Session 3: Hypothesis Testing

Tingfang Wang

June 2024

Contents

Hypothesis Testing

Hypothesis testing for one population

Hypothesis testing for one population is used to make inferences about a population based on sample data. We compare the observed sample statistic to a hypothesized value and assess the likelihood of obtaining such an extreme result assuming the null hypothesis is true.

One Sample z-test for population percentage

The test statistic in this case is:

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \sim N(0,1)$$

Let's perform a one-sample z-test to determine if the population percentage of people who prefer cats over dogs differs significantly from 50%.

```r
# Sample data
pet_data <- c(rep(1, 140), rep(0, 60))

cat_lovers <- 140    # Number of successes
n <- 200        # Sample size

# Calculate the observed proportion
observed_proportion <- cat_lovers / n

# Null proportion (specified value)
null_proportion <- 0.5

# Compute the standard error of the proportion
standard_error <- sqrt((null_proportion * (1 - null_proportion)) / n)

# Calculate the test statistic (Z-score)
test_statistic <- (observed_proportion - null_proportion) / standard_error

# Calculate the two-tailed p-value
p_value <- 2 * pnorm(abs(test_statistic), lower.tail = FALSE)

# Display the test statistic and p-value
test_statistic
```

```
## [1] 5.656854
```

```r
p_value
```

```
## [1] 1.541726e-08
```

```r
prop_1sample_Ztest <- function(data, prop0, alternative){
  n <- length(data) # Sample size
  n_success <- sum(data == 1)  # Number of successes
  observed_proportion <- n_success / n # Calculate the observed proportion
  # Compute the standard error of the proportion
  standard_error <- sqrt((prop0 * (1 - prop0)) / n)
  # Calculate the test statistic (Z-score)
  test_statistic <- (observed_proportion - prop0) / standard_error
  # Calculate the p-value
  if (alternative == "two.sided") {
    p_value <- 2 * pnorm(abs(test_statistic), lower.tail = FALSE)
  }
  else if(alternative == "less") {
    p_value <- pnorm(test_statistic, lower.tail = TRUE)
  }
  else if(alternative == "greater"){
    p_value <- pnorm(abs(test_statistic), lower.tail = FALSE)
  }
  # Return the test statistic and p-value
  return(list("TestStatistice" = test_statistic, "PVlaue" = p_value))
}

prop_1sample_Ztest(data = pet_data, prop0 = 0.5, alternative = "two.sided")
```

```
## $TestStatistice
## [1] 5.656854
##
## $PVlaue
## [1] 1.541726e-08
```

In this example, we specify `null_proportion = 0.5` as the hypothesized population proportion. Replace the `cat_lovers` and `n` variables with your own values, adjust the null proportion if needed, and interpret the results based on the obtained p-value.

The `p-value` is then calculated by comparing the absolute value of the test statistic to the standard normal distribution using the `pnorm()` function. We multiply the p-value by 2 to obtain a two-tailed test.

The calculated p-value is 1.5417258×10^{-8}, which is less than the significance level (e.g., 0.05). Therefore, we reject the null hypothesis and conclude that there is enough evidence to suggest that the population percentage differs significantly from 50%.

Or, we can use the `prop.test()` function. It is commonly used when you want to compare proportions or test whether the proportion of successes differs significantly from a specified value.

The basic syntax of prop.test is as follows:

```
prop.test(x, n, p = NULL, alternative = "two.sided", correct = TRUE)
```

Here's a breakdown of the main arguments:

- `x`: The number of successes or positive outcomes. It can be a vector or a single value.

- `n`: The total number of trials or observations corresponding to x. It can also be a vector of the same length as x.

- `p`: The proportion under the null hypothesis. By default, it is set to NULL, which means that it assumes the proportion of successes is equal to the overall proportion of x divided by n.

- `alternative`: Specifies the alternative hypothesis. It can take one of the following values: "two.sided" (default), "less", or "greater". "two.sided" tests if the proportion differs from the null hypothesis proportion. "less" tests if the proportion is less than the null hypothesis proportion. "greater" tests if the proportion is greater than the null hypothesis proportion.

- `correct`: A logical value indicating whether to apply continuity correction. By default, it is set to TRUE, which is appropriate for large sample sizes. This option must be set to FALSE to make the test mathematically equivalent to the uncorrected z-test of a proportion.

