DE LA SALLE UNIVERSITY - MANILA

Hippopotamus (Hypothesis Testing Calculator)

A Term Project

Presented to Mr. Ramon Stephen L. Ruiz

In Partial Fulfillment of the

Requirements for the Course Programming Logic and Design (PROLOGI)

by

Alabado, Tayshaun Pierce - Signature/Initials

Taraya, Marcus Alden - Signature/Initials

Sy, Win Clarence - Signature/Initials

EQ3

Monday and Thursday, 12:45 PM - 1:45 PM

April 2023

Table of Contents

I. Introduction

A crucial part of statistical analysis is hypothesis testing, which is used to determine the significance of study results and draw inferences about population characteristics. Hypothesis testing involves a number of steps, including formulating a hypothesis, choosing an appropriate statistical test, gathering and processing data, and interpreting the findings. However, carrying out these procedures by hand can be difficult and time-consuming, particularly for people without a strong statistical background.

In order to address this issue, hypothesis testing calculators have gained favor recently. These calculators give customers a quick and efficient way to test hypotheses, allowing for accurate statistical analysis. The usefulness of calculators for hypothesis testing has increased due to the expansion of big data and the rise in demand for data-driven decision-making.

In this research work, we describe the creation and assessment of a calculator for hypothesis testing with a user-friendly interface and precise statistical findings. We offer a variety of statistical tests in our calculator, from easy tests to choices about whether to accept or reject the null hypothesis. We also give users the option to tailor their analyses to meet their unique requirements, including defining the degree of significance, picking the specific test, and entering data.

To evaluate the effectiveness of our calculator, we conducted a series of tests comparing the results obtained from our calculator to those obtained from manual hypothesis testing. Our results indicate that our calculator provides highly accurate and reliable statistical analyses.

Overall, the development of a robust and reliable hypothesis testing calculator has the potential to greatly improve the accuracy and efficiency of statistical analysis, making it an essential tool for researchers and practitioners alike. Through our research, we hope to contribute to the advancement of statistical analysis and data-driven decision-making by providing a powerful and accessible tool for hypothesis testing.

A. Background of the Study

Hypothesis testing is a fundamental statistical tool that is widely used to determine whether a particular hypothesis or claim about a population is supported by the available evidence (Babbie, 2017; Field, 2013). It is a critical component of the scientific method and is used in a variety of fields, including social sciences, business, medicine, and engineering (Rosnow & Rosenthal, 1991; Welch, 1937).

By using hypothesis testing, researchers can draw conclusions about a population from a small sample of data. (Kline, 2015). It entails contrasting an alternative hypothesis, which is the null hypothesis's opposite, with a null hypothesis, which is the presumption that there is no difference or effect in the population. (Patten, 2018). The steps in the hypothesis testing procedure are data collection, computation of a test statistic, and evaluation of the statistical significance of the observed result. (Fisher, 1925).

With the availability of vast volumes of data in modern research, the use of hypothesis testing has grown in importance. (Hinton et al., 2019). However, the computations necessary for hypothesis testing can be challenging and demand a thorough knowledge of statistical theory and procedures. (Maxwell & Delaney, 2004). Furthermore, it can be difficult for some people to grasp how to make conclusions based on probability while conducting hypothesis testing. (Moore & McCabe, 2003).

To address these challenges, the development of hypothesis testing calculators has become an important area of research (Hogg & Tanis, 2018). These calculators allow researchers and practitioners to perform hypothesis tests quickly and accurately, without the need for extensive statistical knowledge (Zhang, 2015). They can be used to test a wide range of hypotheses, from simple comparisons of means to more complex analyses of variance and regression models (Freedman et al., 2007).

The development of hypothesis testing calculators has significant implications for both research and practice. They provide a powerful tool for researchers to analyze their data and draw conclusions about their populations (Cohen, 1994). In addition, they can be used in applied settings, such as business and healthcare, to make decisions based on statistical evidence (Altman & Bland, 1994).

In this project, we aim to develop a hypothesis testing calculator that is accessible and user-friendly, while still providing accurate and reliable results. Our calculator will be designed to handle a wide range of hypothesis tests. By providing a powerful and accessible tool for hypothesis testing, we hope to contribute to the advancement of statistical analysis and promote evidence-based decision-making in a variety of fields.

B. Problem Statement

Hypothesis testing is a critical component of statistical analysis used to determine the significance of research findings and draw conclusions about population parameters. However, performing hypothesis testing manually can be time-consuming, complex, and prone to errors, particularly for individuals without a strong statistical background. While existing hypothesis testing calculators provide a solution, they are often limited in functionality, difficult to use, or provide inaccurate results.

The ideal scenario is that all researchers and practitioners have access to a reliable, user-friendly, and accurate hypothesis testing calculator that can perform a wide range of statistical tests, customize analysis based on specific needs, and provide clear interpretations of results. However, the current reality is that many available hypothesis testing calculators lack the necessary features and accuracy to provide accurate and reliable statistical analyses.

This gap in knowledge and available tools have significant consequences for the accuracy and efficiency of statistical analysis, potentially leading to incorrect conclusions and flawed decision-making. This problem highlights the need for a robust and reliable hypothesis testing calculator that can provide users with accurate and efficient statistical analysis, regardless of their statistical background.

Therefore, this research project aims to develop and evaluate a hypothesis testing calculator that addresses these issues, providing users with a powerful tool for conducting accurate and reliable statistical analysis.

C. Objectives

C.1. General Objective

The general objective of this project is to develop a hypothesis testing calculator that provides users with a user-friendly interface and accurate statistical results, allowing for quick and efficient statistical analysis.

