Exploring Equivalence Testing with the Updated TOSTER R Package

Aaron R. Caldwell

Abstract

This is an article detailing the “avocado TOST” update to the TOSTER R package.

*Keywords:* statistics, bootstrapping, minimal effects test, NHST, TOST

Exploring Equivalence Testing with the Updated TOSTER R Package

# Introduction

Researchers often erroneously declare that no statistical effect exists based on a single “non-significant” p-value (Altman & Bland, 1995). In many of these cases, the data may corroborate the researchers claim but the interpretation of a null hypothesis significance test (NHST), wherein the null hypothesis is “no effect”, is nonetheless incorrect. In order to statistically test for whether there is “no effect” or “no difference” researchers could explore using equivalence testing. A very simple equivalence testing approach is the use of “two one-sided tests” (TOST) (Schuirmann, 1987). In TOST procedures, an upper () and lower () equivalence bound is specified based on the smallest effect size of interest (SESOI). If the TOST is below a prespecified alpha level, then the effect can be considered close enough to zero to be practically equivalent (Lakens, 2017).

Both the complaints about erroneous conclusions regarding equivalence (Altman & Bland, 1995) and proposed statistical solutions (Schuirmann, 1987) have existed for decades now. Yet the problem appears to persist in many applied disciplines. I estimate that the cause of this continued dissonance is due to a lack of education on equivalence testing and struggle for many applied researchers to implement equivalence testing. In my experience, most researchers have received some degree of statistical training in their doctoral or master’s studies, but it is rare that any have idea of equivalence testing. It may also be difficult to implement equivalence testing for many researchers. This may be caused by most statistical software defaulting to a null hypothesis of zero, or even completely lacking an ability to change the null hypothesis.

The TOSTER R package was originally developed in 2018 (CITE Lakens) to introduce experimental psychologists to the concept of equivalence testing and provide an easy-to-use implementation in R. Since then I have made a significant update to the package in order to improve the user interface and expand the tools available within the package. TOSTER (Avocado TOST edition >v0.4.0) is a good tool to introduce the idea of equivalence testing to the average researcher who is already familiar with equivalence testing.

One possible source of this misinterpretation may be due to the fact that many researchers are

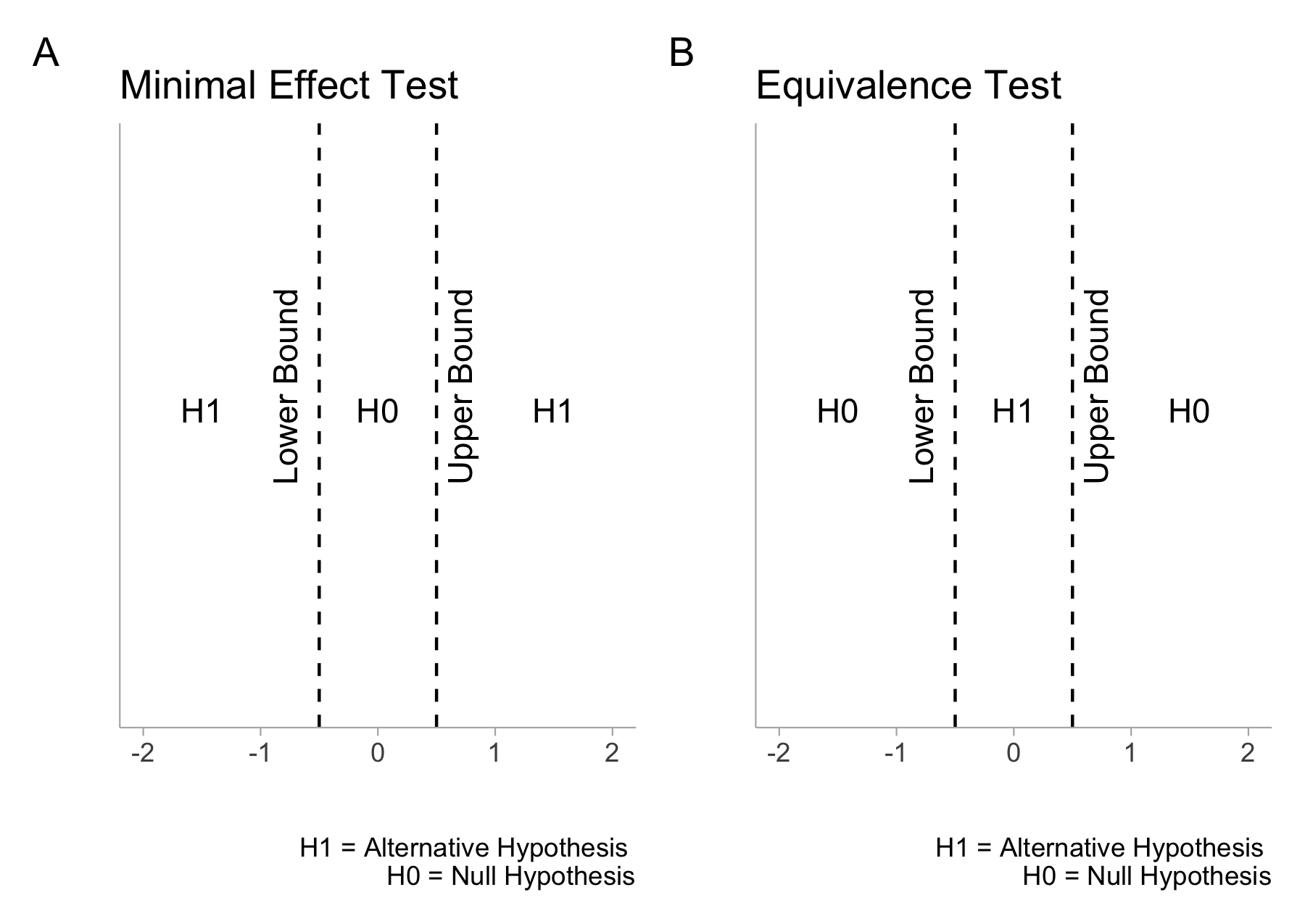
An experienced R programmer may have no problem performing equivalence testing within R but beginners may struggle with both writing the code and interpreting the output.

