median of the data
    else:
        median = data[N//2] # if the N is not an even number, the median is data[N//2]
    return median #returnes the median

#lst_my = [2, 4, 5, 6, 0, 3]
#print(median(lst_my))
# I used the print to check my code right after writing it
```

```python
def Mode(data):
    # Check if the input list is empty, and if so, return an empty list
    if len(data) == 0:
        return []

    mode_count = {} #empty dictionary to store the count of each unique element
    max_count = 0 # Initialize a variable to keep track of the maximum count
    modes = [] # Initialize an empty list to store the modes found in the input list

    for i in data: #check each elemet in the data
        if i in mode_count: # Check if the element exists in the mode_count dictionary
            mode_count[i] += 1 # If it is, add + 1
        else:

            mode_count[i] = 1 # If it's not, add it to the dictionary with a count of 1

        # if the count of one element mode [i] is greater than max_count, then give max_count the value of mode_count[i]
        if mode_count[i] > max_count:
            max_count = mode_count[i]

    # Iterate through the elements and their counts in the mode_count dictionary
    for key, count in mode_count.items():
        if count == max_count: # Check if the count = maximum count
            # If it does, add the element to the modes list
             modes.append(key)

    # Returns the modes as a list
    return modes
```

```python
# function for the range
def range_data(data):
    min_value = None #introduces the min_value
    max_value = None #introduces the max_value

    for i in data: # calls each element of the data
        if max_value == None or i > max_value: #and checks if the element's value is more than the max_value
             max_value = i #if yes, then max_value gets new value. The process continues until the loop ends.
        if min_value == None or i < min_value: #same logic is used for min_value
             min_value = i

    return max_value - min_value # returnes the range, which is max_value - min_value.

# giving the dataset to check the code
#answer is 10-1= 9.
#print(range_data([2, 4, 6, 1, 2, 7, 4, 8, 10]))
```

```python
# write your function for the standard deviation here
def standard_deviation(data):
    Mean = mean(data) #calls the function to fins the mean of the data
    sumtotal = 0 #requiered for Sum.

    for x in data: # uses all elements of the data.
        sumtotal = sumtotal + (x - Mean)**2 #recreated the summation of the formula E((x - Mean)**2)

    #it is sample standard deviation formula, that is why sumtotal/(len(data) -1)
    return (sumtotal/(len(data)-1))**0.5 #sqrt of sumtotal/(len(data)-1)
```

```
print(mean([2, 4, 6, 0, 3, 4]), "built in code:", np.mean([2, 4, 6, 0, 3, 4]))
print(median([2, 4, 5, 6, 0, 3]), "built in code:", np.median([2, 4, 5, 6, 0, 3]))
print(Mode([1, 2, 2, 3, 4, 4, 4, 5])) #checks if it can show only one mode
print(Mode([1, 2, 2, 3, 4, 4])) # when the list has many modes
print(Mode([1, 2, 3, 4])) # if each element is unique
print(range_data([2, 4, 6, 1, 2, 7, 4, 8, 10]), "built in code:", np.ptp([2, 4, 6, 1, 2, 7, 4, 8, 10]))
print(standard_deviation([2, 4, 6, 1]), "built in code:", np.std([2, 4, 6, 1], ddof=1))
```

```
3.1666666666666665 built in code: 3.1666666666666665
3.5 built in code: 3.5
[4]
[2, 4]
[1, 2, 3, 4]
9 built in code: 9
2.217355782608345 built in code: 2.217355782608345
```

```
# call the functions and print the stats here
x_i = round(mean(df_y_1),2)
median_i = round(median(df_y_1),2)
mode_i = Mode(df_y_1)
range_i = round(range_data(df_y_1),2)
s_i = round(standard_deviation(df_y_1),2)

print()

print("mean of physically inactive subgroup = ", x_i)
print("median of physically inactive subgroup = ", median_i)
print("mode of physically inactive subgroup = ", mode_i)
print("range of physically inactive subgroup = ", range_i)
print("ssample sd of physically inactive subgroup = ", s_i)

print()
print()

x_a = round(mean(df_y_2),2)
median_a = round(median(df_y_2),2)
mode_a = Mode(df_y_2)
range_a = round(range_data(df_y_2),2)
s_a = round(standard_deviation(df_y_2),2)
```

