LBA Statistical Inference

Minerva University

CS50: Formal Analyses

Prof. Stan

December 09, 2021

LBA Statistical Inference

Introduction

My chosen dataset was the Medial Temporal Lobe (MTL) and other data for 26 participants. For this paper, I will focus on two variables: the level of anxiety of participants and their metabolic unit score. Through statistical calculations, I will analyze whether the level of metabolic unit score plays a role in the anxiety levels of participants. Analyzing this aspect is important to determine if a low level of metabolic unit can affect someone's levels of anxiety. Investigating this issue is important given the reality that many people sit for hours every day—from kids in school to adults in offices, and not doing exercises overall.

Dataset

My chosen dataset was the Medial Temporal Lobe (MTL) and other data for 26 participants from OpenIntro, a free platform that contains books focused on math subjects. The dataset has drawn its information from a sample of 25 women and 10 men (middle-aged or older) (OpenIntro, n.d.) from a study conducted by Dr. Prabha Siddarth at the University of California at Los Angeles (Friedman, 2018). The database contains information about the gender, and age of the participants, but focuses on how many hours they spend sitting per day, their levels of anxiety and depression based on the Hamilton Rate Scale, and brain characteristics such as the thickness of specific subareas.

For this paper, I will focus on two variables. The first one is the level of anxiety reported by participants using the Hamilton Rating Scale. This variable is quantitative, discrete, and ordinal (it expresses a kind of magnitude—low or high level). This paper will analyze the dependent characteristic of this variable related to the other analyzed variable. The second variable is the metabolic unit score, a quantitative, independent, and continuous variable.[1] These variables were specifically chosen with the purpose of answering the question "are levels of anxiety influenced by high and low levels of metabolic unit scores?". For this analysis, rows with null values on the data set were excluded using Python (see Appendix A).

Analysis

Hypotheses

Before performing a difference of means test to examine if levels of anxiety are influenced by metabolic units, we need to define our null and alternative hypotheses, and compare the

[1] **#variables:** I identified and classified the key variables used for my test, defining dependent, independent variables and other classifications. I also explained the relation between the variables, and how one may affect the other.

subgroups. The sample was divided into “low metabolic units” and “high metabolic units” based on the predefined variable given by the dataset that categorizes the levels of metabolic units into those two groups. The significance value of the test was set to α = 0.05, since committing a Type II error, in this case, seems to have worse consequences (if we had a α = 0.01, we would favor Type II error from happening).

The hypotheses are:

1. Null Hypothesis (H0): People with low levels of metabolic units have the same levels of anxiety as people with high levels of metabolic units.
2. Alternative Hypothesis (HA): People with low levels of metabolic units have higher levels of anxiety than people with high levels of metabolic units.

Since the test only plans to analyze if low metabolic levels cause higher levels of anxiety than high levels of metabolic units, the test will be one-tailed.

Summary statistics

The dataset was read into Python using the pandas’ library. As the first step of the hypothesis test, the descriptive statistics for the defined subgroups were set. Table 1 summarizes those (see Appendix A—which also provides a general summary of other variables).

Table 1: Summary statistics for the anxiety level people with low and high metabolic unit (activity level)

	Anxiety Levels based on Low Metabolic Unit (activity level)	Anxiety Levels based on High Metabolic Unit (activity level)
Count	n_1 = 13	n_2 = 20
Mean	$\bar{x}_1$ = 4.95	$\bar{x}_2$ = 3.53
Median	4.0	2.0
Mode	1.0	2.0
Standard Deviation	s_1 = 4.03	s_2 = 2.66
Range	12.0	9.0

The following histograms (see Appendix B) show the sample distribution of both subgroups and important statistics such as the mean and the median. Analyzing the table and the histograms make it possible to infer the level of anxiety is slightly higher for people with lower metabolic units.[2]

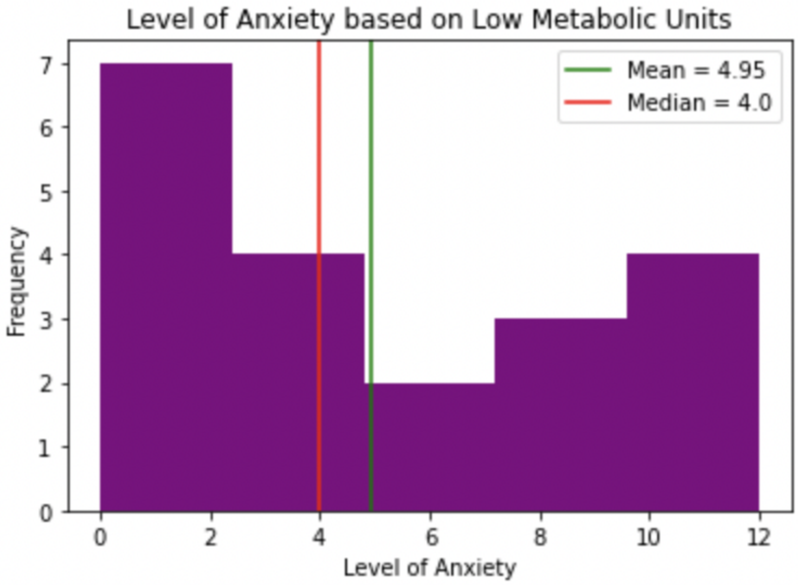


Figure 1: Histogram for the level of anxiety of people with low metabolic units.

