

Hypothesis test for one population proportion

Requirements:

Each observation should be independent of other

Random sample with replacement, but if sample is without replacement should not be more than 10% of the population

The data contains only two categories, such as pass/fail or yes/no

The data should have at least 10 successes and at least 10 failures for normal approximation

$n \cdot p \geq 10$; $n \cdot q \geq 10$

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, H_0 (contains equal sign) and H_1

Step 2: Define significance level (alpha)

Step 3: Calculate Z-test statistic

$$Z_t = \frac{(\hat{p} - p)}{\sqrt{p \cdot q / n}}$$

\hat{p} - sample proportion of success

p and q are hypothetical values (success and failure)

$q = 1 - p$

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put Z-critical value with corresponding alpha from Z-table

Step 5: Interpret results:

If Z-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the Z-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value method (Bayesian Testing):

import scipy.stats as stats

stats.binomtest(k, n, p, alternative='two-sided')

stats.binomtest(k, n, p, alternative='greater')

stats.binomtest(k, n, p, alternative='less')

k - the number of successes

If P-value \leq alpha \Rightarrow Reject H_0 and accept H_1

If P-value $>$ alpha \Rightarrow Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Hypothesis test for one population mean, σ is known

Requirements:

Each observation should be independent of other

Random sample with replacement, but if sample is without replacement should not be more than 10% of the population

Population standard deviation is known

Sample size is equal or greater than 30 or Population is normally distributed

If std is unknown the sample size should be greater than 30

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, H_0 (contains equal sign) and H_1

Step 2: Define significance level (α)

Step 3: Calculate Z-test statistic:

$$Z_t = (X_{\text{bar}} - \mu) / \sigma / \sqrt{n}$$

X_{bar} - sample mean

μ - population hypothetical mean

σ - population standard deviation

n - sample size

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put Z-critical value with corresponding α from Z-table

Step 5: Interpret the results:

If Z-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the Z-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from statsmodels.stats.weightstats import ztest

ztest_Score, p_value= ztest(df["col"],value = null_mean, alternative='two-sided')

ztest_Score, p_value= ztest(df["col"],value = null_mean, alternative='larger')

ztest_Score, p_value= ztest(df["col"],value = null_mean, alternative='smaller')

If P-value $\leq \alpha \Rightarrow$ Reject H_0 and accept H_1

If P-value $> \alpha \Rightarrow$ Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Hypothesis test for one population mean, σ is unknown and sample size is less than 30

Requirements:

Each observation should be independent of other

Random sample with replacement, but if sample is without replacement should not be more than 10% of the population

Population standard deviation is unknown and sample size is less than 30

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, H_0 (contains equal sign) and H_1

Step 2: Define significance level (α)

Step 3: Calculate T-test statistic:

$$T_t = (X_{\text{bar}} - \mu) / s / \sqrt{n}$$

X_{bar} - sample mean

μ - population hypothetical mean

s - sample standard deviation

n - sample size

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put T-critical value with corresponding alpha from T-table and DF

Step 5: Interpret results:

If T-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the T-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from scipy import stats

stats.ttest_1samp(df["col"], null_mean, alternative = "two-sided")

stats.ttest_1samp(df["col"], null_mean, alternative = "greater")

stats.ttest_1samp(df["col"], null_mean, alternative = "less")

If P-value \leq alpha \Rightarrow Reject H_0 and accept H_1

If P-value $>$ alpha \Rightarrow Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Hypothesis test for one variance

Chi-square Test

Requirements:

Each observation should be independent of other

Random sample with replacement, but if sample is without replacement should not be more than 10% of the population

The population should follow a normal distribution

Traditional Method:

Step 1: Define Claim and Opposite, H_0 (contains equal sign) and H_1

Step 2: Define significance level (alpha)

Step 3: Calculate Test statistic:

$$X^2 = ((n - 1) * (s*s)) / (\sigma * \sigma)$$

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail; Keep in mind that distribution starts from Zero and it's only Right-Skewed

On the picture put Chi-squared critical value with corresponding alpha and DF

Example in Python:

for 90% confidence level two tail test

import scipy.stats as stats

X2 = ((n - 1) * (s*s)) / (sigma*sigma)

chi_critical_right = stats.chi2.isf(0.05, df)

chi_critical_left = stats.chi2.isf(0.95, df)

for 90% confidence level one tail test

import scipy.stats as stats

```
X2 = ((n - 1) * (s*s)) / (sigma*sigma)
chi_critical = stats.chi2.isf(0.10, df)
```

