Equivalence testing for linear regression: Supplemental material

Harlan Campbell

University of British Columbia, Department of Statistics

Earth Sciences Building, Room 3182, 2207 Main Mall, Vancouver, British Columbia, Canada, V6T 1Z2

harlan.campbell@stat.ubc.ca

Additional formulas, notation and tables

Least squares estimation

For completeness, we provide details and notation for least squares estimation in a standard linear regression model. We define:

$$\widehat{\beta}_k = ((X^T X)^{-1} X^T y)_k, \text{for } k \text{ in } 1,..., K, \text{ and} \tag{1}$$

$$\hat{\sigma} = \sqrt{\sum_{i=1}^{N} (\hat{\epsilon}_i^2)/(N-K-1)}, \tag{2}$$

where $\hat{\epsilon}_i = \widehat{y}_i - y_i$, and $\widehat{y}_i = X_{i\times}^T \widehat{\beta}$, for i in 1,..., N. We also define R^2_{YX}, the coefficient of determination from the linear regression of Y predicted from X:

$$R^2_{YX} = \frac{\sigma_{XY}^T \Sigma_X^{-1} \sigma_{XY}}{\sigma_Y^2}, \tag{3}$$

where $\sigma_Y^2 = (\beta^T \text{Cov}(X)\beta + \sigma^2)$ is the unconditional variance of Y, (note that: $\sigma_Y^2 \geq \sigma^2$); σ_{XY} is the vector of population covariances between the K different predictors and Y; and Σ_X is the population covariance matrix of the K different predictors. The $\hat{R}^2_{YX}$ statistic estimates the parameter R^2_{YX} from the observed data:

$$\hat{R}^2_{YX} = 1 - \frac{\sum_{i=1}^{N} \hat{\epsilon}_i^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2}, \tag{4}$$

where $\bar{y} = \sum_{i=1}^{N} y_i / N$.

A standard NHST for the k-th predictor, X_k, is stated as:

$\text{H}_0 : \beta_k = 0$, vs.
$\text{H}_1 : \beta_k \neq 0$.

Typically one conducts one of two different (yet mathematically identical) tests. Most commonly a t-test is done to calculate a p-value as follows:

$$p\text{-value}_k = 2 \times F_t\left(\frac{|\widehat{\beta}_k|}{\text{SE}(\widehat{\beta}_k)}, N - K - 1\right), \text{for } k \text{ in } 0,...,K, \tag{5}$$

where we use $F_t(\quad \cdot \quad ; df)$ to denote the cdf of the t-distribution with df degrees of freedom, and where: $\text{SE}(\widehat{\beta}_k) = \hat{\sigma}\sqrt{[(X^T X)^{-1}]_{kk}}$. Alternatively, we can conduct an F-test and, for k in 1,...,K, we will obtain the very same p-value with:

$$p\text{-value}_k = p_F\left((N - K - 1)\frac{\widehat{sr}_k^2}{1 - \hat{R}^2_{YX}}, 1, N - K - 1\right), \tag{6}$$

where $p_f(\cdot \quad ; df_1, df_2)$ is the cdf of the F-distribution with df_1 and df_2 degrees of freedom, and where: $\widehat{sr}_k^2 = \hat{R}^2_{YX} - \hat{R}^2_{YX_{-k}}$. Regardless of whether the t-test or the F-test is employed, if $p\text{-value}_k < \alpha$, we reject the null hypothesis of $\text{H}_0 : \beta_k = 0$ against the alternative $\text{H}_0 : \beta_k \neq 0$.

A valid equivalence test for the standardized difference between two independent means

A valid equivalence test for the standardized difference between two independent means, θ, can be defined by the following null and alternative hypotheses (see Serlin et al. (1993), Weber & Popova (2012)):

$$\mathrm{H}_0 : \theta \leq \Delta_{lower} \quad \text{or:} \quad \theta \geq \Delta_{upper}, \quad \text{vs.}$$
$$\mathrm{H}_1 : \theta > \Delta_{lower} \quad \text{and:} \quad \theta < \Delta_{upper},$$

where $\theta = \mu_d/\sigma$ and the equivalence margin is $(\Delta_{lower}, \Delta_{upper})$. A p-value for this test can then be calculated as p-value=max$(p_d^{lower}, p_d^{upper})$, where:

$$p_d^{lower} = 1 - F_t\left(\frac{\hat{\mu}_d}{\hat{\sigma}_p}\sqrt{\frac{N_1N_2}{N_1+N_2}}, N_1+N_2-2, \Delta_{lower}\sqrt{\frac{N_1N_2}{N_1+N_2}}\right), \quad \text{and} \tag{7}$$

$$p_d^{upper} = 1 - F_t\left(-\frac{\hat{\mu}_d}{\hat{\sigma}_p}\sqrt{\frac{N_1N_2}{N_1+N_2}}, N_1+N_2-2, -\Delta_{upper}\sqrt{\frac{N_1N_2}{N_1+N_2}}\right),$$

where N_1 is the number of observations in the first sample, N_2 is the number of observations in the second sample, where $\hat{\mu}_d$ is the difference between the two sample means, and $\hat{\sigma}_p$, the pooled standard deviation estimate, is calculated from the two samples as:

$$\hat{\sigma}_p = \sqrt{\frac{(N_1-1)\hat{\sigma}_1^2 + (N_2-1)\hat{\sigma}_2^2}{N_1+N_2-2}}, \tag{8}$$

where $\hat{\sigma}_1$ is the estimated standard deviation of the first sample, and $\hat{\sigma}_2$ is the estimated standard deviation of the second sample.

