Bentley University MA214 in R

Content extracted from the How to Data Website

This PDF generated on 03 August 2023

Contents

MA214 is an undergraduate statistics course at Bentley University that builds on the basic managerial statistics course taken by all students. The description from the course catalog can be found here.

It covers hypothesis tests, analysis of variance, multiple regression, and contingency tables.

Review of statistical inference

Two populations

Variance inference

Chi-squares tests

ANOVA

Regression

- How to fit a linear model to two columns of data
- How to compute a confidence interval for the expected value of a response variable
- How to compute R-squared for a simple linear model
- How to predict the response variable in a linear model

Nonparametric tests

- How to create a QQ-plot
- How to test data for normality with Pearson's chi-squared test
- How to test data for normality with the D'Agostino-Pearson test
- How to test data for normality with the Jarque-Bera test

Content last modified on 03 August 2023.

How to compute a confidence interval for a population mean

Description

If we have a set of data that seems normally distributed, how can we compute a confidence interval for the mean? Assume we have some confidence level already chosen, such as $\alpha = 0.05$.

We will use the t-distribution because we have not assumed that we know the population standard deviation, and we have not assumed anything about our sample size. If you know the population standard deviation or have a large sample size (typically at least 30), then you can use z-scores instead; see how to compute a confidence interval for a population mean using z-scores.

Related tasks:

- How to compute a confidence interval for a population mean using z-scores
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)
- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

When applying this technique, you would have a series of data values for which you needed to compute a confidence interval for the mean. But in order to provide code that runs independently, we create some fake data below. When using this code, replace our fake data with your real data.

```r
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
data <- c( 435,542,435,4,54,43,5,43,543,5,432,43,36,7,876,65,5 ) # fake

# If you need the two values stored in variables for later use, do:
answer <- t.test( data, conf.level=1-alpha )
lower_bound <- answer$conf.int[1]
upper_bound <- answer$conf.int[2]

# If you just need to see the results in a report, do this alone:
t.test( data, conf.level=1-alpha )
```

```
    One Sample t-test

data:  data
t = 3.1853, df = 16, p-value = 0.005753
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  70.29848 350.05446
sample estimates:
mean of x
 210.1765
```

Note: The solution above assumes that the population is normally distributed, which is a common assumption in introductory statistics courses, but we have not verified that assumption here.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for a population mean using z-scores

Description

If we have a set of data that seems normally distributed, how can we compute a confidence interval for the mean? Assume we have some confidence level already chosen, such as $\alpha = 0.05$.

We will use the normal distribution, which assumes either that we know the population standard deviation, or we have a large enough sample size (typically at least 30). If neither of these is true in your case, then you can use *t*-scores instead; see how to compute a confidence interval for a population mean.

Related tasks:

- How to compute a confidence interval for a population mean
- How to do a two-sided hypothesis test for a population mean
- How to do a two-sided hypothesis test for two sample means (on website)
- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

When applying this technique, you would have a series of data values for which you needed to compute a confidence interval for the mean. But in order to provide code that runs independently, we create some fake data below. When using this code, replace our fake data with your real data.

We include the population standard deviation below, assuming it is known. See the notes at the end for what to do if you do not know the population standard deviation in your situation.

```r
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
pop.std <- 250      # replace with your population standard devation, if known
data <- c( 435,542,435,4,54,43,5,43,543,5,432,43,36,7,876,65,5 ) # fake

# Compute the sample mean, as an estimate for the population mean.
sample.mean <- mean( data )

# The margin of error then has the following formula.
z.score <- qnorm( alpha / 2, lower.tail=FALSE )
moe <- pop.std * z.score / sqrt( length( data ) )

# The confidence interval is centered on the mean with moe as its radius:
c( sample.mean - moe, sample.mean + moe )
```

```
[1]  91.3362 329.0167
```

Notes:

1. If you do not have the population standard deviation, but your sample is large enough (typically at least 30), you can approximate the population standard deviation with the sample standard deviation, using the code `pop.std <- sd( data )`. If your sample is not that large, then consider using a different technique instead; see how to compute a confidence interval for a population mean.
2. The solution above assumes that the population is normally distributed, which is a common assumption in introductory statistics courses, but we have not verified that assumption here.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for the population proportion

Description

If we have a sample of qualitative data from a normally distributed population, then how do we compute a confidence interval for a population proportion?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

We're going to use some fake data here for illustrative purposes, but you can replace our fake data with your real data in the code below.

```r
# Replace the next two lines of code with your real data
sample_size = 30
sample_proportion = 0.39

# Find the margin of error
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
moe <- qnorm(1-alpha/2, 0, 1) * sqrt(sample_proportion*(1-sample_proportion)/sample_size)

# Find the confidence interval
upper_bound <- sample_proportion + moe
lower_bound <- sample_proportion - moe
lower_bound
upper_bound
```

```
[1] 0.2154641



[1] 0.5645359
```

Our 95% confidence interval is $[0.2155, 0.5645]$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for a mean difference (matched pairs)

Description

Say we have two sets of data that are not independent of each other and come from a matched-pairs experiment, $(x_1, x_1'), (x_2, x_2'), \ldots, (x_n, x_n')$. We want to perform inference on the mean of the differences between these two samples, that is, the mean of $x_1 - x_1', x_2 - x_2', \ldots, x_n - x_n'$, called μ_D. We want to determine if it is significantly different from, greater than, or less than zero (or any other hypothesized value). We can do so with a two-tailed, right-tailed, or left-tailed hypothesis test for matched pairs.

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion
- How to do a hypothesis test for population variance
- How to do a hypothesis test for the difference between means when both population variances are known
- How to do a hypothesis test for the difference between two proportions
- How to do a hypothesis test for the mean with known standard deviation
- How to do a hypothesis test for the ratio of two population variances
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We choose a value, $0 \leq \alpha \leq 1$, as the Type I Error rate, and in this case we will set it to be 0.05.

We're going to use fake fata here, but you can replace our fake data with your real data below. Because the data are matched pairs, the samples must be the same size.

```
# Replace the following example data with your real data
sample.1 <- c(15, 10,  7, 22, 17, 14)
sample.2 <- c( 9,  1, 11, 13,  3,  6)
```

Two-tailed test

In a two-sided hypothesis test, the null hypothesis states that the mean difference is equal to 0 (or some other hypothesized value), $H_0 : \mu_D = 0$.

```
alpha = 0.05
t.test(sample.1, sample.2, alternative = "two.sided",
       mu = 0, paired = TRUE, conf.level = 1-alpha)
```

```
    Paired t-test

data:  sample.1 and sample.2
t = 2.8577, df = 5, p-value = 0.0355
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
  0.7033862 13.2966138
sample estimates:
mean difference 
              7 
```

Our p-value, 0.0355, appears in the third line of the output. It is smaller than α, so we have sufficient evidence to reject the null hypothesis and conclude that the mean difference between the two samples is significantly different from zero.

If we want instead to test whether it is some other value $d \neq 0$, then just use that value as the `mu` parameter to the `t.test` function instead of zero.

Right-tailed test

If instead we want to test whether the mean difference is less than or equal to zero, $H_0 : \mu_D \leq 0$, we can use a right-tailed test, as follows.

```r
t.test(sample.1, sample.2, alternative = "greater",
       mu = 0, paired = TRUE, conf.level = 1-alpha)
```

```
    Paired t-test

data:  sample.1 and sample.2
t = 2.8577, df = 5, p-value = 0.01775
alternative hypothesis: true mean difference is greater than 0
95 percent confidence interval:
 2.06416     Inf
sample estimates:
mean difference 
              7 
```

Our p-value, 0.01775, is smaller than α, so we have sufficient evidence to reject the null hypothesis and conclude that the mean difference between the two samples is significantly greater than zero.

Again, you can use another value $d \neq 0$ in place of `mu = 0` in the code.

Left-tailed test

If instead we want to test whether the mean difference is greater than or equal to zero, $H_0 : \mu_D \geq 0$, we can use a right-tailed test, as follows.

```r
t.test(sample.1, sample.2, alternative = "less",
       mu = 0, paired = TRUE, conf.level = 1-alpha)
```

```
	Paired t-test

data:  sample.1 and sample.2
t = 2.8577, df = 5, p-value = 0.9822
alternative hypothesis: true mean difference is less than 0
95 percent confidence interval:
     -Inf 11.93584
sample estimates:
mean difference 
              7 
```

Our p-value, 0.9822, is larger than α, so we do not have sufficient evidence to reject the null hypothesis; we must continue to assume that the mean difference between the two samples is greater than or equal to zero.

Again, you can use another value $d \neq 0$ in place of `mu = 0` in the code.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for a population proportion

Description

When we have qualitative data, we're often interested in performing inference on population proportions. That is, the proportion (between 0.0 and 1.0) of the population that is in a certain category with respect to the qualitative variables. Given a sample proportion, $\bar{p}$, how can we test whether the population proportion is equal to, greater than, or less than some hypothesized value?

Related tasks:

- How to compute a confidence interval for the population proportion
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for the difference between means when both population variances are known
- How to do a hypothesis test for the difference between two proportions
- How to do a hypothesis test for the mean with known standard deviation
- How to do a hypothesis test for the ratio of two population variances
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We're going to use fake data here for illustrative purposes, but you can replace our fake data with your real data in the code below.

Let's say that we've hypothesized that about one-third of Bostonians are unhappy with the Red Sox' performance. To test this hypothesis, we surveyed 460 Bostonians and found that 76 of them were unhappy with the Red Sox' performance.

We summarize this situation with the following variables. We will do a test with a Type I error rate of $\alpha = 0.05$.

```
n <- 460               # Number of respondents in sample
x <- 76                # Number of respondents in chosen subset
population_prop <- 1/3 # Hypothesized population proportion
```

Two-tailed test

A two-tailed test is for the null hypothesis $H_0 : p = \frac{1}{3}$. We use R's `prop.test()` function and provide it the data from above, requesting a two-tailed test.

```
prop.test(x = x, n = n, p = population_prop, alternative = "two.sided")
```

```
	1-sample proportions test with continuity correction

data:  x out of n, null probability population_prop
X-squared = 57.75, df = 1, p-value = 2.976e-14
alternative hypothesis: true p is not equal to 0.3333333
95 percent confidence interval:
 0.1330899 0.2030664
sample estimates:
        p 
0.1652174 
```

The p-value (shown at the end of the third line of the output) is less than α, so we can reject the null hypothesis. The proportion of Bostonians unhappy with Red Sox performance is different from $\frac{1}{3}$.

R also has a `binom.test()` function that takes the same arguments.

Right-tailed test

A right-tailed test is for the null hypothesis $H_0 : p \leq \frac{1}{3}$. We use R's `prop.test()` function and provide it the data from above, requesting a right-tailed test.

```r
prop.test(x = x, n = n, p = population_prop, alternative = "greater")
```

```
	1-sample proportions test with continuity correction

data:  x out of n, null probability population_prop
X-squared = 57.75, df = 1, p-value = 1
alternative hypothesis: true p is greater than 0.3333333
95 percent confidence interval:
 0.1377034 1.0000000
sample estimates:
        p 
0.1652174 
```

The p-value (shown at the end of the third line of the output) is greater than α, so we cannot reject the null hypothesis. We continue to assume that the proportion of Bostonians unhappy with Red Sox performance is less than or equal to $\frac{1}{3}$.