C.2. Specific Objectives

1. To identify and evaluate the different types of hypothesis tests that will be included in the calculator.
2. To design and develop a user-friendly interface for the hypothesis testing calculator, including the input and output of data and the selection of statistical tests.
3. To implement and test the functionality of the hypothesis testing calculator, ensuring the accuracy and reliability of statistical results.
4. To evaluate the effectiveness of the hypothesis testing calculator through comparisons with manual hypothesis testing, ensuring the calculator's level of precision is comparable to that of experienced statisticians.

D. Significance of the Project

The development of a robust and reliable hypothesis testing calculator has significant implications for both research and practice in various fields, including science, engineering, business, and healthcare. The significance of this project is to improve accuracy and efficiency by providing users with a powerful and accurate tool for hypothesis testing, the calculator can significantly improve the accuracy and efficiency of statistical analysis. This can lead to more accurate conclusions, informed decision-making, and ultimately, better outcomes. Making a user-friendly interface that makes statistical analysis accessible to individuals with limited statistical knowledge, enabling them to perform hypothesis testing with ease. The calculator significantly reduces the time required for statistical analysis, allowing researchers and practitioners to allocate more time to other critical tasks. By providing a reliable and accurate tool for hypothesis testing, the calculator can enhance the quality of research findings, increasing the credibility and validity of research outcomes. The calculator can increase productivity by reducing the time and effort required for statistical analysis, allowing individuals to focus on other critical tasks.

II. Review of Related Literature

A. Hypothesis Testing

B. Reliability

C. Precision

III. Methodology

This project will utilize Python language with **Conda Kernel** since it has powerful libraries that we could use that would help us in this project. **Numpy** and **Math** would be used for calculations. **Scipy** would be used to get the critical value for the problem. **Colorma** would be used for coloring the program so that it would be easier for the user to see the important values. We will be utilizing several functions such as if-else statements. For the formula, we would be getting it from the course **FNDSTAT** we have taken from Term 1 AY2022-2023. We will be using modular functions to simplify the code and avoiding code duplication would make troubleshooting easier. We would be separating each case with its own functions so that it would be more user-friendly.

A. Conceptual Framework – IPO Chart (Input-Process-Output-Chart)

Input	Process	Output
Case A x, u_0, o, n, alpha **Case B** x, u_0, s, n, alpha **Single Population Test Proportions** x, n, p_0, alpha **Case C** x_1, s_1, n_1, x_2, s_2, n_2, d_0, alpha **Case D** x_1, s_1, n_1, x_2, s_2, n_2, d_0, alpha **Case E** x_1, s_1, n_1, x_2, s_2, n_2, d_0, alpha	Import scipy.stats as st **Case A** $z = \frac{\bar{x} - \mu_0}{\left(\sigma/\sqrt{n}\right)}$ **Case B** $t = \frac{\bar{x} - \mu_0}{\left(s/\sqrt{n}\right)}$ $df = n - 1$ **Single Population Test Proportion** $\hat{p} = \frac{x}{n}$ $z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0 q_0}{n}}}$ **Case C** $z = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$ **Case D** $s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$	Test Statistic P-Value **(if any)** Critical Value Degree of Freedom **(if any)** Sample Proportions **(if any)** Pooled Sample Proportion **(if any)** Pooled Standard Deviation **(if any)** Difference **(if any)** Difference Standard Deviation **(if any)** Whether to Reject or Not Reject the Null Hypothesis

Case F d, d_0, s, n, alpha **Two Populations Test Proportions** x_1, n_1, x_2, n_2, alpha **One Population Variances** n, s, o_0, alpha **Two Population Variances** n_1, s_1, n_2, s_2, alpha	$df = n_1 + n_2 - 2$ $t = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$ **Case E** $t = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$ $df = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}$ **Case F** $t = \frac{\bar{d} - d_0}{\left({s_d}/{\sqrt{n}}\right)}$ $df = n - 1$ **Test Proportions for Two Population** $\hat{p}_1 = \frac{x_1}{n_1}$ and $\hat{p}_2 = \frac{x_2}{n_2}$ $\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}$ $z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}\hat{q}\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$ **One Population Variances** $\chi^2 = \frac{(n-1)s^2}{\sigma_0^2}$ $v = n - 1.$ **Two Population Variances** $f = \frac{s_1^2}{s_2^2}$	

	$v_1 = n_1 - 1$ and $v_2 = n_2 - 1$. **Z-Statistic P-Value and Critical Point - Left** p_value = st.norm.cdf(z_statistic) critical_value = st.norm.ppf(1 - alpha) **Z-Statistic P-Value and Critical Point - Right** critical_value = st.norm.ppf(1 - alpha) p_value = 1 - st.norm.cdf(z_statistic) **Z-Statistic P-Value and Critical Point - Both** alpha = alpha/2 critical_value1 = st.norm.ppf(alpha) critical_value2 = st.norm.ppf(1 - alpha) p_value = 2 * (1 - st.norm.cdf(abs(z_statistic))) **T-Statistic P-Value and Critical Point - Left** critical_value = st.t.ppf(alpha, d_f) p_value = st.t.cdf(t_stat, d_f) **T-Statistic P-Value and Critical Point - Right** critical_value = st.t.ppf(1 - alpha, d_f) p_value = st.t.cdf(1- t_stat, d_f)	