An equivalence test for correlations based on Fisher's Z transformation

A p-value from the equivalence test for correlations based on Fisher's Z transformation is calculated as $p_Z = max(\mathbb{p}_Z^{lower}, \mathbb{p}_Z^{upper})$, where:

$$\mathbb{p}_Z^{lower} = 1 - F_Z\left(\frac{\sqrt{N-3}}{2}\ln\left(\left(\frac{1+\widehat{sr}_1}{1-\widehat{sr}_1}\right) - \left(\frac{1+\Delta_{lower}}{1-\Delta_{lower}}\right)\right)\right), \tag{9}$$

and:

$$\mathbb{p}_Z^{upper} = 1 - F_Z\left(\frac{\sqrt{N-3}}{2}\ln\left(\left(\frac{1+\widehat{sr}_1}{1-\widehat{sr}_1}\right) + \left(\frac{1+\Delta_{upper}}{1-\Delta_{upper}}\right)\right)\right),$$

where $F_Z()$ denotes the cdf of the standard normal distribution; see Goertzen & Cribbie (2010) for details.

i	X_1 Received counselling (yes=1; no=0)	X_2 Age (years)	X_3 Household income ($)	Y Anxiety score points
1	0	13	75593	12.1
2	0	15	57954	15.5
3	0	13	61336	13.3
4	1	14	47628	14.3
5	0	14	46564	12.3
6	1	12	74071	13.8
7	1	17	76964	14.5
8	1	15	69060	11.6
9	0	13	86445	12.8
10	0	17	109002	17.5
11	1	16	58179	14.7
12	1	14	21817	16.8
13	1	17	88115	12.5
14	0	17	53816	15.8
15	1	16	54240	17.3
16	0	16	88511	15.8
17	1	16	62305	16.2
18	0	15	43586	13.6
19	0	14	71626	12.3
20	0	14	65222	12.0
21	0	14	68115	14.6
22	1	15	75706	13.2
23	0	13	60587	12.8
24	0	19	80888	16.1
25	0	17	63590	20.1
26	0	13	74636	12.3
27	1	14	89937	15.2
28	1	14	76704	15.0
29	0	16	61481	13.2
30	1	15	90976	15.0
31	0	15	87870	17.9
32	1	15	78968	16.3
33	0	15	72775	14.9
34	1	17	55442	15.6
35	1	15	95213	10.4
36	0	18	55995	19.0
37	0	12	111747	9.5
38	0	16	98652	16.7
39	0	15	63286	19.3
40	1	15	47472	12.3

Table 1. The hypothetical anxiety study dataset. In this hypothetical study, Y might be a student's score on an anxiety assessment questionnaire; X_1 might be a binary variable indicating whether or not the student received counselling services (0 = "did not receive counselling; 1 = "did receive counselling"); X_2 might be a continuous predictor corresponding to the student's age in years; and X_3 might be a continuous predictor corresponding to the student's household income in dollars.

Simulation Study 1

We simulated data in order to compare the operating characteristics of two equivalence tests for the difference between two independent means:

(1) the invalid test (i.e., the test proposed by Lakens (2017)), with null hypothesis $\text{H}_0 : |\mu_d| \geq \Delta \times \hat{\sigma}$; and

(2) the valid test (see equation (7)), with null hypothesis $\text{H}_0 : |\theta| \geq \Delta$, where $\theta = \mu_d/\sigma$.

We considered 6 different values for the total sample size, N, ranging from 54 to 3500 (values representative of sample sizes in large and very large psychological studies (Kühberger et al., 2014; Fraley & Vazire, 2014; Marszalek et al., 2011)), and 4 different values for the upper bound of a symmetric equivalence margin, Δ, ranging from 0.2 to 1.0. We simulated data from a Normal distribution such that the true Cohen's d was equal to 0, or equal to Δ, or equal to 0.15.

For each of the different configurations within the simulation study, we simulated 2,000,000 unique datasets and calculated a p-value with each of the two equivalence tests. We then calculated the proportion of these p-values less than $\alpha = 0.05$. We specifically chose to conduct 2,000,000 simulation runs so as to keep computing time within a reasonable limit while also reducing the amount of Monte Carlo standard error to a very negligible amount (for looking at type 1 error with $\alpha = 0.05$, Monte Carlo SE will be approximately $0.00015 \approx \sqrt{0.05(1-0.05)/2,000,000}$; see Morris et al. (2019)).

The simulation study was done using the R statistical software with default simulation routines (R Core Team, 2020). Results are displayed in Table and suggest that, in practice, using the invalid test can lead to a higher than advertised type 1 error when sample sizes are large and a minor loss of efficiency when sample sizes are small.

N	Δ	$\Pr(p\text{-val} < 0.05 \mid d = \Delta)$		$\Pr(p\text{-val} < 0.05 \mid d = 0)$		$\Pr(p\text{-val} < 0.05 \mid d = 0.15)$	
		invalid test	valid test	invalid test	valid test	invalid test	valid test
54	0.20	0.000	0.000	0.000	0.000	0.000	0.000
54	0.50	0.024	0.029	0.128	0.152	0.110	0.131
54	0.75	0.047	0.050	0.716	0.727	0.647	0.658
54	1.00	0.050	0.050	0.949	0.950	0.915	0.916
80	0.20	0.000	0.000	0.000	0.000	0.000	0.000
80	0.50	0.045	0.048	0.430	0.443	0.352	0.364
80	0.75	0.049	0.050	0.905	0.907	0.833	0.835
80	1.00	0.051	0.050	0.994	0.994	0.981	0.980
180	0.20	0.000	0.000	0.000	0.000	0.000	0.000
180	0.50	0.050	0.050	0.909	0.910	0.750	0.752
180	0.75	0.051	0.050	0.999	0.999	0.990	0.990
180	1.00	0.054	0.050	1.000	1.000	1.000	1.000
540	0.20	0.048	0.049	0.501	0.502	0.135	0.136
540	0.50	0.051	0.050	1.000	1.000	0.992	0.992
540	0.75	0.053	0.050	1.000	1.000	1.000	1.000
540	1.00	0.057	0.050	1.000	1.000	1.000	1.000
1000	0.20	0.050	0.050	0.870	0.870	0.196	0.197
1000	0.50	0.052	0.050	1.000	1.000	1.000	1.000
1000	0.75	0.054	0.050	1.000	1.000	1.000	1.000
1000	1.00	0.058	0.050	1.000	1.000	1.000	1.000
3500	0.20	0.050	0.050	1.000	1.000	0.434	0.434
3500	0.50	0.052	0.050	1.000	1.000	1.000	1.000
3500	0.75	0.055	0.050	1.000	1.000	1.000	1.000
3500	1.00	0.059	0.050	1.000	1.000	1.000	1.000