Again, `binom.test()` takes the same arguments.

Left-tailed test

A left-tailed test is for the null hypothesis $H_0 : p \geq \frac{1}{3}$. We use R's `prop.test()` function and provide it the data from above, requesting a left-tailed test.

```r
prop.test(x = x, n = n, p = population_prop, alternative = "less")
```

```
    1-sample proportions test with continuity correction

data:  x out of n, null probability population_prop
X-squared = 57.75, df = 1, p-value = 1.488e-14
alternative hypothesis: true p is less than 0.3333333
95 percent confidence interval:
 0.0000000 0.1967951
sample estimates:
        p 
0.1652174 
```

The p-value (shown at the end of the third line of the output) is less than α, so we can reject the null hypothesis. The proportion of Bostonians unhappy with Red Sox performance is less than $\frac{1}{3}$.

Again, `binom.test()` takes the same arguments.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for population variance

Description

Assume we want to estimate the variability of a quantity across a population, starting from a sample of data, $x_1, x_2, x_3, \ldots x_k$. How might we test whether the population variance is equal to, greater than, or less than a hypothesized value?

Related tasks:

- How to compute a confidence interval for the population proportion
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion
- How to do a hypothesis test for the difference between means when both population variances are known
- How to do a hypothesis test for the difference between two proportions
- How to do a hypothesis test for the mean with known standard deviation
- How to do a hypothesis test for the ratio of two population variances
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We'll use R's dataset `EuStockMarkets` to do an example. This dataset has information on the daily closing prices of 4 European stock indices. We're going to look at the variability of Germany's DAX closing prices.

Let's load the dataset. (See how to quickly load some sample data (on website).) If using your own data, place it into the `values` variable instead of using the code below.

```
# install.packages("datasets") # If you have not already done this
library(datasets)
EuStockMarkets <- data.frame(EuStockMarkets)
values <- EuStockMarkets$DAX
```

Two-tailed test

We may ask whether the population variance is significantly different from a hypothesized value. Let's test against a variance of 1,000,000.

Our null hypothesis states that the population variance is equal to 1,000,000, $H_0 : \sigma^2 = 1,000,000$. We calculate the test statistic and p-value as follows, using a χ^2 distribution. We can use any α between 0.0 and 1.0 as our Type I Error Rate; we will use $\alpha = 0.05$ here.

```
hyp.var <- 1000000                             # hypothesized variance
df <- length(values) - 1                       # degrees of freedom
test.statistic <- df*var(values)/hyp.var       # test statistic
2*pchisq(test.statistic, df=df, lower.tail=FALSE) # two-tailed p-value
```

```
[1] 3.189769e-07
```

Our p-value, 3.189769×10^{-7}, is smaller than α, so we have sufficient evidence to reject the null hypothesis. The variance of closing prices on Germany's DAX is signficantly different from 1,000,000.

Left-tailed test

What if we wanted to determine if the population variance were significantly less than 1,000,000? Our null hypothesis would therefore be $H_0 : \sigma^2 \geq 1,000,000$.

The computations are very similar to the previous case, but with a different formula for the p-value. We repeat the code that's in common, for ease of use when copying and pasting.

```r
hyp.var <- 1000000                               # hypothesized variance
df <- length(values) - 1                         # degrees of freedom
test.statistic <- df*var(values)/hyp.var         # test statistic
pchisq(test.statistic, df=df, lower.tail=TRUE) # left-tailed p-value
```

```
[1] 0.9999998
```

Our p-value, 0.9999998, is greater than α, so we do not have sufficient evidence to reject the null hypothesis. We should continue to assume that the variance of closing prices on Germany's DAX is greater than or equal to 1,000,000.

Right-tailed test

What if we wanted to determine if the population variance were significantly less than 1,000,000? Our null hypothesis would therefore be $H_0 : \sigma^2 \geq 1,000,000$.

The computations are very similar to the previous case, but with a different formula for the p-value. We repeat the code that's in common, for ease of use when copying and pasting.

```r
hyp.var <- 1000000                                # hypothesized variance
df <- length(values) - 1                          # degrees of freedom
test.statistic <- df*var(values)/hyp.var          # test statistic
pchisq(test.statistic, df=df, lower.tail=FALSE) # right-tailed p-value
```

```
[1] 1.594884e-07
```

Our p-value, 1.594884×10^{-7}, is smaller than α, so have sufficient evidence to reject the null hypothesis. We conclude that the variance of closing prices on Germany's DAX is significantly greater than 1,000,000.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for the mean with known standard deviation

Description

Let's say we are measuring a variable over a population, and we know its standard deviation σ is known, and assume that the variable is normally distributed. We take a sample, $x_1, x_2, x_3, \ldots, x_k$, and compute its mean $\bar{x}$. We want to determine if the sample mean is significantly different from, greater than, or less than some hypothesized value, such as a hypothesized population mean. How do we formulate an appropriate hypothesis test?

Related tasks:

- How to compute a confidence interval for a population mean
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion
- How to do a hypothesis test for population variance
- How to do a hypothesis test for the difference between means when both population variances are known
- How to do a hypothesis test for the difference between two proportions
- How to do a hypothesis test for the ratio of two population variances
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We will use the following fake data, but you can insert your actual data in its place. We have a sample of just 5 values and an assumed population standard deviation of 3.

```
sample <- c(31, 44, 28, 25, 40)  # sample data
pop.std <- 3                      # population standard deviation
```

We also choose a value $0 \leq \alpha \leq 1$ as our Type I error rate. We'll let α be 0.05 here, but you can change that in the code below.

Two-tailed test

In a two-tailed test, we compare the unknown population mean to a hypothesized value m using the null hypothesis $H_0 : \mu = m$. Here we'll use $m = 30$, but you can replace that value in the code below with the hypothesis relevant for your situation.

```
m <- 30                                       # hypothesized mean
n <- length(sample)                           # number of observations
xbar <- mean(sample)                          # sample mean
test.stat <- (xbar - m) / (pop.std/sqrt(n))   # test statistic
2*pnorm(abs(test.stat), 0, 1, lower.tail=FALSE)   # two-tailed p-value
```

```
[1] 0.007290358
```

The p-value, 0.00729, is less than α, so we have evidence to reject the null hypothesis and conclude that the actual population mean μ is significantly different from the hypothesized value of $m = 30$.

Right-tailed test

In a right-tailed hypothesis test, the null hypothesis is that the population mean is greater than or equal to a chosen value, $H_0 : \mu \geq m$.

Most of the code below is the same as above, but we repeat it to make it easy to copy and paste. Only the computation of the p-value changes.

```
m <- 30                                     # hypothesized mean
n <- length(sample)                         # number of observations
xbar <- mean(sample)                        # sample mean
test.stat <- (xbar - m) / (pop.std/sqrt(n)) # test statistic
pnorm(test.stat, 0, 1, lower.tail=FALSE)    # right-tailed p-value
```

```
[1] 0.003645179
```

The p-value, 0.003645, is less than α, so we have evidence to reject the null hypothesis and conclude that the actual population mean μ is significantly less than the hypothesized value of $m = 30$.

Left-tailed test

In a left-tailed hypothesis test, the null hypothesis is that the population mean is less than or equal to a chosen value, $H_0 : \mu \leq m$.

Most of the code below is the same as above, but we repeat it to make it easy to copy and paste. Only the computation of the p-value changes.

```
m <- 30                                     # hypothesized mean
n <- length(sample)                         # number of observations
xbar <- mean(sample)                        # sample mean
test.stat <- (xbar - m) / (pop.std/sqrt(n)) # test statistic
pnorm(test.stat, 0, 1, lower.tail=TRUE)     # left-tailed p-value
```

```
[1] 0.9963548
```

The p-value, 0.99635, is greater than α, so wwe do not have sufficient evidence to conclude that $\mu > m$ and should continue to accept the null hypothesis, that $\mu \leq m$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for a mean difference (matched pairs)

Description

Say we have two sets of data that are not independent of each other and come from a matched-pairs experiment, and we want to construct a confidence interval for the mean difference between these two samples. How do we make this confidence interval? Let's assume we've chosen a confidence level of $\alpha = 0.05$.

Related tasks:

- How to do a hypothesis test for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

We have two samples of data, $x_1, x_2, x_3, \ldots, x_k$ and $x'_1, x'_2, x'_3, \ldots, x'_k$. We're going to use some fake data below just as an example; replace it with your real data.

```
sample.1 <- c(15, 10, 7, 22, 17, 14)
sample.2 <- c(9, 1, 11, 13, 3, 6)
```

The shortest way to create the confidence interval is with R's `t.test()` function. It's just one line of code (after we choose α).

```
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
t.test(sample.1, sample.2, paired = TRUE, conf.level = 1-alpha)
```

```
    Paired t-test

data:  sample.1 and sample.2
t = 2.8577, df = 5, p-value = 0.0355
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
  0.7033862 13.2966138
sample estimates:
mean difference 
              7 
```

If you need the lower and upper bounds later, you can save them as variables as follows.

```r
conf.interval <- t.test(sample.1, sample.2, paired = TRUE, conf.level = 1-alpha)
lower.bound <- conf.interval$conf.int[1]
upper.bound <- conf.interval$conf.int[2]
```

It's also possible to do the computation manually, using the code below.

```r
diff.samples <- sample.1 - sample.2               # differences between the samples
n = length(sample.1)                              # number of observations per sample
diff.mean <- mean(diff.samples)                   # mean of the differences
diff.variance <- var( diff.samples )              # variance of the differences
critical.val <- qt(p = alpha/2, df = n - 1,
    lower.tail=FALSE)                             # critical value
radius <- critical.val*sqrt(diff.variance)/sqrt(n) # radius of confidence interval
c( diff.mean - radius, diff.mean + radius )       # confidence interval
```

```
[1]  0.7033862 13.2966138
```

Either method gives the same result. Our 95% confidence interval is $[0.70338, 13.2966]$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to choose the sample size in a study with two population means

Description

When designing a study, it is important to choose a sample size that is large enough to perform a useful test but that is also economically feasible. How we choose the sample size depends on what test we plan to run on the data from our study. Here, let's say our data will be used to compare two population means. If we are planning such a study, how do we determine how large it should be in order for the test that compares the population means to have a certain power?

Related tasks:

- How to compute the power of a test comparing two population means (on website)

Solution in pure R

Example: Let's say we're designing a study to assess the effectiveness of a new four-week exercise program for weight loss. Assume that weight loss in four-week exercise programs is normally distributed with a standard deviation of around 5 pounds. The goal is that the new exercise program will have a 4-pound higher weight loss than the average program. (Notice that we will be comparing the means of two populations, the weight loss in each of two programs.)

We choose a value $0 \leq \alpha \leq 1$ as the probability of a Type I error in our test that compares the two means. (Recall, Type I error is for a false positive, finding we should reject H_0 when it's actually true). Let's set α to be 0.05 here.

We choose a value $0 \leq \beta \leq 1$ as the probability of a Type II error (false negative, failing to reject H_0 when it's actually false). Let's set β to be 0.2 here. The test's power is $1 - \beta$, or in this case, 0.8.