```
print("mean of physically active subgroup = ", x_a)
print("median of physically active subgroup = ", median_a)
print("mode of physically active subgroup = ", mode_a)
print("range of physically active subgroup = ", range_a)
print("sample sd of physically active subgroup = ", s_a)

print()

#print("Check: built in mean of physically active subgroup = ", np.mean(df_y_2))
#print("Check: median of physically active subgroup = ", np.median(df_y_2))
#print("Check: range of physically active subgroup = ", np.ptp(df_y_2))
#print("Check: sample sd of physically active subgroup = ", np.std(df_y_2))
```

```
mean of physically inactive subgroup =  6.69
median of physically inactive subgroup =  6.5
mode of physically inactive subgroup =  [5, 6]
range of physically inactive subgroup =  4
ssample sd of physically inactive subgroup =  1.4
```

```
mean of physically active subgroup =  7.04
median of physically active subgroup =  7
mode of physically active subgroup =  [7]
range of physically active subgroup =  5
sample sd of physically active subgroup =  1.29
```

Histograms

```
import pandas as pd  #library for data manipulation and analysis. It provides data structures like DataFrames.
import matplotlib.pyplot as plt  #library for creating static, interactive, and animated visualizations in Python.
%matplotlib inline
# Magic command that makes matplotlib plots appear inline within the notebook.

# Histogram of physically inactive students' sleep hours
data_i = np.array(df_y_1)  # Convertes the df_y_1 data into a NumPy array for ease of manipulation

# Calculates the minimum bin width based on the unique values in the data
d_i = np.diff(np.unique(data_i)).min()
left_of_first_bin_i = data_i.min() - float(d_i)/2  # Calculates the left edge of the first bin
right_of_last_bin_i = data_i.max() + float(d_i)/2  # Calculates the right edge of the last bin
# Creates a histogram with specified bin edges, label, color, edge color, and transparency.
plt.hist(data_i, np.arange(left_of_first_bin_i, right_of_last_bin_i + d_i, d_i), label='physically inactive', color='gre
plt.title('Hours of sleep per school day of physically inactive students')  #title of the histogram
plt.xlabel('Hours of sleep per school day')  # x-axis label
plt.ylabel('Number of students')  # y-axis label
plt.legend(loc='upper right')  # puts the legend to the upper right corner of the plot
plt.grid(axis='y')  # Add horizontal gridlines
#figure caption at the bottom of the plot
plt.figtext(0.5, -0.1, "Figure 1. Distribution of school night sleep hours for students who are physically active 0 days
plt.show()
```

```
#similar steps are repeated for physically active students' sleep hours
data_a = np.array(df_y_2)  # Convertes df_y_2 data into a NumPy array.

# Calculates bin edges similar to the previous histogram
d_a = np.diff(np.unique(data_a)).min()
left_of_first_bin_a = data_a.min() - float(d_a)/2
right_of_last_bin_a = data_a.max() + float(d_a)/2
# Creates and displays the histogram for physically active students
plt.hist(data_a, np.arange(left_of_first_bin_a, right_of_last_bin_a + d_a, d_a), label='physically active', color='blue'
plt.title('Hours of sleep per school day of physically active students')
plt.xlabel('Hours of sleep per school day')
plt.ylabel('Number of students')
plt.legend(loc='upper right')
plt.grid(axis='y')
plt.figtext(0.5, -0.1, "Figure 2. Distribution of school night sleep hours for students who are physically active more t
plt.show()
# Histogram comparing physically active and inactive students' sleep hours
plt.hist(data_a, np.arange(left_of_first_bin_a, right_of_last_bin_a + d_a, d_a), label='physically active', color='blue'
plt.hist(data_i, np.arange(left_of_first_bin_i, right_of_last_bin_i + d_i, d_i), label='physically inactive', color='gre