Hypothesis test for two population proportions

Requirements:

Each observation should be independent of other

Two random samples with replacement, but if sample is without replacement should not be more than 10% of the population

The data contains only two categories, such as pass/fail or yes/no

The data in every sample should have at least 10 successes and at least 10 failures for normal approximation $n \cdot p \geq 10$; $n \cdot q \geq 10$

First approach - the proportions are not equal

Un-pooled method

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, $H_0 (p_1 - p_2 = D_0)$ and $H_1 (p_1 - p_2 \neq D_0)$

Step 2: Define significance level (α)

Step 3: Calculate Z-test statistic

$Z_t = ((p1_hat - p2_hat) - D_0) / \sqrt{((p1_hat * q1_hat) / n1) + ((p2_hat * q2_hat) / n2)}$

$D_0 = p_1 - p_2$; hypothetical difference in the proportions

p_hat is sample proportion of success

q_hat is sample proportion of failure

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put Z-critical value with corresponding α from Z-table

Step 5: Interpret results:

If Z-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the Z-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from statsmodels.stats import proportion

proportion.test_proportions_2indep(count1, n1, count2, n2, value = D, alternative = "two-sided", method = "score")

proportion.test_proportions_2indep(count1, n1, count2, n2, value = D, alternative = "larger", method = "score")

proportion.test_proportions_2indep(count1, n1, count2, n2, value = D, alternative = "smaller", method = "score")

If P-value $\leq \alpha \Rightarrow$ Reject H_0 and accept H_1

If P-value $> \alpha \Rightarrow$ Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Second approach - the proportions are equal

Pooled method

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, $H_0 (p_1 - p_2 = 0)$ and H_1

Step 2: Define significance level (α)

Step 3: Calculate Z-test statistic

$$Z_t = (p1_hat - p2_hat) / \sqrt{p_hat_total * (1 - p_hat_total) * (1/n1 + 1/n2)}$$

p_hat_total is $(X1 + X2) / (n1 + n2)$

p_hat is sample proportion of success

q_hat is sample proportion of failure

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put Z-critical value with corresponding α from Z-table

Step 5: Interpret results:

If Z-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the Z-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

```
from statsmodels.stats import proportion
```

```
proportion.test_proportions_2indep(count1, n1, count2, n2, value = None, alternative =  
"two-sided", method = "score")
```

```
proportion.test_proportions_2indep(count1, n1, count2, n2, value = None, alternative = "larger",  
method = "score")
```

```
proportion.test_proportions_2indep(count1, n1, count2, n2, value = None, alternative =  
"smaller", method = "score")
```

If P-value $\leq \alpha \Rightarrow$ Reject H_0 and accept H_1

If P-value $> \alpha \Rightarrow$ Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Hypothesis test for two population means, σ is known

Requirements:

Each observation should be independent of other

Two random samples with replacement, but if sample is without replacement should not be more than 10% of the population

Population standard deviation is known

Sample size is equal or greater than 30

Population is normally distributed

If std is unknown the sample size should be greater than 30

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, H_0 and H_1

Step 2: Define significance level (α)

Step 3: Calculate Z-test statistic:

$$Z_t = (X1_bar - X2_bar) - D_0 / \sqrt{((\sigma_1^2 / n_1) + ((\sigma_2^2 / n_2))}$$

D_0 - Hypothetical difference

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put Z-critical value with corresponding α from Z-table

Step 5: Interpret results:

If Z-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the Z-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from statsmodels.stats.weightstats import ztest

ztest(df1["col"], df2["col"], value = 0, alternative = "two-sided")

ztest(df1["col"], df2["col"], value = 0, alternative = "larger")

ztest(df1["col"], df2["col"], value = 0, alternative = "smaller")

If P-value $\leq \alpha \Rightarrow$ Reject H_0 and accept H_1

If P-value $> \alpha \Rightarrow$ Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Hypothesis test for two population means, σ is unknown and sample size is less than 30

Requirements:

The both data sets are independent

Each observation should be independent of other

Random sample with replacement, but if sample is without replacement should not be more than 10% of the population

Population standard deviation is unknown and sample size is less than 30

Populations are normally distributed

First approach - assumption that the variances are not equal

Traditional Method (Non-Bayesian Testing)