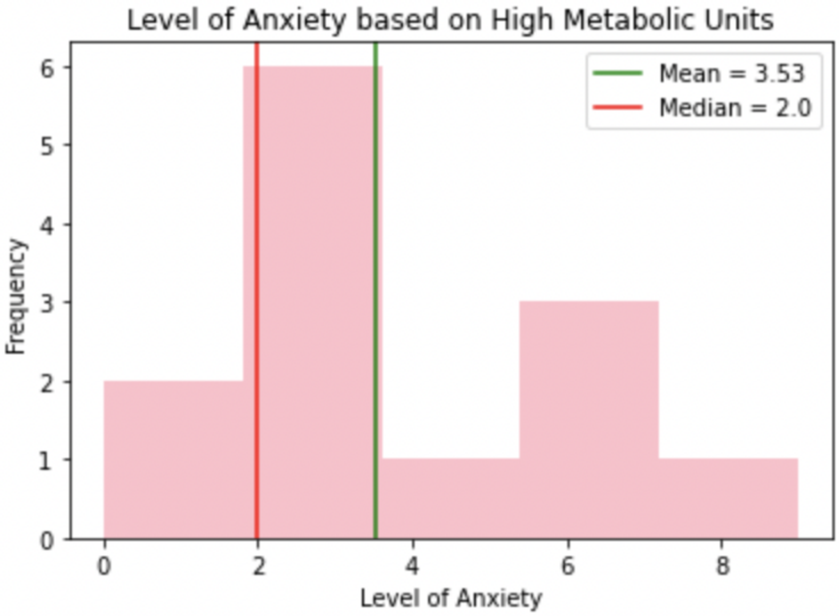


Figure 2: Histogram for the level of anxiety of people with high metabolic units.

[2] **#descriptivestats:** The mean, median, and other key descriptive statistics of the data were presented and later interpreted and analyzed to inform decision-making based on the null and alternative hypothesis. Histograms were also provided highlighting key features of descriptive stats, being interpreted later on.

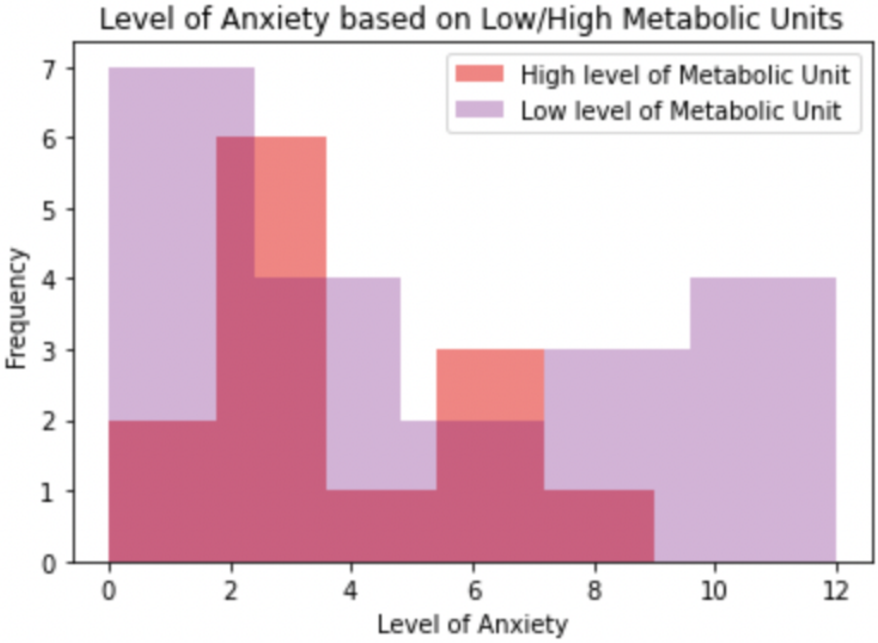


Figure 3: Histogram with the level of anxiety for both people with low and high metabolic units.[3]

Conditions for inference

For the inference, a t-distribution will be used as the standard deviation of the population is not known and the distribution is not normal. Also, we need to analyze if our data meet the needed conditions for a t-distribution to be applied:

- **Normal Distribution Characteristics:** the sample sizes are not greater than 30, lowering the test's plausibility. This fact will be ignored for the sake of calculations, but it is crucial to highlight that the sample sizes should be 30 or greater to meet t-distribution requirements.
- **Randomness:** The main sample of 35 was conveniently chosen in the original study. This does not guarantee randomness, so the subgroup samples are also not necessarily random. As we can not know this aspect will be ignored for the sake of calculations. However, it is important to highlight that the samples need to be random for a well drawn test.
- **Independence (10% rule):** The sample was chosen from the overall population of middle-aged or older people, so it is fair to assume that the sample sizes are less than 10% of the population.[4]

[3] **#dataviz:** I effectively interpreted, analyzed and created a fitting data visualization that met the information I wanted to convey. I chose an appropriate graph to showcase the data, and provided justifications and details about the visualization.