# Basics of Equivalence Testing

## The TOSTER R Package

In an effort to make TOSTER more informative and easier to use, a new function t\_TOST was created. This function operates very similarly to base R’s t.test function, but performs 3 t-tests (one two-tailed and two one-tailed tests). In addition, this function has a generic method where two vectors can be supplied or a formula can be given (e.g.,y ~ group). This function also makes it easier to switch between types of t-tests. All three types (two sample, one sample, and paired samples) can be performed/calculated from the same function. Moreover, the output from this function is verbose, and should make the decisions derived from the function more informative and user-friendly.

Also, t\_TOST is not limited to equivalence tests. Minimal effects testing (MET) is possible. MET is useful for situations where the hypothesis is about a minimal effect and the null hypothesis *is* equivalence.



*Figure* *1.*  Type of Hypothesis

## Vignettes with TOSTER

In the general introduction to this package we detailed how to look at *old* results and how to apply TOST to interpreting those results. However, in many cases, users may have new data that needs to be analyzed. Therefore, t\_TOST can be applied to new data. This vignette will use the bugs data from the jmv R package and the sleep data.

data('sleep')  
library(jmv)  
data('bugs')

### Independent Groups.

For this example, we will use the sleep data. In this data there is a group variable and an outcome extra.

head(sleep)

## extra group ID  
## 1 0.7 1 1  
## 2 -1.6 1 2  
## 3 -0.2 1 3  
## 4 -1.2 1 4  
## 5 -0.1 1 5  
## 6 3.4 1 6

We will assume the data are independent, and that we have equivalence bounds of +/- 0.5. All we need to do is provide the formula, data, and eqbound arguments for the function to run appropriately. In addition, we can set the var.equal argument (to assume equal variance), and the paired argument (sets if the data is paired or not). Both are logical indicators that can be set to TRUE or FALSE. The alpha is automatically set to 0.05 but this can also be adjusted by the user. The Hedges correction is also automatically calculated, but this can be overridden with the bias\_correction argument. The hypothesis is automatically set to “EQU” for equivalence but if a minimal effect is of interest then “MET” can be supplied

# Formula Interface  
res1 = t\_TOST(formula = extra ~ group,  
 data = sleep,  
 low\_eqbound = -.5,  
 high\_eqbound = .5,  
 # SMD CIs will be biased  
 # THis was done to make code run faster  
 smd\_ci = "t")  
  
# x & y interface  
res1a = t\_TOST(x = subset(sleep,group==1)$extra,  
 y = subset(sleep,group==2)$extra,  
 low\_eqbound = -.5,  
 high\_eqbound = .5)

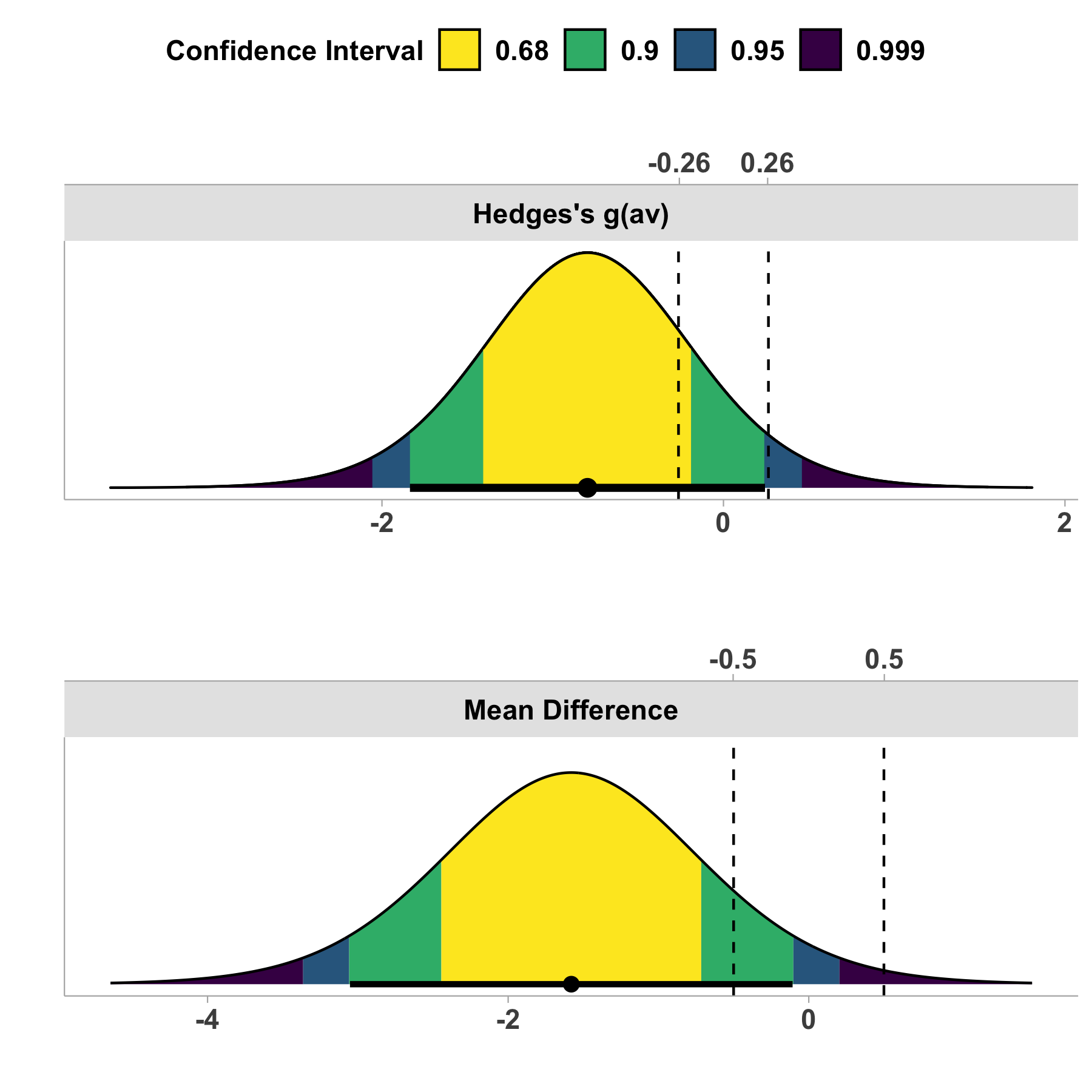
Once the function has run, we can print the results with the print command. This provides a verbose summary of the results.