What should the sample size be for each group?

```r
# sd = standard deviation = 5 pounds
# delta = desired increase = 4 pounds
# sig.level = alpha = 0.05
# power = 1 - beta = 1 - 0.20 = 0.80
# n = NULL so R computes it for us
power.t.test(n = NULL, delta = 4, sd = 5, sig.level = 0.05, power = 0.80)
```

```

     Two-sample t test power calculation 

              n = 25.52463
          delta = 4
             sd = 5
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

Our sample size needs to be 26 participants in order for the power of the study to be 80% with our specified parameters.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for the difference between two means when both population variances are known

Description

If we have samples from two independent populations, and both of the population variances are known, how do we construct a confidence interval for the difference between the population means?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

We're going to use some fake data here to illustrate how to make the confidence interval. Replace our fake data and population variances with your actual data and population variances if you use this code.

```
sample.1 <- c(15, 10, 7, 22, 17, 14)
sample.2 <- c(9, 1, 11, 13, 3, 6)
pop1.variance <- 2.3
pop2.variance <- 3
```

We will need the size and mean of each sample.

```
n.sample1 <- length(sample.1)
n.sample2 <- length(sample.2)
xbar1 <- mean(sample.1)
xbar2 <- mean(sample.2)
```

We can then use that data to create the confidence interval.

```
# Find the critical value from the normal distribution
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
critical.val <- qnorm(p=alpha/2, lower.tail=FALSE)

# Find the lower and upper bounds of the confidence interval
radius <- critical.val*sqrt(pop1.variance/n.sample1 + pop2.variance/n.sample2)
upper.bound <- (xbar1 - xbar2) + radius
lower.bound <- (xbar1 - xbar2) - radius
lower.bound
upper.bound
```

```
[1] 5.157912



[1] 8.842088
```

Our 95% confidence interval for the true difference between the population means is $[5.1579, 8.842]$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for the difference between two means when population variances are unknown

Description

If we have samples from two independent populations and both of the population variances are unknown, how do we compute a confidence interval for the difference between the population means?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

We're going to use some fake data here to illustrate how to make the confidence interval. Replace our fake data with your actual data if you use this code.

```r
sample.1 <- c(15, 10, 7, 22, 17, 14)
sample.2 <- c(9, 1, 11, 13, 3, 6)
```

In the example below, we specify `var.equal = FALSE` to indicate that we cannot assume that the variances are equal. If you know them to be equal in your situation, replace `FALSE` with `TRUE`.

```r
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
conf.interval <- t.test(sample.1, sample.2, var.equal = FALSE, conf.level = 1-alpha)
# If you need the upper and lower bounds later, store them in variables like this:
lower.bound <- conf.interval$conf.int[1]
upper.bound <- conf.interval$conf.int[2]
# Print out the lower and upper bounds
lower.bound
upper.bound
```

```
[1] 0.5852484



[1] 13.41475
```

Our 95% confidence interval for the true difference between these population means is $[0.5852, 13.4147]$.

You can also see the test statistic and p-value by inspecting the result of the `t.test` function we ran above.

```r
conf.interval
```

```
    Welch Two Sample t-test

data:  sample.1 and sample.2
t = 2.4363, df = 9.8554, p-value = 0.0354
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.5852484 13.4147516
sample estimates:
mean of x mean of y
14.166667  7.166667
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for the difference between two proportions

Description

When dealing with qualitative data, we often want to construct a confidence interval for the difference between two population proportions. For example, if we are trying a drug on experimental and control groups of patients, we probably want to compare the proportion of patients who got well in one group versus the other.

How do we make such a comparison using a confidence interval?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

Here is some fake data for the purposes of this illustration. Let's say we conduct a survey of people in Boston and of people in Nashville and ask them if they prefer chocolate or vanilla ice cream. We want to compare the proportions of people from the two cities who like vanilla.

- Out of 150 people in Boston surveyed, 90 prefer vanilla.
- Out of 135 people in Nashville surveyed, 50 prefer vanilla.

We'll let $\bar{p_1}$ represent the proportion of people from Boston who like vanilla and $\bar{p_2}$ represent the proportion of people from Nashville who like vanilla. You can replace the code for this fake data below with your real data.

```r
# number of observations in the samples
n1 <- 150
n2 <- 135
# proportions in the two samples
p_bar1 <- 90/150
p_bar2 <- 50/135
```

We now compute the confidence interval using R's `qnorm` function.

```r
# Find the critical value to compute the confidence interval
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
critical_value <- qnorm(p = alpha/2, lower.tail=FALSE)

# Compute the standard error of the proportions
std_error <- sqrt( p_bar1*(1-p_bar1)/n1 + p_bar2*(1-p_bar2)/n2 )

# Compute the upper and lower bounds of the confidence interval and print them out
upper_bound <- (p_bar1 - p_bar2) + critical_value*std_error
lower_bound <- (p_bar1 - p_bar2) - critical_value*std_error
lower_bound
upper_bound
```

```
[1] 0.1165722



[1] 0.3426871
```

The confidence interval for the difference between these two proportions is $[0.11657, 0.34269]$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for the ratio of two population variances

Description

Let's say we want to compute a confidence interval for two population variances. We take two samples of data, $x_1, x_2, x_3, \ldots, x_k$ and $x'_1, x'_2, x'_3, \ldots, x'_k$, and compute their variances, σ_1^2 and σ_2^2. How do we compute a confidence interval for $\frac{\sigma_1^2}{\sigma_2^2}$?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion

Solution in pure R

We'll use R's dataset EuStockMarkets as an example; of course you should replace this example data with your actual data when using this code. This dataset has information on the daily closing prices of 4 European stock indices. We're going to compare the variability of Germany's DAX and France's CAC closing prices here.

```r
# install.packages("datasets") # if you have not done so already
library(datasets)

# Load in the EuStockMarkets data and convert to a DataFrame
EuStockMarkets <- data.frame(EuStockMarkets)

# Our two samples are its DAX and CAC columns
sample.1 <- EuStockMarkets$DAX
sample.2 <- EuStockMarkets$CAC
```

Now that we have our data loaded we can compute the confidence interval. You can change the confidence level by changing the value of α below.

```r
# The degrees of freedom in each sample is its length minus 1
df_1 = length(sample.1) - 1
df_2 = length(sample.2) - 1

# Compute the ratio of the variances
test.stat.ratio <- var(sample.1)/var(sample.2)

# Find the critical values from the F-distribution
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
lower_critical_value <- 1 / qf(p = alpha/2, df1 = df_1, df2 = df_2, lower.tail = FALSE)
upper_critical_value <- qf(p = alpha/2, df1 = df_2, df2 = df_1, lower.tail = FALSE)

# Compute the confidence interval and print it out
lower_bound <- test.stat.ratio*lower_critical_value
upper_bound <- test.stat.ratio*upper_critical_value
lower_bound
upper_bound
```

```
[1] 3.190589



[1] 3.827044
```

The 95% confidence interval for the ratio of the variances for Germany's DAX and France's CAC is $[3.191, 3.827]$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for the difference between means when both population variances are known

Description

Assume we have two samples, $x_1, x_2, \ldots, x_n$ and $x'_1, x'_2, \ldots, x'_n$, that come from normally distributed populations with known variances, and the two sample means are $\bar{x}$ and $\bar{x}'$, respectively. We might want to ask whether the difference $\bar{x} - \bar{x}'$ is significantly different from, greater than, or less than zero.

Related tasks:

- How to compute a confidence interval for the difference between two means when both population variances are known
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion
- How to do a hypothesis test for population variance
- How to do a hypothesis test for the difference between two proportions
- How to do a hypothesis test for the mean with known standard deviation
- How to do a hypothesis test for the ratio of two population variances
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We're going to use fake data here, but you can replace our fake data with your real data below. You will need not only the samples but also the known population standard deviations.

```
sample1 <- c(5, 8, 10, 3, 6, 2)
sample2 <- c(13, 20, 16, 12, 18, 15)
population1_sd = 2.4
population2_sd = 3
```

We must compute the sizes and means of the two samples.

```
n1 <- length(sample1)
n2 <- length(sample2)
sample1_mean <- mean(sample1)
sample2_mean <- mean(sample2)
```

We choose a value $0 \le \alpha \le 1$ as the probability of a Type I error (a false positive, finding we should reject H_0 when it's actually true). We will use $\alpha = 0.05$ in this example.

Two-tailed test

In a two-tailed test, the null hypothesis is that the difference is zero, $H_0 : \bar{x} - \bar{x}' = 0$. We compute a test statistic and p-value as follows.

```
test_statistic <- (sample1_mean - sample2_mean) /
    sqrt(population1_sd^2/n1 + population2_sd^2/n2)
2*pnorm(abs(test_statistic), lower.tail = FALSE)  # two-tailed p-value
```

```
[1] 1.820494e-10
```

Our p-value is less than α, so we have sufficient evidence to reject the null hypothesis. The difference between the means is significantly different from zero.

Right-tailed test

In the right-tailed test, the null hypothesis is $H_0 : \bar{x} - \bar{x}' \leq 0$. That is, we are testing whether the difference is greater than zero.

The code is very similar to the previous, except only in computing the p-value. We repeat the code that's in common, to make it easier to copy and paste the examples.

```
test_statistic <- (sample1_mean - sample2_mean) /
    sqrt(population1_sd^2/n1 + population2_sd^2/n2)
pnorm(test_statistic, lower.tail = FALSE)  # right-tailed p-value
```

```
[1] 1
```

Our p-value is greater than α, so we do not have sufficient evidence to reject the null hypothesis. We would continue to assume that the difference in means is less than or equal to zero.

Left-tailed test

In a left-tailed test, the null hypothesis is $H_0 : \bar{x} - \bar{x}' \geq 0$. That is, we are testing whether the difference is less than zero.

The code is very similar to the previous, except only in computing the p-value. We repeat the code that's in common, to make it easier to copy and paste the examples.

```
test_statistic <- (sample1_mean - sample2_mean) /
    sqrt(population1_sd^2/n1 + population2_sd^2/n2)
pnorm(test_statistic, lower.tail = TRUE)  # left-tailed p-value
```

```
[1] 9.102468e-11
```

Our p-value is less than α, so we have sufficient evidence to reject the null hypothesis. The difference between the means is significantly less than zero.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for the difference between two proportions

Description

When dealing with qualitative data, we typically measure what proportion of the population falls into various categories (e.g., which religion a survey respondent adheres to, if any). We might want to compare two proportions by measuring their difference, and asking whether it is equal, greater, or less than zero. How can we perform such a test?

Related tasks:

- How to compute a confidence interval for the difference between two proportions
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion
- How to do a hypothesis test for population variance
- How to do a hypothesis test for the difference between means when both population variances are known
- How to do a hypothesis test for the mean with known standard deviation
- How to do a hypothesis test for the ratio of two population variances
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We will use some fake data in this example, but you can replace it with your real data. Imagine we conduct a survey of people in Boston and of people in Nashville and ask them if they prefer chocolate or vanilla ice cream. We get data like the following.

City	Prefer chocolate	Prefer vanilla	Total
Boston	60	90	150
Nashville	85	50	135

We want to compare the proportions of people from the two cities who like vanilla.

Let $\bar{p}_1$ represent the proportion of people from Boston who like vanilla and $\bar{p}_2$ represent the proportion of people from Nashville who like vanilla.

```
n1 <- 150
n2 <- 135
p_bar1 <- 90/150
p_bar2 <- 50/135
```

We choose a value $0 \leq \alpha \leq 1$ as our Type 1 error rate. For this example, we will use $\alpha = 0.05$.