Table 2. Results from Simulation Study 1. Note that the maximum type 1 error rate should not exceed $\alpha = 0.05$. As such, when $\Delta = d$, the probability of a p-value less than 0.05 should not exceed 0.05. When $\Delta > d$, the probability of a p-value less than 0.05 corresponds to the test's statistical power.

Simulation Study 2

We simulated data in order to compare the operating characteristics of two equivalence tests for the difference between two independent means:

(1) the proposed equivalence test for semipartial correlation coefficients (see equation (**??**)) ("sr test"); and

(2) the equivalence test for correlations based on Fisher's Z transformation (see equation (9)) ("Z test").

Both tests are valid for testing the lack of an association between Y and X when $K = 1$ (i.e., for simple linear regression). We considered 6 different values for the total sample size, N, ranging from 54 to 3500 (values representative of sample sizes in large and very large psychological studies (Kühberger et al., 2014; Fraley & Vazire, 2014;

Marszalek et al., 2011)), and 3 different values for the upper bound of a symmetric equivalence margin, Δ, ranging from 0.1 to 0.20. We simulated data from a bivariate Normal distribution such that the true value of sr_1 was equal to 0, or equal to Δ, or equal to 0.05.

For each of the different configurations within the simulation study, we simulated 2,000,000 unique datasets and calculated a p-value with each of the two equivalence tests. We then calculated the proportion of these p-values less than $\alpha = 0.05$. We specifically chose to conduct 2,000,000 simulation runs so as to keep computing time within a reasonable limit while also reducing the amount of Monte Carlo standard error to a very negligible amount (for looking at type 1 error with $\alpha = 0.05$, Monte Carlo SE will be approximately $0.00015 \approx \sqrt{0.05(1-0.05)/2,000,000}$; see Morris et al. (2019)).

The simulation study was done using the R statistical software with default simulation routines (R Core Team, 2020). Results are displayed in Table and suggest that, in practice, both equivalence tests obtain very similar values for the type 1 error and statistical power.

Simulation Study 3

We conducted a simple simulation study in order to better understand the operating characteristics of the proposed equivalence test for the semipartial correlation and to confirm that the proposed formula for approximating statistical power (equation (**??**)) is accurate. The equivalence test in the simulation study targeted sr_1 and considered a symmetric equivalence margin, $(-\Delta, \Delta)$, such that the hypothesis test in question can be stated as: $\text{H}_0 : |sr_1| \geq \Delta$, vs. $\text{H}_1 : |sr_1| < \Delta$.

We considered 4 different values for the total sample size, N, ranging from 54 to 3500 (values representative of sample sizes in large and very large psychological studies (Kühberger et al., 2014; Fraley & Vazire, 2014; Marszalek et al., 2011)), and 3 different values for the upper bound of the symmetric equivalence margin, Δ, ranging from 0.10 to 0.20. We considered two values for K, the number of predictors: $K = 2$ or $K = 4$; and simulated the predictors from a multivariate Normal distribution with

N	Δ	$\Pr(p < 0.05 \mid sr_1 = \Delta)$		$\Pr(p < 0.05 \mid sr_1 = 0)$		$\Pr(p < 0.05 \mid sr_1 = 0.05)$	
		sr test	Z test	sr test	Z test	sr test	Z test
54	0.10	0.000	0.000	0.000	0.000	0.000	0.000
54	0.15	0.000	0.000	0.000	0.000	0.000	0.000
54	0.20	0.000	0.000	0.000	0.000	0.000	0.000
80	0.10	0.000	0.000	0.000	0.000	0.000	0.000
80	0.15	0.000	0.000	0.000	0.000	0.000	0.000
80	0.20	0.016	0.022	0.081	0.107	0.076	0.101
180	0.10	0.000	0.000	0.000	0.000	0.000	0.000
180	0.15	0.038	0.041	0.273	0.286	0.237	0.249
180	0.20	0.046	0.049	0.694	0.707	0.629	0.642
540	0.10	0.048	0.048	0.499	0.503	0.345	0.348
540	0.15	0.048	0.050	0.935	0.937	0.817	0.820
540	0.20	0.047	0.049	0.998	0.998	0.982	0.983
1000	0.10	0.049	0.050	0.870	0.872	0.597	0.599
1000	0.15	0.048	0.050	0.998	0.998	0.968	0.969
1000	0.20	0.048	0.050	1.000	1.000	1.000	1.000
3500	0.10	0.050	0.050	1.000	1.000	0.972	0.972
3500	0.15	0.049	0.050	1.000	1.000	1.000	1.000
3500	0.20	0.049	0.050	1.000	1.000	1.000	1.000

Table 3. Results from Simulation Study 2. Note that the maximum type 1 error rate should not exceed $\alpha = 0.05$. When $\Delta > sr_1$, the probability of a p-value less than 0.05 corresponds to the test's statistical power. The two tests under consideration are the proposed equivalence test for semipartial correlation coefficients ("sr test") and the equivalence test for correlations based on Fisher's Z transformation ("Z test").

a correlation matrix in which all off-diagonal elements were equal to either $\rho_X = 0.1$ or to $\rho_X = 0.2$. Finally, the outcome data, Y, was simulated such that the true value of sr_1 was equal to 0, or equal to Δ, or equal to 0.05.