plt.title('Hours of sleep per school day of students who are physically inactive and physically active')
plt.xlabel('Hours of sleep per school day')
plt.ylabel('Number of students')
plt.legend(loc='upper right')
# Add gridlines to the plot
plt.grid(axis='y')
plt.figtext(0.5,-0.1,"Figure 3. Comparison of School Night Sleep Hours for Physically Inactive Students (in green) and P
plt.show()
```

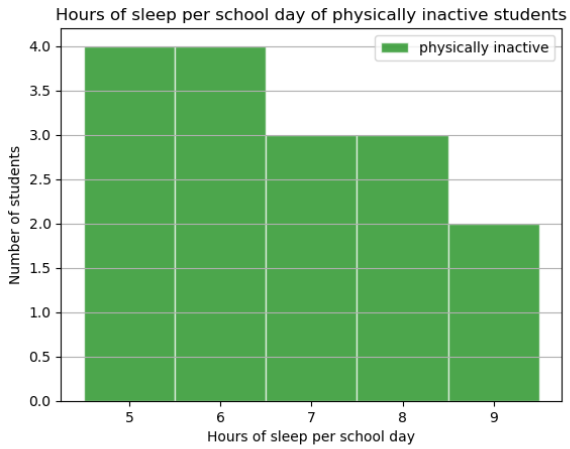

Figure 1. Distribution of school night sleep hours for students who are physically active 0 days per week.

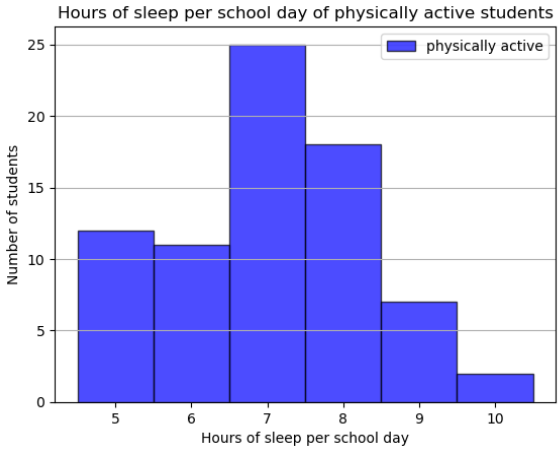

Figure 2. Distribution of school night sleep hours for students who are physically active more than or equal to 1 day per week.

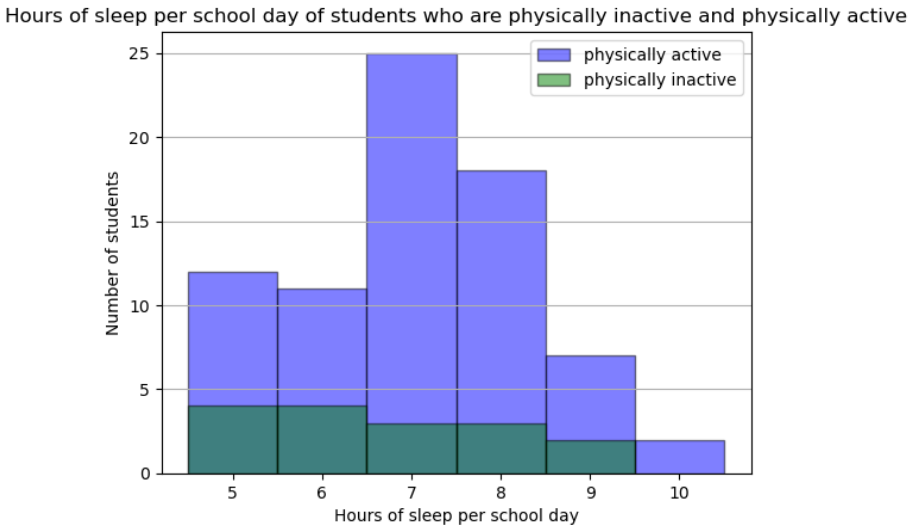

Figure 3. Comparison of School Night Sleep Hours for Physically Inactive Students (in green) and Physically Active Srudents (in blue). Where 'Physically Active' means being active 1-7 days a week, while 'Physically Inactive' studnts are active 0 days a week.

Statistical Inference

```
# Previously calculated statistics: sample sd (s_a and s_i), sample size(n_a and n_i), and mean(x_a and x_i)