Two-tailed test

In a two-tailed test, the null hypothesis states that the difference between the two proportions equals a hypothesized value; let's choose zero, $H_0 : \bar{p}_1 - \bar{p}_2 = 0$. We perform this test by computing a test statistic and p-value as shown below, then comparing the p-value to our chosen α.

```
p_bar <- (90 + 50) / (150 + 135)                # overall proportion
std_error <- sqrt(p_bar*(1-p_bar)*(1/n1+1/n2))  # standard error
test_statistic <- (p_bar1 - p_bar2)/std_error   # test statistic
2*pnorm(q = test_statistic, lower.tail = FALSE) # two-tailed p-value
```

```
[1] 0.0001080269
```

Our p-value, 0.000108, is smaller than α, so we can reject the null hypothesis and conclude that the difference between the two proportions is different from zero.

But we did not need to compare the difference to zero; we could have used any hypothesized difference for comparison. Let's repeat the above test, comparing the difference to 0.15 instead, as an example.

```
hyp.diff = 0.15                                            # hypothesized difference
std_error <- sqrt(p_bar1*(1-p_bar1)/n1
                + p_bar2*(1-p_bar2)/n2)                    # standard error
test_statistic <- ((p_bar1 - p_bar2) - hyp.diff)/std_error # test statistic
2*pnorm(q = test_statistic, lower.tail = FALSE)            # two-tailed p-value
```

```
[1] 0.1674453
```

Our p-value, 0.1674, is greater than α, so we cannot reject the null hypothesis and cannot conclude that the difference between these two proportions is significantly different from 0.15.

Right-tailed test

In a right-tailed test, the null hypothesis states that the difference between the two proportions is less than or equal to a hypothesized value. Let's begin by using zero as our hypothesized value, $H_0 : \bar{p}_1 - \bar{p}_2 \leq 0$.

We repeat some code below that we've seen above, just to make it easy to copy and paste the example elsewhere.

```
p_bar <- (90 + 50) / (150 + 135)                # overall proportion
std_error <- sqrt(p_bar*(1-p_bar)*(1/n1+1/n2))  # standard error
test_statistic <- (p_bar1 - p_bar2)/std_error   # test statistic
pnorm(q = test_statistic, lower.tail = FALSE)   # right-tailed p-value
```

```
[1] 5.401347e-05
```

Our p-value is smaller than α, so we can reject the null hypothesis and conclude that the difference between the two proportions is significantly greater than zero.

But we did not need to compare the difference to zero; we could have used any hypothesized difference for comparison. Let's repeat the above test, comparing the difference to 0.15 instead, as an example.

```
hyp.diff = 0.15                                            # hypothesized difference
std_error <- sqrt(p_bar1*(1-p_bar1)/n1
                + p_bar2*(1-p_bar2)/n2)                    # standard error
test_statistic <- ((p_bar1 - p_bar2) - hyp.diff)/std_error # test statistic
pnorm(q = test_statistic, lower.tail = FALSE)              # right-tailed p-value
```

```
[1] 0.08372266
```

Our p-value, 0.0837, is greater than α, so we cannot reject the null hypothesis and cannot conclude that the difference between these two proportions is significantly greater than 0.15.

Left-tailed test

In a left-tailed test, the null hypothesis states that the difference between the two proportions is greater than or equal to a hypothesized value. Let's begin by using zero as our hypothesized value, $H_0 : \bar{p}_1 - \bar{p}_2 \geq 0$.

We repeat some code below that we've seen above, just to make it easy to copy and paste the example elsewhere.

```
p_bar <- (90 + 50) / (150 + 135)                 # overall proportion
std_error <- sqrt(p_bar*(1-p_bar)*(1/n1+1/n2))   # standard error
test_statistic <- (p_bar1 - p_bar2)/std_error    # test statistic
pnorm(q = test_statistic, lower.tail = TRUE)     # left-tailed p-value
```

```
[1] 0.999946
```

Our p-value, 0.9999, is greater than α, so we cannot reject the null hypothesis and cannot conclude that the difference between the two proportions is significantly less than zero.

But we did not need to compare the difference to zero; we could have used any hypothesized difference for comparison. Let's repeat the above test, comparing the difference to 0.15 instead, as an example.

```
hyp.diff = 0.15                                             # hypothesized difference
std_error <- sqrt(p_bar1*(1-p_bar1)/n1
                + p_bar2*(1-p_bar2)/n2)                     # standard error
test_statistic <- ((p_bar1 - p_bar2) - hyp.diff)/std_error  # test statistic
pnorm(q = test_statistic, lower.tail = TRUE)                # left-tailed p-value
```

```
[1] 0.9162773
```

Our p-value, 0.91627, is greater than α, so we cannot reject the null hypothesis and cannot conclude that the difference between these two proportions is significantly less than 0.15.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a hypothesis test for the ratio of two population variances

Description

Let's say we want to compare the variability of two populations. We take two samples of data, $x_1, x_2, x_3, \ldots, x_k$ from population 1 and $x'_1, x'_2, x'_3, \ldots, x'_k$ from population 2. What hypothesis tests can help us compare the population variances?

Related tasks:

- How to compute a confidence interval for the difference between two proportions
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion
- How to do a hypothesis test for population variance
- How to do a hypothesis test for the difference between means when both population variances are known
- How to do a hypothesis test for the difference between two proportions
- How to do a hypothesis test for the mean with known standard deviation
- How to do a hypothesis test of a coefficient's significance (on website)
- How to do a one-sided hypothesis test for two sample means
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)

Solution in pure R

We'll use R's dataset `EuStockMarkets` to do an example. This dataset has information on the daily closing prices of 4 European stock indices. We're going to compare the variability of Germany's DAX and France's CAC closing prices.

Let's load the dataset. (See how to quickly load some sample data (on website).) If using your own data, place it into the `sample1` and `sample2` variables instead of using the code below.

```
# install.packages("datasets") # If you have not already done so
library(datasets)

# Load the dataset and convert it to a data frame, then extract two columns
EuStockMarkets <- data.frame(EuStockMarkets)
sample.1 <- EuStockMarkets$DAX
sample.2 <- EuStockMarkets$CAC
```

Two-tailed test

For all tests below, we will use $\alpha = 0.05$ as our Type I Error Rate, but any value between 0.0 and 1.0 can be used.

Two-tailed test

We can use a two-tailed test to test whether the two population variances are equal. Specifically, the null hypothesis will be:

$$H_0 : \frac{\sigma_1^2}{\sigma_2^2} = 1$$

```
sample.1.df <- length(sample.1) - 1            # degrees of freedom
sample.2.df <- length(sample.2) - 1            # degrees of freedom
test.statistic <- var(sample.1)/var(sample.2)  # test statistic
2*pf(test.statistic, df1=sample.1.df, df2=sample.2.df, lower.tail=FALSE) # p-value
```

```
[1] 7.729079e-151
```

Our p-value is smaller than our chosen alpha, so we have sufficient evidence to reject the null hypothesis. The ratio of the variance of the closing prices on Germany's DAX and France's CAC is significantly different than 1, so the variances are not equal.

Right-tailed test

In a right-tailed test, the null hypothesis is that the ratio is less than or equal to 1. This is equivalent to asking if $\sigma_1^2 \leq \sigma_2^2$.

$$H_0 : \frac{\sigma_1^2}{\sigma_2^2} \leq 1$$

We repeat below some of the code above to make each example easy to copy and paste.

```
sample.1.df <- length(sample.1) - 1            # degrees of freedom
sample.2.df <- length(sample.2) - 1            # degrees of freedom
test.statistic <- var(sample.1)/var(sample.2)  # test statistic
pf(test.statistic, df1=sample.1.df, df2=sample.2.df, lower.tail=FALSE) # p-value
```

```
[1] 3.86454e-151
```

Our p-value is smaller than our chosen alpha, so we have sufficient evidence to reject the null hypothesis. The ratio of the variance of the closing prices on Germany's DAX and France's CAC is significantly greater than 1, so the variance of closing prices on Germany's DAX is greater than that of closing prices on France's CAC.

To test whether $\sigma_1^2 \geq \sigma_2^2$, simply swap the roles of the two data columns in the above code.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a Kruskal-Wallis test

Description

If we have samples from several independent populations, we might want to test whether the population medians are equal. We may not be able to assume anything about the populations' variances, nor whether they are normally distributed, but we do assume that the populations have distributions that are approximately the same shape. The Kruskal-Wallis Test will allow us to test the medians for equality. It is similar to a One-Way ANOVA but using medians instead of means. How do we perform a Kruskal-Wallis Test?

Related tasks:

- How to do a one-way analysis of variance (ANOVA)
- How to use Bonferroni's Correction method
- How to do a Wilcoxon rank-sum test

Solution in pure R

For the purposes of this example, let's say we have a sample of GPAs from matriculated students at three Ivy League institutions: Harvard, Dartmouth, and Columbia. This is example data, and you can replace it with your actual data when you re-use this code.

R requires that our categories and our numeric sample values be in separate vectors. We could structure our data as follows.

```
gpas <- c( 3.40, 3.66, 3.90, 3.55, 3.90, 3.58,
           3.90, 3.97, 3.92, 3.83, 4.00, 3.68,
           4.00, 3.75, 3.34 )
schools <- c(
    "Harvard", "Harvard", "Harvard", "Harvard", "Harvard", "Harvard",
    "Dartmouth", "Dartmouth", "Dartmouth", "Dartmouth", "Dartmouth", "Dartmouth",
    "Columbia", "Columbia", "Columbia" )
```

The Kruskal-Willis Test uses a null hypothesis that the category medians are equal, $H_0 : m_C = m_H = m_D \leq 0$. We choose α, or the Type I error rate, as 0.05 and run the test as shown below.

```
kruskal.test(gpas, schools)
```

```
    Kruskal-Wallis rank sum test

data:  gpas and schools
Kruskal-Wallis chi-squared = 3.706, df = 2, p-value = 0.1568
```

The p-value, 0.1568, is greater than α, so we fail to reject the null hypothesis. We do not have sufficient evidence to conclude that the median GPAs of matriculated students at these three schools are different from each other.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a one-sided hypothesis test for two sample means

Description

If we have two samples, $x_1, \ldots, x_n$ and $x'_1, \ldots, x'_n$, and we compute the mean of each one, we might want to ask whether one mean is less than the other. Or more precisely, is their difference significantly less than zero?

