Equivalence testing with TOSTER in R: A tutorial

Peder M Isager

May 10, 2018

Introduction

This short tutorial will guide you through the steps reqiured to perform an equivalence test in the free software environment R. The example in this tutorial illustrates the procedure for an independent sample *t*-test. Very little effort is needed to substitute the type of test used (e.g. paired sample t-test, correlation, proportion difference), or substitute a set of data more relevant to the user for the arbitrary values used in the example.

Step by step tutorial

Step 1: Install R

R is available for Linux, MacOS, and Windows, and can be downloaded from The Comprehensive R Archive Network (CRAN). If you are unfamiliar with R, a short introduction can be found here.

Step 2: Install RStudio (optional)

RStudio is a free integrated development environment (IDE) for R. It can be installed from www.rstudio.com Rstudio serves to make the experience of working with R more user-friendly, especially for new users. Its layout is similar to that of the Matlab IDE, or the Spyder IDE for the Python language.

Step 3: Install the TOSTER package

After installation, open R or Rstudio. In the console window, run the following command:

```
install.packages("TOSTER")
```

This will download and install the R package TOSTER from CRAN. The TOSTER package contains a wide variety of functions relevant for conducting equivalence tests, including functions for performing power analysis for an equivalence test. A more comprehensive introduction to the package can be found here.

Step 4: Obtain required statistics from your data

You can conduct a two-one-sided-test of equivalence (TOST) in TOSTER using only summary statistics of data. Let us conduct a TOST for an independent samples t-test as an example. For this we need the group means, standard deviations, and sample sizes from the data. Run the following lines of code in you R console to define these parameters (using some arbitrary values) in R:

```
group1_mean <- 0
group2_mean <- 0.2

group1_stdev <- 0.5
group2_stdev <- 0.4

group1_n <- 32
group2_n <- 34
```

In addtion, we need to define the equivalence bounds (stated in values of Cohen's d). Run the following lines of code in you R console to define these parameters in R:

```
lower_bound <- -1
upper_bound <-  1
```

Step 5: Load the TOSTER package

To use the TOSTER package in your R session, load it by running the following command in the console window:

```
library(TOSTER)
```

Step 6: Run a two-one-sided-test in R

The `TOSTtwo()` function of the TOSTER package allows us to conduct an equivalence test for an independent samples t test, specifying equivalence bounds in values of Cohen's d. If you would like to specify equivalence bounds in unstandardized scale values, you can use the `TOSTtwo.raw()` function instead. To conduct the equivalence test, run the following command in you R console:

```
TOSTtwo(m1 = group1_mean, m2 = group2_mean, sd1 = group1_stdev, sd2 = group2_stdev, n1 = group1_n, n2 =
```

Notice that the values we assigned in step 4 are now being used as input to the `TOSTtwo()` function arguments. This is done here merely to make it clear which argument requires which input. You can also insert numbers directly into the function input if you prefer. If you need more information about the `TOSTtwo()` function, running `help("TOSTtwo")` in your R console will send you to a documentation page fr the function.

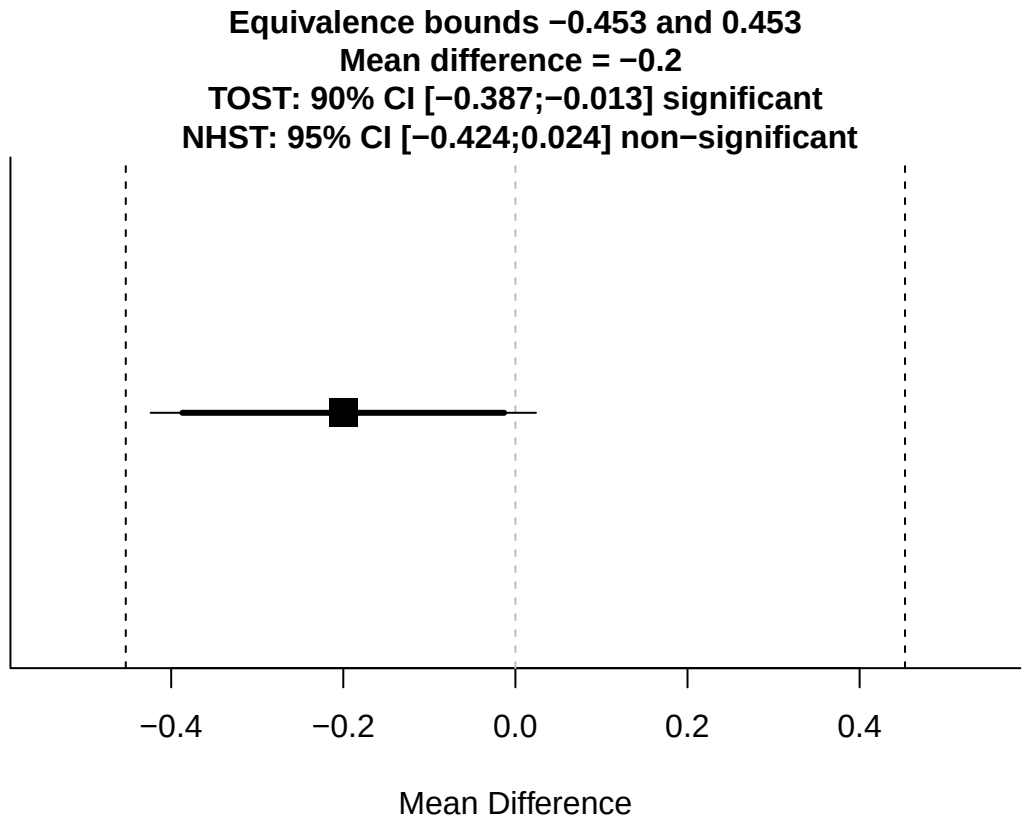
Running this command will provide you with the following output, summarizing the results of the test:

```
## Using alpha = 0.05 Welch's t-test was non-significant, t(59.36101) = -1.78754, p = 0.07895512

##

##
## Using alpha = 0.05 the equivalence test based on Welch's t-test  was significant, t(59.36101) = 2.25

##
```




```
## TOST results:
##   t-value 1  p-value 1 t-value 2    p-value 2       df
## 1  2.259176 0.01377909 -5.834257 1.195383e-07 59.36101
##
## Equivalence bounds (Cohen's d):
##   low bound d high bound d
## 1          -1            1
##
## Equivalence bounds (raw scores):
##   low bound raw high bound raw
## 1    -0.4527693      0.4527693
##
## TOST confidence interval:
##   Lower Limit 90% CI raw Upper Limit 90% CI raw
## 1             -0.3869531            -0.01304693
```

Notice that in the plot, the position of the dashed lines representing the upper and lower equivalence bounds are not consistent with the equivalence bound values we defined in step 4. This is because the `TOSTtwo()` function recalculates the bound values from Cohen's d back into the raw scale (based on the standard deviations provided in the function input) before plotting. If we would have used the `TOSTtwo.raw()` function, the bound values would already have been stated in raw scale, and thus they would have been identical to the position of the stipled lines.