print(res1)

##   
## Welch Two Sample t-test  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):-0.500 & 0.500  
## Alpha Level:0.05  
## The equivalence test was non-significant, t(17.78) = -1.272, p = 8.9e-01  
## The null hypothesis test was non-significant, t(17.78) = -1.861, p = 7.94e-02  
## NHST: don't reject null significance hypothesis that the effect is equal to zero   
## TOST: don't reject null equivalence hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test -1.860813 0.849091 17.77647 0.07939414  
## TOST Lower -1.271948 0.849091 17.77647 0.89010996  
## TOST Upper -2.449678 0.849091 17.77647 0.01245133  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw -1.5800000 0.8490910 -3.053381 -0.1066185 0.9  
## Hedges's g(av) -0.7964846 0.5991937 -1.836233 0.2432637 0.9

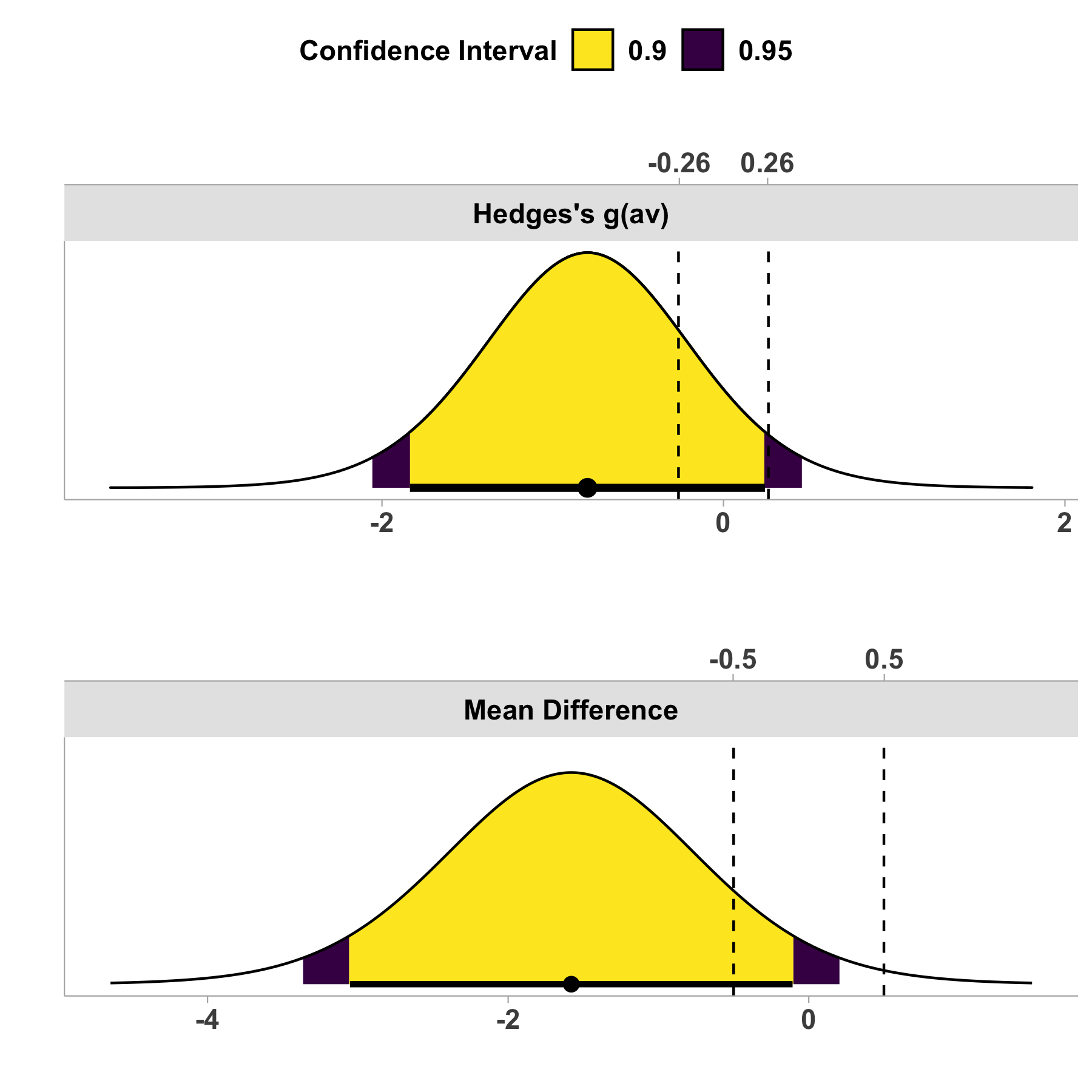
Another nice feature is the generic plot method that can provide a visual summary of the results. All of the plots in this package were inspired by the [concurve](https://cran.r-project.org/package=concurve) R package. There are two types of plots that can be produced. The first, and default, is the consonance density plot (type = "cd").

plot(res1, type = "cd")