For each of the different configurations within the simulation study, we simulated 500,000 unique datasets and calculated a p-value with the proposed equivalence test. We then calculated the proportion of these p-values less than $\alpha = 0.05$. We also used the proposed formula for approximating statistical power for each scenario to calculate the approximate power. We specifically chose to conduct 500,000 simulation runs so as to keep computing time within a reasonable limit while also reducing the amount of Monte Carlo standard error to a very negligible amount (for looking at type 1 error with $\alpha = 0.05$, Monte Carlo SE will be approximately $0.0003 \approx \sqrt{0.05(1-0.05)/500,000}$; see Morris et al. (2019)).

The simulation study was done using the R statistical software with default simulation routines (R Core Team, 2020). Results are displayed in Table and suggest that, in practice, the proposed test ("sr test") has correct type 1 error and that the proposed formula for estimating statistical power ("approx pwr.") is reasonably accurate.

N	Δ	K	ρ_X	$\Pr(p < 0.05 \mid sr_1 = \Delta)$		$\Pr(p < 0.05 \mid sr_1 = 0)$		$\Pr(p < 0.05 \mid sr_1 = 0.05)$	
				sr test	approx pwr.	sr test	approx pwr.	sr test	approx pwr.
54	0.10	2.0	0.10	0.000	0.000	0.000	0.000	0.000	0.000
54	0.15	2.0	0.10	0.000	0.000	0.000	0.000	0.000	0.000
54	0.20	2.0	0.10	0.004	0.000	0.000	0.000	0.000	0.000
180	0.10	2.0	0.10	0.000	0.000	0.000	0.000	0.000	0.000
180	0.15	2.0	0.10	0.044	0.046	0.303	0.301	0.225	0.229
180	0.20	2.0	0.10	0.049	0.050	0.718	0.713	0.601	0.598
540	0.10	2.0	0.10	0.049	0.049	0.528	0.527	0.284	0.285
540	0.15	2.0	0.10	0.049	0.050	0.945	0.942	0.756	0.754
540	0.20	2.0	0.10	0.049	0.050	0.998	0.998	0.970	0.968
3500	0.10	2.0	0.10	0.050	0.050	1.000	1.000	0.910	0.909
3500	0.15	2.0	0.10	0.050	0.050	1.000	1.000	1.000	1.000
3500	0.20	2.0	0.10	0.050	0.050	1.000	1.000	1.000	1.000
54	0.10	4.0	0.10	0.000	0.000	0.000	0.000	0.000	0.000
54	0.15	4.0	0.10	0.000	0.000	0.000	0.000	0.000	0.000
54	0.20	4.0	0.10	0.003	0.000	0.000	0.000	0.000	0.000
180	0.10	4.0	0.10	0.000	0.000	0.000	0.000	0.000	0.000
180	0.15	4.0	0.10	0.044	0.045	0.299	0.299	0.225	0.227
180	0.20	4.0	0.10	0.050	0.050	0.717	0.711	0.600	0.596
540	0.10	4.0	0.10	0.048	0.049	0.525	0.524	0.282	0.284
540	0.15	4.0	0.10	0.049	0.050	0.943	0.941	0.756	0.753
540	0.20	4.0	0.10	0.049	0.050	0.998	0.998	0.971	0.968
3500	0.10	4.0	0.10	0.049	0.050	1.000	1.000	0.909	0.909
3500	0.15	4.0	0.10	0.049	0.050	1.000	1.000	1.000	1.000
3500	0.20	4.0	0.10	0.049	0.050	1.000	1.000	1.000	1.000
54	0.10	2.0	0.25	0.000	0.000	0.000	0.000	0.000	0.000
54	0.15	2.0	0.25	0.000	0.000	0.000	0.000	0.000	0.000
54	0.20	2.0	0.25	0.007	0.010	0.000	0.000	0.000	0.000
180	0.10	2.0	0.25	0.000	0.000	0.000	0.000	0.000	0.000
180	0.15	2.0	0.25	0.045	0.047	0.303	0.301	0.227	0.231
180	0.20	2.0	0.25	0.049	0.050	0.718	0.713	0.602	0.600
540	0.10	2.0	0.25	0.049	0.049	0.528	0.527	0.285	0.286
540	0.15	2.0	0.25	0.049	0.050	0.945	0.942	0.757	0.756
540	0.20	2.0	0.25	0.049	0.050	0.998	0.998	0.971	0.969
3500	0.10	2.0	0.25	0.050	0.050	1.000	1.000	0.911	0.910
3500	0.15	2.0	0.25	0.050	0.050	1.000	1.000	1.000	1.000
3500	0.20	2.0	0.25	0.050	0.050	1.000	1.000	1.000	1.000
54	0.10	4.0	0.25	0.000	0.000	0.000	0.000	0.000	0.000
54	0.15	4.0	0.25	0.000	0.000	0.000	0.000	0.000	0.000
54	0.20	4.0	0.25	0.002	0.000	0.000	0.000	0.000	0.000
180	0.10	4.0	0.25	0.000	0.000	0.000	0.000	0.000	0.000
180	0.15	4.0	0.25	0.043	0.045	0.295	0.295	0.225	0.226
180	0.20	4.0	0.25	0.049	0.050	0.714	0.708	0.600	0.595
540	0.10	4.0	0.25	0.048	0.049	0.521	0.521	0.282	0.283
540	0.15	4.0	0.25	0.049	0.050	0.942	0.940	0.756	0.752
540	0.20	4.0	0.25	0.049	0.050	0.998	0.998	0.971	0.968
3500	0.10	4.0	0.25	0.049	0.050	1.000	1.000	0.909	0.908
3500	0.15	4.0	0.25	0.049	0.050	1.000	1.000	1.000	1.000
3500	0.20	4.0	0.25	0.049	0.050	1.000	1.000	1.000	1.000

Table 4. Results from Simulation Study 3. Note that the maximum type 1 error rate should not exceed $\alpha = 0.05$. When $\Delta > sr_1$, the probability of a p-value less than 0.05 corresponds to the test's statistical power.