Related tasks:

- How to compute a confidence interval for a population mean
- How to do a two-sided hypothesis test for a sample mean (on website)
- How to do a two-sided hypothesis test for two sample means (on website)
- How to do a one-way analysis of variance (ANOVA)
- How to do a hypothesis test for a mean difference (matched pairs)
- How to do a hypothesis test for a population proportion

Solution in pure R

If we call the mean of the first sample $\bar{x}_1$ and the mean of the second sample $\bar{x}_2$, then this is a left-tailed test with the null hypothesis $H_0 : \bar{x}_1 - \bar{x}_2 \geq 0$. We choose a value $0 \leq \alpha \leq 1$ as the probability of a Type I error (false positive, finding we should reject H_0 when it's actually true).

```
# Replace these first three lines with the values from your situation.
alpha <- 0.10
sample1 <- c( 6, 9, 7, 10, 10, 9 )
sample2 <- c( 12, 14, 10, 17, 9 )

# Run a one-sample t-test and print out alpha, the p value,
# and whether the comparison says to reject the null hypothesis.
t.test( sample1, sample2, conf.level=1-alpha, alternative = "less" )
```

```
    Welch Two Sample t-test

data:  sample1 and sample2
t = -2.4617, df = 5.7201, p-value = 0.02549
alternative hypothesis: true difference in means is less than 0
90 percent confidence interval:
      -Inf -1.605229
sample estimates:
mean of x mean of y
      8.5      12.4
```

Although we can deduce the answer to our question from the above output, by comparing the p-value with α manually, we can also ask R to do it.

```
# Is there enough evidence to reject the null hypothesis?
result <- t.test( sample1, sample2, conf.level=1-alpha, alternative = "less" )
result$p.value < alpha
```

```
[1] TRUE
```

In this case, the samples give us enough evidence to reject the null hypothesis at the $\alpha = 0.10$ level. The data suggest that $\bar{x}_1 < \bar{x}_2$.

Here we did not assume that the two samples had equal variance. If in your case they do, you can pass the parameter `var.equal=TRUE` to `t.test`.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a Wilcoxon rank-sum test

Description

Assume we have two independent samples of data, $x_1, x_2, x_3, \ldots x_n$ and $x'_1, x'_2, x'_3, \ldots x'_m$, each from a different population. Also assume that the sample sizes are small or the populations are not normally distributed, but that the two population distributions are approximately the same shape. How can we test whether there is a significant difference between the two medians (or if one is significantly greater than or less than the other)? One method is the Wilcoxon Rank-Sum Test.

Related tasks:

- How to do a Kruskal-Wallis test
- How to do a Wilcoxon signed-rank test
- How to do a Wilcoxon signed-rank test for matched pairs

Solution in pure R

We're going to use fake data for illustrative purposes, but you can replace our fake data with your real data. Say our first sample, $x_1, x_2, x_3, \ldots x_k$, has median m_1, and our second sample, $x'_1, x'_2, x'_3, \ldots x'_k$, has median m_2.

```
# Replace sample1 and sample2 with your data
sample1 <- c(56, 31, 190, 176, 119, 15, 140, 152, 167)
sample2 <- c(45, 36, 78, 54, 12, 25, 39, 48, 52, 70, 85)
```

We choose a value, $0 \leq \alpha \leq 1$, as the Type I Error Rate. We'll let α be 0.05.

Two-tailed test

To test the null hypothesis $H_0 : m_1 - m_2 = 0$, that is, $m_1 = m_2$, we use a two-tailed test:

```
wilcox.test(sample1, sample2, alternative = "two.sided", mu = 0, paired = FALSE)
```

```
    Wilcoxon rank sum exact test

data:  sample1 and sample2
W = 77, p-value = 0.03813
alternative hypothesis: true location shift is not equal to 0
```

Our p-value, 0.03813, is less than $\alpha = 0.05$, so we have sufficient evidence to reject the null hypothesis. The population medians are significantly different from each other.

(The output above is slightly different than the output you would get by running this test in Python, because SciPy uses a normal distribution internally, but R uses a Wilcoxon distribution.)

Right-tailed test

To test the null hypothesis $H_0 : m_1 - m_2 \leq 0$, that is, $m_1 \leq m_2$, we use a right-tailed test:

```
wilcox.test(sample1, sample2, alternative = "greater", mu = 0, paired = FALSE)
```

```
	Wilcoxon rank sum exact test

data:  sample1 and sample2
W = 77, p-value = 0.01906
alternative hypothesis: true location shift is greater than 0
```

Our p-value, 0.01906, is less than $\alpha = 0.05$, so we have sufficient evidence to reject the null hypothesis. The first population medians is significantly greater second.

(The output above is slightly different from the output you would get by running this test in Python, because SciPy uses a normal distribution internally, but R uses a Wilcoxon distribution.)

Left-tailed test

To test the null hypothesis $H_0 : m_1 - m_2 \geq 0$, that is, $m_1 \geq m_2$, we use a left-tailed test:

```r
wilcox.test(sample1, sample2, alternative = "less", mu = 0, paired = FALSE)
```

```
	Wilcoxon rank sum exact test

data:  sample1 and sample2
W = 77, p-value = 0.9845
alternative hypothesis: true location shift is less than 0
```

Our p-value, 0.9845, is greater than α, so we do not have sufficient evidence to reject the null hypothesis. The first population median is not significantly smaller than the second population median.

(The output above is slightly different from the output you would get by running this test in Python, because SciPy uses a normal distribution internally, but R uses a Wilcoxon distribution.)

NOTE: If there are ties in the data and there are fewer than 50 observations in each sample, then R will compute a p-value using the normal approximation, and there will be an error message indicating that the exact p-value cannot be calculated.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a Wilcoxon signed-rank test

Description

Assume we a sample of data, $x_1, x_2, x_3, \dots x_k$ and either the sample size is small or the population is not normally distributed. But we still want to perform tests that compare the sample median to a hypothesized value (equal, greater, or less). One method is the Wilcoxon Signed-Rank Test.

Related tasks:

- How to do a Kruskal-Wallis test
- How to do a Wilcoxon rank-sum test
- How to do a Wilcoxon signed-rank test for matched pairs

Solution in pure R

We're going to use fake data for illustrative purposes, but you can replace our fake data with your real data. Say our sample, $x_1, x_2, x_3, \dots x_k$, has median m.

```
# Replace the next line with your data
sample <- c(19, 4, 23, 16, 1, 8, 30, 25, 13)
```

We choose a value, $0 \le \alpha \le 1$, as the Type I Error Rate. We'll let α be 0.05. In the examples below, we will be comparing the median m to a hypothesized value of $a = 10$, but you can use any value for a.

Two-tailed test

To test the null hypothesis $H_0 : m = a$, we use a two-tailed test:

```
a <- 10
wilcox.test(sample, mu = a, alternative = "two.sided")
```

```
Warning message in wilcox.test.default(sample, mu = a, alternative = "two.sided"):
“cannot compute exact p-value with ties”




    Wilcoxon signed rank test with continuity correction

data:  sample
V = 35, p-value = 0.1544
alternative hypothesis: true location is not equal to 10
```

Our p-value, 0.1544, is greater than $\alpha = 0.05$, so we do not have sufficient evidence to reject the null hypothesis. We may continue to assume the population median is equal to 10.

Right-tailed test

To test the null hypothesis $H_0 : m \ge a$, we use a right-tailed test:

```
wilcox.test(sample, mu = a, alternative = "less")
```

```
Warning message in wilcox.test.default(sample, mu = a, alternative = "less"):
“cannot compute exact p-value with ties”




    Wilcoxon signed rank test with continuity correction

data:  sample
V = 35, p-value = 0.9386
alternative hypothesis: true location is less than 10
```

Our p-value, 0.9386, is greater than $\alpha = 0.05$, so we do not have sufficient evidence to reject the null hypothesis. We may continue to assume the population median is less than (or equal to) 10.

Left-tailed test

To test the null hypothesis $H_0 : m \leq a$, we use a left-tailed test:

```
wilcox.test(sample, mu = a, alternative = "greater")
```

```
Warning message in wilcox.test.default(sample, mu = a, alternative = "greater"):
“cannot compute exact p-value with ties”




    Wilcoxon signed rank test with continuity correction

data:  sample
V = 35, p-value = 0.0772
alternative hypothesis: true location is greater than 10
```

Our p-value, 0.0772, is greater than α, so we do not have sufficient evidence to reject the null hypothesis. We may continue to assume the population median is greater than (or equal to) 10.

NOTE: If there are ties in the data and there are fewer than 50 observations in each sample, then R will compute a p-value using the normal approximation, and there will be an error message indicating that the exact p-value cannot be calculated.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a Wilcoxon signed-rank test for matched pairs

Description

Assume we have two samples of data that come in matched pairs, $x_1, x_2, x_3, \ldots x_k$ and $x'_1, x'_2, x'_3, \ldots x'_k$, which we might pair up as $(x_1, x'_1), (x_2, x'_2), \ldots, (x_k, x'_k)$. The two samples may be from different populations. Also assume that the sample sizes are small or the populations are not normally distributed.

Consider measuring the difference in each pair, $x_1 - x'_1, x_2 - x'_2, \ldots, x_k - x'_k$. We want to perform tests that compare the median of those differences, m_D, to a hypothesized value (equal, greater, or less). One method is the Wilcoxon Signed-Rank Test for Matched Pairs.

Related tasks:

- How to do a Kruskal-Wallis test
- How to do a Wilcoxon rank-sum test
- How to do a Wilcoxon signed-rank test

Solution in pure R

The method we will use is equivalent to subtracting the two samples and then performing the signed-rank test. See how to do a Wilcoxon signed-rank test to compare the two methods.

We're going to use fake data for illustrative purposes, but you can replace our fake data with your real data.

```r
# Replace sample1 and sample2 with your data
sample1 <- c(156, 133, 90, 176, 119, 120, 40, 52, 167, 80)
sample2 <- c(45, 36, 78, 54, 12, 25, 39, 48, 52, 70)
```

We choose a value, $0 \leq \alpha \leq 1$, as the Type I Error Rate. We'll let α be 0.05.

Two-tailed test

To test the null hypothesis $H_0 : m_D = 0$, we use a two-tailed test:

```r
wilcox.test(sample1, sample2, alternative = "two.sided", mu = 0, paired = TRUE)
```

```
    Wilcoxon signed rank exact test

data:  sample1 and sample2
V = 55, p-value = 0.001953
alternative hypothesis: true location shift is not equal to 0
```

Our p-value, 0.00195, is less than $\alpha = 0.05$, so we have sufficient evidence to reject the null hypothesis. The median difference is significantly different from zero.

Right-tailed test

To test the null hypothesis $H_0 : m_D \leq 0$, we use a right-tailed test:

```r
wilcox.test(sample1, sample2, alternative = "greater", mu = 0, paired = TRUE)
```

```
    Wilcoxon signed rank exact test

data:  sample1 and sample2
V = 55, p-value = 0.0009766
alternative hypothesis: true location shift is greater than 0
```

Our p-value, 0.0009766, is less than $\alpha = 0.05$, so we have sufficient evidence to reject the null hypothesis. The median difference is significantly greater than zero.

Left-tailed test

To test the null hypothesis $H_0 : m_D \geq 0$, we use a left-tailed test:

```r
wilcox.test(sample1, sample2, alternative = "less", mu = 0, paired = TRUE)
```

```
    Wilcoxon signed rank exact test

data:  sample1 and sample2
V = 55, p-value = 1
alternative hypothesis: true location shift is less than 0
```

Our p-value, 1.0, is greater than α, so we do not have sufficient evidence to reject the null hypothesis. We should continue to assume that the mean difference may be less than (or equal to) zero.