[4] **#distributions:** I accurately identified an appropriate distribution to be used in the given context, making the necessary assumptions for it to work and describing features of this distribution. I included all relevant calculations, further detailing it on the appendix. I also explained the limitations and consequences of the errors of the conditions for the t-test.

Difference of Means Test

To assess statistical significance the difference of the means of the t-distribution is calculated. Since we already know the mean of each sample group, the following can be done:

- H0: $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$
- HA: $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$

First, we compute the T-score using the following formula (difference of means test): $T = (\bar{x}_1 - \bar{x}_2)/\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$. From that, we find a T-score of 1.210 and a standard error (SE) of 1.165. From those values, we can draw a p-value of 0.124, which is bigger than our 0.05 significance level (see Appendix C). The T-score (1.210) means that this is the number of standard deviations above the mean. The p-value represents the probability of observing data at least as favorable to the alternative as our current dataset, if the null hypothesis is true.[5] Thus, since we found a p-value of 0.124>0.05 we conclude that the data favors the null hypothesis (we fail to reject the null hypothesis).

Since we had this p-value result, it is important to assess its practical and statistical significance. For that, we can measure the effect size using Hedge's g. This method was chosen because of the small sample size being manipulated on the test—Hedge's g calculation corrects the upward bias, more evident in small sample sizes. Computing Hedge's g requires the pooled standard $sp = \sqrt{\frac{(n-1)s_1^2 + (n-2)\, s_2^2}{n_1 + n_2 - 2}}$. The resulting effect size of -0.3861 (see Appendix C) indicates we have a small effect size (it is a negative result because one sample is smaller than the other—to assess the effect size we use the absolute value). In the context of this test, this means there is a small difference in anxiety levels between low and high metabolic units (a statistical but not a practical significance).[6] A bigger sample size of the population and more samplings could give us a clearer picture of this comparison.

Confidence Intervals

To understand the difference between both subgroups, a confidence interval for the mean of the anxiety levels of low metabolic united people and high metabolic united people will be drawn. The confidence intervals will provide plausible ranges for the population mean to

[5] **#probability**: The p-value information is effectively interpreted in the test context, explaining its meaning in a t-test and what it implied for the hypothesis. I demonstrated all key calculations, diving more deep on the appendix.

[6] **#significance**: To assess the significance of the teste a measure of significance was interpreted to explain the effect size considering the sample size. I cleraly distinguished the Type I and Type II error and explained what the effect size meant on the context of the test, considering the concepts of statistical and practical significance.

be contained in. Choosing a 95% confidence interval means that if we calculated the confidence interval from a 100 different samples, about 95 of them would contain the true population mean. This confidence interval is wide enough to contain the true population mean, but is also narrow enough. We compute the standard error as: $SE = \bar{x} \pm t * (s/\sqrt{n})$. The t-score for 95% confidence level is 1.73 for high metabolic united people and 1.78 for low metabolic united ones (see Appendix D). The resulting confidence intervals, rounded to three decimal places, are:

- Levels of anxiety for high level metabolic units: [2.501, 4.558]
- Levels of anxiety for low level metabolic units: [2.960, 6.939]

The fact both intervals overlap means that the means of the samples can be the same, indicating there is a weak correlation between the anxiety levels and metabolic units.[7]

Results and Conclusions

Through the calculations, we can conclude there is no difference between the levels of anxiety of people with low or high levels of metabolic units. The p-value (0.124) and effect size (absolute value = 0.3861) make it clear it is not possible for us to reject the null hypothesis, while the overlapping nature of the confidence intervals further weakens the alternative hypothesis.

This is a strong induction because all calculations served as premises/evidence to the final conclusion and because those are predictions—they only represent the likelihood of scenarios. Since all calculations were followed according to statistical analysis conventions and correct formulas and types of tests, the induction is strengthened. However, as already mentioned, the sample size of both used samples are < 30 and we can not guarantee randomness, meaning the plausibility of the t-test is compromised and the induction process weakened.[8]

In summary, we have shown that people with low levels of metabolic units have approximately the same levels of anxiety as people with high levels of metabolic units.

Word Count: 1472 words.

References

Data Sets. (n.d.). OpenIntro. https://www.openintro.org/data/index.php?data=mtl

[7] **#confidenceintervals:** The means of both samples were estimated using confidence intervals. The meaning of the interval was interpreted and explained, considering overlaps between the results and they mean. The calculations were presented clearly and in detailed steps on the appendix.

[8] **#induction:** I effectively analyzed the premisses that led to the conclusion, pointing their strenght, justifying them and pointing their limitations given the context.

