

What is hypothesis testing in statistics* Hypothesis testing is a statistical method used to make decisions or draw conclusions about a population based on sample data. It is a framework for testing assumptions (or hypotheses) about a parameter of the population, such as the mean, proportion, or variance.

Key Components of Hypothesis Testing: Null Hypothesis (H_0):

The default assumption or statement being tested. Usually represents no effect, no difference, or the status quo (e.g., $H_0: \mu = \mu_0$). Alternative Hypothesis (H_a or H_1):

Represents the opposite of the null hypothesis. Suggests the presence of an effect, difference, or relationship (e.g., $H_a: \mu \neq \mu_0$). Significance Level (α):

The threshold for determining whether to reject H_0 . Common values are 0.05, 0.01, or 0.10, representing the probability of rejecting H_0 when it is true (Type I error). Test Statistic:

A calculated value based on sample data that measures the difference between the observed data and what is expected under H_0 . Examples include z -scores, t -scores, or F -statistics. P-Value:

The probability of observing a test statistic at least as extreme as the one calculated, assuming H_0 is true. A small p -value (typically $< \alpha$) suggests evidence against H_0 . Decision:

Reject H_0 : If the p -value is less than α , there is sufficient evidence to support H_a .

Fail to Reject H_0 : If the p -value is greater than α , there is insufficient evidence to support H_a . Steps in Hypothesis Testing: State the Hypotheses:

Formulate H_0 and H_a clearly. Choose the Significance Level (α):

Set a threshold for decision-making. Select and Compute the Test Statistic:

Choose the appropriate statistical test based on data type and hypothesis. Compute the test statistic using sample data. Determine the P-Value or Critical Value:

Compare the test statistic to critical values or calculate the p -value. Make a Decision:

Based on the comparison, decide to reject or fail to reject H_0 . Interpret the Results:

Provide a conclusion in the context of the original problem. Example: Suppose a company claims their light bulbs last an average of 1,000 hours. A consumer group tests a sample of 50 bulbs and finds an average lifespan of 980 hours with a known standard deviation of 30 hours. Using a 0.05 significance level, the group can test if the mean lifespan differs from 1,000 hours.

This process of testing $H_0: \mu = 1000$ against $H_a: \mu \neq 1000$ is hypothesis testing in action.

#2) What is the null hypothesis, and how does it differ from the alternative hypothesis*

Null Hypothesis (H_0): The null hypothesis is the default assumption or claim that there is no effect, no difference, or no relationship in the population. It represents the status quo or the claim that any observed differences in the data are due to random variation or chance. Example: $H_0: \mu = 100$ (The population mean is 100.) $H_0: p = 0.5$ (The population proportion is 0.5.) $H_0: \sigma_1^2 = \sigma_2^2$ (The variances of two populations are equal.)

Alternative Hypothesis (H_a): The alternative hypothesis is the statement being tested against the null hypothesis. It represents a claim of an effect, a difference, or a relationship that the researcher wants to investigate. Example: $H_a: \mu \neq 100$ (The population mean is not 100.) $H_a: p > 0.5$ (The population proportion is greater than 0.5.) $H_a: \sigma_1^2 \neq \sigma_2^2$ (The variances of two populations are not equal.)

Key Differences Between H_0 and H_a :

Aspect	Null Hypothesis (H_0)	Alternative Hypothesis (H_a)
Definition	Assumes no effect, no difference, or no relationship.	Suggests an effect, difference, or relationship exists.
Purpose	Serves as the baseline or default claim to test.	Represents the researcher's claim or what they aim to prove.
Rejection	Rejected if evidence suggests it is unlikely.	Accepted if H_0 is rejected based on statistical evidence.
Direction	Typically uses an equality (e.g., $=$).	Can be one-tailed ($<$ or $>$) or two-tailed (\neq).

Focus: Protects against Type I error (false positive).

Associated with Type II error (false negative).

Relationship Between H_0 and H_a : These hypotheses are mutually exclusive: only one can be true in the context of the test. They are also exhaustive: they cover all possible outcomes regarding the parameter being tested. Example: A pharmaceutical company wants to test if a new drug increases patient recovery rates compared to a standard recovery rate of 70%:

$H_0: p = 0.7$ (The recovery rate is 70%, no improvement.) $H_a: p > 0.7$ (The recovery rate is greater than 70%, indicating improvement.)

This test would evaluate whether there is enough evidence to reject H_0 and support H_a .

#3) What is the significance level in hypothesis testing, and why is it important?

Significance Level (α) in Hypothesis Testing: The significance level is the threshold used to determine whether a null hypothesis (H_0) should be rejected in a hypothesis test. It is denoted by α and represents the probability of making a Type I error, which occurs when H_0 is rejected even though it is true.

Importance of the Significance Level: Controls Error Risk:

By setting α , researchers control the likelihood of falsely rejecting H_0 (Type I error). For example, if $\alpha = 0.05$, there is a 5% chance of rejecting H_0 when it is true. Establishes

a Decision Rule:

The significance level provides a benchmark for comparing the p-value (the probability of observing the test statistic or something more extreme if H_0 is true). If $p \leq \alpha$, reject H_0 ; otherwise, fail to reject H_0 . Balances Risks:

A lower α reduces the risk of Type I error but increases the risk of a Type II error (failing to reject H_0 when it is false). Researchers choose α based on the context of the study and the consequences of errors. Guides Interpretation:

It helps in determining the strength of evidence against H_0 . For instance, smaller α values (e.g., 0.01) require stronger evidence to reject H_0 , ensuring more confidence in the results.

Common Significance Levels: 0.05 (5%): Most commonly used; balances error risks reasonably.

0.01 (1%): Used in cases where the cost of Type I error is very high (e.g., medical trials, safety studies). 0.10 (10%): Sometimes used in exploratory research where stricter thresholds might be too conservative. Example: Suppose a researcher tests if a new teaching method improves student scores compared to the standard method, with $\alpha = 0.05$. The null hypothesis is $H_0: \mu = 75$ (average score remains 75).

If the test yields a p-value of 0.03: Since $p < \alpha$, the researcher rejects H_0 , concluding the new method likely improves scores. If the p-value were 0.08: Since p

$p > \alpha$, H_0 is not rejected, and there is insufficient evidence to claim improvement. The choice of α ensures this decision process is systematic and consistent.

Double-click (or enter) to edit

#4) What does a P-value represent in hypothesis testing*

What is a P-Value in Hypothesis Testing? The p-value (or probability value) is a key concept in hypothesis testing. It represents the probability of observing a test statistic as extreme as, or more extreme than, the one calculated from the sample data, assuming that the null hypothesis (H_0) is true.

In simpler terms, the p-value quantifies how well the sample data aligns with H_0 . A smaller p-value indicates stronger evidence against H_0 .

Interpretation of the P-Value: Small P-Value ($p \leq \alpha$):

Indicates the observed result is unlikely under H_0 . Suggests rejecting the null hypothesis.

Example: $p = 0.02$ (with $\alpha = 0.05$) implies that there's only a 2% chance of observing the data if H_0 is true. Large P-Value (p

$p > \alpha$):

Indicates the observed result is consistent with H_0 . Suggests failing to reject the null hypothesis. Example: $p = 0.15$ (with $\alpha = 0.05$) implies the evidence is insufficient

to conclude H_0 is false. Key Features of the P-Value: Range: The p-value ranges from 0 to 1.

$p = 0$: Strong evidence against H_0 . $p = 1$: Complete alignment with H_0 .

Dependence on Sample Data:

The p-value reflects the data and test statistic, not the truth or falsity of H_0 . Threshold:

Compared against the significance level (α) to make decisions:

Common thresholds: $\alpha = 0.05$, $\alpha = 0.01$. Common Misinterpretations of the P-

Value: Not the Probability That H_0 is True:

The p-value does not provide the probability that H_0 is correct or false. Not the Probability of Making an Error:

The p-value is not the probability of making a Type I or Type II error. Does Not Measure Practical Significance:

A small p-value indicates statistical significance, but it does not guarantee practical or meaningful effects. Example: A researcher tests if a coin is fair ($H_0: p = 0.5$, where p is the probability of heads). The coin is flipped 50 times, and heads appear 35 times. Using a hypothesis test, the p-value is calculated as $p = 0.02$.

Interpretation: If the coin were truly fair, the probability of observing 35 or more heads in 50 flips is 2%. Since $p < 0.05$, the researcher rejects H_0 , concluding the coin is likely biased.

The p-value helps quantify evidence and supports a systematic decision-making process in hypothesis testing.

#5) How do you interpret the P-value in hypothesis testing?

What is a P-Value in Hypothesis Testing? The p-value (or probability value) is a key concept in hypothesis testing. It represents the probability of observing a test statistic as extreme as, or more extreme than, the one calculated from the sample data, assuming that the null hypothesis (H_0) is true.

In simpler terms, the p-value quantifies how well the sample data aligns with H_0 . A smaller p-value indicates stronger evidence against H_0 .

Interpretation of the P-Value: Small P-Value ($p \leq \alpha$):

Indicates the observed result is unlikely under H_0 . Suggests rejecting the null hypothesis.

Example: $p = 0.02$ (with $\alpha = 0.05$) implies that there's only a 2% chance of observing the data if H_0 is true. Large P-Value ($p > \alpha$):

Indicates the observed result is consistent with H_0 . Suggests failing to reject the null hypothesis. Example: $p = 0.15$ (with $\alpha = 0.05$) implies the evidence is insufficient to conclude H_0 is false. Key Features of the P-Value: Range: The p-value ranges from 0 to 1.

$p = 0$ $p=0$: Strong evidence against H_0 . $p = 1$ $p=1$: Complete alignment with H_0 .

Dependence on Sample Data:

The p-value reflects the data and test statistic, not the truth or falsity of H_0 . Threshold:

Compared against the significance level (α) to make decisions:

Common thresholds: $\alpha = 0.05$, $\alpha = 0.01$. Common Misinterpretations of the P-Value: Not the Probability That H_0 is True:

The p-value does not provide the probability that H_0 is correct or false. Not the Probability of Making an Error:

The p-value is not the probability of making a Type I or Type II error. Does Not Measure Practical Significance:

A small p-value indicates statistical significance, but it does not guarantee practical or meaningful effects. Example: A researcher tests if a coin is fair ($H_0: p = 0.5$, where p is the probability of heads). The coin is flipped 50 times, and heads appear 35 times. Using a hypothesis test, the p-value is calculated as $p = 0.02$.

Interpretation: If the coin were truly fair, the probability of observing 35 or more heads in 50 flips is 2%. Since $p < 0.05$, the researcher rejects H_0 , concluding the coin is likely biased. The p-value helps quantify evidence and supports a systematic decision-making process in hypothesis testing.

#6)What are Type 1 and Type 2 errors in hypothesis testing*

In hypothesis testing, Type I and Type II errors represent the two possible mistakes that can occur when making decisions about the null hypothesis (H_0).

Type I Error (α): Definition: A Type I error occurs when the null hypothesis (H_0) is rejected, even though it is actually true. Example: Concluding a new drug is effective when it actually has no effect. Probability: The significance level (α) determines the probability of making a Type I error. Common choices for α : 0.05 (5%), 0.01 (1%). Type II Error (β): Definition: A Type II error occurs when the null hypothesis (H_0) is not rejected, even though it is actually false.

Example: Failing to detect that a new drug is effective when it actually works. Probability: Denoted by β , which depends on factors like sample size, effect size, and significance level (α). $1 - \beta$ is called the power of the test, representing the probability of correctly rejecting H_0 when it is false.

#7)What is the difference between a one-tailed and a two-tailed test in hypothesis testing

Difference Between One-Tailed and Two-Tailed Tests in Hypothesis Testing: In hypothesis testing, the choice between a one-tailed and a two-tailed test depends on the nature of the research question and how the alternative hypothesis (H_a) is formulated.

1. One-Tailed Test: A one-tailed test evaluates whether a parameter is either greater than or less than a specific value, but not both. The rejection region is located in only one tail of the distribution. Used when the research question is directional, meaning the alternative hypothesis specifies a particular direction of effect. Key Features: Hypotheses: Null hypothesis (H_0): $\mu = \mu_0$

Alternative hypothesis (H_a): For a right-tailed test: μ

$$\mu > \mu_0$$

For a left-tailed test: $\mu < \mu_0$

Example: Testing if a new drug increases recovery rates: $H_0: \mu = 70\%$ (recovery rate is 70%) $H_a: \mu$

$\mu > 70\%$ (recovery rate is greater than 70%)

2. Two-Tailed Test: A two-tailed test evaluates whether a parameter is different from a specific value, considering both directions (greater than or less than). The rejection regions are located in both tails of the distribution

#8) What is the Z-test, and when is it used in hypothesis testing?

A Z-test is a statistical test used to determine whether there is a significant difference between a sample statistic (e.g., sample mean or proportion) and a population parameter, or between two sample statistics. It relies on the assumption that the data are normally distributed or that the sample size is sufficiently large for the Central Limit Theorem to apply.

When is a Z-Test Used? A Z-test is appropriate in the following scenarios:

Known Population Standard Deviation (σ):

The population standard deviation is known, and the sample size is large ($n \geq 30$), or the data are normally distributed for smaller samples. Comparing Means or Proportions:

To test the mean of a single population ($H_0: \mu = \mu_0$). To compare the means of two populations ($H_0: \mu_1 = \mu_2$). To test the proportion of a single population ($H_0: p = p_0$). To compare proportions between two populations ($H_0: p_1 = p_2$).