R code

All data and code has been saved in csv format and is available in the OSF repository at https://osf.io/5yr92/, DOI 10.17605/OSF.IO/5YR92.

```
###########################################
## Useful functions
###########################################
equiv_corrZ<-function(var1, var2, delta_upper, delta_lower = NA) {

  if(is.na(delta_lower)){delta_lower <-(-delta_upper)}

  corxy<-cor(var1,var2)
  n<-length(var1)

  ##### Run a two t-test procedure for equivalance with Fisher's z transformation ####
  zei_lower <- log((1-delta_lower)/(1+delta_lower))/2
  zei_upper <- log((1+delta_upper)/(1-delta_upper))/2
  zcorxy<-log((1+corxy)/(1-corxy))/2
  equivt1_fz<-(zcorxy+ zei_lower)/(1/sqrt(n-3))
  pvalue1_fz<-1-pnorm(equivt1_fz)
  equivt2_fz<-(zcorxy- zei_upper)/(1/sqrt(n-3))
  pvalue2_fz<-pnorm(equivt2_fz)

  the_results <- c(pvalue_equiv_z=max(c(pvalue1_fz, pvalue2_fz), na.rm=TRUE))
  return(the_results)
}
###########################################

###########################################
equivBeta <- function(Y = rnorm(100),
                      Xmatrix = cbind(rnorm(100), rnorm(100)),
                      DELTA_upper = 0.1,
                      DELTA_lower = -0.1){
  if(is.na(DELTA_lower)[1]){DELTA_lower <-(-DELTA_upper)}
  Xmatrix <- cbind(Xmatrix)
  X <- cbind(1, Xmatrix)
  N <- dim(cbind(X[,-1]))[1]
  K <- dim(cbind(X[,-1]))[2]
  if(length(DELTA_lower)==1){DELTA_lower <- rep(DELTA_lower, K+1)}
  if(length(DELTA_upper)==1){DELTA_upper <- rep(DELTA_upper, K+1)}
```

```
  lmmod <- summary(lm(Y~X[,-1]))
  beta_hat <- lmmod$coef[,1]
  SE_beta_hat <- lmmod$coef[,2]

  mysigma<-summary(lm(Y~X[,-1]))$sigma
  mysigma*sqrt(solve(t(X)%*%X)[k,k])

  pval <- p_lower <- p_upper <- rep(0,K)
  for(k in 1:(K+1)){
    p_lower[k] <- pt((beta_hat[k] - DELTA_lower[k])/SE_beta_hat[k], N-K-1, 0,
                     lower.tail=FALSE)
    p_upper[k] <- pt((-beta_hat[k] + DELTA_upper[k])/SE_beta_hat[k], N-K-1, 0,
                     lower.tail=FALSE)
    pval[k] <- max(c(p_lower[k],p_upper[k]))
  }

  names(beta_hat) <- paste("beta", c(1:dim(X)[2])-1, sep="_")
  names(pval) <- paste("pval", c(1:dim(X)[2])-1, sep="_")
  DELTA = cbind(DELTA_lower, DELTA_upper)
  rownames(DELTA) <- paste("DELTA", c(1:dim(X)[2])-1, sep="_")
  return(list(beta = beta_hat, pval = pval, DELTA = DELTA))
}
############################################

############################################
equivBetaPower <- function(DELTA_upper, DELTA_lower, N, K, SEbetak, true_beta=0){
  ncp1 <- (DELTA_upper-true_beta)/SEbetak
  ncp2 <- (DELTA_lower-true_beta)/SEbetak
  Tstatstar <- qt(1-0.05, N-K-1)
  power = pt(+ncp1-Tstatstar, N-K-1, lower.tail=TRUE) - pt(+ncp2+Tstatstar,
             N-K-1, lower.tail=TRUE)
  return(power)
}
############################################

############################################
equivSR <- function(Y= rnorm(100),
         Xmatrix= cbind(rnorm(100),rnorm(100)),
         DELTA_upper= 0.1,
```

```
       DELTA_lower= NA){

Xmatrix <-cbind(Xmatrix)
X <- cbind(1,Xmatrix)
N <- dim(Xmatrix)[1]
K <- dim(Xmatrix)[2]
kvec= 1:K
if(is.na(DELTA_lower[1])){DELTA_lower <-(-DELTA_upper)}
if(length(DELTA_upper)!=K){DELTA_upper <- rep(DELTA_upper[1], K)}
if(length(DELTA_lower)!=K){DELTA_lower <- rep(DELTA_lower[1], K)}

lmmod <- summary(lm(Y~X[,-1]))
R2 <- lmmod$r.squared
if(K==1){R2Xkmink <- 0; diffR2k<-R2; R2Ymink<-0}
if(K>1){
  R2Ymink <- apply(cbind(kvec),1,function(k)summary(lm(Y~Xmatrix[,-k]))$r.squared)
  R2Xkmink <- apply(cbind(kvec),1,function(k)summary(lm(Xmatrix[,k]~Xmatrix[,-k]))$r.squared)
  diffR2k <- unlist(lapply(c(kvec), function(k) {R2-summary(lm(Y~Xmatrix[,-k]))$r.squared}))
}
lmmod_scale <- summary(lm(scale(Y)~scale(X[,-1])-1))
SPC <- lmmod_scale$coef[,1]*sqrt(1-R2Xkmink)
# should be equal in abs:
c(sqrt(diffR2k),SPC)

# equation (11) of Dudgeon (2016)
SIGMA2_SPC <- (R2^2 - 2*R2 + R2Ymink + 1 - R2Ymink^2)/(N-K-1)
SE_SPC <- sqrt(SIGMA2_SPC)

CI90_upper_squared <- CI95_upper <- CI95_lower <- CI90_upper <-
  CI90_lower <- pval <- pval1 <- pval2 <- rep(0, length(kvec))
for(k in kvec){
  pval1[k] <- pt((SPC[k]-DELTA_lower[k])/SE_SPC[k], N-K-1, lower.tail=FALSE)
  pval2[k] <- pt((DELTA_upper[k]-SPC[k])/SE_SPC[k], N-K-1, lower.tail=FALSE)
  CI90_upper[k] <- SPC[k] - qt(0.05,df=N-K-1)* SE_SPC[k]
  CI90_lower[k] <- SPC[k] + qt(0.05,df=N-K-1)* SE_SPC[k]

  CI95_upper[k] <- SPC[k] - qt(0.025,df=N-K-1)* SE_SPC[k]
  CI95_lower[k] <- SPC[k] + qt(0.025,df=N-K-1)* SE_SPC[k]

  CI90_upper_squared[k] <- (SPC[k] - qt(0.1,df=N-K-1)* SE_SPC[k])^2
```

```
    pval[k] <- max(c(pval1[k], pval2[k]))
  }

  CI90 <- cbind(CI90_lower, CI90_upper)
  CI95 <- cbind(CI95_lower, CI95_upper)
  LEAD <- apply(abs(CI90),1,max)

  names(SPC) <- paste("sr", 1:K, sep="_")
  names(SE_SPC) <- paste("SE_sr", 1:K, sep="_")
  names(pval) <- paste("pval", 1:K, sep="_")
  rownames(CI90) <- paste("CI90", 1:K, sep="_")
  rownames(CI95) <- paste("CI95", 1:K, sep="_")
  names(LEAD) <- paste("LEAD", 1:K, sep="_")