Friedman, R. A. (2018, April 19). *Opinion | Standing Up at Your Desk Could Make You Smarter*. The New York Times. https://www.nytimes.com/2018/04/19/opinion/standing-up-at-your-desk-could-make-you-smarter.html

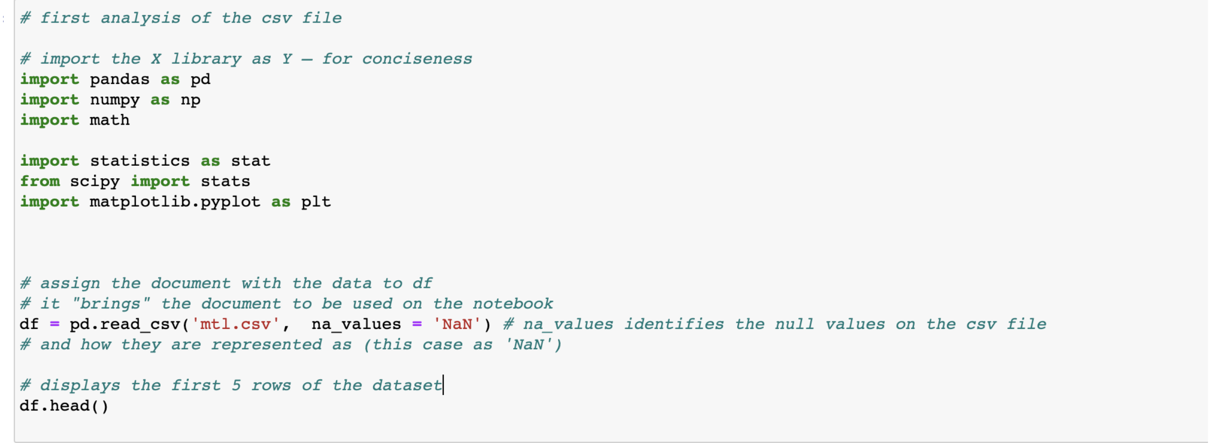
Reflection

For this assignment, I used the feedback for #dataviz from the “Variables with LBA” assignment. The grading comment advised me to always include captions and highlight important statistics points such as the mean. Based on that, I included captions for the visualizations and highlighted important features such as the mean and the median. The feedback not only helped me do a better job on this assignment, but it also made me realize what a good #dataviz is and what key features should be included to ensure clear communication.

Appendix

The full Jupyther notebooks including the Python code and comments can be accessed on the zipped folder submitted as the secondary file.

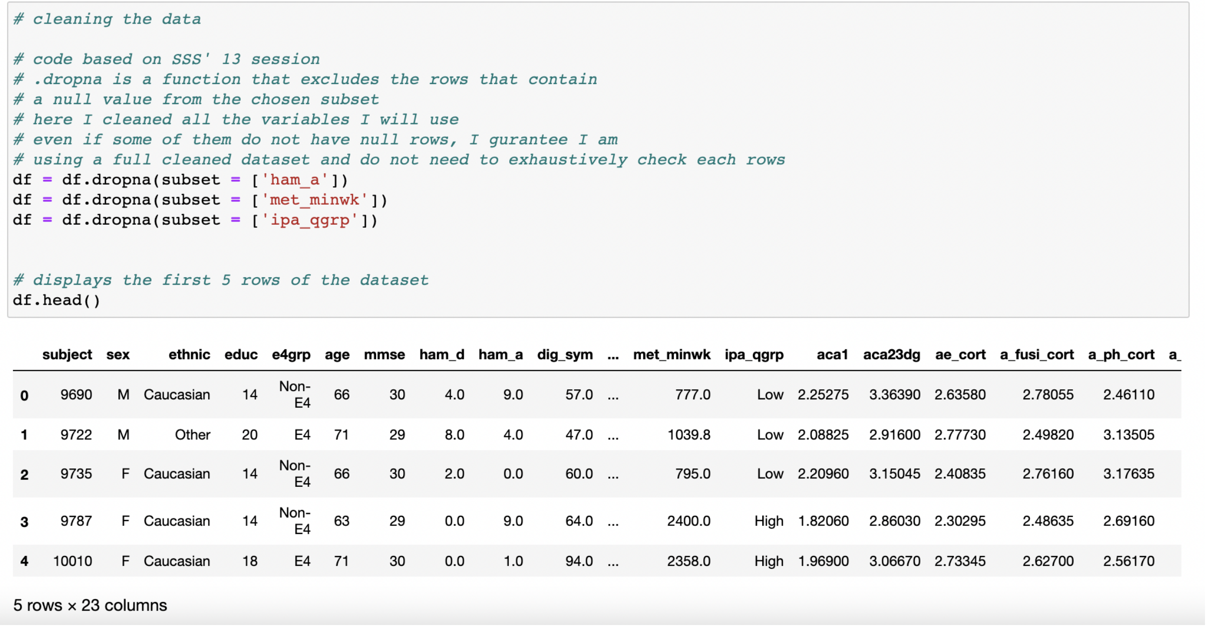
Appendix A: Import, Analyze, and Visualize Data



```
# first analysis of the csv file

# import the X library as Y - for conciseness
import pandas as pd
import numpy as np
import math

import statistics as stat
from scipy import stats
import matplotlib.pyplot as plt



# assign the document with the data to df
# it "brings" the document to be used on the notebook
df = pd.read_csv('mtl.csv',  na_values = 'NaN') # na_values identifies the null values on the csv file
# and how they are represented as (this case as 'NaN')

# displays the first 5 rows of the dataset
df.head()
```