Step 1: Define Claim and Opposite, H_0 and H_1

Step 2: Define significance level (α)

Step 3: Calculate T-test statistic:

$$T_t = (X1_bar - X2_bar) - D_0 / \sqrt{((s_1^2 / n_1) + ((s_2^2 / n_2))}$$

D_0 - Hypothetical difference

s_1 and s_2 - sample standard deviation

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put T-critical value with corresponding alpha from T-table and DF
Degree of Freedom = $\min(n_1 - 1; n_2 - 1)$; take the minimum of these two

Step 5: Interpret results:

If T-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the T-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from scipy import stats

stats.ttest_ind(df1["col"], df2["col"], equal_var = False, alternative = "two-sided")

stats.ttest_ind(df1["col"], df2["col"], equal_var = False, alternative = "greater")

stats.ttest_ind(df1["col"], df2["col"], equal_var = False, alternative = "less")

If P-value \leq alpha \Rightarrow Reject H_0 and accept H_1

If P-value $>$ alpha \Rightarrow Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Second approach - assumption that the variances are equal

Traditional Method (Non-Bayesian Testing)

Step 1: Define Claim and Opposite, H_0 and H_1

Step 2: Define significance level (alpha)

Step 3: Calculate T-test statistic:

$$T_t = \frac{(X_1\text{-bar} - X_2\text{-bar}) - D_0}{\text{np.sqrt}(\frac{((n_1 - 1) * (s_1 * s_1)) + ((n_2 - 1) * (s_2 * s_2))}{(n_1 + n_2 - 2)}) * \text{np.sqrt}((1 / n_1) + (1 / n_2))}$$

s_1 and s_2 - sample standard deviation

D_0 - Hypothetical difference

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put T-critical value with corresponding alpha from T-table and DF

Degree of Freedom = $n_1 + n_2 - 2$

Step 5: Interpret results:

If T-test statistic is in Rejection Region \Rightarrow Reject H_0 and accept H_1

If the T-test statistic is in the Fail to Rejection Region \Rightarrow We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from scipy import stats

stats.ttest_ind(df1["col"], df2["col"], equal_var = True, alternative = "two-sided")

stats.ttest_ind(df1["col"], df2["col"], equal_var = True, alternative = "greater")

stats.ttest_ind(df1["col"], df2["col"], equal_var = True, alternative = "less")

If P-value \leq alpha \Rightarrow Reject H_0 and accept H_1

If P-value $>$ alpha \Rightarrow Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Third Approach - when populations are not normally distributed

A **Mann-Whitney U test (compare the medians)** is used to compare the differences between two independent samples when the sample distributions are not normally distributed and the sample sizes are small ($n < 30$)

It is considered to be the nonparametric equivalent to the two sample T-test

`t, pvalue = stats.mannwhitneyu(data1,data2, alternative = "two-sided")`
If the sample size is < 20, set method = "exact" to get the exact pvalue

Paired T-Test

Requirements:

If the values in one sample affect the values in the other sample, then the samples are dependent

Paired differences need to follow a normal distribution

Example- blood pressure before and after the specific medicine

Traditional Method (Non-Bayesian Testing):

Step 1: Define Claim and Opposite, H_0 and H_1

Step 2: Define significance level (alpha)

Step 3: Calculate T-test statistic:

$$T_t = (X1_bar - X2_bar) - D_0 / sd / np.sqrt(n)$$

D_0 - Hypothetical difference

Sd - standard deviation difference; $S_1 - S_2$

Step 4: Draw a picture: according to H_1 it is left-tail, right-tail or two-tail

On the picture put T-critical value with corresponding alpha from T-table and DF

Degree of Freedom = $n - 1$

Step 5: Interpret results:

If T-test statistic is in Rejection Region => Reject H_0 and accept H_1

If the T-test statistic is in the Fail to Rejection Region => We know nothing! There is not enough evidence to accept H_1

P-value Method (Bayesian Testing):

from scipy import stats

`stats.ttest_rel(bp_before, bp_after, alternative="two-sided")`

`stats.ttest_rel(bp_before, bp_after, alternative="greater")`

`stats.ttest_rel(bp_before, bp_after, alternative="less")`

If P-value <= alpha => Reject H_0 and accept H_1

If P-value > alpha => Fail to Reject H_0 ; We know nothing! There is not enough evidence to accept H_1

Wilcoxon Matched Pair Test (Wilcoxon signed-rank test)

Is a non-parametric hypothesis test that compares the medians of two paired groups and tells if they are identically distributed or not.