# Difference of means T-score calculation
std_error_diff = np.sqrt((s_i**2 / n_i) + (s_a**2 / n_a))
t_score_diff = (x_a - x_i) / std_error_diff

# Degrees of freedom for the difference of means
df_diff = min(n_i - 1, n_a - 1)

# P-value for the difference of means
p_value_diff = 2 * stats.t.sf(np.abs(t_score_diff), df_diff)

# Confidence intervals for physically inactive and active studnts sleep hours per night
confidence_level = 0.95
alpha = 1 - confidence_level
t_multiplier_i = stats.t.ppf(1 - alpha/2, n_i - 1)
t_multiplier_a = stats.t.ppf(1 - alpha/2, n_a - 1)

ci_i = (round(x_i - t_multiplier_i * s_i / np.sqrt(n_i), 2), round(x_i + t_multiplier_i * s_i / np.sqrt(n_i), 2))
ci_a = (round(x_a - t_multiplier_a * s_a / np.sqrt(n_a), 2), round(x_a + t_multiplier_a * s_a / np.sqrt(n_a), 2))

# Pooled standard deviation
pooled_sd = np.sqrt(((n_a - 1) * s_a**2 + (n_i - 1) * s_i**2) / (n_a + n_i - 2))

# Effect size (Cohen's d): means difference / pooled sd.
d = (x_a - x_i) / pooled_sd
```

```
print("t-score:", round(t_score_diff, 2))
print("degree of freedom:", df_diff)
print("p-value:", p_value_diff)
print("confidence interval of physically active people:",ci_a)
print("confidence interval of physically inactive people:",ci_i)
print("Pooled standard deviation:", round(pooled_sd, 2))
print("Cohen's d:", round(d, 2))
```

```
t-score: 0.92
degree of freedom: 15
p-value: 0.37206334558295917
confidence interval of physically active people: (6.74, 7.34)
confidence interval of physically inactive people: (5.94, 7.44)
Pooled standard deviation: 1.31
Cohen's d: 0.27
```

Reflection

Testing: To ensure the accuracy of my results, I used Python's statistical libraries for calculations and testing of my descriptive stats functions. To verify the correctness of confidence intervals and p-values, I used the M26 website, which calculates these values by taking a URL link of the dataset. This methodical approach provided a double-check against potential computational errors, assuring the validity of the confidence intervals and p-values derived.

Using feedback: In a previous assignment, I got a comment about the bin sizes of histograms and that integers should be in the center of each bin. This time, I paid attention to this important detail that allows a clear visualization of the distribution. Also, I got a comment on the use of #induction that I should explain the reliability and the strength of the induction in interpreting the results and note why this interpretation is induction. I believe that I used comments properly to make my applications of #dataviz and #induction stronger.

Acknowledgments: I stayed after Session 22 to learn about #induction used in the conclusion part. I thank Nazym Zhumaiygaiyeva for searching the internet for information about

histogram bins' size and locations. Also, I want to note the M26 student who shared his website to check the results of statistical inference (El Statistician, 2023).

AI Statement: I used Grammarly to detect grammar and punctuation errors in the assignment.[10]

[10] **#professionalism:** The paper carefully attributes all data sources, follows appropriate statistical methods, and presents findings in a structured and formal manner. This strict observance of professional practices in documentation and communication not only enhances the paper's credibility but also demonstrates a commitment to the ethical dissemination of information.

References

Abdikarim, A. (2023). FA Assignment 2: Describing Data. [Unpublished assignment for FA50], Minerva University

El Statistician. (2023). El-Statistician.vercel.app. Retrieved December 10, 2023, from https://el-statistician.vercel.app/i/ci

Hallal, P. C., Victora, C. G., Azevedo, M. R., & Wells, J. C. K. (2006). Adolescent Physical Activity and Health. Sports Medicine, 36(12), 1019–1030. https://doi.org/10.2165/00007256-200636120-00003

Wolfson, A. R., & Carskadon, M. A. (2003). Understanding adolescent's sleep patterns and school performance: a critical appraisal. Sleep Medicine Reviews, 7(6), 491–506. https://doi.org/10.1016/s1087-0792(03)90003-7