	T-Statistic P-Value and Critical Point - Both alpha = alpha/2 critical_value1 = st.t.ppf(alpha, d_f) critical_value2 = st.t.ppf(1 - alpha, d_f) p_value = (1 - st.t.cdf(abs(t_stat), d_f)) * 2 **Single Population Variance P-Value** p_value = st.chi2.sf(chi_squared, n - 1) **Single Population Variance - Left** critical_value = st.chi2.ppf(alpha, n - 1) **Single Population Variance - Right** critical_value = st.chi2.ppf(1 - alpha, n - 1) **Single Population Variance - Both** alpha = alpha/2 critical_value1 = st.chi2.ppf(alpha, n - 1) critical_value2 = st.chi2.ppf(1 - alpha, n - 1) **Two Population Variance P-Value** p_value = st.f.sf(f_stats, v_1, v_2) **Two Population Variance -**	

	Left critical_value = st.f.ppf(alpha, v_1, v_2) **Two Population Variance - Right** critical_value = st.f.ppf(1 - alpha, v_1, v_2) **Two Population Variance - Both** alpha = alpha/2 critical_value1 = st.f.ppf(alpha, v_1, v_2) critical_value2 = st.f.ppf(1 - alpha, v_1, v_2)	

Table 1.1

We can see that in the IPO figure found in Table 1.1, We can see that for the input, the user would input the data needed for the specific hypothesis test that he/she would like to do. In the processes, we can see that there is a formula for each test to get the test statistics (z or t or X or f), degree of freedom (df and v), proportions (p), pooled standard deviation (s_p^2), p-value (p_value), and critical values (critical_value, critical_value1, critical_value2). For the output, It would print out the following: test statistic, p-value (if any), critical value, degree of freedom (if any), sample proportions (if any), pooled sample proportion (if any), pooled standard deviation (if any), the difference (if any), difference standard deviation (if any), and whether to reject or not reject the null hypothesis