NOTE: If there are ties in the data and there are fewer than 50 observations in each sample, then R will compute a *p*-value using the normal approximation, and there will be an error message indicating that the exact *p*-value cannot be calculated.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for a single population variance

Description

Let's say we want to compute a confidence interval for the variability of a population. We take a sample of data, $x_1, x_2, x_3, \ldots, x_k$ and compute its variance, s^2. How do we construct a confidence interval for the population variance σ^2?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the expected value of a response variable
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

We'll use R's dataset EuStockMarkets here. This dataset has information on the daily closing prices of 4 European stock indices. We're going to look at the variability of Germany's DAX closing prices. Feel free to replace this sample data with your actual data if you use this code.

```r
# install.packages("datasets") # if you have not done so already
library(datasets)

# Load stock market data, convert to DataFrame, and choose the DAX column.
EuStockMarkets <- data.frame(EuStockMarkets)
sample <- EuStockMarkets$DAX
```

Now that we have our sample data loaded, let's go ahead and make the confidence interval.

```r
# Find the critical values from the right and left tails of the Chi-square distribution
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
n <- length(sample)
left_critical_val <- qchisq(p = alpha/2, df = n-1, lower.tail=FALSE)
right_critical_val <- qchisq(p = 1-alpha/2, df = n-1, lower.tail=FALSE)

# Find the upper and lower bounds of the confidence interval and print them out
lower_bound <- ((n - 1)*var(sample))/left_critical_val
upper_bound <- ((n - 1)*var(sample))/right_critical_val
lower_bound
upper_bound
```

```
[1] 1104642



[1] 1256248
```

Our 95% confidence interval for the standard deviation of Germany's DAX closing prices is [1104642, 1256248].

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform a chi-squared test on a contingency table

Description

If we have a contingency table showing the frequencies observed in two categorical variables, how can we run a χ^2 test to see if the two variables are independent?

Solution in pure R

Here we will use a 2×4 matrix to store a contingency table of education vs. gender, taken from Penn State University's online stats review website. You should use your own data. (Note: R's `table` function is useful for creating contingency tables from data.)

```
data <- matrix( c( 60, 54, 46, 41, 40, 44, 53, 57 ), ncol = 4,
                dimnames=list( c('F','M'), c('HS','BS','MS','PhD') ),
                byrow =TRUE)
data
```

```
  HS BS MS PhD
F 60 54 46 41
M 40 44 53 57
```

The χ^2 test's null hypothesis is that the two variables are independent. We choose a value $0 \le \alpha \le 1$ as the probability of a Type I error (false positive, finding we should reject H_0 when it's actually true).

R provides a `chisq.test` function that does exactly what we need.

```
results <- chisq.test( data )
results
```

```

    Pearson's Chi-squared test

data:  data
X-squared = 8.0061, df = 3, p-value = 0.04589
```

We can manually compare the p-value to an α we've chosen, or ask R to do it.

```
alpha <- 0.05            # or choose your own alpha here
results$p.value < alpha  # reject the null hypothesis?
```

```
[1] TRUE
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a goodness of fit test for a multinomial experiment

Description

If we have historical values for multiple population proportions, plus more recent samples from those same populations, we may want to compare to see if the proportions appear to have changed. This is called a goodness of fit test for a multinomial experiment. How can we execute it?

Solution in pure R

Let's say we have a dataset with the previous population proportions for four categories. (This is contrived data, but the code below can be used on your actual data.)

Category	Frequency	Proportion
A	43	0.25
B	62	0.36
C	52	0.30
D	16	0.09

We have also taken a more recent sample and found the number of observations from it that belong to each category. We want to determine if the proportions coming from the recent sample are equal to the previous proportions.

R expects that we will have two vectors, one with the expected number of observations in each group (from the previous, or hypothesized proportions) and the other with the actual number of observations in each group (from the more recent sample). R also expects that the total number of observations in each vector is the same. We'll create two vectors below with the fake data from above, but you can replace them with your real data

```r
# our fake data:
old.observations <- c(43, 62, 52, 16)
new.observations <- c(56, 80, 12, 25)
# now organized into a data frame:
categories <- c("A", "B", "C", "D")
data <- data.frame(categories, old.observations, new.observations)
```

We set the null hypothesis to be that the proportions of each category from the recent sample are equal to the previous proportions.

$$H_0 : p_A = 0.25 \text{ and } \; p_B = 0.36 \text{ and } \; p_C = 0.30 \text{ and } \; p_D = 0.09.$$

We choose a value $0 \leq \alpha \leq 1$ as our Type 1 error rate. We'll let α be 0.05 here.

```r
# Run the Chi-Square test, giving the test statistic and p-value
chisq.test(data$new.observations, p=data$old.observations, rescale.p=TRUE)
```

```
    Chi-squared test for given probabilities

data:  data$new.observations
X-squared = 44.988, df = 3, p-value = 9.308e-10
```

Our p-value is less than α, so we have sufficient evidence to reject the null hypothesis. It does appear that the proportion of at least one of the four categories is significantly different now from what it was previously.

If instead you provided the population proportions as the old observations, that is, a vector of values that sum to 1, you can omit the `rescale.p` argument.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a one-way analysis of variance (ANOVA)

Description

If we have multiple independent samples of the same quantity (such as students' SAT scores from several different schools), we may want to test whether the means of each of the samples are the same. Analysis of Variance (ANOVA) can determine whether any two of the sample means differ significantly. How can we do an ANOVA?

Related tasks:

- How to do a two-sided hypothesis test for two sample means (on website) (which is just an ANOVA with only two samples)
- How to do a two-way ANOVA test with interaction
- How to do a two-way ANOVA test without interaction
- How to compare two nested linear models (on website)
- How to conduct a mixed designs ANOVA (on website)
- How to conduct a repeated measures ANOVA (on website)
- How to perform an analysis of covariance (ANCOVA)
- How to do a Kruskal-Wallis test

Solution in pure R

R expects you to have all the samples in one vector, and the groups they came from in a separate, categorical vector. So, for example, if we had SAT scores from four different schools (named A, B, C, and D), then our data might be arranged like this.

```
SAT.scores <- c(
    1100, 1250, 1390, 970, 1510, 1010, 1050, 1090, 1110,
    900, 1550, 1300, 1270, 1210, 900, 850, 1110, 1070, 910, 920
)
school.names <- c(
    'A', 'A', 'A', 'A', 'A', 'B', 'B', 'B', 'B',
    'C', 'C', 'C', 'C', 'C', 'D', 'D', 'D', 'D', 'D', 'D'
)
```

ANOVA tests the null hypothesis that all group means are equal. You choose α, the probability of Type I error (false positive, finding we should reject H_0 when it's actually true). I will use $\alpha = 0.05$ in this example.

```
# Run a one-way ANOVA and print a summary of all the output
result <- aov( SAT.scores ~ school.names )
summary( result )
```

```
             Df Sum Sq Mean Sq F value Pr(>F)
school.names  3 321715  107238   3.689 0.0342 *
Residuals    16 465140   29071
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value reported in that output is 0.0433. You could manually check whether $p < \alpha$. Since it is, we would reject H_0, and therefore conclude that at least one pair of means is statistically significantly different.

Or you could ask R to do the comparison for you, but getting the p-value from the ANOVA summary is fiddly:

```
alpha <- 0.05
p.value <- unname( unlist( summary( result ) ) )[9]
p.value < alpha
```

```
[1] TRUE
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a two-way ANOVA test with interaction

Description

When we analyze the impact that two factors have on a response variable, we often consider the possible relationship between the two factors. That is, does their interaction term affect the response variable? A two-way ANOVA test with interaction can answer that question.

Related tasks:

- How to do a one-way analysis of variance (ANOVA)
- How to do a two-way ANOVA test without interaction
- How to compare two nested linear models (on website) using ANOVA
- How to conduct a mixed designs ANOVA (on website)
- How to conduct a repeated measures ANOVA (on website)
- How to perform an analysis of covariance (ANCOVA)

Solution in pure R

We're going to use R's `esoph` dataset, about esophageal cancer cases. We will focus on the impact of age group (`agegp`) and alcohol consumption (`alcgp`) on the number of cases of the cancer (`ncases`). We ask, does the interaction between these two factors affect the number of cases?

First, we load in the dataset. (See how to quickly load some sample data (on website).)

```
# install.packages("datasets") # if you have not already done this
library(datasets)
data <- esoph
head(data)
```

```
  agegp alcgp     tobgp    ncases ncontrols
1 25-34 0-39g/day 0-9g/day 0      40
2 25-34 0-39g/day 10-19    0      10
3 25-34 0-39g/day 20-29    0       6
4 25-34 0-39g/day 30+      0       5
5 25-34 40-79     0-9g/day 0      27
6 25-34 40-79     10-19    0       7
```

Next, we create a model that includes the response variable we care about, plus the two categorical variables we will be testing, as well as their interaction term.

```
# the * below means multiplication, to create an interaction term
model <- aov(ncases ~ agegp*alcgp, data = data)
```

A two-way ANOVA with interaction tests the following three null hypotheses.

1. There is no interaction between the two categorical variables. (If we reject this we do not test the other two hypotheses.)
2. The mean response is the same across all groups of the first factor. (In our example, that says the mean `ncases` is the same for all age groups.)
3. The mean response is the same across all groups of the second factor. (In our example, that says the mean `ncases` is the same for all alcohol consumption groups.)

We choose a value, $0 \leq \alpha \leq 1$, as the Type I Error Rate. Let's let $\alpha = 0.05$ here.

```
summary(model)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
agegp        5  261.2   52.24  14.048 2.89e-09 ***
alcgp        3   52.7   17.57   4.723  0.00486 ** 
agegp:alcgp 15  107.6    7.17   1.928  0.03633 *  
Residuals   64  238.0    3.72                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value for the interaction of age group and alcohol consumption is in the third row, final column, 0.03633. It is less than α, so we can reject the null hypothesis that age group and alcohol consumption do not interact with regards to the number of esophageal cancer cases. That is, we have reason to believe that their interaction does effect the outcome.

As we stated when we listed the hypotheses to test, since we rejected the first null hypothesis, we will not test the other two. In the case where you failed to reject the first null hypothesis, you could consider each p-value in the first two rows of the above table, one for each of the two factors.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to do a two-way ANOVA test without interaction

Description

When we analyze the impact that two factors have on a response variable, we may know in advance that the two factors do not interact. How can we use a two-way ANOVA test to test for an effect from each factor without including an interaction term for the two factors?

Related tasks:

- How to do a one-way analysis of variance (ANOVA)
- How to do a two-way ANOVA test without interaction
- How to compare two nested linear models (on website) using ANOVA
- How to conduct a mixed designs ANOVA (on website)
- How to conduct a repeated measures ANOVA (on website)
- How to perform an analysis of covariance (ANCOVA)

Solution in pure R

We're going to use R's `esoph` dataset, about esophageal cancer cases. We will focus on the impact of age group (`agegp`) and alcohol consumption (`alcgp`) on the number of cases of the cancer (`ncases`). We ask, does either of these two factors affect the number of cases?

First, we load in the dataset. (See how to quickly load some sample data (on website).)

```r
# install.packages("datasets") # if you have not already done this
library(datasets)
data <- esoph
head(data)
```

```
  agegp alcgp     tobgp    ncases ncontrols
1 25-34 0-39g/day 0-9g/day 0      40
2 25-34 0-39g/day 10-19    0      10
3 25-34 0-39g/day 20-29    0       6
4 25-34 0-39g/day 30+      0       5
5 25-34 40-79     0-9g/day 0      27
6 25-34 40-79     10-19    0       7
```

Next, we create a model that includes the response variable we care about, plus the two categorical variables we will be testing. We simply omit the interaction term. (If you wish to include it, see how to do a two-way ANOVA test with interaction.)

```r
# the * below means multiplication, to create an interaction term
model <- aov(ncases ~ agegp + alcgp, data = data)
```

A two-way ANOVA with interaction tests the following two null hypotheses.