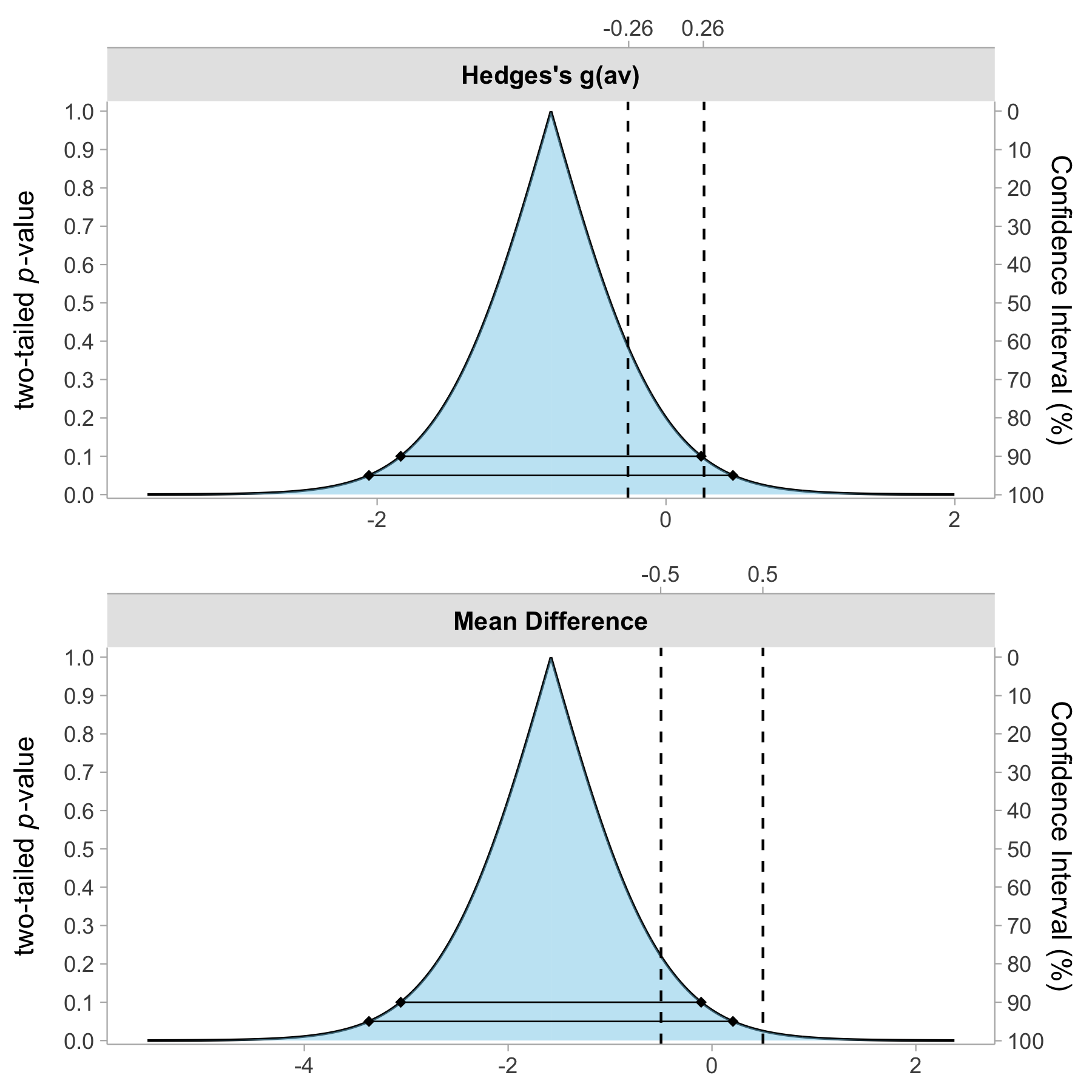
The shading pattern can be modified with the ci\_shades.

plot(res1, type = "cd",  
 ci\_shades = c(.9,.95))



Consonance plots, where all confidence intervals can be simultaneous plotted, can also be produced. The advantage here is multiple confidence interval lines can plotted at once.

plot(res1, type = "c",  
 ci\_lines = c(.9,.95))



### Paired Samples.

To perform a paired samples TOST, the process does not change much. We could process the test the same way by providing a formula. All we would need to then is change paired to TRUE.

res2 = t\_TOST(formula = extra ~ group,  
 data = sleep,  
 paired = TRUE,  
 low\_eqbound = -.5,  
 high\_eqbound = .5)  
res2

##   
## Paired t-test  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):-0.500 & 0.500  
## Alpha Level:0.05  
## The equivalence test was non-significant, t(9) = -2.777, p = 9.89e-01  
## The null hypothesis test was significant, t(9) = -4.062, p = 2.83e-03  
## NHST: reject null significance hypothesis that the effect is equal to zero   
## TOST: don't reject null equivalence hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test -4.062128 0.3889587 9 0.0028328902  
## TOST Lower -2.776644 0.3889587 9 0.9892407566  
## TOST Upper -5.347611 0.3889587 9 0.0002319027  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw -1.580000 0.3889587 -2.293005 -0.8669947 0.9  
## Hedges's g(z) -1.173925 0.4109805 -1.804551 -0.4977326 0.9

However, we may have two vectors of data that are paired. So instead we may want to just provide those separately rather than using a data set and setting the formula. This can be demonstrated with the “bugs” data.

res3 = t\_TOST(x = bugs$LDHF,  
 y = bugs$LDLF,  
 paired = TRUE,  
 low\_eqbound = -1,  
 high\_eqbound = 1)  
res3

##   
## Paired t-test  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):-1.000 & 1.000  
## Alpha Level:0.05  
## The equivalence test was non-significant, t(90) = 2.655, p = 9.95e-01  
## The null hypothesis test was significant, t(90) = 6.649, p = 2.22e-09  
## NHST: reject null significance hypothesis that the effect is equal to zero   
## TOST: don't reject null equivalence hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test 6.648618 0.2504032 90 2.223690e-09  
## TOST Lower 10.642177 0.2504032 90 6.681389e-18  
## TOST Upper 2.655058 0.2504032 90 9.953114e-01  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw 1.6648352 0.2504032 1.2486748 2.0809956 0.9  
## Hedges's g(z) 0.6911376 0.1166772 0.4986594 0.8802135 0.9