B. Hierarchy Chart

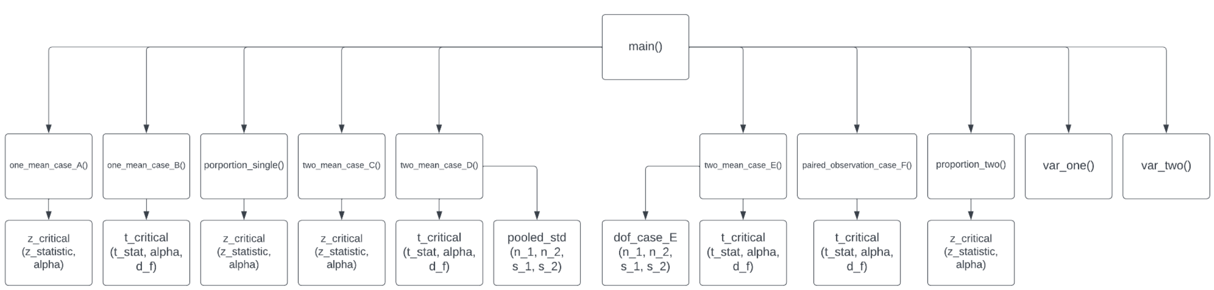


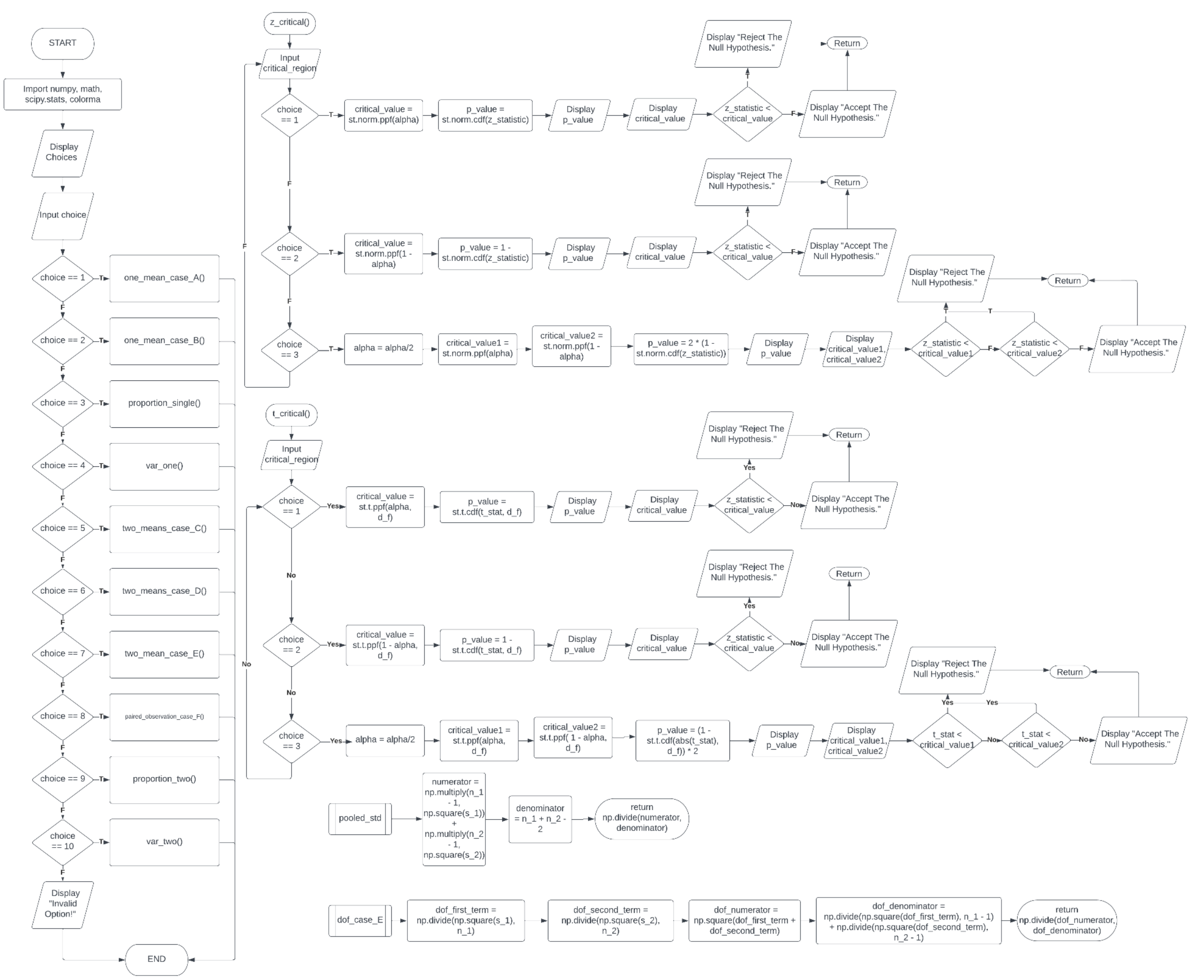
Figure 1.1

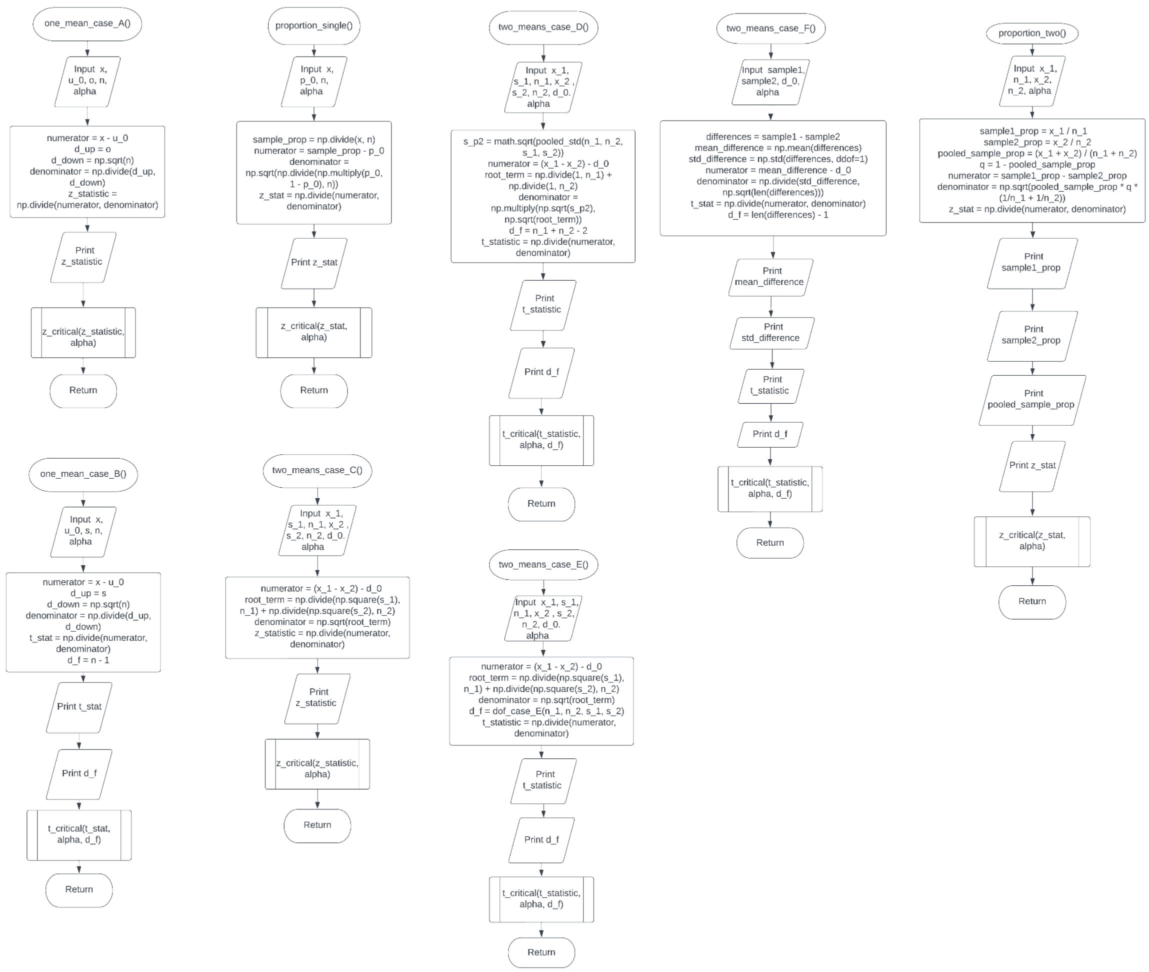
In the hierarchy chart found in Figure 1.1, we can see how each module are leaked to one another so that the program would be able to perform each selected tasks. It is important to have a hierarchy chart as this enables us to perform faster troubleshooting since we would know which are connected to which meaning that we will be able to find the broken function easily. We can also see that are several same functions being called and there is another example of the advantages of this which are that we would have fewer repeated codes meaning that modifying and adding new lines would be more efficient and troubleshooting would also be easier. An example would be the main() is connected to one_mean_case_A() and it is connected to z_critical (Z_statistic, alpha) to get the critical points and determine whether the program should reject or not reject the null hypothesis.