Large Sample Sizes:

For sample sizes $n \geq 30$, the sampling distribution of the test statistic approaches normality, even if the data are not normally distributed. Hypothesis Testing:

When testing hypotheses about population parameters. Types of Z-Tests: One-Sample Z-Test:

Tests whether the mean of a single sample differs from a known population mean. Example:

Testing if the average height of a group of students differs from the national average. Two-

Sample Z-Test:

Compares the means of two independent samples. Example: Testing whether the average income of two cities is the same. Z-Test for Proportions:

Tests whether a sample proportion differs from a known population proportion or compares proportions between two samples. Example: Testing if the proportion of voters favoring a candidate differs from 50%. Steps to Perform a Z-Test: State the Hypotheses:

Null hypothesis (H_0): Assumes no difference or effect. Alternative hypothesis (H_a): Assumes a difference or effect. Calculate the Z-Statistic:

For a single mean:

$$Z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$$

$$\bar{x} - \mu$$

where \bar{x} is the sample mean, μ is the population mean, σ is the population standard deviation, and n is the sample size.

For proportions:

$$Z = \frac{\hat{p} - p_0}{\sqrt{p_0(1-p_0) / n}}$$

$$\hat{p} - p_0$$

where \hat{p} is the sample proportion, p_0 is the population proportion, and n is the sample size.

Determine the P-Value:

Use standard normal tables or statistical software to find the p-value corresponding to the calculated Z-statistic. Compare with α :

If $p \leq \alpha$, reject H_0 . Otherwise, fail to reject H_0 . Example:

#9)How do you calculate the Z-score, and what does it represent in hypothesis testing*

What is a Z-Score? A Z-score represents the number of standard deviations a data point, sample mean, or statistic is from the mean of the distribution. In hypothesis testing, it measures how extreme the observed data are under the assumption that the null hypothesis (H_0) is true.

Formula for Z-Score: The formula depends on the context:

For a Single Data Point (x):

$$Z = \frac{x - \mu}{\sigma}$$

x : Observed value. μ : Population mean. σ : Population standard deviation. For a Sample Mean (\bar{x}):

$$Z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$$

$$\bar{x} - \mu$$

\bar{x} : Sample mean. μ : Population mean (under H_0). σ : Population standard deviation.

n : Sample size. For Proportions:

$$Z = \frac{\hat{p} - p_0}{\sqrt{p_0(1-p_0)}} / \sqrt{n}$$

$$\hat{p} - p_0$$

\hat{p} : Sample proportion. p_0 : Population proportion (under H_0). n : Sample size. What Does the Z-Score Represent? Measure of Extremeness:

Indicates how far the observed value is from the mean, measured in standard deviations.

Standardized Scale:

Z-scores are standardized, meaning they can be compared across different datasets or distributions. Evidence Against the Null Hypothesis (H_0):

A high absolute Z-score (e.g., Z

$Z > 2$ or $Z < -2$) suggests the observed data are unusual under H_0 . Determines P-Value:

The Z-score is used to calculate the p-value, which quantifies the probability of observing data as extreme as the test statistic under H_0 . Steps to Calculate a Z-Score in Hypothesis

#10}What is the T-distribution, and when should it be used instead of the normal distribu

The t-distribution (or Student's t-distribution) is a probability distribution used in statistics when making inferences about a population mean, especially when the sample size is small and/or the population standard deviation (σ) is unknown. It is similar to the normal distribution but has heavier tails, meaning it accounts for more variability in the data.

Key Characteristics of the T-Distribution: Shape:

Bell-shaped and symmetric around zero, like the normal distribution. Has heavier tails, which allow for greater uncertainty in estimates. Degrees of Freedom (df):

The t-distribution depends on $df = n - 1$, where n is the sample size. As df increases, the t-distribution approaches the normal distribution. Variance:

Greater than 1 for small df , decreasing as df increases. Equal to 1 when df is infinite (i.e., the t-distribution becomes the normal distribution). When to Use the T-Distribution Instead of the Normal Distribution: Unknown Population Standard Deviation (σ):

When σ is unknown and the sample standard deviation (s) is used as an estimate. Small Sample Size ($n < 30$):

The t-distribution is more appropriate for small samples because it accounts for the additional uncertainty from estimating σ . Hypothesis Testing and Confidence Intervals:

Used in single-sample, paired-sample, and two-sample t-tests. Used to construct confidence intervals for the mean when σ is unknown. T-Distribution vs. Normal Distribution: Feature T-Distribution Normal Distribution Applicability Used when σ is unknown and/or $n < 30$.

Used when σ is known or $n \geq 30$. Shape Bell-shaped with heavier tails. Bell-shaped with thinner tails. Degrees of Freedom Depends on $df = n - 1$. Does not depend on df . Approach to Normality Approaches normal distribution as $df \rightarrow \infty$. Fixed shape regardless of sample size. Formula for T-Statistic:

#11)What is the difference between a Z-test and a T-test*

Differences Between a Z-Test and a T-Test The Z-test and T-test are both statistical tests used to compare sample data to population parameters or to compare two samples. The main differences lie in the assumptions, use cases, and formulas.

Key Differences Feature Z-Test T-Test Population Standard Deviation (σ) Requires σ to be known. Used when σ is unknown (uses sample s as an estimate). Sample Size Appropriate for large samples ($n \geq 30$). Suitable for small samples ($n < 30$). Distribution Based on the normal distribution. Based on the t-distribution. Degrees of Freedom (df) Not required; fixed standard normal distribution. Depends on $df = n - 1$ for single-sample t-tests. Tail Behavior Thinner tails (less conservative for small samples). Heavier tails (accounts for variability in small samples). Applications - Test of population mean/proportion when σ is known.

- Comparing large independent samples. - Test of population mean when σ is unknown.
- Comparing means of small independent or paired samples. Formulas Z-Test:

For a single mean: $Z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$

$\bar{x} - \mu$

For proportions: $Z = \frac{\hat{p} - p_0}{\sqrt{p_0(1-p_0)}} \sqrt{n}$

$\hat{p} - p_0$

T-Test:

For a single mean: $t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$

$\bar{x} - \mu$

When to Use a Z-Test or T-Test Z-Test:

Large sample sizes ($n \geq 30$). Population standard deviation (σ) is known. Examples: Testing if the mean height of students in a large class differs from 170 cm. Comparing

proportions of survey responses in a large dataset. T-Test:

Small sample sizes ($n < 30$). Population standard deviation (σ) is unknown. Examples: Testing if a new drug affects blood pressure based on a small clinical trial. Comparing the mean scores of two small groups of students. Example to Illustrate the Difference Scenario: You want to test whether the mean weight of apples in a sample differs from 150 grams.

Z-Test:

Known $\sigma = 10$, sample size $n = 40$, sample mean $\bar{x} = 145$: $Z = \frac{145 - 150}{10 / \sqrt{40}} = -3.16$

Look up the Z-score to determine significance. T-Test:

Unknown σ , sample standard deviation $s = 10$, $n = 15$, $\bar{x} = 145$: $t = \frac{145 - 150}{10 / \sqrt{15}} = -1.94$

Use the t-distribution with $df = 15 - 1 = 14$ to determine significance. Summary Test Use Case Z-Test Large samples, known population

#12)What is the T-test, and how is it used in hypothesis testing*

The T-test is a statistical method used to determine whether there is a significant difference between the means of two groups, or between a sample mean and a population mean. It is commonly applied when the population standard deviation (σ) is unknown, and the sample size is small ($n < 30$).

Types of T-Tests One-Sample T-Test:

Compares the mean of a single sample to a known population mean. Example: Testing if the average height of students in a class is equal to the national average. Independent (Two-Sample) T-Test:

Compares the means of two independent groups. Example: Testing if the average test scores of two different schools are significantly different. Paired (Dependent) T-Test:

Compares the means of two related groups (e.g., before-and-after measurements). Example: Testing the effect of a training program on employee productivity. How is the T-Test Used in Hypothesis Testing?

1. Formulate Hypotheses Null Hypothesis (H_0): Assumes no difference between the means. Alternative Hypothesis (H_a): Assumes a significant difference exists.
2. Calculate the T-Statistic The formula depends on the type of t-test:

One-Sample T-Test:

$$t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$$

$$\bar{x} - \mu$$

\bar{x} : Sample mean. μ : Population mean. s : Sample standard deviation. n : Sample size.

Independent Two-Sample T-Test:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad t = n_1$$

$$s_1^2$$

$$\bullet \quad n_2$$

$$s_2^2$$

$$\bar{x}$$

$$1 - \bar{x}$$

$$2$$

$$\bar{x}_1, \bar{x}_2$$

$$1, \bar{x}$$

2: Means of two groups. s_1, s_2 : Standard deviations of two groups. n_1, n_2 : Sample sizes of two groups. Paired T-Test:

$$t = \frac{\bar{d} - s_d}{\sqrt{n}} \quad t = s_d / n$$

$$\bar{d}$$

\bar{d} : Mean of the differences between paired observations. s_d : Standard deviation of the differences. n : Number of paired observations.

3. Determine Degrees of Freedom (df) For one-sample or paired t-test: $df = n - 1$.

For two-sample t-test: $df = n_1$

+ $n_2 - 2$ (if variances are equal).

4. Find the Critical Value or P-Value Compare the calculated t-statistic to the critical value from a t-distribution table for the given df and significance level (α). Alternatively, calculate the p-value corresponding to the t-statistic.

5. Make a Decision If $|t|$

$|t| > \text{critical value}$ or $p \leq \alpha$, reject H_0 . Otherwise, fail to reject H_0 . Example:

One-Sample T-Test

#13)What is the relationship between Z-test and T-test in hypothesis testing*

Relationship Between Z-Test and T-Test in Hypothesis Testing The Z-test and T-test are closely related statistical methods used to test hypotheses about population means or proportions. Both tests assess whether observed data significantly differ from a hypothesized value or between groups. However, their use depends on specific conditions regarding sample size and knowledge of population parameters.

Key Relationships Both Test Means and Proportions:

Both Z-tests and T-tests are used to compare: A sample mean to a population mean. Two sample means. Proportions (in the case of Z-tests). Dependence on Sample Size:

Z-Test: Appropriate for large sample sizes ($n \geq 30$ $n \geq 30$) or when the population standard deviation (σ) is known. **T-Test:** Used when the sample size is small ($n < 30$ $n < 30$) and the population standard deviation (σ) is unknown. Underlying Distributions:

Z-Test: Relies on the normal distribution. **T-Test:** Relies on the t-distribution, which adjusts for the uncertainty of using the sample standard deviation (s) as an estimate for σ . Convergence:

As the sample size increases ($n \rightarrow \infty$ $n \rightarrow \infty$), the t-distribution approaches the normal distribution. For large samples, the T-test becomes nearly identical to the Z-test. Test Statistic Similarity:

The formulas for the test statistics are similar, differing primarily in whether the population standard deviation (σ) or the sample standard deviation (s) is used: **Z-Test:** $Z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$

$$\bar{x} - \mu$$

T-Test: $t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$

$$\bar{x} - \mu$$

Interpretation:

Both tests yield a statistic (Z or t) that measures how extreme the sample data are relative to the null hypothesis. Both tests result in a p-value, which is compared to a significance level (α) to make a decision. When to Use Each Test Criterion Z-Test T-Test Population Standard Deviation (σ) Known Unknown Sample Size (n) Large ($n \geq 30$ $n \geq 30$) Small ($n < 30$ $n < 30$) Distribution Normal T-distribution Examples Comparing a large sample mean to a population mean. Comparing a small sample mean to a population mean.

Start coding or [generate](#) with AI.

14. What is a confidence interval, and how is it used to interpret statistical results? A confidence interval (CI) is a range of values, derived from sample data, that is likely to contain the true population parameter (e.g., mean, proportion) with a specified level of confidence (e.g., 95%).

Interpretation: A 95% confidence interval means that if we were to take 100 different samples and compute a confidence interval for each, approximately 95 of them would contain the true population parameter. Use: It provides a measure of the uncertainty or precision of an estimate.

15. What is the margin of error, and how does it affect the confidence interval? The margin of error (MOE) is the maximum expected difference between the sample statistic and the true population parameter due to sampling variability.

Effect: A smaller margin of error results in a narrower confidence interval, indicating more precise estimates. A larger margin of error results in a wider confidence interval, indicating less precision.

16. How is Bayes' Theorem used in statistics, and what is its significance? Bayes' Theorem describes the probability of an event based on prior knowledge of conditions related to the event. It updates prior probabilities based on new evidence.

$$P(A|B) = P(B|A) \cdot P(A) / P(B)$$

Use in Statistics: Updating probabilities as new data becomes available. Applied in fields such as machine learning, medical diagnosis, and decision-making. Significance: It helps incorporate prior knowledge into probabilistic models.

17. What is the Chi-square distribution, and when is it used? The Chi-square distribution is a probability distribution that arises when a sum of squared independent standard normal variables is calculated. It is skewed for small degrees of freedom (df) and becomes symmetric as df increases.

Use: Goodness-of-fit tests. Tests of independence (e.g., contingency tables). Variance analysis of normally distributed data.

18. What is the Chi-square goodness of fit test, and how is it applied? The Chi-square goodness-of-fit test determines whether observed categorical data matches an expected distribution.

Steps: Formulate hypotheses: H_0 : Observed data matches the expected distribution. H_a : Observed data does not match. Compute the test statistic: $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$

where O_i = observed frequency, E_i = expected frequency. Compare the statistic to the critical value from the Chi-square distribution table.

