EMMA ding
emmading.com
info@datainterviewpro.com

Z-test for Proportions

Lesson Structure

Why Use a Z-test for Proportions
One-Proportion Z-test
Two-Proportions Z-test

📌 Interview Questions

- Compare CTRs of an ad of the two groups of users
- Derive a confidence interval for the probability of getting heads from a series of coin tosses

▼ Why Use a Z-test for Proportions

The typical one and two-sample proportions tests are of this form

$$T = \frac{d}{s}$$

where d is the difference between a proportion and a constant or the difference between two proportions and s is an estimated standard deviation of d.

💡 The Slutsky's theorem: As long as the denominator s converges in probability to that unknown standard deviation, σ_d (a fairly weak condition), then $\frac{d}{s} \sim N(0, 1)$

Therefore, we have some justification for treating T as asymptotically normal, but we have no justification for treating it as t-distributed.

Theoretically, we don't use t-tests to test proportions and there's no good argument that t-distribution should be better than the z-distribution as an approximation to the distribution of T. But t-tests are sometimes used to test proportions.

Not entirely wrong as the results from a t-test is similar to that of a z-test for large samples.

▼ One-Proportion Z-test

Compare a proportion of a population to a constant.

Let p be the success rate of a large number n of independent Bernoulli trials.

Let $\hat{p}$ be the observed success rate, that is the number of observed successes over the total number of trials.

💡 When the sample contains at least 10 successes and 10 failures, it would be reasonable to use the normal approximation of a binomial distribution.

$$n\hat{p} \sim Binomial(n, p) \sim N(np, npq)$$

$$\hat{p} \sim N(p, \frac{p(1-p)}{n})$$

▼ Hypothesis

$H_0 : p = \ p_0$

$H_1 : p \neq p_0$

▼ Z-statistic

Under H_0, $Z \sim N(0, 1)$.

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

▼ E.g. Estimate the click-through rate $p \in (0, 1)$ of users on Ads

Suppose that we have an algorithm for Ad selection and we'd like to estimate the click-through rate $p \in (0, 1)$ of users on the Ads selected by this algorithm.

Given that we have access to 1000 users and the observed click-through rate is $\hat{p} = 0.2$. We set a significance level α of 5%.

$H_0 : p = 0.15$

```python
p_0 = 0.15
n = 1000
p_hat = 0.2

sigma = (p_0 * (1-p_0) / n)**0.5

observed_z_score = (p_hat - p_0) / sigma
critical_z_score = stats.norm.ppf(0.975)
print(observed_z_score)
print(critical_z_score)
```

```
4.428074427700477
1.959963984540054
```

▼ Confidence interval for a proportion

A level $(1-\alpha) * 100\%$ confidence interval for p

$$\hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

▼ E.g. Confidence interval of click-through rate on Ads

```python
# Build a confidence interval at 95% confidence level for the true proportion
sigma = (p_hat * (1-p_hat) / n)**0.5
margin_of_error = critical_z_score * sigma

lower = p_hat - margin_of_error
upper = p_hat + margin_of_error
confidence_interval = (lower, upper)
print(confidence_interval)
```

```
(0.17520819870781754, 0.22479180129218249)
```

▼ Two-Proportions Z-test

Compare the proportions p_1 and p_2 of two populations.

▼ Hypothesis

$H_0 : p_1 = p_2$

$H_1 : p_1 \neq p_2$

Under the null hypothesis, the two proportions are the same.

▼ Z-statistic

$\hat{p}_1 \sim \ N(p_1, \frac{p_1(1-p_1)}{n_1})$

$\hat{p}_2 \sim \ N(p_2, \frac{p_2(1-p_2)}{n_2})$

$\hat{p}_1 - \hat{p}_2 \sim \ N(p_1 - p_2, \frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2})$

Under H_0, $Z \sim N(0, 1)$.

$$Z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_1} + \frac{1}{n_2})}} \text{ and } \hat{p} = \frac{k_1 + k_2}{n_1 + n_2} = \frac{n_1\hat{p}_1 + n_2\hat{p}_2}{n_1 + n_2}$$

This pooled estimate $\hat{p}$ is similar to a weighted mean, but with two proportions.

💡 In many statistical programs, the default is to estimate the two proportions separately (i.e., unpooled).

$$Z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}}$$

▼ **E.g. Compare CTRs of two algorithms.**

Suppose we have 2 algorithms that are using different strategies to show Ads.

	Clicks	Impressions	CTR
Algorithm 1	30	900	0.033
Algorithm 2	20	1000	0.02

```python
# two-sample proportion confidence interval at 95% confidence level
n_x = 900
p_x = 0.033

n_y = 1000
p_y = 0.02

d = p_x - p_y

# pooled proportion
p = (n_x * p_x + n_y * p_y) / (n_x + n_y)
pooled_stdev = p * (1 - p) *  (1 / n_x + 1 / n_y) ** 0.5

# unpooled
unpooled_stdev = p_x * (1 - p_x) / n_x + p_y * (1 - p_y) / n_y ** 0.5

# z-statistic
observed_z_score = d / pooled_stdev
critical_z_score = stats.norm.ppf(0.975)
print(observed_z_score)
print(critical_z_score)
```

```
11.106992302731543
1.959963984540054
```

▼ Confidence interval for the difference between 2 proportions

Typically, we used **unpooled** proportions instead of pooled estimate of proportions.

> 💡 While the hypothesis testing procedure is based on the H_0, the confidence interval approach is not based on this assumption.

A level $(1-\alpha) * 100\%$ confidence interval of $p_1 - p_2$

$$\hat{p}_1 - \hat{p}_2 \pm z_{\alpha/2}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}$$

▼ E.g. Confidence interval of the difference between CTRs of two algorithms.

```
# Generate a CI at 95% CL for the difference of the two population means
margin_of_error = critical_z_score * unpooled_stdev
lower = d - margin_of_error
upper = d + margin_of_error
# Build a confidence interval at 95% CL for the true proportion
confidence_interval = (lower, upper)
print(confidence_interval)
```

```
(0.011715707947004552, 0.01428429205299545)
```