  return(list(sr = unname(SPC), SE_sr= SE_SPC , pval= pval,
              CI90=CI90, CI95=CI95, LEAD=LEAD))
}
############################################

############################################
equivSRPower <- function(DELTA_upper, DELTA_lower, N, K, SESRk, true_SR=0){
  ncp1 <- (DELTA_upper-true_SR)/SESRk
  ncp2 <- (DELTA_lower-true_SR)/SESRk
  Tstatstar <- qt(1-0.05, N-K-1)
  power = pt(+ncp1-Tstatstar, N-K-1, lower.tail=TRUE) - pt(+ncp2+Tstatstar,
            N-K-1, lower.tail=TRUE)
  return(power)
}
############################################

############################################
BFstandardBeta <- function(Y= yvec, Xmatrix= Xmat, BFthres=3, random=FALSE){

  K<-dim(Xmatrix)[2]
  mydata<-data.frame(Y, Xmatrix)
  colnames(mydata) <- c(c("yvector"),paste("X",1:K,sep=""))
  BFmod <- regressionBF(yvector~. , data= mydata)

  BF<-result<-rep(0,K)
  for(k in 1:K){
    whichk<-paste("X",k,sep="")
```

```
    BF_without_k <-BFmod[!grepl(whichk,names(BFmod)$numerator)][
      which.max(nchar(names(BFmod)$numerator[!grepl(whichk,names(BFmod)$numerator)]))]
    BF_full <- BFmod[which.max(nchar(names(BFmod)$numerator))]
    BF[k] <- exp(as.numeric(slot(BF_without_k,
                "bayesFactor")[1]))/exp(as.numeric(slot(BF_full,"bayesFactor")[1]))
    if(BF[k]<= 1/BFthres){result[k]<-"positive" }
    if(BF[k]>BFthres){result[k]<-"negative"}
    if(BF[k]> 1/BFthres & BF[k]<BFthres){result[k]<-"inconclusive"}
  }
  return(list(BF=c(BF), BFthres=c(BFthres),conclusion= result))
}
############################################

############################################
## Hypothetical anxiety study example:
############################################
equivBetaPower(DELTA_upper=2, DELTA_lower=-2, N=40, K=3, SEbetak=0.63)
#0.8540956

# Note: hypothetical data was created with the following code:
#set.seed(123)
#anx <- cbind(1,sample(c(0,1),40,TRUE),round(rnorm(40,15,1.8)), round(rnorm(40,68000,20000)))
#score <- round(anx%*%c(8,0.6,0.5,-0.00001) + rnorm(40,0,2.3),1)

anxiety_data <- read.csv("anxiety.csv")[,-1]
score <- anxiety_data[,1]
anx <- as.matrix(anxiety_data[,-1])

# Study results:
coefficients(summary(lm(score~anx-1)))[,1:2]
confint((lm(score~anx-1)), level=0.95)
confint((lm(score~anx-1)), level=0.90)

# Equivalence test for regression coef:
equivBeta(Y = score, Xmatrix = anx[,-1], DELTA_lower = -2, DELTA_upper = 2)

# or "by hand":
beta_hat <- coefficients(summary(lm(score~anx-1)))[2,1]
SE_beta_hat <- coefficients(summary(lm(score~anx-1)))[2,2]
DELTA_lower <- -2
```

```
DELTA_upper <- 2
N <- 40
K <- 3
p_lower <- pt((beta_hat - DELTA_lower)/SE_beta_hat, N-K-1, 0, lower.tail=FALSE)
p_upper <- pt((-beta_hat + DELTA_upper)/SE_beta_hat, N-K-1, 0, lower.tail=FALSE)
pval <- max(c(p_lower,p_upper))
pval
# 0.017733

# LEAD (or "equivalence confidence interval"):
# 90% CI:
CI90 <- c(beta_hat-qt(1-0.05,N-K-1)*SE_beta_hat, beta_hat+qt(1-0.05,N-K-1)*SE_beta_hat)
LEAD <- max(abs(CI90))
LEAD
#  1.674596
equivBeta(Y = score, Xmatrix = anx[,-1], DELTA_lower = -LEAD, DELTA_upper = LEAD)$pval[2]
#  0.05

############################################
## Salaries example
############################################
Salaries <- read.csv("Salaries.csv")[,-1]
# or obtain data from:
# library(carData)

### simple linear regression:
y <- Salaries$salary
X <- model.matrix(lm(salary ~ sex, data=Salaries))

summary(lm(salary ~ sex, data=Salaries))$coef[1:2,1:2]
# Estimate Std. Error
# (Intercept) 101002.41   4809.386
# sexMale      14088.01   5064.579
summary(lm(salary ~ sex, data=Salaries))$sigma
# 30034.61
salaries_equiv <- equivSR(Y = y, Xmatrix = X[,-1], DELTA_upper = 0.5, DELTA_lower = -0.5)
salaries_equiv["sr"]
# 0.1386102
salaries_equiv["SE_sr"]
# 0.04934876
```

```
equivBeta(Y = y, Xmatrix = X[,-1], DELTA_upper = 5000,  DELTA_lower = -5000)$pval[2]
# 0.9632451

# the same equivalence test done "by hand":
beta_hat <- coefficients(summary(lm(salary~sex,data=Salaries)))[2,1]
SE_beta_hat <- coefficients(summary(lm(salary~sex,data=Salaries)))[2,2]
DELTA_lower <- -5000
DELTA_upper <- 5000
N <- 397
K <- 1
p_lower <- pt((beta_hat - DELTA_lower)/SE_beta_hat, N-K-1, 0, lower.tail=FALSE)
p_upper <- pt((-beta_hat + DELTA_upper)/SE_beta_hat, N-K-1, 0, lower.tail=FALSE)
pval <- max(c(p_lower,p_upper))
pval

equivSR(Y = y, Xmatrix = X[,-1], DELTA_upper = 0.1, DELTA_lower = -0.1)$pval
# 0.7827741
# the same equivalence test done "by hand":
p_lower <- pt((sr1- (-0.10))/SEsr1, 397-1-1, lower.tail=FALSE)
p_upper <- pt((0.10-sr1)/SEsr1,397-1-1, lower.tail=FALSE)
pval <- max(c(p_lower,p_upper))
pval

library(BayesFactor)
sdata <- data.frame(salary = Salaries$salary,
            sex = as.numeric(as.factor(Salaries$sex)) - 1)
regressionBF(salary ~ sex, data = sdata)
# 4.525
linearReg.R2stat(N = 397, p = 1, R2 = summary(lm(salary ~ sex,
        data=Salaries))$r.squared, simple = TRUE)
# 4.525
lmBF(salary ~ sex, data = Salaries)
#  6.177
lmBF(salary ~ sex, data = sdata)