19. What is the F-distribution, and when is it used in hypothesis testing? The F-distribution is a probability distribution used in hypothesis testing, especially in comparing variances or analyzing variance.

Use: Analysis of Variance (ANOVA). Testing the equality of two population variances.

Characteristics: Positively skewed. Defined by two degrees of freedom (df_1, df_2).

20. What is an ANOVA test, and what are its assumptions? The Analysis of Variance (ANOVA) tests whether there are significant differences between the means of three or more groups.

Assumptions: The populations are normally distributed. Homogeneity of variances (equal variances across groups). Observations are independent.

21. What are the different types of ANOVA tests? One-Way ANOVA:

Compares the means of three or more groups based on one independent variable. Example:

Comparing test scores of students from different schools. Two-Way ANOVA:

Examines the effect of two independent variables on a dependent variable. Example: Testing the effect of gender and teaching method on test scores. Repeated Measures ANOVA:

Used when the same subjects are measured under different conditions. Example: Measuring blood pressure before and after a treatment.

22. What is the F-test, and how does it relate to hypothesis testing? The F-test is used to compare variances or test hypotheses about means (as part of ANOVA).

Relation to Hypothesis Testing: Null Hypothesis (H_0): Variances or means are equal.

Alternative Hypothesis (H_a): At least one variance or mean is different. Formula: $F = \frac{\text{Variance of group 1}}{\text{Variance of group 2}}$

If the calculated F-value exceeds the critical value from the F-distribution table, H_0 is rejected.

#PRACTICAL QUESTIONS

#1) Write a Python program to perform a Z-test for comparing a sample mean to a known population mean.

```
#interpret the results@
import scipy.stats as stats
import math
```

```
def z_test(sample_mean, population_mean, population_std, sample_size, significance_level=0.05):
    # Calculate the Z-score
    z_score = (sample_mean - population_mean) / (population_std / math.sqrt(sample_size))
```

```
    # Calculate the p-value (two-tailed test)
    p_value = 2 * (1 - stats.norm.cdf(abs(z_score)))
```

```
    # Interpret the result
    print(f"Z-score: {z_score}")
    print(f"P-value: {p_value}")
```

```
    if p_value < significance_level:
        print(f"At a significance level of {significance_level}, we reject the null hypothesis")
        print("There is enough evidence to suggest the sample mean is significantly different from the population mean.")
```

```

else:
    print(f"At a significance level of {significance_level}, we fail to reject the nu
    print("There is not enough evidence to suggest the sample mean is significantly d

# Example usage
# Known population mean and standard deviation
population_mean = 50
population_std = 5

# Sample data
sample_mean = 48
sample_size = 30

# Significance level (default is 0.05)
significance_level = 0.05

# Perform the Z-test
z_test(sample_mean, population_mean, population_std, sample_size, significance_level)

🔍 Z-score: -2.1908902300206643
P-value: 0.02845973691631065
At a significance level of 0.05, we reject the null hypothesis.
There is enough evidence to suggest the sample mean is significantly different from t

```

Double-click (or enter) to edit

```

##$@ Simulate random data to perform hypothesis testing and calculate the corresponding P-
import numpy as np
from scipy.stats import ttest_ind

# Set random seed for reproducibility
np.random.seed(42)

# Simulate random data for two groups
# Group 1: Mean = 50, Standard Deviation = 10, Sample Size = 30
group1 = np.random.normal(loc=50, scale=10, size=30)

# Group 2: Mean = 55, Standard Deviation = 10, Sample Size = 30
group2 = np.random.normal(loc=55, scale=10, size=30)

# Perform two-sample t-test
t_stat, p_value = ttest_ind(group1, group2)

# Print results
print("Group 1 Mean:", np.mean(group1))
print("Group 2 Mean:", np.mean(group2))
print("T-statistic:", t_stat)
print("P-value:", p_value)

# Interpret the p-value
alpha = 0.05
if p_value < alpha:

```

```
print("Reject the null hypothesis: There is a significant difference between the groups")
else:
```

```
print("Fail to reject the null hypothesis: No significant difference between the groups")
```



```
Group 1 Mean: 48.118531041489625
```

```
Group 2 Mean: 53.788375297100565
```

```
T-statistic: -2.3981151520102415
```

```
P-value: 0.019717941865758228
```

```
Reject the null hypothesis: There is a significant difference between the groups.
```

Double-click (or enter) to edit

```
#Implement a one-sample Z-test using Python to compare the sample mean with the population mean
```

```
import numpy as np
```

```
from scipy.stats import norm
```

```
# Sample data
```

```
sample_data = np.random.normal(loc=52, scale=8, size=50) # Sample with mean=52, std=8, n=50
```

```
# Population parameters
```

```
population_mean = 50 # The population mean
```

```
population_std = 8 # The population standard deviation
```

```
# Calculate sample mean and sample size
```

```
sample_mean = np.mean(sample_data)
```

```
sample_size = len(sample_data)
```

```
# Calculate the standard error of the mean (SEM)
```

```
sem = population_std / np.sqrt(sample_size)
```

```
# Calculate the Z-score
```

```
z_score = (sample_mean - population_mean) / sem
```

```
# Calculate the p-value for a two-tailed test
```

```
p_value = 2 * (1 - norm.cdf(abs(z_score)))
```

```
# Print results
```

```
print("Sample Mean:", sample_mean)
```

```
print("Population Mean:", population_mean)
```

```
print("Z-Score:", z_score)
```

```
print("P-value:", p_value)
```

```
# Interpret the result
```

```
alpha = 0.05 # Significance level
```

```
if p_value < alpha:
```

```
    print("Reject the null hypothesis: The sample mean is significantly different from the population mean")
```

```
else:
```

```
    print("Fail to reject the null hypothesis: The sample mean is not significantly different from the population mean")
```



```
Sample Mean: 51.74404075654454
```

```
Population Mean: 50
```

```
Z-Score: 1.5415288070229534
```


P-value: 0.12318813743726431

Fail to reject the null hypothesis: The sample mean is not significantly different fr

Double-click (or enter) to edit

```
#Perform a two-tailed Z-test using Python and visualize the decision region on a plot
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Sample data
sample_data = np.random.normal(loc=52, scale=8, size=50) # Sample with mean=52, std=8, n

# Population parameters
population_mean = 50 # Population mean
population_std = 8 # Population standard deviation

# Calculate sample mean and sample size
sample_mean = np.mean(sample_data)
sample_size = len(sample_data)

# Calculate the standard error of the mean (SEM)
sem = population_std / np.sqrt(sample_size)

# Calculate the Z-score
z_score = (sample_mean - population_mean) / sem

# Calculate the p-value for a two-tailed test
p_value = 2 * (1 - norm.cdf(abs(z_score)))

# Set significance level
alpha = 0.05

# Print results
print("Sample Mean:", sample_mean)
print("Population Mean:", population_mean)
print("Z-Score:", z_score)
print("P-value:", p_value)

# Decision rule interpretation
if p_value < alpha:
    decision = "Reject the null hypothesis: The sample mean is significantly different fr
else:
    decision = "Fail to reject the null hypothesis: The sample mean is not significantly

print(decision)

# Visualization of the decision region
x = np.linspace(-4, 4, 1000) # Values for the Z-distribution plot
y = norm.pdf(x, loc=0, scale=1) # Standard normal distribution

# Plot the Z-distribution
```

```
plt.plot(x, y, label="Standard Normal Distribution", color='blue')

# Fill the rejection regions (two-tailed test)
plt.fill_between(x, y, where=(x <= -1.96) | (x >= 1.96), color='red', alpha=0.5, label="R")

# Plot the Z-score
plt.axvline(z_score, color='green', linestyle='--', label=f"Z-Score = {z_score:.2f}")

# Mark the critical values ( $\pm 1.96$  for  $\alpha = 0.05$ )
plt.axvline(-1.96, color='black', linestyle='--', label="Critical Value (-1.96)")
plt.axvline(1.96, color='black', linestyle='--', label="Critical Value (+1.96)")

# Add labels and title
plt.title('Two-Tailed Z-Test Visualization')
plt.xlabel('Z-Score')
plt.ylabel('Density')
plt.legend(loc='best')

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



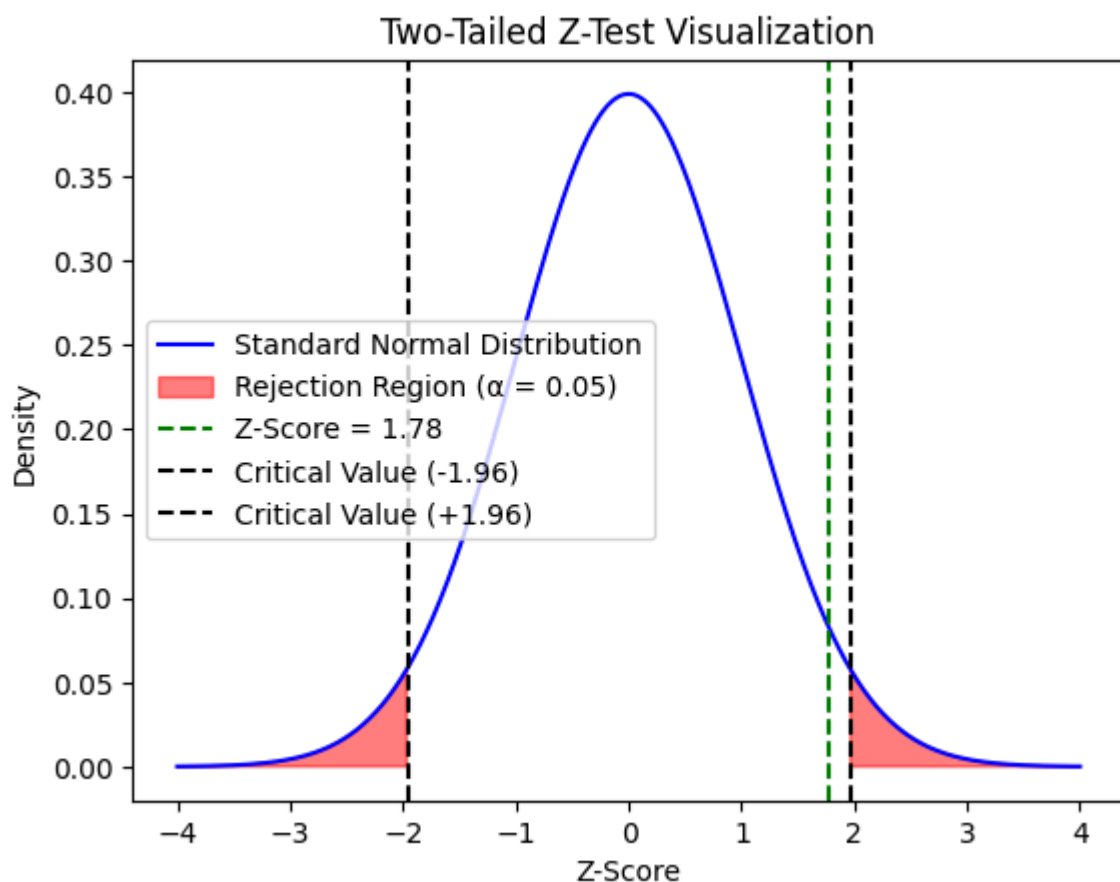
Sample Mean: 52.01003917074095

Population Mean: 50

Z-Score: 1.7766404101018904

P-value: 0.07562742244695086

Fail to reject the null hypothesis: The sample mean is not significantly different fr



```
#Create a Python function that calculates and visualizes Type 1 and Type 2 errors during
import numpy as np
import matplotlib.pyplot as plt
```

```

from scipy.stats import norm

def visualize_errors(population_mean, population_std, sample_size, alpha=0.05, true_mean=None):
    # Set default value for true mean if not provided
    if true_mean is None:
        true_mean = population_mean

    # Generate random sample from the population
    sample_data = np.random.normal(loc=true_mean, scale=population_std, size=sample_size)

    # Calculate sample mean and standard error of the mean (SEM)
    sample_mean = np.mean(sample_data)
    sem = population_std / np.sqrt(sample_size)

    # Perform Z-test
    z_score = (sample_mean - population_mean) / sem
    p_value = 2 * (1 - norm.cdf(abs(z_score))) # Two-tailed test

    # Calculate Type I and Type II errors
    # Type I Error: Reject null hypothesis when it is true (in the rejection region)
    # Type II Error: Fail to reject null hypothesis when it is false (in the acceptance region)

    # Critical values for two-tailed Z-test at alpha = 0.05
    critical_value = norm.ppf(1 - alpha / 2)

    # Type I error occurs if we reject the null hypothesis when it is true (outside the critical values)
    type_1_error = z_score > critical_value or z_score < -critical_value

    # Type II error occurs if we fail to reject the null hypothesis when the true mean is different from the population mean
    # We'll simulate a scenario where the true mean is different from the population mean
    if true_mean != population_mean:
        # Recalculate Type II error based on true mean
        z_score_type_ii = (sample_mean - true_mean) / sem
        type_ii_error = abs(z_score_type_ii) < critical_value
    else:
        type_ii_error = False # No Type II error if true mean equals population mean

    # Visualization
    x = np.linspace(-4, 4, 1000) # Values for the Z-distribution plot
    y = norm.pdf(x, loc=0, scale=1) # Standard normal distribution

    # Plot the Z-distribution
    plt.plot(x, y, label="Standard Normal Distribution", color='blue')

    # Fill the rejection regions (Type I error region)
    plt.fill_between(x, y, where=(x <= -critical_value) | (x >= critical_value), color='red')

