EMMA ding
emmading.com
info@datainterviewpro.com

Z-test for Means

Lesson Structure

📌 Interview Questions

- Mainly application questions
- Walkthrough the whole hypothesis testing process

When to Use the Z-test

- Infer properties of a population mean/proportion from a large enough sample of the population.
- Comparing population means/proportions from 2 different samples.

Assumptions of Z-test

The test statistic $Z \sim N(0, 1)$

$$Z = \frac{\text{sample mean} - \mu_0}{\text{sample standard deviation}}$$

where μ_0 is a constant or the sample mean of a second population to which we would like to compare our population mean.

There are multiple ways to ensure that this quantity Z is normally distributed.

- The easiest is to assume that the **population is normally distributed and that the population variance is known**. Then the denominator is constant, and Z is normal. In practice, we rarely know the true variance of the population, and that the sample points are exactly normally distributed.
- A most common scenario: We do not know the exact distribution of each sample point, but we know that they are **observed independently of each other, that we have more than 30 sample points, and that the population likely has a finite variance**. Then by the **Central Limit Theorem**, the fact

that the sample variance is a consistent estimator of the true variance, and Z is approximately standard normal.

▼ Z-test for One-Sample Mean

Let X be a large random sample of size n from a population with mean μ and standard deviation σ.

▼ Hypotheses

$H_0 : \mu = \mu_0$,

$H_1 : \mu \neq \mu_0$.

▼ Z-statistic

Based on CLT: $\bar{X} \sim N(\mu, \sigma^2/n)$

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

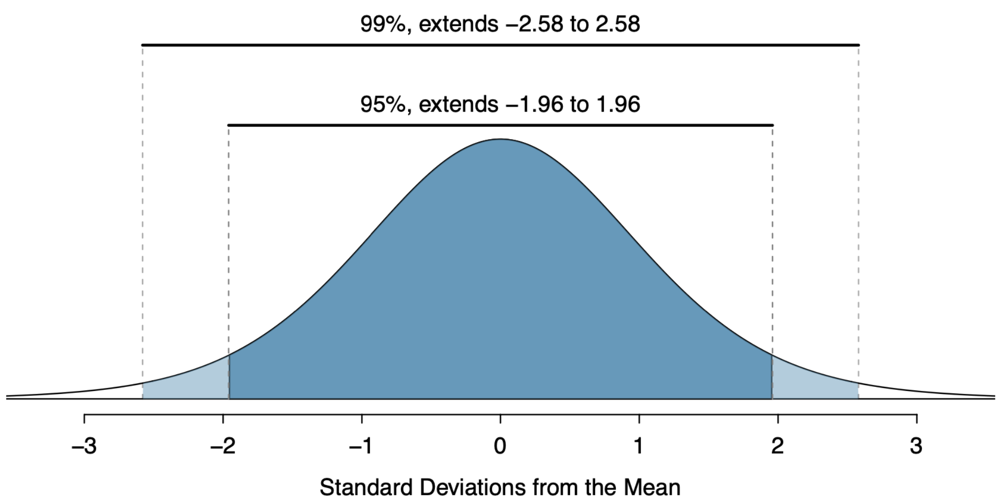
$$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$$

if σ is unknown (the case in almost all applications) and the sample size is large, we can replace it with the sample standard deviation s, then:

$$Z = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \text{where } s = \sqrt{\frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2}{n-1}}$$

▼ Observed Z-score vs the critical Z-score

- When $\alpha = 0.05$, then $z_{\alpha/2} = z_{.025} \approx 1.96$.



- if observed Z > critical Z: reject H_0; otherwise, fail to reject H_0

▼ **E.g. Test the height of men.**

Suppose that we sample the height of 1000 men. We generate this data from a normal with mean 172 cm and a standard deviation 4.5 cm, but as expected in practice, we assume that we do not have access to these parameters.