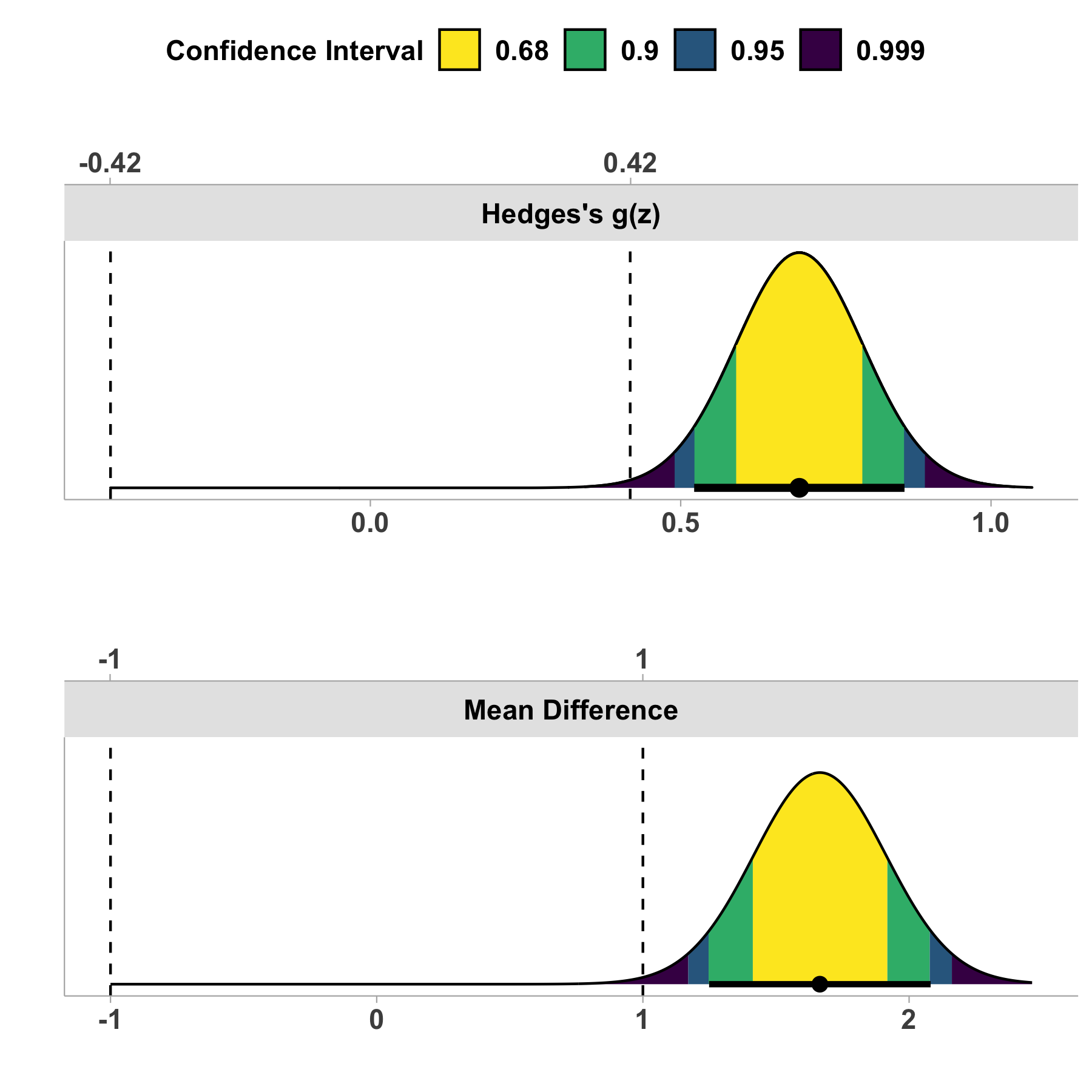
We may want to perform a Minimal Effect Test with the hypothesis argument set to “MET”.

res3a = t\_TOST(x = bugs$LDHF,  
 y = bugs$LDLF,  
 paired = TRUE,  
 hypothesis = "MET",  
 smd\_ci = "t",  
 low\_eqbound = -1,  
 high\_eqbound = 1)  
res3a

##   
## Paired t-test  
## Hypothesis Tested: Minimal Effect  
## Equivalence Bounds (raw):-1.000 & 1.000  
## Alpha Level:0.05  
## The minimal effect test was significant, t(90) = 10.642, p = 4.69e-03  
## The null hypothesis test was significant, t(90) = 6.649, p = 2.22e-09  
## NHST: reject null significance hypothesis that the effect is equal to zero   
## TOST: reject null MET hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test 6.648618 0.2504032 90 2.223690e-09  
## TOST Lower 10.642177 0.2504032 90 1.000000e+00  
## TOST Upper 2.655058 0.2504032 90 4.688649e-03  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw 1.6648352 0.2504032 1.2486748 2.080996 0.9  
## Hedges's g(z) 0.6911376 0.1019557 0.5216911 0.860584 0.9

Again, we can plot the effects from the t\_TOST result.

plot(res3a)



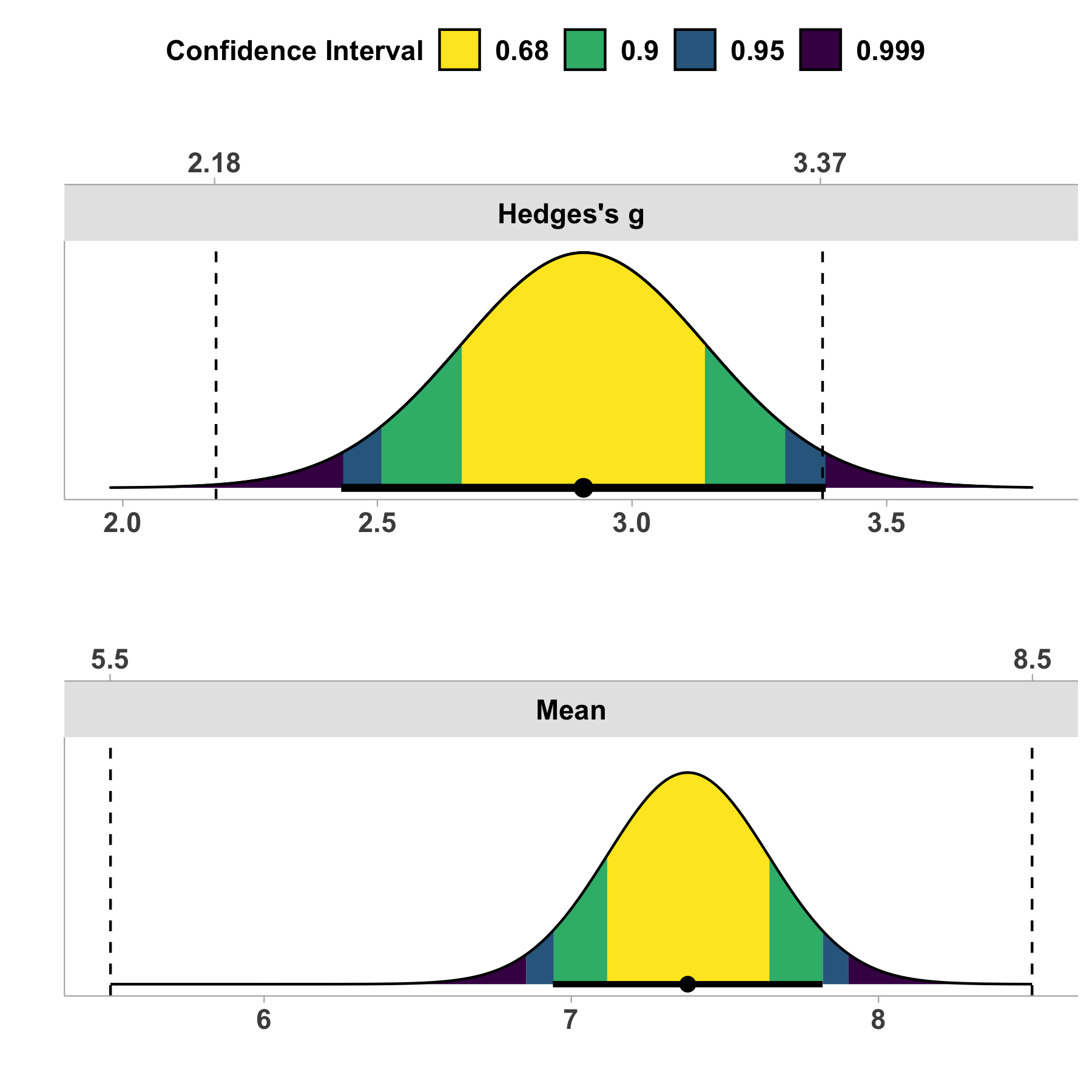
### One Sample t-test.