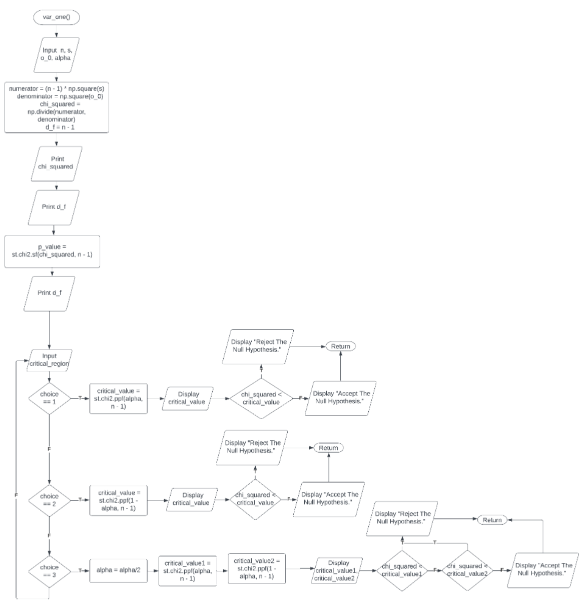
C. Flowchart

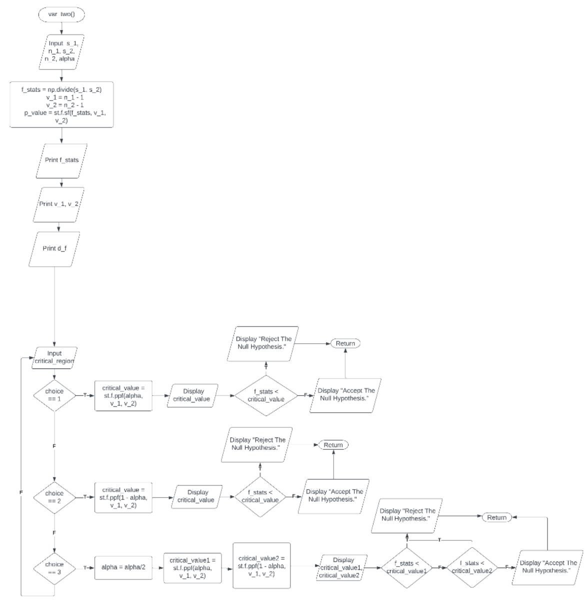
The Google Drive folder link below contains the flowchart for this project since when imported into the word file, it would become very blurry and unreadable.

https://drive.google.com/drive/folders/10UjIdV-SWk1_epcxUHT7WwWsRHHYuHc1?usp=share_link