1. The mean response is the same across all groups of the first factor. (In our example, that says the mean `ncases` is the same for all age groups.)
2. The mean response is the same across all groups of the second factor. (In our example, that says the mean `ncases` is the same for all alcohol consumption groups.)

We choose a value, $0 \leq \alpha \leq 1$, as the Type I Error Rate. Let's let $\alpha = 0.05$ here.

```
summary(model)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
agegp        5  261.2   52.24  11.943 1.28e-08 ***
alcgp        3   52.7   17.57   4.016   0.0103 *  
Residuals   79  345.6    4.37                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value for the alcohol consumption factor is in the first row, final column, 1.029452×10^{-2}. It is less than α, so we can reject the null hypothesis that alcohol consumption does not affect the number of esophageal cancer cases. That is, we have reason to believe that it does affect the number of cases.

The p-value for the age group factor is in the second row, final column, 8.907998×10^{-9}. It is less than α, so we can reject the null hypothesis that age group does not affect the number of esophageal cancer cases. Again, we have reason to believe that it does affect the number of cases.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute Fisher’s confidence intervals

Description

If we run a one-way ANOVA test and find that there is a significant difference between population means, we might want to know which means are actually different from each other. One way to do so is with Fisher’s Least Significant Difference Confidence Intervals, which forms a confidence interval for each pair of samples. How do we go about making these confidence intervals?

Solution in pure R

We will use some fake data for the purposes of an example, but you can replace it with your real data in the code below. Consider an ice cream shop’s sales data over several weekends.

```
num.transactions <- c(91, 134, 98, 105, 93, 89, 145, 132, 109,
                      94, 105, 99, 84, 128, 120, 115, 118)
days <- c("Fri", "Sun", "Sun", "Sat", "Fri", "Fri", "Sat", "Sun", "Sun",
          "Fri", "Sat", "Sat", "Fri", "Sun", "Fri", "Sat", "Sun")
```

Let’s assume that you have already performed an ANOVA on this data, as shown below. (If you’re not familiar with ANOVA, see how to do a one-way ANOVA test.) Let’s assume that we chose α to be 0.05.

```
model <- aov(num.transactions ~ days)
summary(model)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
days         2   1965   982.7   4.348  0.034 *
Residuals   14   3164   226.0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the p-value in the `Pr(>F)` column, we can see that, at the 5% significance level, there are significant differences between the mean number of transactions at the ice cream shop across these weekend days.

We’ll use the `LSD.test` function (Least Significant Difference) from R’s `agricolae` package to get the confidence intervals for each pair of days. Let’s use $\alpha = 0.05$ again so that we get 95% confidence intervals.

```
# install.packages("agricolae") # if you have not already done so
library(agricolae)

test <- LSD.test(model, alpha=0.05, "days")
test
```

```
$statistics
   MSerror Df     Mean       CV
  226.0333 14 109.3529 13.74851

$parameters
        test p.ajusted name.t ntr alpha
  Fisher-LSD      none   days   3  0.05

$means
    num.transactions      std r      se       LCL      UCL Min Max    Q25 Q50
Fri         95.16667 12.67149 6 6.13777  82.00246 108.3309  84 120  89.50  92
Sat        113.80000 18.36301 5 6.72359  99.37933 128.2207  99 145 105.00 105
Sun        119.83333 14.23259 6 6.13777 106.66913 132.9975  98 134 111.25 123
       Q75
Fri  93.75
Sat 115.00
Sun 131.00

$comparison
NULL

$groups
    num.transactions groups
Sun        119.83333      a
Sat        113.80000     ab
Fri         95.16667      b

attr(,"class")
[1] "group"
```

The portion of this lengthy output on which to focus is the `$groups` section. If the categories share a letter in the “groups” column, then their means are not significantly different from each other. Therefore:

- Sunday and Saturday share the letter “a,” so we know that the number of transactions on these two days are not significantly different from each other.
- The same goes for Saturday and Friday, which share the letter “b.”
- But Sunday and Friday do not share a letter, so the number of transactions on these two days is significantly different.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform an analysis of covariance (ANCOVA)

Description

Recall that covariates are variables that may be related to the outcome but are unaffected by treatment assignment. In a randomized experiment with one or more observed covariates, an analysis of covariance (ANCOVA) addresses this question: How would the mean outcome in each treatment group change if all groups were equal with respect to the covariate? The goal is to remove any variability in the outcome associated with the covariate from the unexplained variability used to determine statistical significance.

Related tasks:

- How to do a one-way analysis of variance (ANOVA)
- How to compare two nested linear models (on website)
- How to conduct a mixed designs ANOVA (on website)
- How to conduct a repeated measures ANOVA (on website)

Solution in pure R

The solution below uses an example dataset about car design and fuel consumption from a 1974 Motor Trend magazine. (See how to quickly load some sample data (on website).)

```
df <- mtcars
df$vs <- as.factor(df$vs)
```

Let's use ANCOVA to check the effect of the engine type (0 = V-shaped, 1 = straight, in the variable `vs`) on the miles per gallon when considering the weight of the car as a covariate. We will use the `ancova` function from the `pingouin` package to conduct the test.

```
cov.model <- lm(mpg ~ wt + vs, data = df)
anova(cov.model)
```

```
          Df Sum Sq    Mean Sq     F value    Pr(>F)
wt         1 847.72525 847.725250 109.704168 2.284396e-11
vs         1  54.22806  54.228061   7.017656 1.292580e-02
Residuals 29 224.09388   7.727375         NA           NA
```

The p-value for each variable can be found in the final column of the output, called `Pr(>F)`.

The p-value for the `wt` variable tests the null hypothesis, "The quantities `wt` and `mpg` are not related." Since it is below 0.05, we reject the null hypothesis, and conclude that `wt` is significant in predicting `mpg`.

The p-value for the `vs` variable tests the null hypothesis, "The quantities `vs` and `mpg` are not related if we hold `wt` constant." Since it is below 0.05, we reject the null hypothesis, and conclude that `vs` is significant in predicting `mpg` even among cars with equal weight (`wt`).

If we wish to create a 2-factor ANCOVA model, we can test to see if the engine type (0 = V-shaped, 1 = straight) and transmission type (0 = automatic, 1 = manual) have an effect on the Miles/gallon per car when considering the weight of the car as a covariate.

```
cov.model.2 <- lm(mpg ~ wt + vs + am, data = df)
anova(cov.model.2)
```

```
          Df Sum Sq     Mean Sq    F value    Pr(>F)
wt         1 847.725250 847.725250 109.729918 3.420018e-11
vs         1  54.228061  54.228061   7.019303 1.310627e-02
am         1   7.778149   7.778149   1.006807 3.242621e-01
Residuals 28 216.315728   7.725562         NA           NA
```

The *p*-values are again in the final column of output. They show that at the 5% significance level, we would conclude that engine type (`vs`) significantly impacts the Miles/gallon per car while accounting for the weight of the car (`wt`) but the transmission type (`am`) does not.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform post-hoc analysis with Tukey's HSD test

Description

If we run a one-way ANOVA test and find that there is a significant difference between population means, we might want to know which means are actually different from each other. One way to do so is with Tukey's Honestly Significant Differences (HSD) method. It creates confidence intervals for each pair of samples, while controlling for Type I error rate across all pairs. Thus the resulting intervals are a little wider than those produced using Fisher's LSD method. How do we make these confidence intervals, with an appropriate visualization?

Solution in pure R

We load here the same data that appears in the solution for how to perform pairwise comparisons. That solution used ANOVA to determine which pairs of groups have significant differences in their means; follow its link for more details.

```r
# Load an inbuilt data set called InsectSprays and assign it to the variable df
df <- InsectSprays
head( df, 10 )
```

```
   count spray
1  10    A
2   7    A
3  20    A
4  14    A
5  14    A
6  12    A
7  10    A
8  23    A
9  17    A
10 20    A
```

We now want to perform an unplanned comparison test on the data to determine the magnitudes of the differences between pairs of groups. We do this by applying Tukey's HSD approach to perform pairwise comparisons and generate confidence intervals that maintain a specified experiment-wide error rate. We use R's built-in `TukeyHSD` function, and we give it the same ANOVA results that we computed in the solution for how to perform pairwise comparisons (on website).

```r
aov1 <- aov(count ~ spray, data = df)
TukeyHSD(aov1, "spray", ordered=TRUE, conf.level = 0.95)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = count ~ spray, data = df)

$spray
          diff       lwr       upr     p adj
E-C  1.4166667 -3.282742  6.116075 0.9488669
D-C  2.8333333 -1.866075  7.532742 0.4920707
A-C 12.4166667  7.717258 17.116075 0.0000000
B-C 13.2500000  8.550591 17.949409 0.0000000
F-C 14.5833333  9.883925 19.282742 0.0000000
D-E  1.4166667 -3.282742  6.116075 0.9488669
A-E 11.0000000  6.300591 15.699409 0.0000000
B-E 11.8333333  7.133925 16.532742 0.0000000
F-E 13.1666667  8.467258 17.866075 0.0000000
A-D  9.5833333  4.883925 14.282742 0.0000014
B-D 10.4166667  5.717258 15.116075 0.0000002
F-D 11.7500000  7.050591 16.449409 0.0000000
B-A  0.8333333 -3.866075  5.532742 0.9951810
F-A  2.1666667 -2.532742  6.866075 0.7542147
F-B  1.3333333 -3.366075  6.032742 0.9603075
```

Because the above table contains a lot of information, it's often helpful to visualize these intervals. R lets us do so by simply calling `plot` on the above table. We add a few plotting parameters to improve its appearance.

```
plot( TukeyHSD(aov1, "spray", ordered=TRUE, conf.level = 0.95),
      las=1, cex.axis=0.9 )
```

Confidence intervals that cross the vertical, dashed line at $x = 0$ are those in which the means across those groups may be equal. Other intervals have mean differences whose 95% confidence intervals do not include zero.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to use Bonferroni's Correction method

Description

If we run a one-way ANOVA test and find that there is a significant difference between population means, we might want to know which means are actually different from each other. One way to do so is with the Bonferroni correction. This method runs a t-test for each pair of categories using a conservative confidence level.

Related tasks:

- How to do a one-way analysis of variance (ANOVA)
- How to do a two-sided hypothesis test for two sample means (on website) (which is just an ANOVA with only two samples)
- How to do a Kruskal-Wallis test

Solution in pure R

Let's assume that you have already done an analysis of variance (ANOVA). (See how to do a one-way analysis of variance (ANOVA) for details.)

As an example, we will use the fake data below, which looks at the number of transactions at an ice cream shop on the weekends. Let's assume that we chose α to be 0.05 in that ANOVA.

```r
# Store our fake data in vectors.  (You can replace this with your real data.)
num.transactions <- c(91, 134, 98, 105, 93, 89, 145, 132, 109,
                      94, 105, 99, 84, 128, 120, 115, 118)
days <- c("Fri", "Sun", "Sun", "Sat", "Fri", "Fri", "Sat", "Sun", "Sun",
          "Fri", "Sat", "Sat", "Fri", "Sun", "Fri", "Sat", "Sun")

# Perform an ANOVA and print a summary.
model <- aov(num.transactions ~ days)
summary(model)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)  
days         2   1965   982.7   4.348  0.034 *
Residuals   14   3164   226.0                 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The top-right value in the output is the p-value for the test, 0.034. Because it is below our chosen significance level of $\alpha = 0.05$, there are significant differences between the mean number of transactions at the ice cream shop across at least two of these weekend days. But specifically which two, or is it more than two?

We'll use the `PostHocTest()` function in the `DescTools` package, and specify that we want to use the Bonferroni method to make the confidence intervals for each pair of days. Let's let α be equal to 0.05 again, but the Bonferroni correction implies that the overall probability of a Type I Error in *any* of the tests below is now at most 0.05, rather than each one being 0.05 separately.