In other cases we may just have a one sample test. If that is the case all we have to do is supply the x argument for the data. For this test we may hypothesis that the mean of LDHF is not more than 1.5 points greater or less than 7.

res4 = t\_TOST(x = bugs$LDHF,  
 hypothesis = "EQU",  
 smd\_ci = "t",  
 eqb = c(5.5,8.5))  
res4

##   
## One Sample t-test  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):5.500 & 8.500  
## Alpha Level:0.05  
## The equivalence test was significant, t(90) = -4.244, p = 2.66e-05  
## The null hypothesis test was significant, t(90) = 27.942, p = 3.91e-46  
## NHST: reject null significance hypothesis that the effect is equal to zero   
## TOST: reject null equivalence hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test 27.942403 0.2640833 90 3.906969e-46  
## TOST Lower 7.115638 0.2640833 90 1.298575e-10  
## TOST Upper -4.244416 0.2640833 90 2.658068e-05  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw 7.379121 0.2640833 6.940225 7.818017 0.9  
## Hedges's g 2.904671 0.2394719 2.428918 3.380423 0.9

plot(res4)



### Using Summary Statistics.

In some cases you may only have access to the summary statistics. Therefore, we created a function, tsum\_TOST,to perform the same tests just based on the summary statistics. This involves providing the function with a number of different arguments.

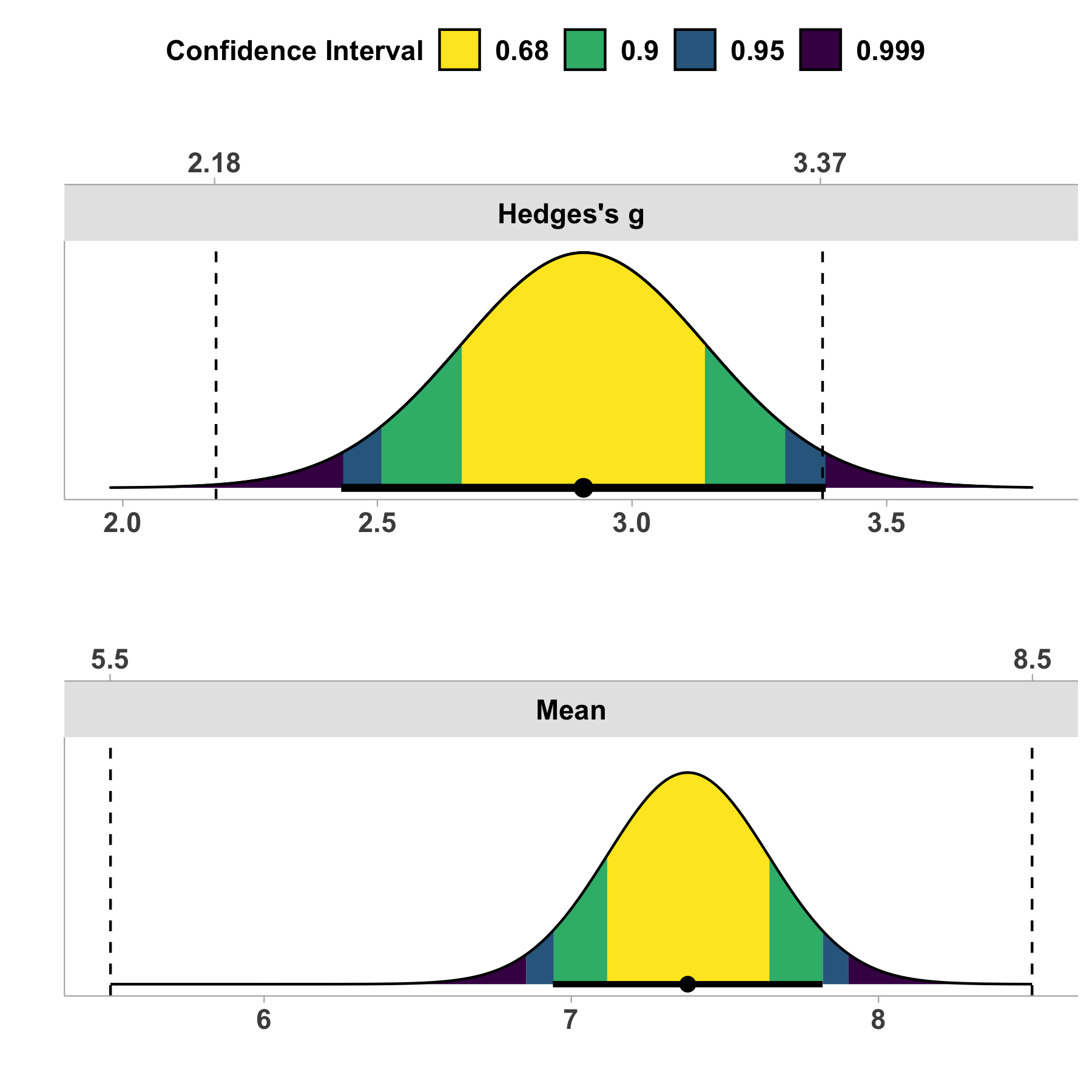
* n1 & n2 the sample sizes (only n1 needs to be provided for one sample case)
* m1 & m2 the sample means
* sd1 & sd2 the sample standard deviation
* r12 the correlation between each if paired is set to TRUE