To perform a one-sample z-test to determine if the population percentage of people who prefer cats over dogs differs significantly from 50%, we can do

```
# Sample data
cat_lovers <- 140    # Number of successes
n <- 200

test <- prop.test(x = 140, n = 200, p = 0.5, alternative = "two.sided", correct = FALSE)

# Print out the result
test
```

```
## 
##  1-sample proportions test without continuity correction
## 
## data:  140 out of 200, null probability 0.5
## X-squared = 32, df = 1, p-value = 1.542e-08
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.6332093 0.7592526
## sample estimates:
##   p
## 0.7
```

```r
# Display the test statistic and p-value
test$p.value
```

```
## [1] 1.541726e-08
```

One Sample z-test for population average

When the sample size is larger, we use z-test, the test statistic in this case is:

$$z = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \sim N(0,1)$$

Let's perform a one-sample z-test when the sample size is large to determine if the average income of a population is 13000.

```r
set.seed(135)
# Define a sample in a vector (40 observations)
income_data <- round(runif(n = 40, min = 11, max = 16)*1000, 0)

avg_1sample_Ztest <- function(data, mu0, alternative){
  # Calculate the test statistic (Z-score)
  test_statistic <- (mean(data) - mu0) / (sd(data) / sqrt(length(data)))
  # Calculate the p-value
  if (alternative == "two.sided") {
    p_value <- 2 * pnorm(abs(test_statistic), lower.tail = FALSE)
  }
  else if(alternative == "less") {
    p_value <- pnorm(test_statistic, lower.tail = TRUE)
  }
  else if(alternative == "greater"){
    p_value <- pnorm(abs(test_statistic), lower.tail = FALSE)
  }
  # Return the test statistic and p-value
  return(list("TestStatistice" = test_statistic, "PVlaue" = p_value))
}
avg_1sample_Ztest(data = income_data, mu0 = 13000, alternative = "two.sided")
```

```
## $TestStatistice
## [1] 2.45119
## 
## $PVlaue
## [1] 0.01423846
```

In the code above, we have a sample stored in the data vector, and we assume that the hypothesized population mean is 13000.

To perform the z-test, we calculate the test statistic `z` by subtracting the hypothesized population mean from the sample mean and dividing it by the standard error of the sample mean (standard deviation divided by the square root of the sample size).

The `p-value` is then calculated by comparing the absolute value of the test statistic to the standard normal distribution using the `pnorm()` function. We multiply the p-value by 2 to obtain a two-tailed test.

The calculated p-value is 1.5417258×10^{-8}, if we set the significance level as 0.05, then the `p-value` is less than the significance level (e.g., 0.05). Therefore, we reject the null hypothesis and conclude that there is enough evidence to suggest that the population average income differs significantly from 13000.

If we set the significance level as 0.01, then the `p-value` is greater than the significance level (e.g., 0.01). Therefore, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the population average income differs significantly from 13000.

Or, we can use `t.test()` in R to perform the z-test when the sample size is large as a t-distribution with a degrees of freedom greater than 30 is very similar with a standard normal distribution.

```r
# Install and load the stats package
#install.packages('stats')
library(stats)

# Apply the one sample z test using t.test function
result <- t.test(x = income_data,
                 mu = 13000,
                 alternative = "two.sided",
                 conf.level = 0.95)

# Print out the result
result
```

```
##
##  One Sample t-test
##
## data:  income_data
## t = 2.4512, df = 39, p-value = 0.01882
## alternative hypothesis: true mean is not equal to 13000
## 95 percent confidence interval:
##  13091.67 13957.13
## sample estimates:
## mean of x
##   13524.4
```

```r
# Print the p-value
result$p.value
```

```
## [1] 0.01882357
```

In this example, we specify $mu = 13000$ as the hypothesized population mean. The `alternative` argument is set to "two.sided" to test for a two-sided alternative hypothesis. If you want to test for a one-sided alternative hypothesis, you can set alternative to "less" or "greater", `conf.level` defines the confidence level, in default it is 95%.

The calculated p-value is 0.0188236, which is less than the significance level (e.g., 0.05). Therefore, we reject the null hypothesis and conclude that there is enough evidence to suggest that the population average income differs significantly from 13000.