$H_0 : \mu = 160.$

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

```
# Simulate n height sample points
n = 1000
sigma = 4.5
population_mean = 172
x = stats.norm.rvs(loc=population_mean, scale=sigma, size=n)
```

```
# Test if mean is close to mu_0 using a Z-test with significance level 5%
mu_0 = 160
sample_sd = x.std() / n**0.5
observed_z_score = (x.mean() - mu_0) / (sigma / n**0.5)

critical_z_score = stats.norm.ppf(0.975)
print('observed_z_score = ', observed_z_score)
print('critical_z_score = ', critical_z_score)
```

```
observed_z_score =  84.76636175428682
critical_z_score =  1.959963984540054
```

▼ **Confidence interval for a population mean**

Let $X_1, ..., X_n$ be independent samples from a population with mean μ and suppose that $n > 30$ so that we can apply the CLT. If the standard deviation is known, then a level $(1 - \alpha) * 100\%$ confidence interval for μ is

$$\bar{X} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

Point estimate: $\bar{X}$

Margin of error: $z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$

If σ is unknown to us, we can use the sample standard deviation s.

$s = \sqrt{\frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)^2}{n-1}}$

▼ **E.g. Confidence interval for the height of men.**

```python
# Build confidence interval at 5% significance level for the population mean
margin_of_error = critical_z_score * (sigma / n**0.5)
lower = x.mean() - margin_of_error
upper = x.mean() + margin_of_error
confidence_interval = (lower, upper)
print(confidence_interval)
```

```
(171.7835569803829, 172.34137250945702)
```

Comparing the Means of Two Populations

Let X_1 be a large random sample of size n_1 from a population with mean μ_1 and standard deviation σ_1.

Let X_2 be a large random sample of size n_2 from a population with mean μ_2 and standard deviation σ_2.

The population mean of interest becomes the difference of the population means: $\mu_1 - \mu_2$.

Hypotheses

$H_0 : \mu_1 = \mu_2,$

$H_1 : \mu_1 \neq \mu_2.$

Z-statistic

$X_1 \sim \ N(\mu_1, \sigma_1^2)$

$X_2 \sim \ N(\mu_2, \sigma_2^2)$

$X_1 - X_2 \sim \ N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2)$

$\bar{X}_1 - \bar{X}_2 \sim N(\mu_1 - \mu_2, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$

$\frac{\bar{X}_1 - \bar{X}_2 - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0, 1)$

Under H_0, $Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0, 1)$

E.g. Compare the height of men from two different countries

```python
# two-sample Z-test example.
n_x = 1000
n_y = 900
sigma_x = 4.5
sigma_y = 3
mu_x = 172
mu_y = 169

x = stats.norm.rvs(loc=mu_x, scale=sigma_x, size=n_x)
y = stats.norm.rvs(loc=mu_y, scale=sigma_y, size=n_y)

# Test if mean height is close to mu_0 using a Z-test with CL 95%
mu_0 = 0
d = x.mean() - y.mean()
sigma_d = (x.var()/n_x + y.var()/n_y)**0.5

observed_z_score = (d - mu_0) / sigma_d
critical_z_score = stats.norm.ppf(0.975)
print(observed_z_score)
print(critical_z_score)
```

```
17.21098208079005
1.959963984540054
```

▼ **Confidence interval for the difference between 2 means**

A level $(1-\alpha) * 100\%$ confidence interval for the difference between the two population means $\mu_1 - \mu_2$ is:

$$\bar{X}_1 - \bar{X}_2 \pm z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

When the values of σ_1 and σ_2 are unknown, they can be estimated with the sample standard deviations s_1 and s_2.

▼ **E.g. Confidence interval for the difference of heights of men.**

```python
# two-sample confidence interval at 95% confidence level
margin_of_error = critical_z_score * sigma_d
lower = d - margin_of_error
upper = d + margin_of_error
two_sample_conf_int = (lower, upper)
print(two_sample_conf_int)
```

```
(2.616204608952039, 3.2886406099288545)
```