Paired differences do not need to follow a normal distribution

H₀: Sample distributions are equal

H₁: Sample distributions are not equal

from scipy import stats

stats.wilcoxon(data_before, data_after)

If P-value ≤ alpha ⇒ Reject H₀ and accept H₁

If P-value > alpha ⇒ Fail to Reject H₀; We know nothing! There is not enough evidence to accept H₁

Hypothesis test for two population variances

F- Ratio Test for equality of variances

Requirements:

Two independent samples

Each observation should be independent of other

Random sample with replacement, but if sample is without replacement should not be more than 10% of the population

The populations should follow a normal distribution

Traditional Method:

Step 1: Define Claim and Opposite, H₀ ($\sigma^2_1 = \sigma^2_2$) and H₁ ($\sigma^2_1 \neq \sigma^2_2$)

Step 2: Define significance level (alpha)

Step 3: Calculate Test statistic:

$F = \text{variance}_{1(\text{larger})} / \text{variance}_{2(\text{smaller})}$

Step 4: Draw a picture: it is always right-tail; F-distribution is always an upper-tailed test

On the picture put F-critical value from the calculator

DF numerator = n₁ - 1

DF denominator = n₂ - 1

Step 5: Interpret results:

If Test statistic is in Rejection Region ⇒ Reject H₀ ($\sigma^2_1 = \sigma^2_2$) and accept H₁

If the Test statistic is in the Fail to Rejection Region ⇒ We know nothing! There is not enough evidence to accept H₁

Example in Python:

from scipy import stats

F = var₁ / var₂

F_critical = stats.f.isf(alpha, df_numerator, df_denominator)

Other tests for comparison two or more variances

If the sample size is small (less than 30), the first step is always to test the normality of the population. Q-Q plot and **One sample Kolmogorov-Smirnov Test (KS Test)** can be used for that. The Kolmogorov–Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution.

If the p-value is larger than 0.05, that means that you can not reject the null. You can say that the sample is from Normal Distribution with a confidence level of 95%.

```
import scipy.stats as statst,  
t, pvalue = stats.kstest(sample, 'norm')
```

$H_0 (\sigma^2_1 = \sigma^2_2 = \sigma^2_3)$ and $H_1 (\sigma^2_1 \neq \sigma^2_2 \neq \sigma^2_3)$ for all of these tests (Kolmogorov-Smirnov Test and Mann-Whitney U Test are only for two populations)

If the pvalue is less than alpha, it means that at least one pair of variances has unequal variances; reject the Null Hypothesis for equality

from scipy import stats

stats.bartlett(data1, data2, data3) - **Bartlett's test**, if we know that all populations are normally distributed; Two or more independent samples;

stats.levene(data1, data2, data3, center = 'mean') - **Levene's test**, if we know that populations are almost normal; Two or more independent samples;

stats.levene(data1, data2, data3, center = 'median') - **Brown-Forsythe Test**, if we know that populations are sort of normal; Two or more independent samples;

t, pvalue = stats.ks_2samp(data1, data2) - **Kolmogorov-Smirnov Test** (KS Test for 2 samples), if we know that these two populations are not at all normal;

t, pvalue = stats.mannwhitneyu(data1, data2) - **Mann-Whitney U Test** (Nonparametric version of 2-sample t test), if we know that these two populations are not at all normal;

stats.kruskal(data1, data2, data3) - **Kruskal-Wallis Test (Non-parametric one-way ANOVA)** For two or more independent samples if we know that populations are not at all normal; All groups should have the same distribution. The null hypothesis is that the medians of all groups are equal and the alternative hypothesis is that at least one population median of one group is different from the population median of at least one other group. Note that rejecting the null hypothesis does not indicate which of the groups differs. It simply tells us that not all of the group medians are equal.

Tukey's Test between groups is required to determine which groups are different.

First create a DataFrame to hold data for score and groups. Finally Interpret all pvalues - the smaller pvalue means that there is statistically significant difference between the means of these two groups;

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

```
tukey = pairwise_tukeyhsd(endog=df['score'], groups=df['group'], alpha = 0.05)
```