

19.7: Bayesian t-tests

The second type of statistical inference problem discussed in this book is the comparison between two means, discussed in some detail in the chapter on t-tests (Chapter 13). If you can remember back that far, you'll recall that there are several versions of the t-test. The `BayesFactor` package contains a function called `ttestBF()` that is flexible enough to run several different versions of the t-test. I'll talk a little about Bayesian versions of the independent samples t-tests and the paired samples t-test in this section.

19.7.1 Independent samples t-test

The most common type of t-test is the independent samples t-test, and it arises when you have data that look something like this:

```
load( "./rbook-master/data/harpo.Rdata" )
head(harpo)
```

```
##   grade   tutor
## 1    65 Anastasia
## 2    72 Bernadette
## 3    66 Bernadette
## 4    74 Anastasia
## 5    73 Anastasia
## 6    71 Bernadette
```

In this data set, we have two groups of students, those who received lessons from Anastasia and those who took their classes with Bernadette. The question we want to answer is whether there's any difference in the grades received by these two groups of student. Back in [Chapter 13](#) I suggested you could analyse this kind of data using the `independentSamplesTTest()` function in the `lsr` package. For example, if you want to run a Student's t-test, you'd use a command like this:

```
independentSamplesTTest(
  formula = grade ~ tutor,
  data = harpo,
  var.equal = TRUE
)
```

```
##
## Student's independent samples t-test
##
## Outcome variable: grade
## Grouping variable: tutor
##
## Descriptive statistics:
##           Anastasia Bernadette
## mean      74.533      69.056
## std dev.   8.999      5.775
##
## Hypotheses:
## null:      population means equal for both groups
## alternative: different population means in each group
##
## Test results:
## t-statistic: 2.115
## degrees of freedom: 31
## p-value: 0.043
##
## Other information:
## two-sided 95% confidence interval: [0.197, 10.759]
## estimated effect size (Cohen's d): 0.74
```

Like most of the functions that I wrote for this book, the `independentSamplesTTest()` is very wordy. It prints out a bunch of descriptive statistics and a reminder of what the null and alternative hypotheses are, before finally getting to the test results. I wrote it that way deliberately, in order to help make things a little clearer for people who are new to statistics.

Again, we obtain a p-value less than 0.05, so we reject the null hypothesis.

What does the Bayesian version of the t-test look like? Using the `ttestBF()` function, we can obtain a Bayesian analog of Student's independent samples t-test using the following command:

```
ttestBF( formula = grade ~ tutor, data = harpo )
```

```
## Bayes factor analysis
## -----
## [1] Alt., r=0.707 : 1.754927 ±0%
##
## Against denominator:
## Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

Notice that format of this command is pretty standard. As usual we have a `formula` argument in which we specify the outcome variable on the left hand side and the grouping variable on the right. The `data` argument is used to specify the data frame containing the variables. However, notice that there's no analog of the `var.equal` argument. This is because the `BayesFactor` package does not include an analog of the Welch test, only the Student test.²⁷⁰ In any case, when you run this command you get this as the output:

So what does all this mean? Just as we saw with the `contingencyTableBF()` function, the output is pretty dense. But, just like last time, there's not a lot of information here that you actually need to process. Firstly, let's examine the bottom line. The `BFindepSample` part just tells you that you ran an independent samples t-test, and the `JZS` part is technical information

that is a little beyond the scope of this book.²⁷¹ Clearly, there's nothing to worry about in that part. In the line above, the text `Null, mu1-mu2 = 0` is just telling you that the null hypothesis is that there are no differences between means. But you already knew that. So the only part that really matters is this line here:

```
[1] Alt., r=0.707 : 1.754927 @plusorminus0%
```

Ignore the `r=0.707` part: it refers to a technical detail that we won't worry about in this chapter.²⁷² Instead, you should focus on the part that reads `1.754927`. This is the Bayes factor: the evidence provided by these data are about 1.8:1 in favour of the alternative.

Before moving on, it's worth highlighting the difference between the orthodox test results and the Bayesian one. According to the orthodox test, we obtained a significant result, though only barely. Nevertheless, many people would happily accept $p=.043$ as reasonably strong evidence for an effect. In contrast, notice that the Bayesian test doesn't even reach 2:1 odds in favour of an effect, and would be considered very weak evidence at best. In my experience that's a pretty typical outcome. Bayesian methods usually require more evidence before rejecting the null.

19.7.2 Paired samples t-test

Back in Section 13.5 I discussed the `chico` data frame in which students grades were measured on two tests, and we were interested in finding out whether grades went up from test 1 to test 2. Because every student did both tests, the tool we used to analyse the data was a paired samples t-test. To remind you of what the data look like, here's the first few cases:

```
load("./rbook-master/data/chico.rdata")
head(chico)
```

```
##      id grade_test1 grade_test2
## 1 student1      42.9      44.6
## 2 student2      51.8      54.0
## 3 student3      71.7      72.3
## 4 student4      51.6      53.4
## 5 student5      63.5      63.8
## 6 student6      58.0      59.3
```

We originally analysed the data using the `pairedSamplesTTest()` function in the `lsr` package, but this time we'll use the `ttestBF()` function from the `BayesFactor` package to do the same thing. The easiest way to do it with this data set is to use the `x` argument to specify one variable and the `y` argument to specify the other. All we need to do then is specify `paired=TRUE` to tell R that this is a paired samples test. So here's our command:

```
ttestBF(
  x = chico$grade_test1,
  y = chico$grade_test2,
  paired = TRUE
)
```

```
## Bayes factor analysis
## -----
## [1] Alt., r=0.707 : 5992.05 ±0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

At this point, I hope you can read this output without any difficulty. The data provide evidence of about 6000:1 in favour of the alternative. We could probably reject the null with some confidence!

This page titled [19.7: Bayesian t-tests](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Danielle Navarro](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.

- [17.7: Bayesian t-tests](#) by [Danielle Navarro](#) is licensed [CC BY-SA 4.0](#). Original source: <https://bookdown.org/ekothe/navarro26/>.