# 4.525

### multiple linear regression:
y <- Salaries$salary
X <- model.matrix(lm(salary ~ sex + yrs.since.phd  + yrs.service + discipline + rank,
                     data=Salaries))
```

```
# NHST p-values
mod1 <- summary(lm(salary ~ sex + yrs.since.phd  + yrs.service + discipline +
           rank, data=Salaries))
mod1$coef[-1,4]
# 2.158412e-01  2.697855e-02  2.142543e-02  1.878412e-09  1.983251e-03  2.296130e-23
mod1$coef[,1:2]

SPCobj <- equivSR(Y = y, Xmatrix = X[,-1], DELTA_upper = 0.1, DELTA_lower = -0.1)
round(SPCobj$pval,4)
# 0.0761 0.3249 0.3577 0.9997 0.6699 1.0000
round(SPCobj$LEAD,4)
# 0.1080 0.1446 0.1480 0.2913 0.1780 0.3999
mod1$sigma
# 22538.65
mod1$r.squared
# 0.4546766

# Bayes Factors
BFs <- (BFstandardBeta(Y= y, Xmatrix=X[,-1])$BF)
BFs
# 3.860594e+00 7.352492e-01 6.036516e-01 1.540123e-07 7.331954e-02 5.592721e-21
round(cbind(mod1$coef[-1,4],SPCobj$pval,BFs,1/BFs),3)
#                            BFs
#sexMale       0.216 0.076 3.861 2.590000e-01
#yrs.since.phd 0.027 0.325 0.735 1.360000e+00
#yrs.service   0.021 0.358 0.604 1.657000e+00
#disciplineB   0.000 1.000 0.000 6.492987e+06
#rankAsstProf  0.002 0.670 0.073 1.363900e+01
#rankProf      0.000 1.000 0.000 1.138788e+15
############################################
## Mindset Theory example
############################################
mind <- read.csv("Mindset.csv")
mind[,"cog"]<-rowMeans(cbind(scale(mind[,"Cattell.Score"]), scale(mind[,"Letter.Sets.Score"])))

# Testing Premise 1: people with growth mind-sets hold learning goals
coefficients(summary(lm(X1.Learning.Goal~Mindset.Score, data=mind)))[2,]
Z <- equiv_corrZ(mind[,"X1.Learning.Goal"], mind[,"Mindset.Score"], 0.2, -Inf)
S <- equivSR(mind[,"X1.Learning.Goal"],
```

```
        mind[,"Mindset.Score"], DELTA_upper=0.2, DELTA_lower=-Inf)

res1 <- c((S$sr)[1], Z, S$pval)
res1
# 0.09774130     0.01450918     0.01582211

# Testing Premise 2: people with fixed mind-sets hold performance goals
coefficients(summary(lm(X2.Performance.Goal~Mindset.Score, data=mind)))[2,]
Z <- equiv_corrZ(mind[,"X2.Performance.Goal"], mind[,"Mindset.Score"], Inf, -0.2)
S <- equivSR(mind[,"X2.Performance.Goal"],
      mind[,"Mindset.Score"], DELTA_upper=Inf, DELTA_lower=-0.20)
res2 <- c((S$sr)[1], Z, S$pval)
res2
#  -0.10894803     0.02576879     0.02749891

# Testing Premise 3: people with fixed mind- sets hold performance-avoidance goals
coefficients(summary(lm(X3.Performance.Avoidance.Goal~Mindset.Score, data=mind)))[2,]
Z <- equiv_corrZ(mind[,"X3.Performance.Avoidance.Goal"], mind[,"Mindset.Score"], Inf, -0.2)
S <- equivSR(mind[,"X3.Performance.Avoidance.Goal"],
      mind[,"Mindset.Score"], DELTA_upper=Inf, DELTA_lower=-0.20)
res3 <- c((S$sr)[1], Z, S$pval)
res3
#  -0.0391385402   0.0003229064   0.0004179885

# Testing Premise 4: people with fixed mind-sets believe
# that talent alone| without effort|creates success
coefficients(summary(lm(X4.Belief.in.Talent~Mindset.Score, data=mind)))[2,]
Z <- equiv_corrZ(mind[,"X4.Belief.in.Talent"], mind[,"Mindset.Score"], Inf, -0.2)
S <- equivSR(mind[,"X4.Belief.in.Talent"],
       mind[,"Mindset.Score"], DELTA_upper=Inf, DELTA_lower=-0.20)
res4 <- c((S$sr)[1], Z, S$pval)
res4
#  -0.061215284    0.001588981    0.001906768

# Testing Premise 5: people with growth mind-sets persist to overcome challenges
coefficients(summary(lm(X5.Response.To.Challenge~Mindset.Score, data=mind)))[2,]
Z <- equiv_corrZ(mind[,"X5.Response.To.Challenge"], mind[,"Mindset.Score"], 0.2, -Inf)
S <- equivSR(mind[,"X5.Response.To.Challenge"],
      mind[,"Mindset.Score"], DELTA_upper=0.20, DELTA_lower=-Inf)
res5 <- c((S$sr)[1], Z, S$pval)
```

```
res5
#  0.055873780    0.001100171    0.001343043

# Testing Premise 6: people with growth mind-sets are more resilient following failure
coefficients(summary(lm(X6.Raven.Test.Score~Mindset.Score, data=mind)))[2,]
Z <- equiv_corrZ(mind[,"X6.Raven.Test.Score"], mind[,"Mindset.Score"], 0.2, -Inf)
S <- equivSR(mind[,"X6.Raven.Test.Score"],
      mind[,"Mindset.Score"], DELTA_upper=0.20, DELTA_lower=-Inf)
res6 <- c((S$sr)[1], Z, S$pval)
res6
#  -1.217740e-01   5.976098e-12   1.525199e-11

# Testing Premise 6a: people with growth mind-sets are
# more resilient following failure when controlling for cognitive ability
summary(lm(X6.Raven.Test.Score~Mindset.Score+ cog, data=mind))
Z1 <- NA
S <- equivSR(mind[,"X6.Raven.Test.Score"],
       as.matrix(mind[,c("Mindset.Score", "cog")]), DELTA_upper=0.20, DELTA_lower=-Inf)
res6a <- c(S$sr[1], Z1, S$pval[1])
res6a
# -5.492464e-02            NA  1.109186e-09

mindset_results <- round(rbind(res1, res2, res3, res4, res5, res6, res6a), digits=3)
mindset_results
#              pvalue_equiv_z pval_1
# res1   0.098          0.015  0.016
# res2  -0.109          0.026  0.027
# res3  -0.039          0.000  0.000
# res4  -0.061          0.002  0.002
# res5   0.056          0.001  0.001
# res6  -0.122          0.000  0.000
# res6a -0.055             NA  0.000
```