	subject	sex	ethnic	educ	e4grp	age	mmse	ham_d	ham_a	dig_sym	...	met_minwk	ipa_qgrp	aca1	aca23dg	ae_cort	a_fusi_cort	a_ph_cort	a_
0	9690	M	Caucasian	14	Non-E4	66	30	4.0	9.0	57.0	...	777.0	Low	2.25275	3.36390	2.63580	2.78055	2.46110	
1	9722	M	Other	20	E4	71	29	8.0	4.0	47.0	...	1039.8	Low	2.08825	2.91600	2.77730	2.49820	3.13505	
2	9735	F	Caucasian	14	Non-E4	66	30	2.0	0.0	60.0	...	795.0	Low	2.20960	3.15045	2.40835	2.76160	3.17635	
3	9787	F	Caucasian	14	Non-E4	63	29	0.0	9.0	64.0	...	2400.0	High	1.82060	2.86030	2.30295	2.48635	2.69160	
4	10010	F	Caucasian	18	E4	71	30	0.0	1.0	94.0	...	2358.0	High	1.96900	3.06670	2.73345	2.62700	2.56170	

5 rows × 23 columns



```
# cleaning the data

# code based on SSS' 13 session
# .dropna is a function that excludes the rows that contain
# a null value from the chosen subset
# here I cleaned all the variables I will use
# even if some of them do not have null rows, I gurantee I am
# using a full cleaned dataset and do not need to exhaustively check each rows
df = df.dropna(subset = ['ham_a'])
df = df.dropna(subset = ['met_minwk'])
df = df.dropna(subset = ['ipa_qgrp'])


# displays the first 5 rows of the dataset
df.head()
```

	subject	sex	ethnic	educ	e4grp	age	mmse	ham_d	ham_a	dig_sym	...	met_minwk	ipa_qgrp	aca1	aca23dg	ae_cort	a_fusi_cort	a_ph_cort	a_
0	9690	M	Caucasian	14	Non-E4	66	30	4.0	9.0	57.0	...	777.0	Low	2.25275	3.36390	2.63580	2.78055	2.46110	
1	9722	M	Other	20	E4	71	29	8.0	4.0	47.0	...	1039.8	Low	2.08825	2.91600	2.77730	2.49820	3.13505	
2	9735	F	Caucasian	14	Non-E4	66	30	2.0	0.0	60.0	...	795.0	Low	2.20960	3.15045	2.40835	2.76160	3.17635	
3	9787	F	Caucasian	14	Non-E4	63	29	0.0	9.0	64.0	...	2400.0	High	1.82060	2.86030	2.30295	2.48635	2.69160	
4	10010	F	Caucasian	18	E4	71	30	0.0	1.0	94.0	...	2358.0	High	1.96900	3.06670	2.73345	2.62700	2.56170	

5 rows × 23 columns

```
# to get descriptive stats informations

# code based on SSS' 13 session
# .describe() function shows 8 descriptive stats informations

anx_describe = df['ham_a'].describe() # for anxiety levels
act_describe = df['met_minwk'].describe() # for metabolic units
print(anx_describe)
print(act_describe)


# since .describe() does not give us all needed/important information
# we need to compute the following informations separatly

# this code was adapted from the one done by me for the Variables with LBA assignement
# calculate range
def do_range(data):
    # transforms the function's parameter into a list
    if type(data) != list:
        is_data = data.tolist()
    else:
        is_data = data

    # sort the list
    is_data.sort()

    # calculates the range of the data set using the range's formulae
    the_range = is_data[-1] - is_data[0]

    return the_range

# uses the mode() function from pandas
# calculate the mode of the dataset rows
anx_mode = df['ham_a'].mode() # for anxiety levels
act_mode = df['met_minwk'].mode() # for metabolic units

# using statistics library to deal with lists
low_mode = stat.mode(low_met) # for the list with anxiety levels of low metabolic units
high_mode = stat.mode(high_met) # for the list with anxiety levels of high metabolic units


# print all the modes
print('The anxiety level mode is:', anx_mode)
print('The levels of metabolic unit mode is:', act_mode)
print('The anxiety level mode of low leveled metabolic unit people is:', low_mode)
print('The anxiety level mode of high leveled metabolic unit people is:', high_mode)
```

```
# uses the median() function from pandas
# calculate the median of the dataset rows
anx_median = df['ham_a'].median()
act_median = df['met_minwk'].median()

# using statistics library to deal with lists
low_median = stat.median(low_met) # for the list with anxiety levels of low metabolic units
high_median = stat.median(high_met) # for the list with anxiety levels of high metabolic units

# print all the medians
print('The anxiety level median is:', anx_median)
print('The levels of metabolic unit median is:', act_median)
print('The anxiety level median of low leveled metabolic unit people is:', low_median)
print('The anxiety level median of high leveled metabolic unit people is:', high_median)

# calls do_range function
# calculates the dataset range per row
anx_range = do_range(df['ham_a'])
act_range = do_range(df['met_minwk'])
low_range = do_range(low_met)
high_range = do_range(high_met)

# print all the ranges
print('The anxiety level range is:', anx_range)
print('The levels of metabolic unit range is:', act_range)
print('The anxiety level range of low leveled metabolic unit people is:', low_range)
print('The anxiety level range of high leveled metabolic unit people is:', high_range)

# mean and stdev of lists using the statistics library
low_mean = stat.mean(low_met)
high_mean = stat.mean(high_met)
low_stdev = stat.stdev(low_met)
high_stdev = stat.stdev(high_met)

# print of mean and stdev
print('Means:', low_mean, high_mean)
print('Stdevs:', low_stdev, high_stdev)
```

```
count    33.000000
mean      4.393939
std       3.578926
min       0.000000
25%       1.000000
50%       4.000000
75%       7.000000
max      12.000000
Name: ham_a, dtype: float64
count      35.000000
mean     1521.257143
std      1225.681486
min        99.000000
25%       693.000000
50%      1039.800000
75%      2218.500000
max      5112.000000
Name: met_minwk, dtype: float64
The anxiety level mode is: 0    1.0
1    2.0
dtype: float64
The levels of metabolic unit mode is: 0    693.0
dtype: float64
The anxiety level mode of low leveled metabolic unit people is: 1.0
The anxiety level mode of high leveled metabolic unit people is: 2.0
The anxiety level median is: 4.0
The levels of metabolic unit median is: 1039.8
The anxiety level median of low leveled metabolic unit people is: 4.0
The anxiety level median of high leveled metabolic unit people is: 2.0
The anxiety level range is: nan
The levels of metabolic unit range is: 5013.0
The anxiety level range of low leveled metabolic unit people is: 12.0
The anxiety level range of high leveled metabolic unit people is: 9.0
Means: 4.95 3.5384615384615383
Stdevs: 4.032434291565019 2.665063620734804
```