    # Plot the Z-score for the sample
    plt.axvline(z_score, color='green', linestyle='--', label=f"Sample Z-Score = {z_score}")

    # Plot the critical values
    plt.axvline(-critical_value, color='black', linestyle='--', label=f"Critical Value (-)")
    plt.axvline(critical_value, color='black', linestyle='--', label=f"Critical Value (+)")

    # Type II error visualization if true mean is different from population mean

```

```
if true_mean != population_mean:
    # Calculate the Z-score for the true mean and fill Type II error region
    z_score_type_ii = (sample_mean - true_mean) / sem
    plt.fill_between(x, y, where=(x <= critical_value) & (x >= -critical_value), color='red')

# Add labels and title
plt.title('Hypothesis Testing: Type I and Type II Errors')
plt.xlabel('Z-Score')
plt.ylabel('Density')
plt.legend(loc='best')

# Show the plot
plt.show()

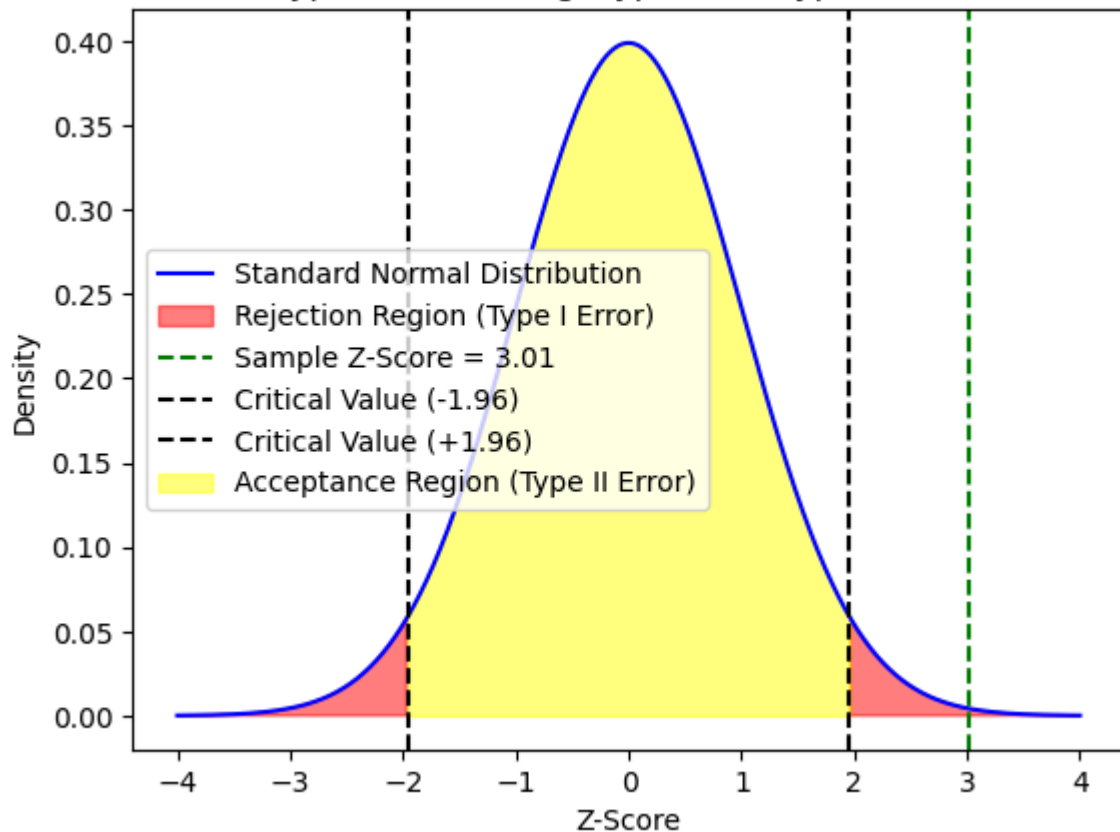
# Return results
return {
    'z_score': z_score,
    'p_value': p_value,
    'type_1_error': type_1_error,
    'type_ii_error': type_ii_error
}

# Example usage:
population_mean = 50
population_std = 8
sample_size = 50
true_mean = 52 # Change this to simulate a different true mean

results = visualize_errors(population_mean, population_std, sample_size, alpha=0.05, true_mean=true_mean)
print("Z-Score:", results['z_score'])
print("P-Value:", results['p_value'])
print("Type I Error:", results['type_1_error'])
print("Type II Error:", results['type_ii_error'])
```



Hypothesis Testing: Type I and Type II Errors



Z-Score: 3.012079666054751

P-Value: 0.002594644938979318

Type I Error: True

Type II Error: True

Write a Python program to perform an independent T-test and interpret the results@

```
import numpy as np
```

```
from scipy.stats import ttest_ind
```

```
def independent_t_test(group1, group2, alpha=0.05):
```

```
    """
```

```
    Perform an independent T-test and interpret the results.
```

```
    Parameters:
```

- group1: Array-like, data for the first sample/group.
- group2: Array-like, data for the second sample/group.
- alpha: Significance level for the test (default is 0.05).

```
    Returns:
```

- t_stat: The T-statistic
- p_value: The P-value of the test
- result: Interpretation of the hypothesis test

```
    """
```

```
    # Perform independent T-test
```

```
    t_stat, p_value = ttest_ind(group1, group2)
```

```
    # Interpret the results
```

```
    if p_value < alpha:
```

```
        result = "Reject the null hypothesis: The means of the two groups are significant"
```

```

else:
    result = "Fail to reject the null hypothesis: The means of the two groups are not

# Print the results
print("T-Statistic:", t_stat)
print("P-Value:", p_value)
print(result)

return t_stat, p_value, result

# Example usage:
# Simulating two independent samples with different means
np.random.seed(42) # Set seed for reproducibility

# Group 1: Mean = 50, Standard Deviation = 10, Sample Size = 30
group1 = np.random.normal(loc=50, scale=10, size=30)

# Group 2: Mean = 55, Standard Deviation = 10, Sample Size = 30
group2 = np.random.normal(loc=55, scale=10, size=30)

# Perform the independent T-test
t_stat, p_value, result = independent_t_test(group1, group2, alpha=0.05)

```



T-Statistic: -2.3981151520102415

P-Value: 0.019717941865758228

Reject the null hypothesis: The means of the two groups are significantly different.

```

#Simulate data and perform both Z-test and T-test, then compare the results using Python@
import numpy as np
from scipy.stats import norm, ttest_ind

```

```

def z_test(sample_data, population_mean, population_std, alpha=0.05):
    """

```

Perform a Z-test for a sample mean vs. a population mean.

Parameters:

- sample_data: Array-like, the sample data.
- population_mean: The population mean to test against.
- population_std: The population standard deviation.
- alpha: Significance level for the test (default 0.05).

Returns:

- z_score: The Z-statistic
 - p_value: The p-value for the Z-test
- """

```

sample_mean = np.mean(sample_data)
sample_size = len(sample_data)

```

```

# Standard Error of the Mean (SEM)
sem = population_std / np.sqrt(sample_size)

```

```

# Z-score calculation

```

```

z_score = (sample_mean - population_mean) / sem
p_value = 2 * (1 - norm.cdf(abs(z_score))) # Two-tailed test

return z_score, p_value

def t_test(group1, group2, alpha=0.05):
    """
    Perform an independent T-test for two groups.

    Parameters:
    - group1: Array-like, data for the first sample/group.
    - group2: Array-like, data for the second sample/group.
    - alpha: Significance level for the test (default 0.05).

    Returns:
    - t_stat: The T-statistic
    - p_value: The p-value for the T-test
    """
    t_stat, p_value = ttest_ind(group1, group2)
    return t_stat, p_value

# Simulate data for the example:
np.random.seed(42) # Set seed for reproducibility

# Group 1: Mean = 50, Standard Deviation = 10, Sample Size = 30
group1 = np.random.normal(loc=50, scale=10, size=30)

# Group 2: Mean = 55, Standard Deviation = 10, Sample Size = 30
group2 = np.random.normal(loc=55, scale=10, size=30)

# Population parameters (for Z-test)
population_mean = 50 # The population mean to test against
population_std = 10 # The population standard deviation

# Perform Z-test
z_score, z_p_value = z_test(group1, population_mean, population_std)

# Perform T-test
t_stat, t_p_value = t_test(group1, group2)

# Compare results
print("Z-Test Results:")
print("Z-Score:", z_score)
print("P-Value:", z_p_value)
print("Reject the null hypothesis" if z_p_value < 0.05 else "Fail to reject the null hypo")

print("\nT-Test Results:")
print("T-Statistic:", t_stat)
print("P-Value:", t_p_value)
print("Reject the null hypothesis" if t_p_value < 0.05 else "Fail to reject the null hypo")

```



Z-Test Results:

Z-Score: -1.0305229898218837

P-Value: 0.3027645658813325

Fail to reject the null hypothesis

T-Test Results:

T-Statistic: -2.3981151520102415

P-Value: 0.019717941865758228

Reject the null hypothesis

Write a Python function to calculate the confidence interval for a sample mean and export numpy as np

from scipy.stats import norm

```
def confidence_interval(sample_data, confidence_level=0.95):
```

```
    """
```

```
    Calculate the confidence interval for a sample mean.
```

```
    Parameters:
```

```
    - sample_data: Array-like, the sample data.
```

```
    - confidence_level: The confidence level for the interval (default is 95%).
```

```
    Returns:
```

```
    - ci_lower: The lower bound of the confidence interval.
```

```
    - ci_upper: The upper bound of the confidence interval.
```

```
    """
```

```
    # Sample statistics
```

```
    sample_mean = np.mean(sample_data)
```

```
    sample_std = np.std(sample_data, ddof=1) # Using sample standard deviation (ddof=1)
```

```
    sample_size = len(sample_data)
```

```
    # Z-score for the given confidence level
```

```
    z_score = norm.ppf(1 - (1 - confidence_level) / 2)
```

```
    # Standard error (SE)
```

```
    standard_error = sample_std / np.sqrt(sample_size)
```

```
    # Margin of error
```

```
    margin_of_error = z_score * standard_error
```

```
    # Confidence interval
```

```
    ci_lower = sample_mean - margin_of_error
```

```
    ci_upper = sample_mean + margin_of_error
```

```
    return ci_lower, ci_upper
```

```
# Example usage:
```

```
np.random.seed(42) # For reproducibility
```

```
# Simulate a sample with mean = 50, standard deviation = 10, and sample size = 30
```

```
sample_data = np.random.normal(loc=50, scale=10, size=30)
```

```
# Calculate the 95% confidence interval
```

```
ci_lower, ci_upper = confidence_interval(sample_data, confidence_level=0.95)
```

```
print(f"95% Confidence Interval: ({ci_lower:.2f}, {ci_upper:.2f})")
```


➡ 95% Confidence Interval: (44.90, 51.34)

#Write a Python program to calculate the margin of error for a given confidence level using numpy and scipy

```
import numpy as np
from scipy.stats import norm
```

```
def margin_of_error(sample_data, confidence_level=0.95):
    """
    Calculate the margin of error for a sample mean at a given confidence level.

    Parameters:
    - sample_data: Array-like, the sample data.
    - confidence_level: The confidence level for the margin of error (default is 95%).

    Returns:
    - margin_of_error: The margin of error for the sample mean.
    """
    # Sample statistics
    sample_mean = np.mean(sample_data)
    sample_std = np.std(sample_data, ddof=1) # Using sample standard deviation (ddof=1)
    sample_size = len(sample_data)

    # Z-score for the given confidence level
    z_score = norm.ppf(1 - (1 - confidence_level) / 2)

    # Standard error (SE)
    standard_error = sample_std / np.sqrt(sample_size)

    # Margin of error
    margin_of_error = z_score * standard_error

    return margin_of_error

# Example usage:
np.random.seed(42) # For reproducibility

# Simulate a sample with mean = 50, standard deviation = 10, and sample size = 30
sample_data = np.random.normal(loc=50, scale=10, size=30)

# Calculate the margin of error for a 95% confidence level
moe = margin_of_error(sample_data, confidence_level=0.95)

print(f"Margin of Error (95% confidence level): {moe:.2f}")
```

➡ Margin of Error (95% confidence level): 3.22

#Implement a Bayesian inference method using Bayes' Theorem in Python and explain the procedure

```
def bayesian_inference(prior, likelihood, evidence):
    """
    Perform Bayesian inference using Bayes' Theorem.

    Parameters:
```

- prior: The prior probability of the hypothesis ($P(H)$).
- likelihood: The likelihood of observing the evidence given the hypothesis ($P(E|H)$).
- evidence: The total probability of the evidence ($P(E)$).

Returns:

- posterior: The posterior probability of the hypothesis given the evidence ($P(H|E)$).