D. Pseudocode

```
def lines():
        Display"----------------------------------------------"

def z_critical(z_statistic, alpha)
        Input critical region
        If critical region = 1 Then:
                critical value = st.norm.ppf alpha
                p value =  st.norm.ppfz_statistic


                Display  p_value
                Display critical value


                If z_statistic < critical_value then:
                        Display "Reject the null hypothesis."
                else:
                        Display"Accept the null hypothesis."
        Else If critical region = 2  Then:
```

```
        critical value = 1- st.norm.ppfalpha
        p value = 1- st.norm.ppf z_statistic

        Display  p_value
        Display critical value


        If z_statistic > critical_value then:
            Display "Reject the null hypothesis."
        else:
            Display"Accept the null hypothesis."


    Else If critical region = 2  Then:
        alpha = alpha/2
        critical value1 =  st.norm.ppf alpha
        critical value2 = st.norm.ppf 1–alpha
        p value = 2*(1- st.norm.ppf z_statistic)


        Display  p_value
        Display critical value1
        Display critical value2


        If z_statistic < critical_value1 or z_statistic > critcial_value2 then:
            Display "Reject the null hypothesis."
        else:
            Display"Accept the null hypothesis."

def t_critical (T_stat, alpha, df)
    Input critical_region

    If critical_region = 1 Then:
        critical_value = st.t.ppt (alpha, df)
        p_value = st.t.cdf (t_stat df)

        Display p_value
        Display critical_value

        If T_stat < critical_value then:
            Display "Reject the null hypothesis"
        Else:
            Display "Accept the null hypothesis"
```

```
    Else If critical_region = 2  Then:
        critical_value = st.t.ppt (1 - alpha, df)
        p_value = st.t.cdf (1 - t_stat df)

        Display p_value
        Display critical_value

        If T_stat > critical_value then:
            Display "Reject the null hypothesis"
        Else:
            Display "Accept the null hypothesis"


    Else If critical_region = 3  Then:
        alpha = alpha/2
        critical_value1 = st.t.ppt (alpha, df)
        critical_value2 = st.t.ppt (1 - alpha, df)
        p_value = st.t.cdf (1 - t_stat df) *2

        Display p_value
        Display critical_value1
        Display critical_value2

        If T_stat < critical_value1 or T_stat > critical_value2 then:
            Display "Reject the null hypothesis"
        Else:
            Display "Accept the null hypothesis"
def pooled_stn(n_1, n_2, s_1, s_2):
    numerator = (n_1 - 1) * s_1^2 + (n_2 - 1) * s_2^2
    denominator = n_1 + n_2 - 2
    return numerator / denominator


def dof_case_E(n_1, n_2, s_1, s_2):
    dof_first term = s_1^2 / n_1
    dof_second term = s_2^2 / n_2
    dof_numerator = (dof_first term + dof_second term)^2
    dof_denominator = (dof_first term)^2 /(n_1-1)   +   (dof_second term)^2 / (n_2)

    return dof_numerator / dof_denominator

def one_mean_case_A():
    Input x
```

```
    Input u_0
    Input o
    Input n
    Input alpha

    numerator = x - u_0
    d_up = o
    d_down = sqrt(n)
    denominator = d_up / d_down
    z_statistic = numerator / denominator
    Display z_statistic
    Call z_critical(z_statistic, alpha)

def one_mean_case_B():
    Input x
    Input u_0
    Input o
    Input n
    Input alpha

    numerator = x - u_0
    d_up = o
    d_down = sqrt(n)
    denominator = d_up / d_down
    t_stat = numerator / denominator
    d_f = n - 1

    Display t_stat
    Display d_f
    Call t_critical(t_stat, alpha, d_f)

def proportion_single():
    Input x
    Input n
    Input p_0
    Input alpha

    sample_prop = x / n
    numerator = sample_prop - p_0
    denominator = sqrt( (p_0* 1-p_0) / n)
    z_stat = numerator / denominator

    Display z_stat

    Call z_critical(z_statistic, alpha)
```

```
def two_means_case_C():
    Input x_1
    Input s_1
    Input n_1
    Input x_2
    Input s_2
    Input n_2
    Input d_0
    Input alpha

    numerator = (x_1 - x_2) - d_0
    root_term = (1 / n_1) + (1 / n_2)
    denominator = sqrt(root_term)
    z_statistic = (numerator / denominator)

    Display z_statistic

    Call z_critical(z_statistic, alpha)

def two_means_case_D():
    Input x_1
    Input s_1
    Input n_1
    Input x_2
    Input s_2
    Input n_2
    Input d_0
    Input alpha

    s_p2 = sqrt(pooled_std(n_1, n_2, s_1, s_2))
    numerator = (x_1 - x_2) - d_0
    root_term = (1 / n_1) + (1 / n_2)
    denominator = sqrt(s_p2) * sqrt(root_term)
    d_f = n_1 + n_2 - 2
    t_statistic = (numerator / denominator)

    Display t_statistic
    Display s_p2
    Display d_f

    Call t_critical(t_statistic, alpha, d_f)

def two_means_case_E():
    Input x_1
    Input s_1
    Input n_1
```

```
    Input x_2
    Input s_2
    Input n_2
    Input d_0
    Input alpha

    numerator = (x_1 - x_2) - d_0
    root_term = ((s_1)^2 / n_1) + ((s_2)^2 / n_2)
    denominator = sqrt(root_term)

    d_f = dof_case_E(n_1, n_2, s_1, s_2)

    t_statistic = (numerator / denominator)

    Display t_statsitic
    Display d_f

    Call t_critical(t_statistic, d_f)


def one_mean_case_F():
    Input sample 1
    Input sample 2
    Input d_0
    Input alpha

    differences = sample1 - sample2
    mean_difference = np.mean(differences)
    std_difference = np.std(differences, ddof=1)


    numerator = mean_difference - d_0
    denominator = std_difference / sqrt(differences)
    t_stat = numerator / denominator
    d_f = differences - 1

    Display mean_differences
    Display std_differences
    Display t_stat
    Display d_f

    Call t_critical(t_stat, alpha, d_f)


def proportion_two():
    Input x_1
    Input n_1
```

```
        Input x_2
        Input n_2
        Input alpha

        sample1_prop = (x_1 / n_1)
        sample2_prop = (x_2 / n_2)
        pooled_sample_prop = (x_1 + x_2) / (n_1 + n_2)
        q = 1 - pooled_sample_prop

        numerator = sample1_prop - sample2_prop
        denominator = sqrt(pooled_sample_prop * q * (1/n_1 + 1/n_2)
        z_stat = numerator / denominator

        Display sample1_prop
        Display sample2_prop
        Display pooled_sample_prop
        Display z_stat

        Call z_critical(z_stat, alpha)


def var_one():
        Input n
        Input s
        Input o_0
        Input alpha

        numerator = (n - 1) * s^2
        denominator = o_0^2
        chi_squared = numerator / denominator
        d_f = n - 1

        Display chi_squared
        Display d_f

        p_value = st.chi2.sf(chi_squared, n - 1)
        Display p_value

        Input critical_region

        If critical_region == 1 then:
                critical_value = st.chi2.ppf(alpha, n - 1)
                Display critical_value
                if chi_squared < critical_value then:
                        Display “Reject the Null Hypothesis”
                else:
                        Display “Accept the Null Hypothesis”
```

```
    Elif critical_region == 2 then:

        critical_value = st.chi2.ppf(1 - alpha, n - 1)
        Display critical_value
        if chi_squared > critical_value then:
            Display "Reject the Null Hypothesis"
        else:
            Display "Accept the Null Hypothesis"

    Elif critical_region == 3 then:
        critical_value1 = st.chi2.ppf(alpha, n - 1)
        critical_value2 = st.chi2.ppf(1 - alpha, n - 1)
        Display critical_value1
        Display critical_value2
        if chi_squared < critical_value or chi_squared > critical_value2 then:
            Display "Reject the Null Hypothesis"
        else:
            Display "Accept the Null Hypothesis"

def var_two():
    Input n_1
    Input s_1
    Input n_2
    Input s_2
    Input alpha

    f_stats = s_1 / s_2
    v_1 = n_1 - 1
    v_2 = n_2 - 1
    p_value = st.f.sf(f_stats, v_1, v_2)

    Display f_stats
    Display v_1
    Display v_2
    Display p_value

    Input critical_region

    If critical_region == 1 then:
        critical_value = st.chi2.ppf(alpha, v_1, v_2)
        Display critical_value
        if chi_squared < critical_value then:
            Display "Reject the Null Hypothesis"
        else:
            Display "Accept the Null Hypothesis"

    Elif critical_region == 2 then:
```

```
        critical_value = st.chi2.ppf(1 - alpha, v_1, v_2)
        Display critical_value
        if chi_squared > critical_value then:
            Display “Reject the Null Hypothesis”
        else:
            Display “Accept the Null Hypothesis”

    Elif critical_region == 3 then:
        alpha = alpha/2
        critical_value1 = st.chi2.ppf(alpha, v_1, v_2)
        critical_value2 = st.chi2.ppf(1 - alpha, v_1, v_2)
        Display critical_value1
        Display critical_value2
        if chi_squared < critical_value or chi_squared > critical_value2 then:
            Display “Reject the Null Hypothesis”
        else:
            Display “Accept the Null Hypothesis”

Display “Welcome to Hippopotamus - Hypothesis Testing Calculator”
Display “There following are the choices for Single Population:”
Display “1. Case A: Single Population Where Sigma is Known or Sample is Large”
Display “2. Case B: Sigma is Unknown or Sample is Small”
Display “3. Test for Proportions: Single Population”
Display “4. Variance For Single Population
lines()
Display “The following are the choices for Two Population:”
Display “5. Case C: Two Mean Big Sample Size or Population Stnadard Deviation is Given”
Display “6. Case D: Two Mean Small Sample Size and Population Sample Standard Deviation
assumed Equal”
Display”7. Case E: Two Mean Small Sample Size and Population Sample Standard Deviation
not Equal”
Display “8. Case F: Paired Observation”
Display “9. Test for Proportions: Two Population”
Display “10. Variance For Two Populations”
lines()
Input choice

If choice == 1 then:
    one_mean_case_A()
Elif choice == 2 then:
    one_mean_case_B()
Elif choice == 3 then:
    proportion_single()
Elif choice == 4 then:
```

```
    var_one()
Elif choice == 5 then:
    two_means_case_C()
Elif choice == 6 then:
    two_means_case_D()
Elif choice == 7 then:
    two_means_case_E()
Elif choice == 8 then:
    paired_observation_case_F()
Elif choice == 9 then:
    proportion_two()
Elif choice == 10 then:
    var_two()
Else:
    Display "Invalid Option!"
```