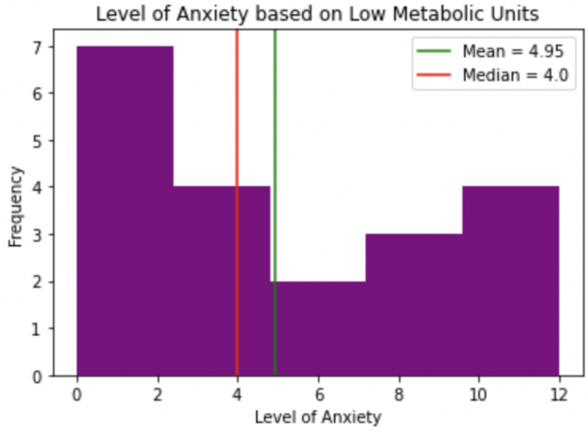
Table 2: General statistics for the used variables

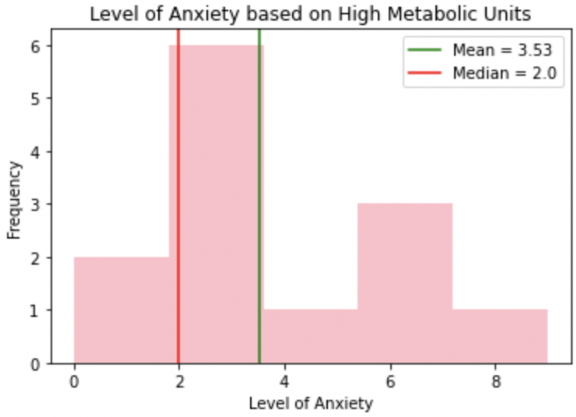
	Anxiety Level	Metabolic Unit (activity level)
Count	33	33
Mean	4.39	1468.18
Median	4	1039.8
Mode	1.0	693
Standard Deviation	3.57	1181.59
Range	12	5013