```

# Bayes' Theorem:  $P(H|E) = P(E|H) * P(H) / P(E)$ 
posterior = (likelihood * prior) / evidence
return posterior

```

Given values for the disease example:

$P_H = 0.01$ # Prior probability of having the disease ($P(H)$)

$P_{E_given_H} = 0.95$ # Likelihood of a positive test given the person has the disease ($P(E|H)$)

$P_{E_given_not_H} = 0.05$ # Likelihood of a positive test given the person does not have the disease ($P(E|\neg H)$)

Probability of evidence ($P(E)$):

$P(E) = P(E|H) * P(H) + P(E|\neg H) * P(\neg H)$


$P_E = (P_{E_given_H} * P_H) + (P_{E_given_not_H} * (1 - P_H))$

Calculate the posterior probability ($P(H|E)$)

$P_{H_given_E} = \text{bayesian_inference}(P_H, P_{E_given_H}, P_E)$

Print the result

`print(f"Posterior probability $P(H|E)$: {P_H_given_E:.4f}")`

 Posterior probability $P(H|E)$: 0.1610

Perform a Chi-square test for independence between two categorical variables in Python

`import numpy as np`

`import pandas as pd`

`from scipy.stats import chi2_contingency`

Sample data: A contingency table of observed counts

For example, data about preferences for two types of beverages (Tea vs Coffee) across two genders (Male vs Female)

```

data = {
    'Tea': [30, 10], # Tea preferences for Male and Female
    'Coffee': [20, 40] # Coffee preferences for Male and Female
}

```

Convert the data into a pandas DataFrame (contingency table)

`df = pd.DataFrame(data, index=['Male', 'Female'])`

Perform the Chi-square test for independence

`chi2, p, dof, expected = chi2_contingency(df)`

Output the results

`print(f"Chi-square statistic: {chi2:.4f}")`

`print(f"P-value: {p:.4f}")`

`print(f"Degrees of freedom: {dof}")`

`print("Expected counts table:")`

`print(expected)`

```
# Interpretation
if p < 0.05:
    print("The result is statistically significant. We reject the null hypothesis of inde
else:
    print("The result is not statistically significant. We fail to reject the null hypoth
```

```
⇒ Chi-square statistic: 15.0417
P-value: 0.0001
Degrees of freedom: 1
Expected counts table:
[[20. 30.]
 [20. 30.]]
The result is statistically significant. We reject the null hypothesis of independenc
```

#Write a Python program to calculate the expected frequencies for a Chi-square test based
import numpy as np

```
def calculate_expected_frequencies(observed_data):
    """
    Calculate the expected frequencies for a Chi-square test based on observed data.

    Parameters:
    - observed_data: A 2D numpy array or list representing the observed frequencies in a

    Returns:
    - expected: A 2D numpy array representing the expected frequencies.
    """
    # Convert the observed data to a numpy array
    observed_data = np.array(observed_data)

    # Calculate the row sums and column sums
    row_sums = np.sum(observed_data, axis=1)
    col_sums = np.sum(observed_data, axis=0)

    # Calculate the total sum (N)
    total_sum = np.sum(observed_data)

    # Calculate expected frequencies for each cell in the contingency table
    expected = np.outer(row_sums, col_sums) / total_sum

    return expected

# Example observed data (2x2 contingency table)
# For example, data about preferences for two types of beverages (Tea vs Coffee) across t
observed_data = [
    [30, 20], # Male: 30 prefer Tea, 20 prefer Coffee
    [10, 40]  # Female: 10 prefer Tea, 40 prefer Coffee
]

# Calculate the expected frequencies
expected_frequencies = calculate_expected_frequencies(observed_data)
```

```
# Print the observed and expected frequencies
print("Observed Frequencies:")
print(np.array(observed_data))
print("\nExpected Frequencies:")
print(expected_frequencies)
```

```
➞ Observed Frequencies:
[[30 20]
 [10 40]]
```

```
Expected Frequencies:
[[20. 30.]
 [20. 30.]]
```

```
#Perform a goodness-of-fit test using Python to compare the observed data to an expected
import numpy as np
from scipy.stats import chisquare
```

```
# Example observed data (frequencies of outcomes in a dice roll experiment)
observed_data = [15, 18, 22, 19, 13, 13] # Observed frequencies for 6 outcomes (1-6)
```

```
# Expected frequencies for a fair die (uniform distribution: each outcome should occur wi
# Assuming we rolled the die 100 times, the expected frequency for each outcome is 100 /
expected_data = [100 / 6] * 6 # Expected frequencies for a fair die
```

```
# Perform the Chi-square goodness-of-fit test
chi2_stat, p_value = chisquare(observed_data, expected_data)
```

```
# Output the results
print(f"Chi-square statistic: {chi2_stat:.4f}")
print(f"P-value: {p_value:.4f}")
```

```
# Interpretation of the result
if p_value < 0.05:
    print("The result is statistically significant. We reject the null hypothesis that th
else:
    print("The result is not statistically significant. We fail to reject the null hypoth
```

```
➞ Chi-square statistic: 3.9200
P-value: 0.5610
The result is not statistically significant. We fail to reject the null hypothesis th
```

```
#Create a Python script to simulate and visualize the Chi-square distribution and discuss
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2
```

```
# Function to simulate and visualize the Chi-square distribution
def plot_chi_square_distribution(degrees_of_freedom, num_samples=10000):
    """
    Simulate and plot the Chi-square distribution for a given degrees of freedom.
```

```
Parameters:
- degrees_of_freedom: The degrees of freedom for the Chi-square distribution.
- num_samples: The number of random samples to generate (default is 10000).
"""

# Generate random samples from the Chi-square distribution
chi_square_samples = np.random.chisquare(degrees_of_freedom, num_samples)

# Plot the histogram of the samples
plt.figure(figsize=(8, 6))
plt.hist(chi_square_samples, bins=50, density=True, alpha=0.6, color='g', label=f'df

# Plot the theoretical Chi-square distribution using scipy.stats
x = np.linspace(0, np.max(chi_square_samples), 1000)
plt.plot(x, chi2.pdf(x, degrees_of_freedom), 'r-', lw=2, label='Theoretical Chi-squar

# Add titles and labels
plt.title(f"Chi-square Distribution (df = {degrees_of_freedom})", fontsize=16)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Density', fontsize=12)
plt.legend(loc='upper right')
plt.grid(True)

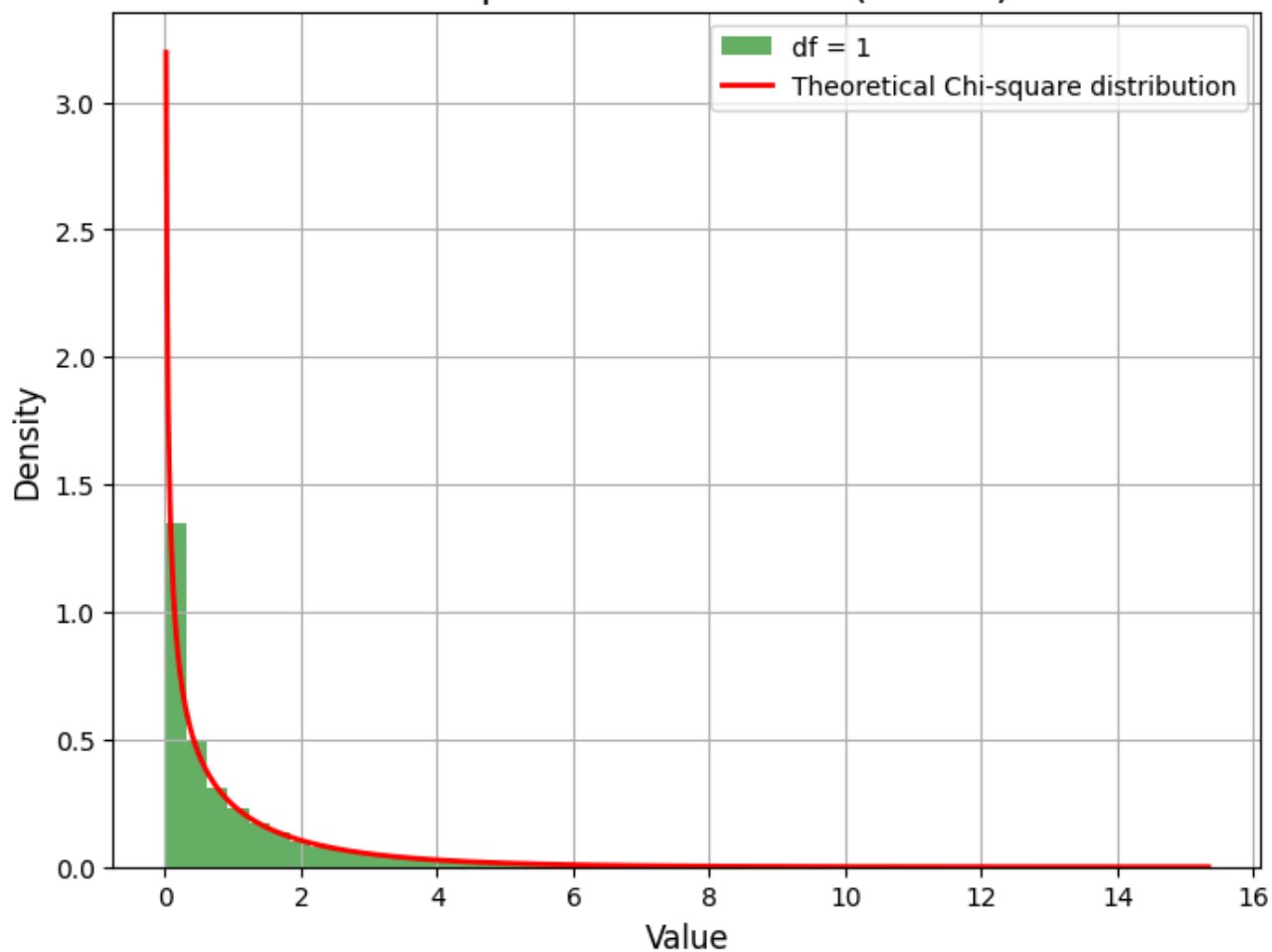
# Show the plot
plt.show()

# Simulate and visualize the Chi-square distribution for different degrees of freedom
df_values = [1, 2, 5, 10] # Different degrees of freedom to explore