IV. Results

Test #1 - Example 8.3.1 from Hypothesis Testing of Single Mean LibreText Statistics

```
Welcome to Hippotamus - Hypothesis Testing Calculator
----------------------------------------------
There following are the choices for Single Population:
----------------------------------------------
1. Case A: Single Population Where Sigma is Known or Sample is Large
2. Case B: Sigma is Unknown or Sample is Small
3. Test for Proportions: Single Population
4. Variance For Single Population
----------------------------------------------
There following are the choices for Two Population:
----------------------------------------------
5. Case C: Two Mean Big Sample Size or Population Stnadard Deviation is Given
6. Case D: Two Mean Small Sample Size and Population Sample Standard Deviation assumed Equal
7. Case E: Two Mean Small Sample Size and Population Sample Standard Deviation not Equal
8. Case F: Paired Observation
9. Test for Proportions: Two Population
10. Variance For Two Populations
----------------------------------------------
Enter your choice: 1
----------------------------------------------
You have Chosen Case A: Single Population where sigma is known or sample is large.
Enter the Sample Mean: 169
Enter the Population Mean: 179
Enter the Population Standard Deviation: 10.39
Enter the Sample Size: 5
Enter the Significance Level: 0.05
----------------------------------------------
The Z Score is -2.1521.
----------------------------------------------
Enter the Critical Region (1) Left, (2) Right, (3) Both: 1
----------------------------------------------
The p-value is 0.0156933725.
The Critical Value is -1.6449.
Reject The Null Hypothesis.
```

Test #2 - Example from FNDSTAT Homework 2 by Ms. Cherry Frondozo

```
Welcome to Hippotamus - Hypothesis Testing Calculator
----------------------------------------------
There following are the choices for Single Population:
----------------------------------------------
1. Case A: Single Population Where Sigma is Known or Sample is Large
2. Case B: Sigma is Unknown or Sample is Small
3. Test for Proportions: Single Population
4. Variance For Single Population
----------------------------------------------
There following are the choices for Two Population:
----------------------------------------------
5. Case C: Two Mean Big Sample Size or Population Stnadard Deviation is Given
6. Case D: Two Mean Small Sample Size and Population Sample Standard Deviation assumed Equal
7. Case E: Two Mean Small Sample Size and Population Sample Standard Deviation not Equal
8. Case F: Paired Observation
9. Test for Proportions: Two Population
10. Variance For Two Populations
----------------------------------------------
Enter your choice: 9
----------------------------------------------
You have Chosen Test for Proportions: Two Population.
Enter the 1st Number of Successes: 280
Enter the 1st Sample Size: 500
Enter the 2nd Number of Successes: 230
Enter the 2nd Sample Size: 500
Enter the Significance Level: 0.01
The 1st Sample Proportion is 0.56.
The 2nd Sample Proportion is 0.46.
The Pooled Sample Proportion is 0.51.
The Z-Statistic is 3.1629.
----------------------------------------------
Enter the Critical Region (1) Left, (2) Right, (3) Both: 2
----------------------------------------------
The p-value is 0.0007810022.
The Critical Value is 2.3263.
Reject The Null Hypothesis.
```