The results from above can be replicated with the tsum\_TOST

res\_tsum = tsum\_TOST(  
 m1 = mean(bugs$LDHF, na.rm=TRUE),  
 sd1 = sd(bugs$LDHF, na.rm=TRUE),  
 n1 = length(na.omit(bugs$LDHF)),  
 hypothesis = "EQU",  
 smd\_ci = "t",  
 low\_eqbound = 5.5,  
 high\_eqbound = 8.5  
)  
  
res\_tsum

##   
## One-sample t-Test  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):5.500 & 8.500  
## Alpha Level:0.05  
## The equivalence test was significant, t(90) = -4.244, p = 2.66e-05  
## The null hypothesis test was significant, t(90) = 27.942, p = 3.91e-46  
## NHST: reject null significance hypothesis that the effect is equal to zero   
## TOST: reject null equivalence hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test 27.942403 0.2640833 90 3.906969e-46  
## TOST Lower 7.115638 0.2640833 90 1.298575e-10  
## TOST Upper -4.244416 0.2640833 90 2.658068e-05  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw 7.379121 0.2640833 6.940225 7.818017 0.9  
## Hedges's g 2.904671 0.2394719 2.428918 3.380423 0.9

plot(res\_tsum)



# Robust Methods for Equivalence Testing

Why this may be useful Describe these aren’t tests of medians but symmetry tests (myth busting) Paired samples may want to be rank transformed

In this package there are currently 2 functions that provide robust alternatives to the t\_TOST function.

## Tests of Symmetry

The Wilcoxon group of tests (includes Mann-Whitney U-test) provide a non-parametric test of differences between groups, or within samples, based on ranks. This provides a test of location shift, which is a fancy way of saying differences in the center of the distribution (i.e., in parametric tests the location is mean). With TOST, there are two separate tests of directional location shift to determine if the location shift is within (equivalence) or outside (minimal effect). The exact calculations can be explored via the documentation of the wilcox.test function.

In the TOSTER package, we accomplish this with the wilcox\_TOST function. Overall, this function operates extremely similar to the t\_TOST function. However, the standardized mean difference (SMD) is *not* calculated. Instead the rank-biserial correlation is calculated for *all* types of comparisons (e.g., two sample, one sample, and paired samples). Also, there is no plotting capability at this time for the output of this function.

As an example, we can use the sleep data to make a non-parametric comparison of equivalence.

data('sleep')  
library(TOSTER)  
  
test1 = wilcox\_TOST(formula = extra ~ group,  
 data = sleep,  
 paired = FALSE,  
 low\_eqbound = -.5,  
 high\_eqbound = .5)  
  
  
print(test1)

##   
## Wilcoxon rank sum test with continuity correction  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):-0.500 & 0.500  
## Alpha Level:0.05  
## The equivalence test was non-significant W = 20.000, p = 8.94e-01  
## The null hypothesis test was non-significant W = 25.500, p = 6.93e-02  
## NHST: don't reject null significance hypothesis that the effect is equal to zero   
## TOST: don't reject null equivalence hypothesis  
##   
## TOST Results   
## statistic p.value  
## NHST 25.5 0.06932758  
## TOST Lower 34.0 0.89385308  
## TOST Upper 20.0 0.01287404  
##   
## Effect Sizes   
## estimate lower.ci upper.ci conf.level  
## Median of Differences -1.346388 -3.3999651 -0.09995341 0.9  
## Rank-Biserial Correlation -0.490000 -0.7492521 -0.10053222 0.9

## Boostrap TOST

The boostrap refers to resampling with replacement and can be used statistical estimation and inference. Bootsrapping techniques are very useful because they are considered somewhat robust to the violations of assumptions for a simple t-test. Therefore we added a bootstrap option, boot\_t\_TOST to the package to provide another robust alternative to the t\_TOST function.

In this function we provide a percentile bootstrap solution outlined by Efron and Tibshirani (1993) (see chapter 16, page 220). The bootstrapped p-values are derived from the “studentized” version of a test of mean differences (Efron & Tibshirani, 1993). Overall, the results should be similar to the results of t\_TOST. **However**, for paired samples, the Cohen’s d(rm) effect size *cannot* be calculated at this time.

### Two Sample Algorithm.

1. Form B bootstrap data sets from x\* and y\* wherein x\* is sampled with replacement from and y\* is sampled with replacement from
2. t is then evaluated on each sample, but the mean of each sample (y or x) and the overall average (z) are subtracted from each
3. An approximate p-value can then be calculated as the number of bootstrapped results greater than the observed t-statistic from the sample.

The same process is completed for the one sample case but with the one sample solution for the equation outlined by . The paired sample case in this bootstrap procedure is equivalent to the one sample solution because the test is based on the difference scores.