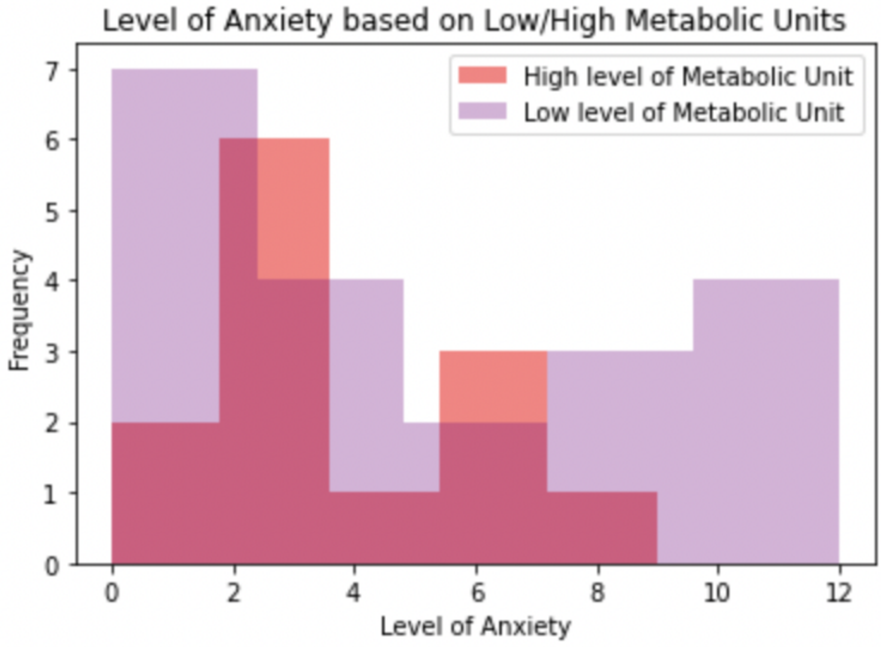
Appendix B: Examine the Subgroups

```
# dataviz

# plotting a histogram
# histogram function: takes into consideration the data to be plotted
# number of bins and color
plt.hist(low_met, bins=5, color='purple')

# create the lines of mean and median on the visualization
# takes into consideration the value where the line should be drawn
plt.axvline(low_mean, color='g', label='Mean = 4.95')
plt.axvline(low_median, color='r', label='Median = 4.0')

# set the position of the legend
plt.legend(loc='upper right')
# x-axis label
plt.xlabel('Level of Anxiety')
# frequency label
plt.ylabel('Frequency')
# plot title
plt.title('Level of Anxiety based on Low Metabolic Units')

# function to show the plot
plt.show()


# plotting a histogram
# histogram function: takes into consideration the data to be plotted
# number of bins and color
plt.hist(high_met, bins=5, color='pink')

# create the lines of mean and median on the visualization
# takes into consideration the value where the line should be drawn
plt.axvline(high_mean, color='g', label = 'Mean = 3.53')
plt.axvline(high_median, color='r', label = 'Median = 2.0')

# set the position of the legend
plt.legend(loc='upper right')
# x-axis label
plt.xlabel('Level of Anxiety')
# frequency label
plt.ylabel('Frequency')
# plot title
plt.title('Level of Anxiety based on High Metabolic Units')

# function to show the plot
plt.show()
```

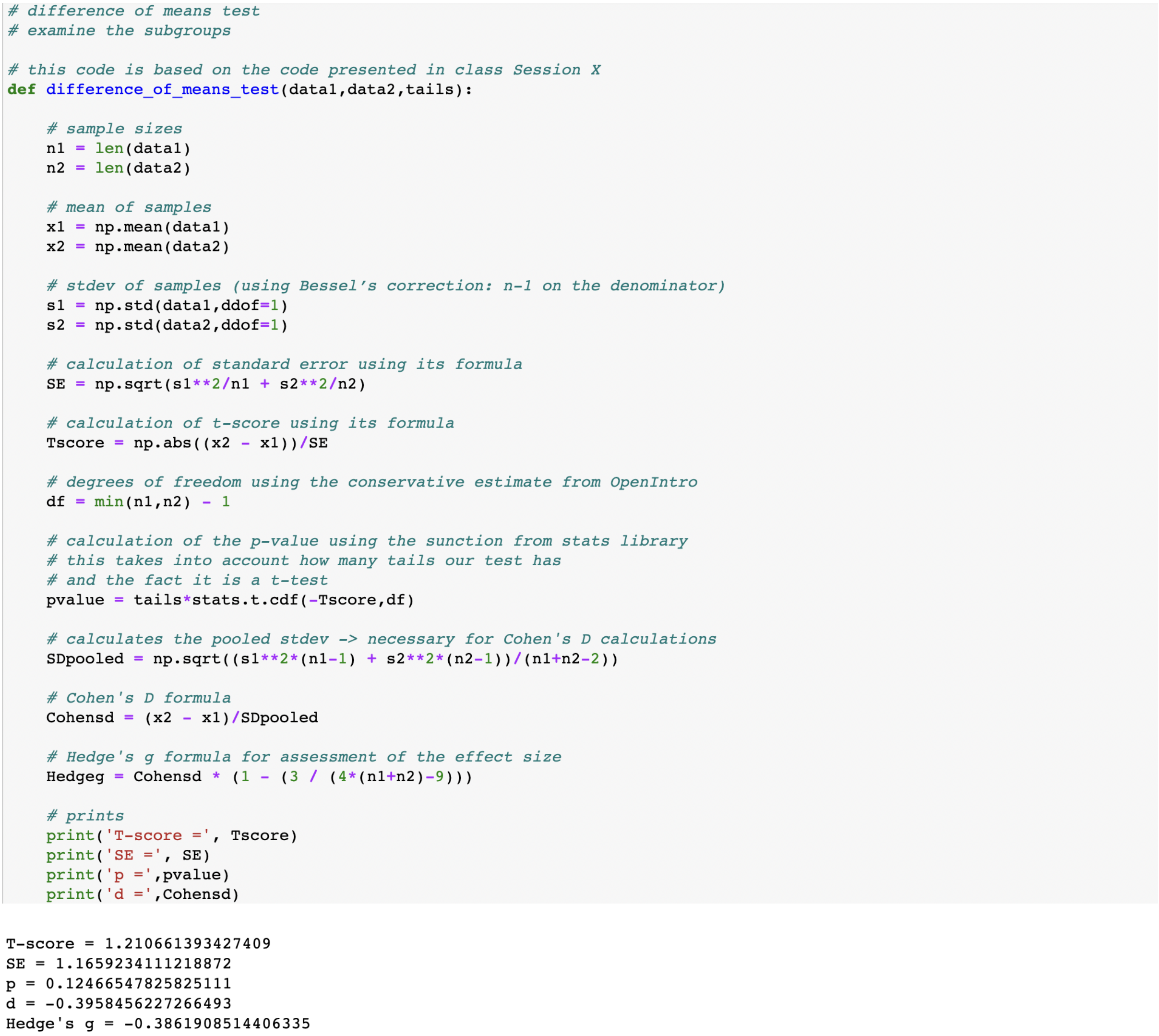
```
# plotting two "merged" histograms
# histogram function: takes into consideration the data to be plotted
# number of bins and color
plt.hist(high_met, bins=5, alpha=0.5, label='High level of Metabolic Unit', color='red')
plt.hist(low_met, bins= 5, alpha=0.3, label='Low level of Metabolic Unit', color='purple')

# set the position of the legend
plt.legend(loc='upper right')
# x-axis label
plt.xlabel('Level of Anxiety')
# frequency label
plt.ylabel('Frequency')
# plot title
plt.title('Level of Anxiety based on Low/High Metabolic Units')

# function to show the plot
plt.show()
```