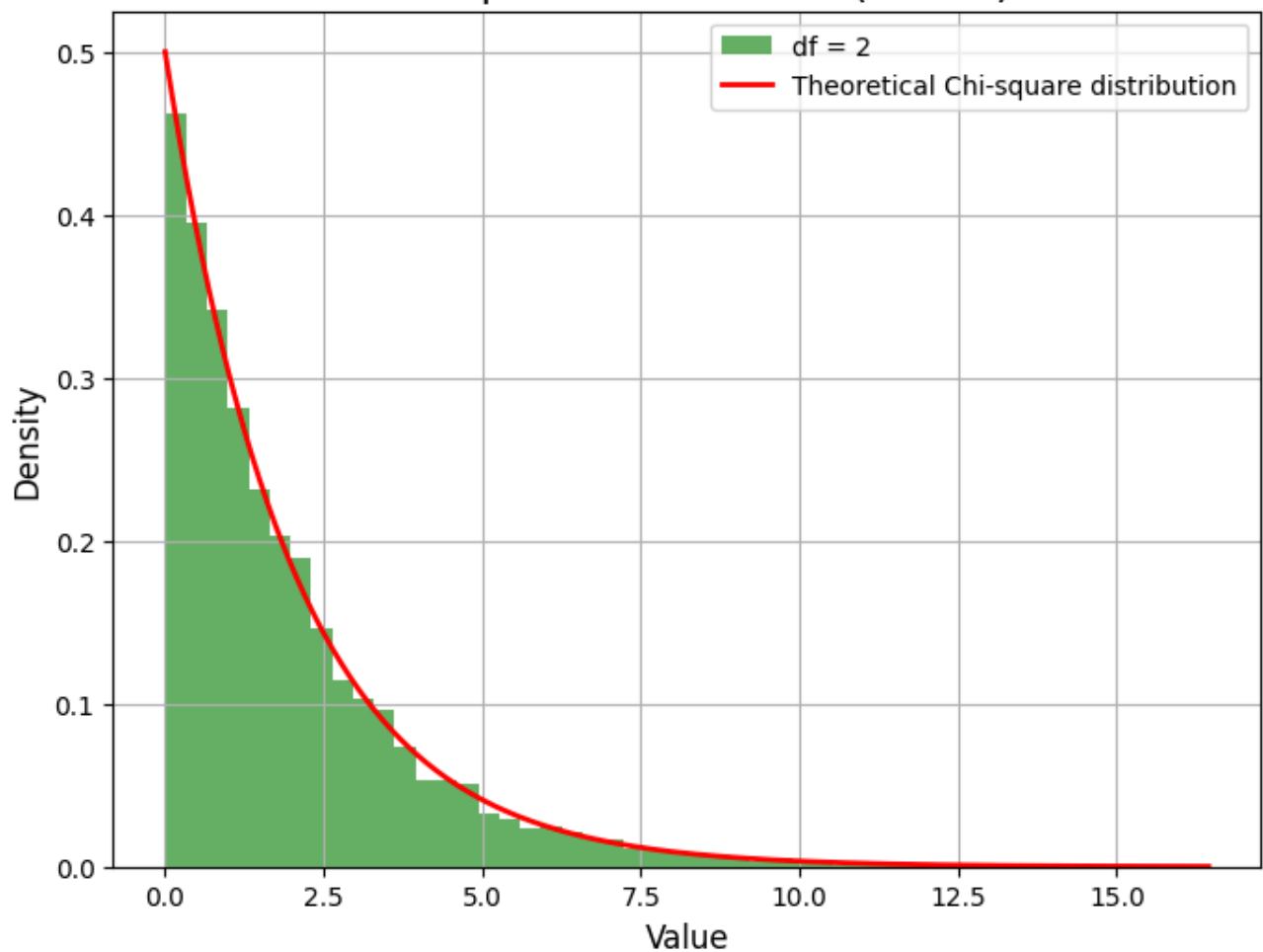
for df in df_values:
    plot_chi_square_distribution(df)
```



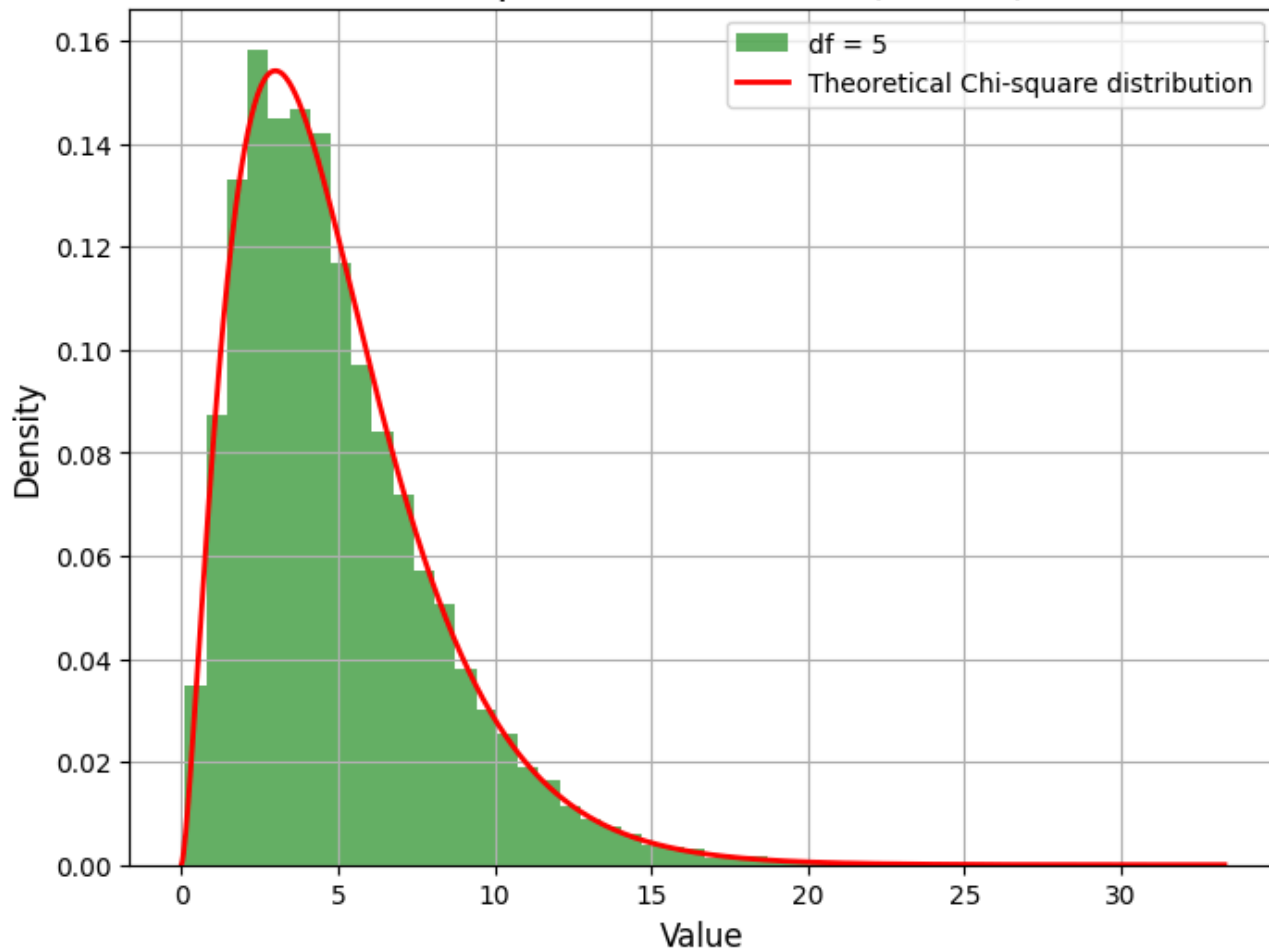
Chi-square Distribution (df = 1)



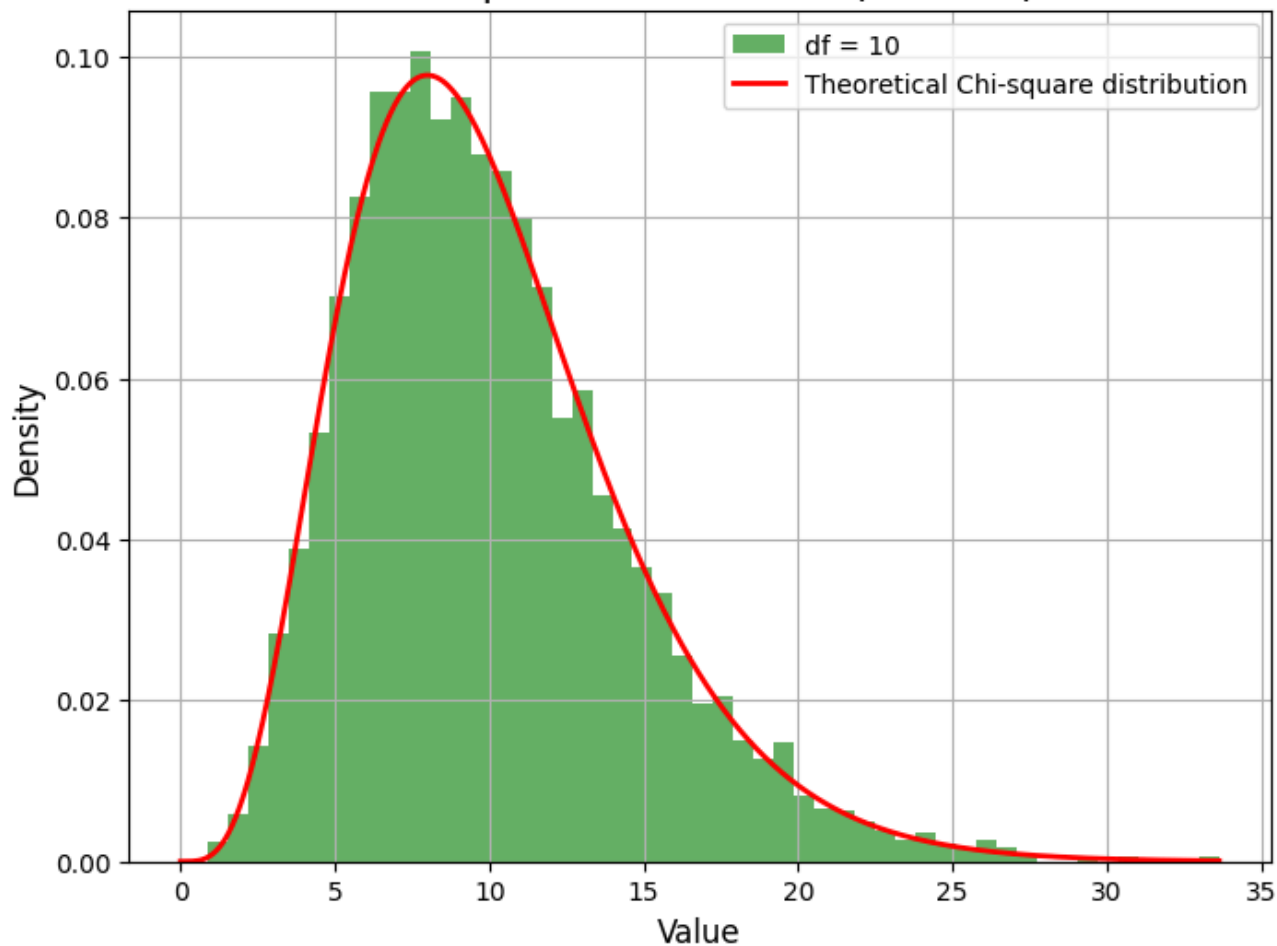
Chi-square Distribution (df = 2)



Chi-square Distribution (df = 5)



Chi-square Distribution (df = 10)




```

#D Implement an F-test using Python to compare the variances of two random samples
import numpy as np
from scipy.stats import f

# Function to perform the F-test and calculate the p-value
def f_test(sample1, sample2):
    # Calculate the sample variances
    var1 = np.var(sample1, ddof=1) # ddof=1 to get sample variance
    var2 = np.var(sample2, ddof=1)

    # Calculate the F-statistic (larger variance in the numerator)
    if var1 > var2:
        f_statistic = var1 / var2
        df1 = len(sample1) - 1 # degrees of freedom for sample1
        df2 = len(sample2) - 1 # degrees of freedom for sample2
    else:
        f_statistic = var2 / var1
        df1 = len(sample2) - 1
        df2 = len(sample1) - 1

    # Calculate the p-value using the F-distribution's cumulative distribution function
    p_value = 1 - f.cdf(f_statistic, df1, df2)

    return f_statistic, p_value

# Simulating two random samples
np.random.seed(42) # For reproducibility
sample1 = np.random.normal(loc=50, scale=10, size=100) # Sample 1 with mean=50, std=10
sample2 = np.random.normal(loc=50, scale=15, size=100) # Sample 2 with mean=50, std=15

# Perform the F-test
f_statistic, p_value = f_test(sample1, sample2)

# Output the results
print(f"F-statistic: {f_statistic:.4f}")
print(f"P-value: {p_value:.4f}")

# Interpretation
alpha = 0.05 # Significance level
if p_value < alpha:
    print("The result is statistically significant. We reject the null hypothesis and con
else:
    print("The result is not statistically significant. We fail to reject the null hypoth

⇒ F-statistic: 2.4811
P-value: 0.0000
The result is statistically significant. We reject the null hypothesis and conclude t

```

```

#Write a Python program to perform an ANOVA test to compare means between multiple groups
#interpret the resultsD
import numpy as np
import scipy.stats as stats

```

```

# Function to perform ANOVA and interpret the results
def anova_test(*groups):
    """
    Perform an ANOVA test to compare means between multiple groups.

    Parameters:
    - groups: Variable length argument for the groups (arrays or lists).

    Returns:
    - f_statistic: The computed F-statistic.
    - p_value: The computed p-value.
    """
    # Perform ANOVA using scipy.stats.f_oneway
    f_statistic, p_value = stats.f_oneway(*groups)

    return f_statistic, p_value

# Example data for multiple groups
group1 = [23, 21, 19, 20, 22, 23, 24]
group2 = [30, 32, 29, 31, 30, 33, 34]
group3 = [35, 36, 37, 36, 35, 34, 38]

# Perform the ANOVA test
f_statistic, p_value = anova_test(group1, group2, group3)

# Output the results
print(f"F-statistic: {f_statistic:.4f}")
print(f"P-value: {p_value:.4f}")

# Interpretation of the result
alpha = 0.05 # Significance level
if p_value < alpha:
    print("The result is statistically significant. We reject the null hypothesis and con
else:
    print("The result is not statistically significant. We fail to reject the null hypoth

```



F-statistic: 132.0172

P-value: 0.0000

The result is statistically significant. We reject the null hypothesis and conclude t



```

#Perform a one-way ANOVA test using Python to compare the means of different groups and p
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt

```