One Sample t-test for population average

When the sample size is small(less than 30), we use t-test, the test statistic in this case is:

$$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \sim t_{n-1}$$

Let's perform a one-sample t-test when the sample size is small to determine if a population average income is 13000.

```r
# Install and load the stats package
#install.packages('stats')
library(stats)

# Define a sample in a vector (12 observations)
data <- c(12, 14, 15, 16, 11, 13, 17, 14, 15, 16, 12, 13)*1000

# Hypothesized population mean
mu <- 13000

# Perform the one sample t test using t.test function
result <- t.test(x = data,
                 mu = 13000,
                 alternative = "two.sided",
                 conf.level = 0.95)
# Print out the result
result
```

```
##
##  One Sample t-test
##
## data:  data
## t = 1.8638, df = 11, p-value = 0.08924
## alternative hypothesis: true mean is not equal to 13000
## 95 percent confidence interval:
##   12819.08 15180.92
## sample estimates:
## mean of x
##     14000
```

```r
# Print the p-value
result$p.value
```

```
## [1] 0.08924403
```

In this example, we specify $mu = 13000$ as the hypothesized population mean. The `alternative` argument is set to "two.sided" to test for a two-sided alternative hypothesis. If you want to test for a one-sided alternative hypothesis, you can set `alternative` to "less" or "greater", `conf.level` defines the confidence level, in default it is 95%.

The calculated p-value is 0.089244, which is greater than the significance level (e.g., 0.05). Therefore, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the population average income differs significantly from 13000.

Hypothesis testing for two populations

Hypothesis testing for two populations is a statistical procedure used to compare the means or proportions of two separate populations. It involves formulating null and alternative hypotheses, collecting data from both populations, performing statistical tests, and drawing conclusions based on the results.

Two sample z-test for population averages

The two-sample z-test for population means in R is used to compare the means of two independent populations when the population variances are known or the sample sizes are sufficiently large.

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

or,

$$z = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \sim N(0, 1)$$

1. Population variances are known

   Here's an example in R that demonstrates how to perform a z-test use `z.test()` function when the population variuances are known.

```
# Install and load the stats package
#install.packages('BSDA')
library(BSDA)
```

```
## Warning: package 'BSDA' was built under R version 4.3.3
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'BSDA'
```

```
## The following object is masked from 'package:datasets':
##
##     Orange
```

```
# Sample data
sample1 <- c(12.5, 13.1, 14.2, 11.9, 12.8)
sample2 <- c(11.2, 12.1, 13.5, 11.8, 12.6)

# Known population standard deviations
sigma1 <- 1.5
sigma2 <- 1.8
```

```r
# Perform z-test
result <- z.test(x = sample1,
                 y = sample2,
                 sigma.x = sigma1,
                 sigma.y = sigma2,
                 alternative = "two.sided",
                 conf.level = 0.95)

# Print out the result
result
```

```
##
##  Two-sample z-Test
##
## data:  sample1 and sample2
## z = 0.62986, p-value = 0.5288
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.393758  2.713758
## sample estimates:
## mean of x mean of y
##     12.90     12.24
```

```r
# Print the p-value
result$p.value
```

```
## [1] 0.5287874
```

In this example, we have two independent samples stored in the `sample1` and `sample2` vectors. We assume known population standard deviations of `sigma1 = 1.5` for the first sample and `sigma2 = 1.8` for the second sample. We then use the `z.test()` function to perform the two-sample z-test.

The calculated p-value is 0.5287874, which is greater than the significance level (e.g., 0.05). Therefore, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the two population means differ significantly.

2. Population variances are unknown

We can still use the `z.test()` function just by replacing the population standard deviations with the sample standard deviations. Here's an example in R that demonstrates how to perform a z-test when the population variances are unknown but the sample sizes are large.

Suppose we want to compare the average income of two populations: group A and group B. We have independent random samples from each group.

```r
set.seed(25886)
# Sample data
group_a_income <- round(runif(n = 40, min = 25000, max = 40000), 0)
group_b_income <- round(runif(n = 50, min = 30000, max = 45000), 0)

# Sample standard deviation
sigma1 <- sd(group_a_income)
sigma2 <- sd(group_b_income)
```

```r
# Perform z-test
result <- z.test(x = group_a_income,
                 y = group_b_income,
                 sigma.x = sigma1,
                 sigma.y = sigma2,
                 alternative = "two.sided",
                 conf.level = 0.95)

# Print out the result
result
```

```
##
##  Two-sample z-Test
##
## data:  group_a_income and group_b_income
## z = -4.3354, p-value = 1.455e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5486.208 -2070.142
## sample estimates:
## mean of x mean of y
##  33100.82  36879.00
```

```r
# Print the p-value
result$p.value
```

```
## [1] 1.454641e-05
```

The calculated p-value is 1.5417258×10^{-8}, which is less than the significance level (e.g., 0.05). Therefore, we reject the null hypothesis and conclude that there is enough evidence to suggest that the average income differs significantly between group A and group B.