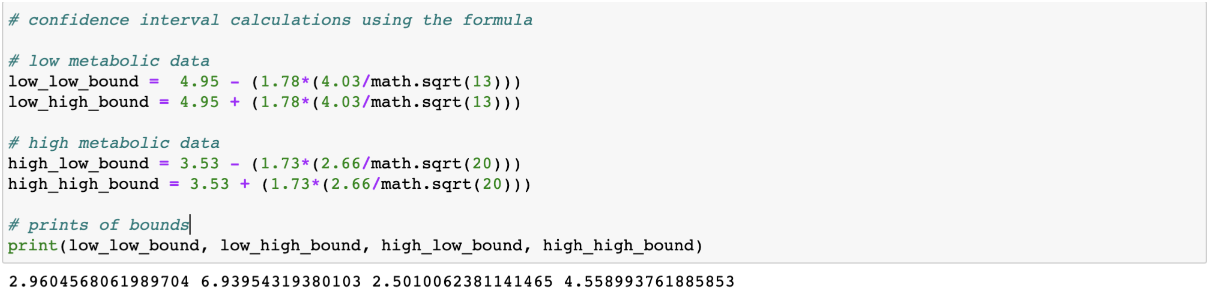
Appendix C: Difference of Means Test



```
# difference of means test
# examine the subgroups

# this code is based on the code presented in class Session X
def difference_of_means_test(data1,data2,tails):

    # sample sizes
    n1 = len(data1)
    n2 = len(data2)

    # mean of samples
    x1 = np.mean(data1)
    x2 = np.mean(data2)

    # stdev of samples (using Bessel’s correction: n-1 on the denominator)
    s1 = np.std(data1,ddof=1)
    s2 = np.std(data2,ddof=1)

    # calculation of standard error using its formula
    SE = np.sqrt(s1**2/n1 + s2**2/n2)

    # calculation of t-score using its formula
    Tscore = np.abs((x2 - x1))/SE

    # degrees of freedom using the conservative estimate from OpenIntro
    df = min(n1,n2) - 1

    # calculation of the p-value using the sunction from stats library
    # this takes into account how many tails our test has
    # and the fact it is a t-test
    pvalue = tails*stats.t.cdf(-Tscore,df)

    # calculates the pooled stdev -> necessary for Cohen's D calculations
    SDpooled = np.sqrt((s1**2*(n1-1) + s2**2*(n2-1))/(n1+n2-2))

    # Cohen's D formula
    Cohensd = (x2 - x1)/SDpooled

    # Hedge's g formula for assessment of the effect size
    Hedgeg = Cohensd * (1 - (3 / (4*(n1+n2)-9)))

    # prints
    print('T-score =', Tscore)
    print('SE =', SE)
    print('p =',pvalue)
    print('d =',Cohensd)
```

```
T-score = 1.210661393427409
SE = 1.1659234111218872
p = 0.12466547825825111
d = -0.3958456227266493
Hedge's g = -0.3861908514406335
```

Appendix D: Confidence Interval



```
# confidence interval calculations using the formula

# low metabolic data
low_low_bound =  4.95 - (1.78*(4.03/math.sqrt(13)))
low_high_bound = 4.95 + (1.78*(4.03/math.sqrt(13)))

# high metabolic data
high_low_bound = 3.53 - (1.73*(2.66/math.sqrt(20)))
high_high_bound = 3.53 + (1.73*(2.66/math.sqrt(20)))

# prints of bounds
print(low_low_bound, low_high_bound, high_low_bound, high_high_bound)
```

```
2.9604568061989704 6.93954319380103 2.5010062381141465 4.558993761885853
```