```

# Function to perform one-way ANOVA and plot the results
def perform_anova_and_plot(*groups):
    """

```

Perform a one-way ANOVA and visualize the results using a box plot.

Parameters:

- groups: Variable length argument for the groups (arrays or lists).

Returns:

- f_statistic: The computed F-statistic.
- p_value: The computed p-value.

"""

Perform one-way ANOVA

f_statistic, p_value = stats.f_oneway(*groups)

Plot boxplot to visualize the distributions of the groups

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

plt.boxplot(groups, vert=True, patch_artist=True,
 boxprops=dict(facecolor='skyblue', color='black'),
 whiskerprops=dict(color='black', linewidth=1.5),
 capprops=dict(color='black', linewidth=1.5),
 flierprops=dict(markerfacecolor='red', marker='o', markersize=8))

plt.title("One-Way ANOVA: Comparison of Group Means", fontsize=16)

plt.xlabel("Groups", fontsize=12)

plt.ylabel("Values", fontsize=12)

plt.xticks([1, 2, 3], ['Group 1', 'Group 2', 'Group 3'])

plt.grid(True)

plt.show()

return f_statistic, p_value

Example data for multiple groups

group1 = [23, 21, 19, 20, 22, 23, 24]

group2 = [30, 32, 29, 31, 30, 33, 34]

group3 = [35, 36, 37, 36, 35, 34, 38]

Perform the ANOVA test and visualize the results

f_statistic, p_value = perform_anova_and_plot(group1, group2, group3)

Output the results

print(f"F-statistic: {f_statistic:.4f}")

print(f"P-value: {p_value:.4f}")

Interpretation of the result

alpha = 0.05 # Significance level

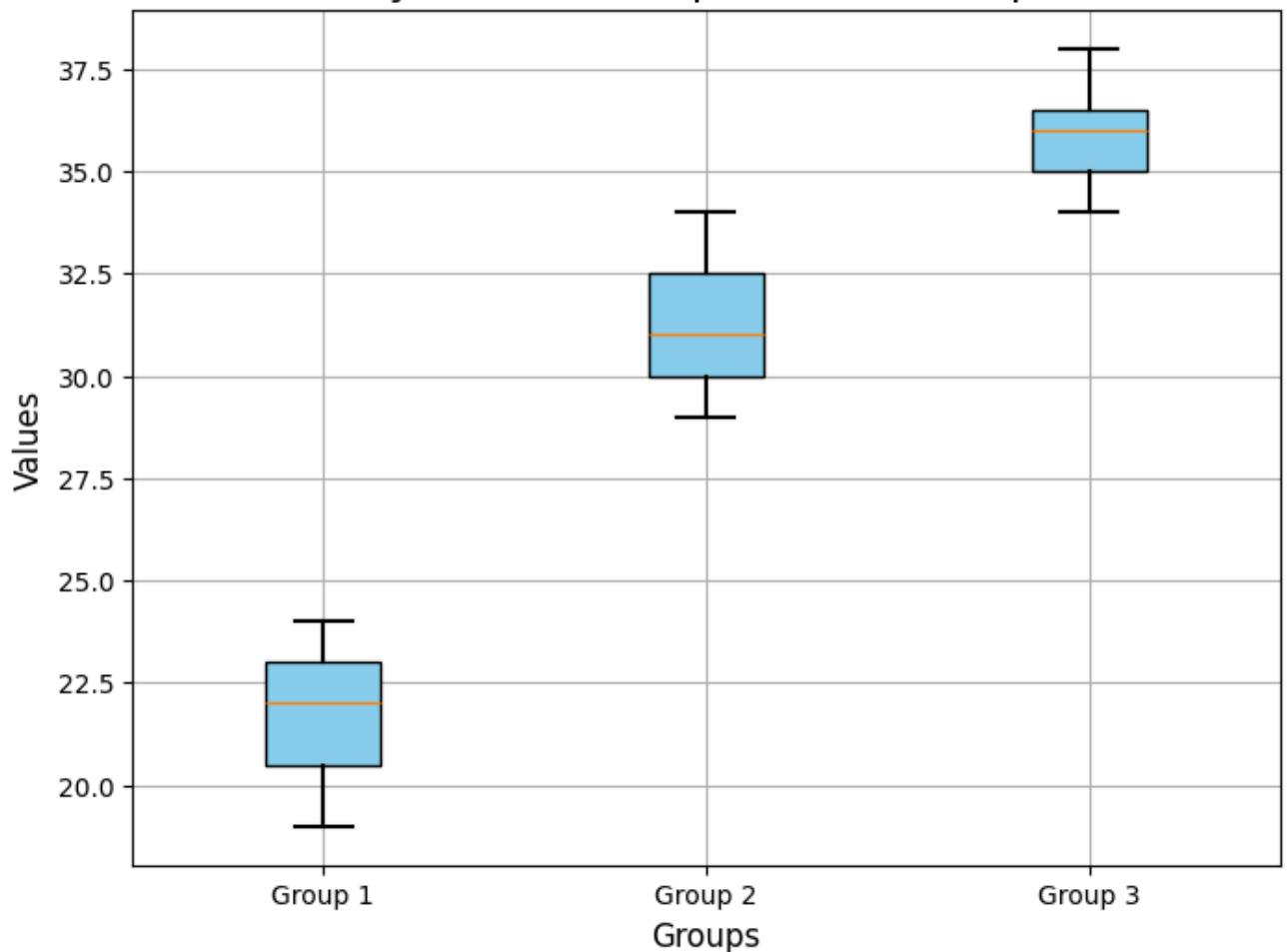
if p_value < alpha:

 print("The result is statistically significant. We reject the null hypothesis and con
else:

 print("The result is not statistically significant. We fail to reject the null hypoth



One-Way ANOVA: Comparison of Group Means



F-statistic: 132.0172

P-value: 0.0000

The result is statistically significant. We reject the null hypothesis and conclude t

#Write a Python function to check the assumptions (normality, independence, and equal var

```
import numpy as np
```

```
import scipy.stats as stats
```

Function to check the assumptions for ANOVA

```
def check_anova_assumptions(*groups):
```

```
    """
```

Check the assumptions for performing an ANOVA test: normality, equal variance, and in

Parameters:

- groups: Variable length argument for the groups (arrays or lists).

Returns:

- normality_tests: Results of normality tests for each group.

- levene_test: Result of the Levene's test for equal variances.

```
    """
```

1. Test for normality using Shapiro-Wilk test

```
normality_tests = {}
```

```
for i, group in enumerate(groups):
```

```
    stat, p_value = stats.shapiro(group)
```

```
    normality_tests[f"Group {i+1} Normality Test (Shapiro-Wilk)"] = (stat, p_value)
```

```

# 2. Test for equal variances using Levene's test
stat, p_value = stats.levene(*groups)
levене_test = (stat, p_value)

# Print the results of normality tests
print("Normality Test Results (Shapiro-Wilk):")
for group, result in normality_tests.items():
    print(f"{group} - Statistic: {result[0]:.4f}, p-value: {result[1]:.4f}")

# Print the result of Levene's test for equal variances
print("\nLevene's Test for Equal Variance:")
print(f"Statistic: {levене_test[0]:.4f}, p-value: {levене_test[1]:.4f}")

# Return the test results
return normality_tests, levене_test

# Example data for multiple groups
group1 = [23, 21, 19, 20, 22, 23, 24]
group2 = [30, 32, 29, 31, 30, 33, 34]
group3 = [35, 36, 37, 36, 35, 34, 38]


# Check assumptions for ANOVA
normality_tests, levене_test = check_anova_assumptions(group1, group2, group3)

# Interpretation of the results:
alpha = 0.05 # Significance level for the tests

# Check normality assumption
for group, result in normality_tests.items():
    if result[1] < alpha:
        print(f"Reject the null hypothesis for normality in {group}. The group is not nor
    else:
        print(f"Fail to reject the null hypothesis for normality in {group}. The group is

# Check equal variance assumption (Levene's Test)
if levене_test[1] < alpha:
    print("\nLevene's test indicates unequal variances between the groups. The assumption
else:
    print("\nLevene's test indicates equal variances between the groups. The assumption o


```

 Normality Test Results (Shapiro-Wilk):
 Group 1 Normality Test (Shapiro-Wilk) - Statistic: 0.9547, p-value: 0.7724
 Group 2 Normality Test (Shapiro-Wilk) - Statistic: 0.9547, p-value: 0.7724
 Group 3 Normality Test (Shapiro-Wilk) - Statistic: 0.9666, p-value: 0.8733

Levene's Test for Equal Variance:
 Statistic: 0.5000, p-value: 0.6147
 Fail to reject the null hypothesis for normality in Group 1 Normality Test (Shapiro-w
 Fail to reject the null hypothesis for normality in Group 2 Normality Test (Shapiro-w
 Fail to reject the null hypothesis for normality in Group 3 Normality Test (Shapiro-w

Levene's test indicates equal variances between the groups. The assumption of equal v

```

#Perform a two-way ANOVA test using Python to study the interaction between two factors a
#resultsD
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
import matplotlib.pyplot as plt
import seaborn as sns

# Function to perform two-way ANOVA and visualize the results
def two_way_anova(data, dependent_var, factor1, factor2):
    """
    Perform a two-way ANOVA to study the interaction between two factors.

    Parameters:
    - data: DataFrame containing the dependent variable and factors.
    - dependent_var: The name of the dependent variable.
    - factor1: The name of the first factor.
    - factor2: The name of the second factor.

    Returns:
    - anova_results: The ANOVA table with the F-statistics and p-values.
    """
    # Fit the model using OLS (ordinary least squares) regression
    model = ols(f'{dependent_var} ~ C({factor1}) * C({factor2})', data=data).fit()

    # Perform the ANOVA test
    anova_table = sm.stats.anova_lm(model, typ=2)

    # Print the results of the ANOVA
    print(anova_table)

    # Visualize the interaction between the factors
    sns.set(style="whitegrid")
    plt.figure(figsize=(8, 6))
    sns.interactionplot(data=data, x=factor1, trace=factor2, response=dependent_var,
                        markers=["o", "s"], colors=["blue", "green"])
    plt.title('Interaction Plot between Factors')
    plt.xlabel(factor1)
    plt.ylabel(dependent_var)
    plt.show()

    return anova_table

# Example Data for Two-Way ANOVA:
# Simulated data with two factors and one dependent variable
np.random.seed(42)

# Generate sample data
factor1 = ['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C']
factor2 = ['X', 'Y', 'Z', 'X', 'Y', 'Z', 'X', 'Y', 'Z']
dependent_var = [23, 27, 25, 29, 31, 30, 26, 28, 27] # Dependent variable (e.g., outcome)

# Create a DataFrame

```

```

data = pd.DataFrame({
    'Factor1': factor1,
    'Factor2': factor2,
    'Outcome': dependent_var
})

# Perform the two-way ANOVA and visualize the results
anova_results = two_way_anova(data, 'Outcome', 'Factor1', 'Factor2')

# Interpretation of results based on p-values:
alpha = 0.05 # Significance level
for factor in anova_results.index:
    p_value = anova_results.loc[factor, 'PR(>F)']
    if p_value < alpha:
        print(f"Reject the null hypothesis for {factor}. There is a significant effect.")
    else:
        print(f"Fail to reject the null hypothesis for {factor}. There is no significant

```

→ /usr/local/lib/python3.11/dist-packages/statsmodels/regression/linear_model.py:1717:
return np.dot(wresid, wresid) / self.df_resid

ValueError Traceback (most recent call last)
<ipython-input-21-71ad98134d99> in <cell line: 0>()
60
61 # Perform the two-way ANOVA and visualize the results
---> 62 anova_results = two_way_anova(data, 'Outcome', 'Factor1', 'Factor2')
63
64 # Interpretation of results based on p-values:

----- 5 frames -----
/usr/local/lib/python3.11/dist-packages/numpy/lib/function_base.py in
asarray_chkfinite(a, dtype, order)
628 a = asarray(a, dtype=dtype, order=order)
629 if a.dtype.char in typecodes['AllFloat'] and not np.isfinite(a).all():
--> 630 raise ValueError(
631 "array must not contain infs or NaNs")
632 return a

ValueError: array must not contain infs or NaNs

Next steps: [Explain error](#)

```

#Write a Python program to visualize the F-distribution and discuss its use in hypothesis
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import f

# Function to plot F-distribution for various degrees of freedom
def plot_f_distribution(df1_values, df2_values, x_range=(0, 5), num_points=1000):
    """
    Visualize the F-distribution for different degrees of freedom (df1, df2).

    Parameters:
    - df1_values: List of numerator degrees of freedom (df1) to visualize.

```

```
- df2_values: List of denominator degrees of freedom (df2) to visualize.
- x_range: Range of x values for the plot (default: 0 to 5).
- num_points: Number of points for plotting (default: 1000).
"""

x = np.linspace(x_range[0], x_range[1], num_points)

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

# Plot F-distribution for various df1 and df2 combinations
for df1 in df1_values:
    for df2 in df2_values:
        y = f.pdf(x, df1, df2)
        plt.plot(x, y, label=f'df1={df1}, df2={df2}')

# Adding title, labels, and legend
plt.title("F-Distribution", fontsize=16)
plt.xlabel("F-Statistic", fontsize=12)
plt.ylabel("Density", fontsize=12)
plt.legend(title="Degrees of Freedom", loc="upper right")
plt.grid(True)

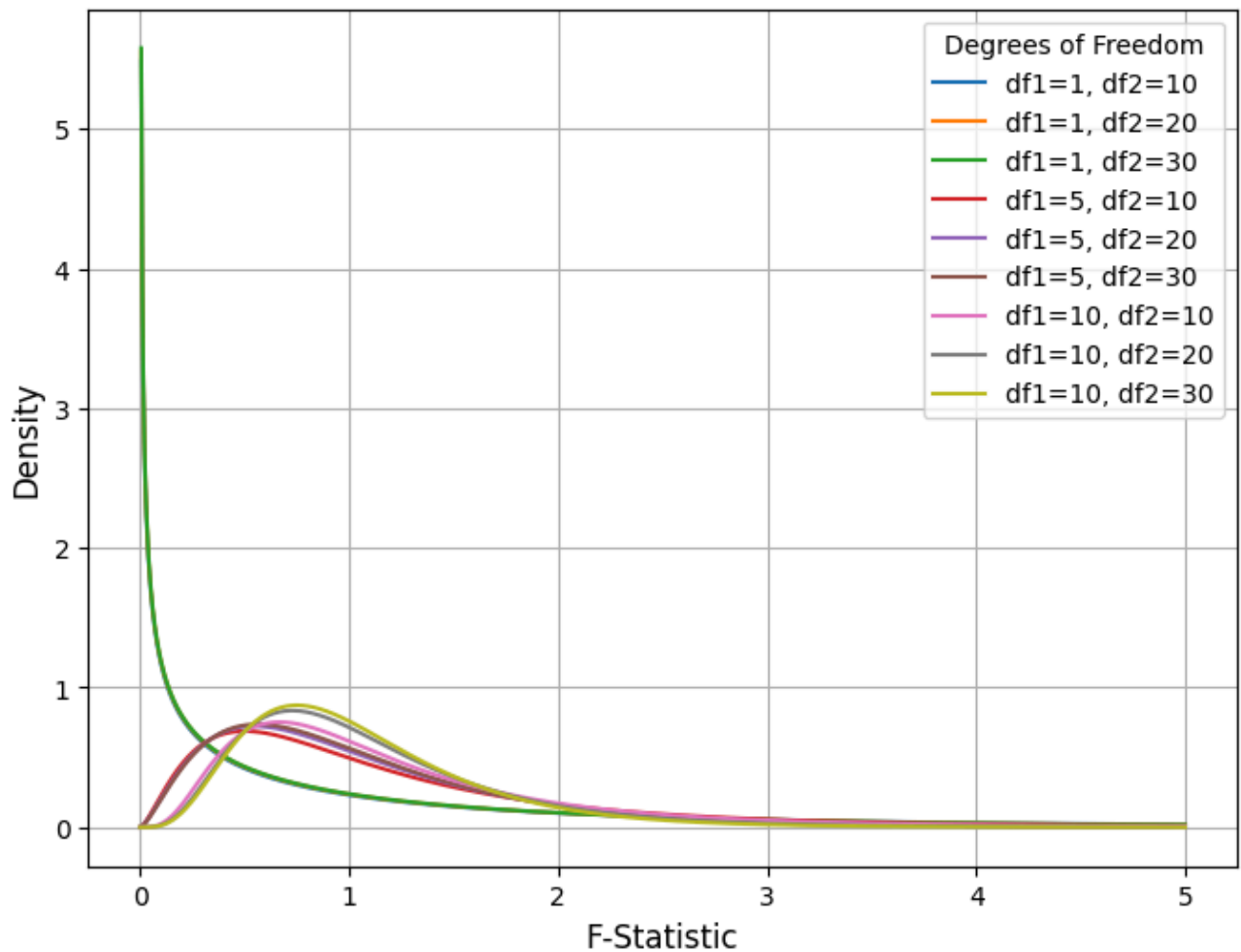
# Show the plot
plt.show()