Two sample t-test for population averages

The two-sample t-test for means in R is used to compare the means of two independent populations when the sample sizes are small. The test statistics are:

When the population variances are unknown but equal,

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t_{(n_1+n_2-2)}$$

where,

$$S_p = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}$$

When the population variances are unknown and not equal,

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \sim t_\nu$$

9

where,

$$\nu = \frac{(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2})^2}{\frac{S_1^4}{n_1^2(n_1-1)} + \frac{S_2^4}{n_2^2(n_2-1)}}$$

1. Equal Unknown Population variances

   Here's an example in R that demonstrates how to perform t-test for the difference of population averages when the population variance are the same using the `t.test()` function:

```
# Install and load the stats package
#install.packages('stats')
#library(stats)

# Sample data
sample1 <- c(1.2, 2.5, 3.1, 2.8, 1.9)
sample2 <- c(1.7, 2.3, 2.9, 3.4, 2.1)

# Perform t-test
result <- t.test(x = sample1,
                 y = sample2,
                 mu = 0, # hypothesized difference
                 alternative = "two.sided",
                 conf.level = 0.95,
                 var.equal = TRUE)

# Print out the result
result
```

```
##
##  Two Sample t-test
##
## data:  sample1 and sample2
## t = -0.39717, df = 8, p-value = 0.7016
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.2251059  0.8651059
## sample estimates:
## mean of x mean of y
##      2.30      2.48
```

```
# Print the p-value
result$p.value
```

```
## [1] 0.7016306
```

   In this code, we provide two samples sample1 and sample2 as input to the `t.test()` function. By setting `var.equal = TRUE`, the `t.test()` function assumes equal variances and uses the pooled variance estimate in the calculation of the test statistic.

   The calculated p-value is 0.7016306, which is greater than the significance level (e.g., 0.05). Therefore, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the two population means differ significantly.

2. Unequal Unknown Population variances

Here's an example in R that demonstrates how to perform t-test for the difference of population averages when the population variance are not the same using the `t.test()` function:

```r
# Install and load the stats package
#install.packages('stats')
#library(stats)

# Sample data
sample1 <- c(1.2, 2.5, 3.1, 2.8, 1.9)
sample2 <- c(1.7, 2.3, 2.9, 3.4, 2.1)

# Perform t-test
result <- t.test(x = sample1,
                 y = sample2,
                 mu = 0, # hypothesized difference
                 alternative = "two.sided",
                 conf.level = 0.95,
                 var.equal = FALSE)

# Print out the result
result
```

```
##
##  Welch Two Sample t-test
##
## data:  sample1 and sample2
## t = -0.39717, df = 7.8869, p-value = 0.7018
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.227721  0.867721
## sample estimates:
## mean of x mean of y
##      2.30      2.48
```

```r
# Print the p-value
result$p.value
```

```
## [1] 0.7017762
```

In this code, we provide two samples sample1 and sample2 as input to the `t.test()` function. By setting `var.equal = FALSE`, the t.test() function assumes unequal variances in the calculation of the confidence interval.

The calculated p-value is 0.7017762, which is greater than the significance level (e.g., 0.05). Therefore, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the two population means differ significantly.

Permutation Test

Permutation tests are non-parametric tests that solely rely on the assumption of exchangeability. A permutation test involves two or more samples. The null hypothesis is that all samples come from the same distribution.

To get a p-value, we randomly sample (without replacement) possible permutations of our variable of interest. The p-value is the proportion of samples that have a test statistic larger than that of our observed data.