```r
# install.packages("DescTools") # If you have not already installed it
library(DescTools)

# Run the test and print the confidence intervals for each pair of days
PostHocTest(model, method = "bonferroni", conf.level = 0.95)
```

```
  Posthoc multiple comparisons of means : Bonferroni
    95% family-wise confidence level

$days
             diff     lwr.ci   upr.ci   pval
Sat-Fri 18.633333  -6.108523 43.37519 0.1798
Sun-Fri 24.666667   1.076232 48.25710 0.0392 *
Sun-Sat  6.033333 -18.708523 30.77519 1.0000

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the output, R has highlighted the second row for us by placing a `*` after it. That is the one row where the p-value (in the final column) is below our chosen $\alpha = 0.05$.

Therefore, the only significant difference in mean number of transactions is between Sundays and Fridays. Notice also that the confidence interval in that row (from `lwr.ci` to `upr.ci`) does not include zero. (In that particular row, the confidence interval is $(1.076232, 48.25710)$.)

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to fit a linear model to two columns of data

Description

Let's say we have two columns of data, one for a single independent variable x and the other for a single dependent variable y. How can I find the best fit linear model that predicts y based on x?

In other words, what are the model coefficients β_0 and β_1 that give me the best linear model $\hat{y} = \beta_0 + \beta_1 x$ based on my data?

Related tasks:

- How to compute R-squared for a simple linear model
- How to fit a multivariate linear model (on website)
- How to predict the response variable in a linear model

Solution in pure R

This solution uses fake example data. When using this code, replace our fake data with your real data.

```r
# Here is the fake data you should replace with your real data.
xs <- c( 393, 453, 553, 679, 729, 748, 817 )
ys <- c(  24,  25,  27,  36,  55,  68,  84 )

# If you need the model coefficients stored in variables for later use, do:
model <- lm( ys ~ xs )
beta0 = model$coefficients[1]
beta1 = model$coefficients[2]

# If you just need to see the coefficients, do this alone:
lm( ys ~ xs )
```

```
Call:
lm(formula = ys ~ xs)

Coefficients:
(Intercept)           xs
   -37.3214       0.1327
```

The linear model in this example is approximately $y = 0.133x - 37.32$.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute a confidence interval for the expected value of a response variable

Description

If we have a simple linear regression model, $y = \beta_0 + \beta_1 x + \epsilon$, where ϵ is some random error, then given any x input, y can be veiwed as a random variable because of ϵ. Let's consider its expected value. How do we construct a confidence interval for that expected value, given a value for the predictor x?

Related tasks:

- How to compute a confidence interval for a mean difference (matched pairs)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a population mean
- How to compute a confidence interval for a single population variance
- How to compute a confidence interval for the difference between two means when both population variances are known
- How to compute a confidence interval for the difference between two means when population variances are unknown
- How to compute a confidence interval for the difference between two proportions
- How to compute a confidence interval for the population proportion
- How to compute a confidence interval for the ratio of two population variances

Solution in pure R

Let's assume that you already have a linear model. We construct an example one here from some fabricated data.

```r
# Make the linear model
x <- c(34, 9, 78, 60, 22, 45, 83, 59, 25)
y <- c(126, 347, 298, 309, 450, 187, 266, 385, 400)
model <- lm(y ~ x)
```

Construct a data frame containing just one entry, the value of the independent variable for which you want to compute the confidence interval. That data frame can then be passed to R's `predict` function to get a confidence interval for the expected value of y.

```r
# Use your chosen value of x below:
data <- data.frame(x=40)
# Compute the confidence interval for y:
predict(model, data, interval="confidence", level=0.95) # or choose a different confidence level; here we use 0
```

```
  fit      lwr     upr
1 313.7217 226.648 400.7954
```

Our 95% confidence interval is $[226.648, 400.7954]$. We can be 95% confident that the true average value of y, given that x is 40, is between 226.648 and 400.7954.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute R-squared for a simple linear model

Description

Let's say we have fit a linear model to two columns of data, one for a single independent variable x and the other for a single dependent variable y. How can we compute R^2 for that model, to measure its goodness of fit?

Related tasks:

- How to fit a linear model to two columns of data
- How to compute adjusted R-squared (on website)

Solution in pure R

We assume you have already fit a linear model to the data, as in the code below, which is explained fully in a separate task, how to fit a linear model to two columns of data.

```
xs <- c( 393, 453, 553, 679, 729, 748, 817 )
ys <- c(  24,  25,  27,  36,  55,  68,  84 )
model <- lm( ys ~ xs )
```

You can get a lot of information about your model from its summary.

```
summary( model )
```

```
Call:
lm(formula = ys ~ xs)

Residuals:
      1       2       3       4       5       6       7 
  9.163   2.199  -9.072 -16.795  -4.431   6.047  12.890 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) -37.32142   18.99544  -1.965  0.10664   
xs            0.13272    0.02959   4.485  0.00649 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.62 on 5 degrees of freedom
Multiple R-squared:  0.8009,	Adjusted R-squared:  0.7611 
F-statistic: 20.12 on 1 and 5 DF,  p-value: 0.006486
```

In particular, it contains the R^2 value.

```
summary( model )$r.squared
```

```
[1] 0.8009488
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to predict the response variable in a linear model

Description

If we have a linear model and a value for each explanatory variable, how do we predict the corresponding value of the response variable?

Related tasks:

- How to fit a linear model to two columns of data
- How to fit a multivariate linear model (on website)

Solution in pure R

Let's assume that you've already built a linear model. We do an example below with fake data, but you can use your own actual data. For more information on the following code, see how to fit a multivariate linear model (on website).

```
x1 <- c( 2,  7,  4,  3, 11, 18,   6, 15,   9,  12)
x2 <- c( 4,  6, 10,  1, 18, 11,   8, 20,  16,  13)
x3 <- c(11, 16, 20,  6, 14,  8,   5, 23,  13,  10)
y  <- c(24, 60, 32, 29, 90, 45, 130, 76, 100, 120)
model <- lm(y ~ x1 + x2 + x3)
```

Let's say we want to estimate y given that $x_1 = 5$, $x_2 = 12$, and $x_3 = 50$. We can use R's `predict()` function as shown below.

```
predict(model, newdata = data.frame(x1 = 5, x2 = 12, x3 = 50))
```

```
        1 
-91.71014 
```

For the given values of the explanatory variables, our predicted response variable is -91.71014.

Note that if you want to compute the predicted values for all the data on which the model was trained, simply call `predict(model)` with no new data, and it defaults to using the training data.

```
predict(model)
```

```
        1         2         3         4         5         6         7         8 
 47.57012  24.35988  42.21531  47.27614 110.86526  70.03098  95.12690  70.91291 
        9        10 
106.52987  91.11264 
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to create a QQ-plot

Description

We often want to know whether a set of data is normally distributed, so that we can deduce what inference tests are appropriate to conduct. If we have a set of data and want to figure out if it comes from a population that follows a normal distribution, one tool that can help is a QQ plot. How do we make and interpret one?

Related tasks:

- How to test data for normality with Pearson's chi-squared test
- How to test data for normality with the D'Agostino-Pearson test
- How to test data for normality with the Jarque-Bera test

Solution in pure R

We're going to use some fake data here by generating random numbers, but you can replace our fake data with your real data in the code below.

```r
# Replace this with your data, such as a variable or column in a DataFrame
values <- c(4, 90, 85, 49, 34, 23, 17, 10, 20, 59, 100, 112, 46, 10, 4, 39, 24, 77, 63, 23, 67, 109, 70)
```

If the data is normally distributed, then we expect that the QQ plot will show the observed values (black circles) falling very clsoe to the red line (the quantiles for the normal distribution).

```r
# Make a QQ plot for the data
qqnorm(values, pch = 1)
# Add the reference line representing what the data should look like if normally distributed
qqline(values, col = "red", lwd = 2)
```

Our observed values fall pretty close to the reference line. In this case, we expected that, because we created fake data that was normally distributed. But for real data, it may not stay so close to the red line.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to test data for normality with Pearson's chi-squared test

Description

We often want to know whether a set of data is normally distributed, so that we can deduce what inference tests are appropriate to conduct. If we have a set of data and want to figure out if it comes from a population that follows a normal distribution, one tool that can help is Pearson's χ^2 test. How do we perform it?

Related tasks:

- How to create a QQ-plot
- How to test data for normality with the D'Agostino-Pearson test
- How to test data for normality with the Jarque-Bera test

Solution in pure R

We're going to use some fake restaurant data, but you can replace our fake data with your real data in the code below. The values in our fake data represent the amount of money that customers spent on a Sunday morning at the restaurant.

```
# Replace your data here
spending <- c(34, 12, 19, 56, 54, 34, 45, 37, 13, 22, 65, 19,
              16, 45, 19, 50, 36, 23, 28, 56, 40, 61, 45, 47, 37)

mean(spending)
sd(spending)
```

```
[1] 36.52



[1] 15.77213
```

We will now conduct a test of the following null hypothesis: The data comes from a population that is normally distributed with mean 36.52 and standard deviation 15.77.

We will use a value $\alpha = 0.05$ as our Type I error rate. The `pearson.test()` function in the `nortest` package can perform Pearson's χ^2 test for normality.

```
# install.packages("nortest") # if you have not already done so
library(nortest)
pearson.test(spending)
```

```
    Pearson chi-square normality test

data:  spending
P = 3.48, p-value = 0.6264
```

The p-value is 0.6264, which is greater than $\alpha = 0.05$, so we fail to reject our null hypothesis. We would continue to operate under our original assumption that the data come from a normally distributed population.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to test data for normality with the D'Agostino-Pearson test

Description

We often want to know whether a set of data is normally distributed, so that we can deduce what inference tests are appropriate to conduct. If we have a set of data and want to figure out if it comes from a population that follows a normal distribution, one tool that can help is the D'Agostino-Pearson test (sometimes also called the D'Agostino-Pearson omnibus test, or the D'Agostino-Pearson k^2 test). How do we perform it?

Related tasks:

- How to create a QQ-plot
- How to test data for normality with Pearson's chi-squared test
- How to test data for normality with the Jarque-Bera test

Solution in R

How to Data does not yet contain a solution for this task in R.

How to test data for normality with the Jarque-Bera test

Description

We often want to know whether a set of data is normally distributed, so that we can deduce what inference tests are appropriate to conduct. If we have a set of data and want to figure out if it comes from a population that follows a normal distribution, one tool that can help is the Jarque-Bera test for normality. How do we perform it?

Related tasks:

- How to create a QQ-plot
- How to test data for normality with the D'Agostino-Pearson test
- How to test data for normality with Pearson's chi-squared test

Solution in R

How to Data does not yet contain a solution for this task in R.