### Example of Bootrapping.

Again, we can use the sleep data to see the bootstrapped results. Notice that the plots show how the resampling via boostrapping indicates the instability of Hedges’ d(z).

data('sleep')  
  
test1 = boot\_t\_TOST(formula = extra ~ group,  
 data = sleep,  
 paired = TRUE,  
 low\_eqbound = -.5,  
 high\_eqbound = .5,  
 R = 999)  
  
  
print(test1)

##   
## Bootstrapped Paired t-test  
## Hypothesis Tested: Equivalence  
## Equivalence Bounds (raw):-0.500 & 0.500  
## Alpha Level:0.05  
## The equivalence test was non-significant, t(9) = -2.777, p = 1e+00  
## The null hypothesis test was significant, t(9) = -4.062, p = 0e+00  
## NHST: reject null significance hypothesis that the effect is equal to zero   
## TOST: don't reject null equivalence hypothesis  
##   
## TOST Results   
## t SE df p.value  
## t-test -4.062128 0.3889587 9 0  
## TOST Lower -2.776644 0.3889587 9 1  
## TOST Upper -5.347611 0.3889587 9 0  
##   
## Effect Sizes   
## estimate SE lower.ci upper.ci conf.level  
## Raw -1.580000 0.3745159 -2.291000 -1.0790000 0.9  
## Hedges's g(z) -1.173925 0.6680162 -2.759268 -0.9238074 0.9

# Equivalence Testing with ANOVAs

For an open access tutorial paper explaining how to set equivalence bounds, and how to perform and report equivalence for ANOVA models see Campbell and Lakens (2021). These functions are meant to be omnibus tests, and additional testing may be necessary. For example, comparison of the estimated marginal means, in addition to or as an alternative of with may be prudent.

## F-test Calculations

Statistical equivalence testing (or “omnibus non-inferiority testing” as Campbell and Lakens (2021)) for *F*-tests are special use case of the cumulative distribution function of the non-central *F* distribution. As Campbell and Lakens (2021) state, these type of questions answer the question: “Can we reject the hypothesis that the total proportion of variance in outcome Y attributable to X is greater than or equal to the equivalence bound ?”

### Hypothesis Tests.

In TOSTER we go a tad farther and calculate a generalization of the non-centrality parameter that allows the equivalence test for *F*-tests to be applied to variety of designs.

Campbell and Lakens (2021) calculate the *p*-value as:

However, this approach could not be applied to factorial ANOVA and the paper only outlines how to apply this approach to a one-way ANOVA and an extension to Welch’s one-way ANOVA.

The non-centrality parameter (ncp = ) can be calculated with the equivalence bound and the degrees of freedom:

The *p*-value for the equivalence test () could then be calculated from traditional ANOVA results and the distribution function:

## Example of Equivalence ANOVA Test

Using the InsectSprays data set in R and the base R aov function we can demonstrate how this omnibus equivalence testing can be applied with TOSTER.

From the initial analysis we an see a clear “significant” effect (the p-value listed is zero but it just very small) of the factor spray. However, we *may* be interested in testing if the effect is practically equivalent. I will arbitrarily set the equivalence bound to a partial eta-squared of 0.35 ()

library(TOSTER)  
# Get Data  
data("InsectSprays")  
# Build ANOVA  
aovtest = aov(count ~ spray,  
 data = InsectSprays)  
  
# Display overall results  
knitr::kable(broom::tidy(aovtest),  
 caption = "Traditional ANOVA Test")

Table 1: Traditional ANOVA Test

| term | df | sumsq | meansq | statistic | p.value |
| --- | --- | --- | --- | --- | --- |
| spray | 5 | 2668.833 | 533.76667 | 34.70228 | 0 |
| Residuals | 66 | 1015.167 | 15.38131 | NA | NA |

We can then use the information in the table above to perform an equivalence test using the equ\_ftest function. This function returns an object of the S3 class htest and the output will look very familiar to the the t-test. The main difference is the estimates, and confidence interval, are for partial .

equ\_ftest(Fstat = 34.70228,  
 df1 = 5,  
 df2 = 66,  
 eqbound = 0.35)

##   
## Equivalence Test from F-test  
##   
## data: Summary Statistics  
## F = 34.702, df1 = 5, df2 = 66, p-value = 1  
## 95 percent confidence interval:  
## 0.5806263 0.7804439  
## sample estimates:  
## [1] 0.724439

Based on the results above we would conclude there is a significant effect of “spray” and the differences due to spray are *not* statistically equivalent. In essence, we reject the traditional null hypothesis of “no effect” but accept the null hypothesis of the equivalence test.

The equ\_ftest is very useful because all you need is very basic summary statistics. However, if you are doing all your analyses in R then you can use the equ\_anova function. This function accepts objects produced from stats::aov, car::Anova and afex::aov\_car (or any ANOVA from derived from afex).

equ\_anova(aovtest,  
 eqbound = 0.35)

## effect df1 df2 F.value p.null pes eqbound p.equ  
## 1 spray 5 66 34.70228 3.182584e-17 0.724439 0.35 0.9999965

# Visualizing Equivalence

Add plots from the package

# Acknowledgement(s)

An unnumbered section, e.g. \section\*{Acknowledgements}, may be used for thanks, etc. if required and included *in the non-anonymous version* before any Notes or References.

# Disclosure statement

The author is the author of the TOSTER package.

# Funding

No funding was provided for this work.

# Notes on contributor(s)

Daniel Lakens provided a review of many of the materials that have been incorporated into the update of TOSTER, and was the original author of this package.

# Nomenclature/Notation

* ANOVA: Analysis of Variance
* SMD: Standardized mean difference (e.g., Cohen’s d)
* TOST: Two-one sided tests of equivalence

# Notes

The R package is also (partially) implemented in jamovi as the TOSTER module.

# References

Altman, D. G., & Bland, J. M. (1995). Statistics notes: Absence of evidence is not evidence of absence. *BMJ*, *311*(7003), 485. <https://doi.org/10.1136/bmj.311.7003.485>

Campbell, H., & Lakens, D. (2021). Can we disregard the whole model? Omnibus non-inferiority testing for R2 in multi-variable linear regression and in ANOVA. *British Journal of Mathematical and Statistical Psychology*, *74*(1), e12201. <https://doi.org/10.1111/bmsp.12201>

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Boca Raton, Florida, USA: Chapman & Hall/CRC.

Lakens, D. (2017). Equivalence tests: A practical primer for t-tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, *1*, 1–8. <https://doi.org/10.1177/1948550617697177>

Schuirmann, D. J. (1987). A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, *15*(6), 657–680. <https://doi.org/10.1007/BF01068419>