Let's perform a permutation test to compare the average income between two groups (group A and group B).

```r
set.seed(123)
# Sample data
group_a_income <- round(runif(n = 40, min = 25000, max = 40000), 0)
group_b_income <- round(runif(n = 50, min = 30000, max = 45000), 0)

# Sample size
n_a <- length(group_a_income)
n_b <- length(group_b_income)

# Sample means
sample_mean_a <- mean(group_a_income)
sample_mean_b <- mean(group_b_income)

# Sample mean difference
observed_difference <- sample_mean_a - sample_mean_b

# Combine the two sample
combined_income <- c(group_a_income, group_b_income)

# Set the number of permutations
num_permutations <- 10000

# Resampling and get the differences
permutation_differences <- replicate(num_permutations, {
  permuted_income <- sample(x = combined_income, replace = FALSE)
  permuted_group_a <- permuted_income[1:n_a]
  permuted_group_b <- permuted_income[(n_a + 1):(n_a + n_b)]
  mean(permuted_group_a) - mean(permuted_group_b)
})

# Calculate the p-value
p_value <- sum(abs(permutation_differences) >= abs(observed_difference)) /
  num_permutations
p_value
```

```
## [1] 2e-04
```

```r
# Define a function to perform the permutation tests
permutation_test <- function(x, y, n_permutations, seed){
  # Set up the seed for re-sampling
  set.seed(seed)
  # Sample size
  n_x <- length(x)
  n_y <- length(y)
  # Sample means
  mean_x <- mean(x)
  mean_y <- mean(y)
  # Sample mean difference
```

```r
  observed_difference <- mean_x - mean_y
  # Combine the two sample
  combined_data <- c(x, y)
  # Re-sampling and get the differences
  permutation_differences <- replicate(n_permutations, {
    permuted_data <- sample(x = combined_data, replace = FALSE)
    permuted_x <- permuted_data[1:n_x]
    permuted_y <- permuted_data[(n_x + 1):(n_x + n_y)]
    mean(permuted_x) - mean(permuted_y)
    })
  # Calculate the p-value
  p_value <- sum(abs(permutation_differences) >= abs(observed_difference)) /
    n_permutations
  # Return p-value
  return(p_value)
}
permutation_test(x = group_a_income,
                 y = group_b_income,
                 n_permutations = 10000,
                 seed = 123)
```

```
## [1] 2e-04
```

The calculated p-value is 2×10^{-4}, which is less than the significance level (e.g., 0.05). Thus, we reject the null hypothesis and conclude that there is enough evidence to suggest that the income of group A and group B are from a same distribution using the permutation test.

Bootstrap Test

Bootstrap tests are non-parametric methods used to estimate the sampling distribution of a statistic by re-sampling(with replacement) from the observed data.

Let's perform a bootstrap test to compare the average income between two groups (group A and group B).

```r
set.seed(123)
# Sample data
group_a_income <- round(runif(n = 40, min = 25000, max = 40000), 0)
group_b_income <- round(runif(n = 50, min = 30000, max = 45000), 0)

# Define a function to perform the bootstrap tests
bootstrap_test <- function(x, y, n_bootstraps, seed){
  # Set up the seed for re-sampling
  set.seed(seed)
  # Sample size
  n_x <- length(x)
  n_y <- length(y)
  # Sample means
  mean_x <- mean(x)
  mean_y <- mean(y)
  # Sample mean difference
  observed_difference <- mean_x - mean_y
  # Re-sampling and get the differences
```

```r
  bootstrapped_differences <- replicate(n_bootstraps, {
    bootstrap_x <- sample(x, replace = TRUE)
    bootstrap_y <- sample(y, replace = TRUE)
    mean(bootstrap_x) - mean(bootstrap_y)
    })
  # Calculate the p-value
  p_value <- sum(abs(bootstrapped_differences) >=
                   abs(observed_difference)) / n_bootstraps
  # Return p-value
  return(p_value)
}
bootstrap_test(x = group_a_income,
               y = group_b_income,
               n_bootstraps = 10000,
               seed = 123)
```

```
## [1] 0.4831
```

The calculated p-value is 2×10^{-4}, which is great than the significance level (e.g., 0.05). Hence, we fail to reject the null hypothesis and conclude that there is not enough evidence to suggest that the average income differs significantly between group A and group B using the bootstrap test.

Conclusion

In this document, we covered various statistical concepts and provided examples with R code. We discussed hypothesis testing, permutation tests, and bootstrap tests. These techniques allow us to make inferences about population parameters and assess the significance of observed differences or associations. Remember to choose appropriate methods based on the nature of your data and the specific research question.