References

Fraley, R. C., & Vazire, S. (2014). The N-pact factor: Evaluating the quality of empirical journals with respect to sample size and statistical power. *PLoS ONE*, *9*(10), e109019. https://doi.org/10.1371/journal.pone.0109019.

Goertzen, J. R., & Cribbie, R. A. (2010). Detecting a lack of association: An equivalence testing approach. *British Journal of Mathematical and Statistical Psychology*, *63*(3), 527–537. doi: 10.1348/000711009X475853.

Kühberger, A., Fritz, A., & Scherndl, T. (2014). Publication bias in psychology: a diagnosis based on the correlation between effect size and sample size. *PLoS ONE*, *9*(9), e105825. https://doi.org/10.1371/journal.pone.0105825.

Lakens, D. (2017). Equivalence tests: a practical primer for t-tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, *8*(4), 355–362. doi: 10.1177/1948550617697177.

Marszalek, J. M., Barber, C., Kohlhart, J., & Cooper, B. H. (2011). Sample size in psychological research over the past 30 years. *Perceptual and Motor Skills*, *112*(2), 331–348. doi: 10.2466/03.11.PMS.112.2.331-348.

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. https://doi.org/10.1002/sim.8086.

R Core Team. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Serlin, R. C., Lapsley, D. K., Keren, G., & Lewis, C. (1993). Rational appraisal of psychological research and the good-enough principle. In G. Keren & C. Lewis (Eds.), *A handbook for data analysis in the behavioral sciences: Methodological issues* (pp. 199–228). Hillsdale, NJ: Psychology Press.

Weber, R., & Popova, L. (2012). Testing equivalence in communication research: Theory and application. *Communication Methods and Measures*, *6*(3), 190–213. https://doi.org/10.1080/19312458.2012.703834.