Test 3 - Example 9.1.2 from Comparison of Two Population Means- Large, Independent Samples LibreText Statistics

```
Welcome to Hippotamus - Hypothesis Testing Calculator
-----------------------------------------------
There following are the choices for Single Population:
-----------------------------------------------
1. Case A: Single Population Where Sigma is Known or Sample is Large
2. Case B: Sigma is Unknown or Sample is Small
3. Test for Proportions: Single Population
4. Variance For Single Population
-----------------------------------------------
There following are the choices for Two Population:
-----------------------------------------------
5. Case C: Two Mean Big Sample Size or Population Stnadard Deviation is Given
6. Case D: Two Mean Small Sample Size and Population Sample Standard Deviation assumed Equal
7. Case E: Two Mean Small Sample Size and Population Sample Standard Deviation not Equal
8. Case F: Paired Observation
9. Test for Proportions: Two Population
10. Variance For Two Populations
-----------------------------------------------
-----------------------------------------------
You have Chosen Case C: Two Mean Big Sample Size or Population Stnadard Deviation is Given.
Enter the 1st Sample Mean: 3.51
Enter the 1st Sample Standard Deviation: 0.51
Enter the 1st Sample Size: 174
Enter the 2nd Sample Mean: 3.24
Enter the 2nd Sample Standard Deviation: 0.52
Enter the 2nd Sample Size: 355
Enter the Difference Between the Population Means: 0
Enter the Significance Level: 0.01
The Z-Statistic is 5.6839.
-----------------------------------------------
Enter the Critical Region (1) Left, (2) Right, (3) Both: 2
-----------------------------------------------
The p-value is 6.6e-09.
The Critical Value is 2.3263.
Reject The Null Hypothesis.
```

V. Discussion of Results

VI. Analysis, Conclusion and Future Directives

Future directives include your recommendations on the improvements that can be made on the same topic. Future directives is optional.

References

Altman, D. G., & Bland, J. M. (1994). Diagnostic tests 2: Predictive values. *BMJ (Clinical research ed.), 309(6947)*, 102. https://doi.org/10.1136/bmj.309.6947.102

Babbie, E. R. (2017). Basics of social research. *Cengage Learning*.

Cohen, J. (1994). The earth is round (p < .05). *American Psychologist, 49(12)*, 997–1003. https://doi.org/10.1037/0003-066X.49.12.997

Freedman, D., Pisani, R., & Purves, R. (2007). Statistics. *WW Norton & Company*.

Field, A. (2013). Discovering statistics using IBM SPSS statistics. *Sage*.

Fisher, R. A. (1925). Statistical methods for research workers. *Oliver and Boyd*.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2019). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29(6), 82-97*. https://doi.org/10.1109/MSP.2012.2205597

Hogg, R. V., & Tanis, E. A. (2018). Probability and statistical inference. *Pearson*.

Kaur, H., & Singh, H. (2020). Development of hypothesis testing calculator for medical researchers. *Journal of Medical Systems, 44(12)*, 1-11.

Kline, R. B. (2015). Principles and practice of structural equation modeling. *Guilford publications*.

Mathematics and Statistics Department. (2022). FNDSTAT Hypothesis Testing Formula Sheet For Variances. De La Salle University. https://drive.google.com/drive/folders/1Y1mhqy8yeHZB9YC7WBPkQCNMH8Icvorh?usp=sharing

Mathematics and Statistics Department. (2022). FNDSTAT Hypothesis Testing Formula Sheet for Two Populations. De La Salle University. https://drive.google.com/drive/folders/1Y1mhqy8yeHZB9YC7WBPkQCNMH8Icvorh?usp=sharing

Mathematics and Statistics Department. (2022). FNDSTAT Hypothesis Testing Formula Sheet for Single Populations. De La Salle University. https://drive.google.com/drive/folders/1Y1mhqy8yeHZB9YC7WBPkQCNMH8Icvorh?usp=sharing

Maxwell, S. E., & Delaney, H. D. (2004). Designing experiments and analyzing data: A model comparison perspective. *Psychology Press*. https://doi.org/10.4324/9781315642956

Moore, D. S., & McCabe, G. P. (2003). Introduction to the practice of statistics. *WH Freeman/Times Books/Henry Holt & Co*.

Patten, M. L. (2018). Understanding research methods: An overview of the essentials. *Routledge*.

Rosnow, R. L., & Rosenthal, R. (1991). Effect sizes for experimenting psychologists. *Canadian Journal of Experimental Psychology, 45(3)*, 308. https://awspntest.apa.org/doi/10.1037/h0087427

Welch, B. L. (1938). The Significance of the Difference Between Two Means when the Population Variances are Unequal. *Biometrika, 29(3/4)*, 350–362. https://doi.org/10.2307/2332010

Zhang, Z. (2015). A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods, 47(2)*, 800-816. https://doi.org/10.3758/bf03193146

Appendices

A. User's Manual

Please write brief instructions on how to use the program, e.g. username and password and other information that you think will be helpful to the user, screenshots can also be included.

B. Source Code – include comments in your source codes

C. **Work breakdown** – itemize the work done by each member of the group.

Example:

Student Name	Tasks Assigned	Percentage of the Work Contribution
Student 1	Module 1 – login authentication Review of related literature	20%

	Please add other tasks assigned to the student	
Student 2	Module 2 – the process of printing the receipt Methodology – Flowchart Please add other tasks assigned to the student	40%
Student 3	Module 3 – the process of quiz evaluation Methodology – Pseudocode Please add other tasks assigned to the student	40%

D. Personal Data Sheet – personal data sheet of each member with 2x2 ID picture, you may opt to limit the information to appear in your data sheet to those which you are comfortable sharing.