# Example: Visualizing F-distribution for various degrees of freedom
df1_values = [1, 5, 10] # Numerator degrees of freedom
df2_values = [10, 20, 30] # Denominator degrees of freedom

plot_f_distribution(df1_values, df2_values)
```




F-Distribution



#D Perform a one-way ANOVA test in Python and visualize the results with boxplots to comp

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import f_oneway
```

```
# Generate sample data for three groups (group1, group2, group3)
np.random.seed(42)
group1 = np.random.normal(50, 10, 30) # Mean = 50, Std = 10, n = 30
group2 = np.random.normal(55, 10, 30) # Mean = 55, Std = 10, n = 30
group3 = np.random.normal(60, 10, 30) # Mean = 60, Std = 10, n = 30
```

```
# Perform One-Way ANOVA
f_stat, p_value = f_oneway(group1, group2, group3)
```

```
# Display the results
print(f"F-statistic: {f_stat}")
print(f"P-value: {p_value}")
```

```
# Visualize the results with boxplots
plt.figure(figsize=(8, 6))
sns.boxplot(data=[group1, group2, group3], notch=True, patch_artist=True)
plt.xticks([0, 1, 2], ['Group 1', 'Group 2', 'Group 3'])
plt.title('Boxplot of Group Means')
```

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

```
# Interpretation based on p-value
```

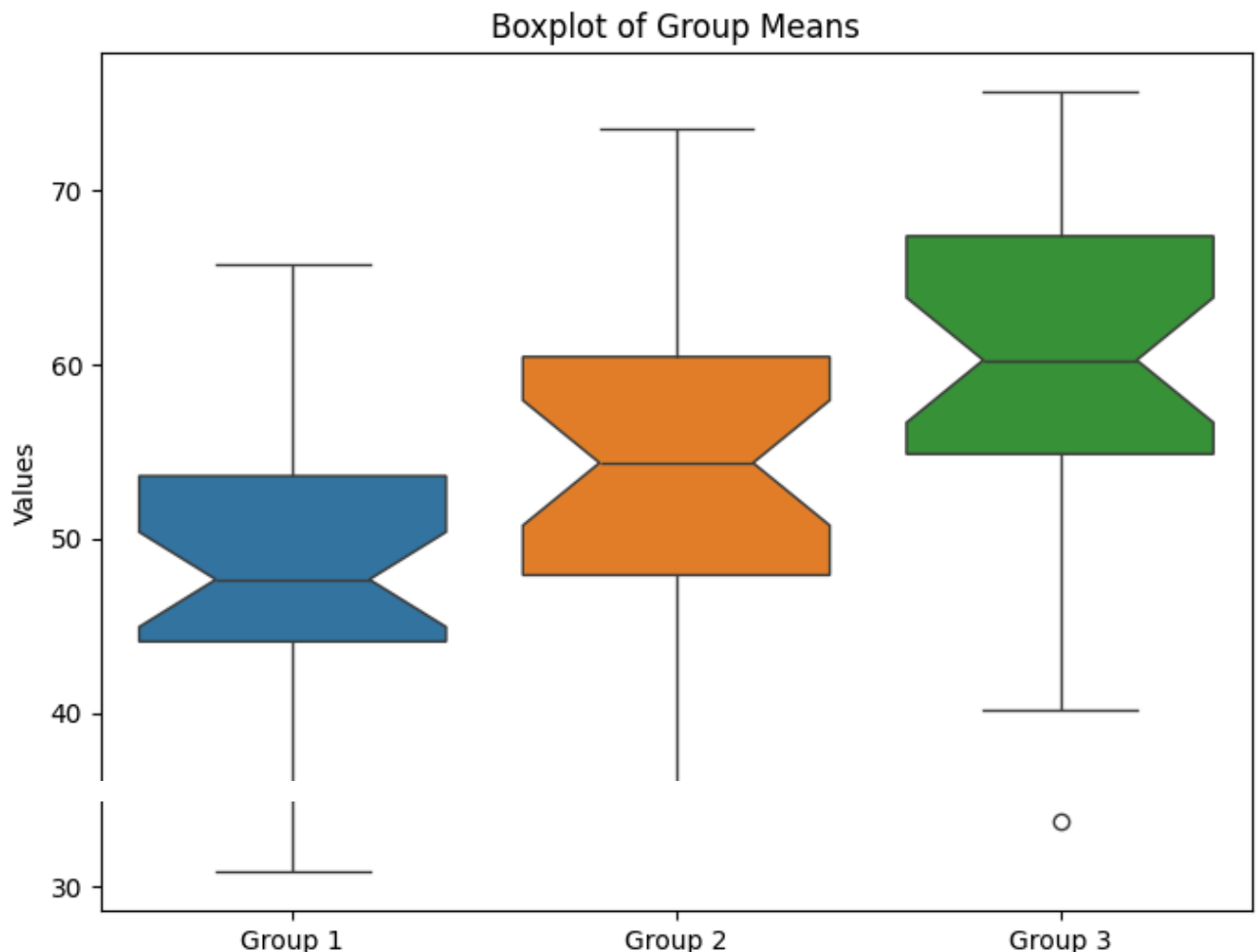
```
alpha = 0.05
```

```
if p_value < alpha:
```

```
    print("Reject the null hypothesis: There is a significant difference between the grou  
else:
```

```
    print("Fail to reject the null hypothesis: There is no significant difference between
```

```
⇒ F-statistic: 12.20952551797281  
P-value: 2.1200748140507065e-05
```



Reject the null hypothesis: There is a significant difference between the group means

```
#Simulate random data from a normal distribution, then perform hypothesis testing to eval  
import numpy as np  
from scipy.stats import ttest_ind
```

```
# Step 1: Simulate random data from a normal distribution
```

```
np.random.seed(42) # Set seed for reproducibility
```

```
group1 = np.random.normal(loc=50, scale=10, size=100) # Group 1: Mean = 50, Std = 10
```

```
group2 = np.random.normal(loc=55, scale=10, size=100) # Group 2: Mean = 55, Std = 10
```

```
# Step 2: Perform a two-sample t-test to compare the means
```

```
t_stat, p_value = ttest_ind(group1, group2)
```

```
# Step 3: Display the results
print(f"T-statistic: {t_stat}")
print(f"P-value: {p_value}")

# Interpretation based on p-value
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference between the grou
else:
    print("Fail to reject the null hypothesis: There is no significant difference between

T-statistic: -4.7546959435052765
P-value: 3.8191352626795345e-06
Reject the null hypothesis: There is a significant difference between the group means
```

```
#D Perform a hypothesis test for population variance using a Chi-square distribution and
import numpy as np
from scipy.stats import chi2

# Step 1: Simulate random sample data (assuming the population variance is 25)
np.random.seed(42)
sample_data = np.random.normal(loc=50, scale=np.sqrt(25), size=30) # Mean = 50, Variance

# Step 2: Calculate sample variance
sample_variance = np.var(sample_data, ddof=1)

# Step 3: Set the population variance under the null hypothesis
population_variance_0 = 25 # The population variance we are testing against

# Step 4: Calculate the test statistic
n = len(sample_data)
chi_square_statistic = (n - 1) * sample_variance / population_variance_0

# Step 5: Perform the Chi-square test
# For a two-tailed test, we need the critical values at alpha = 0.05 for a two-tailed tes
alpha = 0.05
df = n - 1 # Degrees of freedom

# Get the critical values for a two-tailed test
critical_value_lower = chi2.ppf(alpha / 2, df)
critical_value_upper = chi2.ppf(1 - alpha / 2, df)

# Step 6: Compare the test statistic with the critical values
print(f"Chi-square Statistic: {chi_square_statistic}")
print(f"Critical Value Lower: {critical_value_lower}")
print(f"Critical Value Upper: {critical_value_upper}")

# Step 7: Interpret the results
if chi_square_statistic < critical_value_lower or chi_square_statistic > critical_value_u
    print("Reject the null hypothesis: The population variance is significantly different
else:
```

```
print("Fail to reject the null hypothesis: The population variance is not significant")
```

```

Chi-square Statistic: 23.4903355116039
Critical Value Lower: 16.04707169536489
Critical Value Upper: 45.72228580417452
Fail to reject the null hypothesis: The population variance is not significantly diff

```

```

#Write a Python script to perform a Z-test for comparing proportions between two datasets
import numpy as np
from scipy.stats import norm

```

```

# Step 1: Define sample data
# For example, we have two groups: Group 1 and Group 2
# Group 1: 80 successes out of 200 trials
# Group 2: 60 successes out of 150 trials

```

```

x1 = 80 # Number of successes in Group 1
n1 = 200 # Number of trials in Group 1

```

```

x2 = 60 # Number of successes in Group 2
n2 = 150 # Number of trials in Group 2

```

```

# Step 2: Calculate sample proportions
p1 = x1 / n1 # Proportion of successes in Group 1
p2 = x2 / n2 # Proportion of successes in Group 2

```

```

# Step 3: Calculate the pooled sample proportion
p = (x1 + x2) / (n1 + n2)

```

```

# Step 4: Calculate the standard error of the difference in proportions
se = np.sqrt(p * (1 - p) * (1 / n1 + 1 / n2))

```

```

# Step 5: Calculate the Z-test statistic
z_stat = (p1 - p2) / se

```

```

# Step 6: Calculate the p-value for a two-tailed test
p_value = 2 * (1 - norm.cdf(abs(z_stat))) # Two-tailed test

```

```

# Step 7: Display the results
print(f"Z-statistic: {z_stat}")
print(f"P-value: {p_value}")

```

```

# Step 8: Interpretation based on p-value
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: The proportions are significantly different.")
else:
    print("Fail to reject the null hypothesis: The proportions are not significantly diff

```

```

Z-statistic: 0.0
P-value: 1.0
Fail to reject the null hypothesis: The proportions are not significantly different.

```

```

#Implement an F-test for comparing the variances of two datasets, then interpret and visu
import numpy as np
from scipy.stats import f
import matplotlib.pyplot as plt

# Step 1: Simulate two datasets with different variances
np.random.seed(42)

# Group 1: Sample with variance 25 (std = 5)
group1 = np.random.normal(loc=50, scale=5, size=100)

# Group 2: Sample with variance 36 (std = 6)
group2 = np.random.normal(loc=50, scale=6, size=100)

# Step 2: Calculate the sample variances
var1 = np.var(group1, ddof=1)
var2 = np.var(group2, ddof=1)

# Step 3: Calculate the F-statistic
F_statistic = var1 / var2 # F-statistic as the ratio of variances
print(f"Calculated F-statistic: {F_statistic}")

# Step 4: Perform the F-test and calculate the p-value
# Degrees of freedom for each sample
df1 = len(group1) - 1 # Degrees of freedom for group1
df2 = len(group2) - 1 # Degrees of freedom for group2

# p-value for a two-tailed test using the F-distribution
p_value = 2 * min(f.cdf(F_statistic, df1, df2), 1 - f.cdf(F_statistic, df1, df2))

# Step 5: Display results
print(f"F-statistic: {F_statistic}")
print(f"P-value: {p_value}")

# Step 6: Visualize the distributions of the two groups and F-distribution
plt.figure(figsize=(12, 6))

# Plot the histograms of the two groups
plt.subplot(1, 2, 1)
plt.hist(group1, bins=20, alpha=0.6, label='Group 1 (Var=25)', color='blue')
plt.hist(group2, bins=20, alpha=0.6, label='Group 2 (Var=36)', color='orange')
plt.legend()
plt.title('Histograms of Group 1 and Group 2')

# Plot the F-distribution
x = np.linspace(0, 4, 1000)
y = f.pdf(x, df1, df2)
plt.subplot(1, 2, 2)
plt.plot(x, y, label=f'F-distribution (df1={df1}, df2={df2})', color='green')
plt.axvline(x=F_statistic, color='red', linestyle='--', label=f'F-statistic: {F_statistic}')
plt.legend()
plt.title('F-distribution and F-statistic')

```

```
plt.tight_layout()
plt.show()
```

```
# Step 7: Interpret the result based on p-value
```

```
alpha = 0.05
```

```
if p_value < alpha:
```

```
    print("Reject the null hypothesis: The variances of the two groups are significantly  
else:
```

```
    print("Fail to reject the null hypothesis: The variances of the two groups are not si
```

```
⇒ Calculated F-statistic: 0.6297599051193526  
F-statistic: 0.6297599051193526  
P-value: 0.022335339458168898
```

