Appendix 1: R Code

```
# Appendix 1 Code for Evaluating Equivalence and Null Hypotheses in the Organizational Sciences
# This code, the demo data, and convenient testing functions are also available on GitHub
# in a repository called jmstanto/eqvtesting. Install the package with the following steps:
# install.packages("devtools")
# devtools::install_github("jmstanto/eqvtesting")

#  Create Figure 1
require("ggplot2")
# Data frame with 3 variables: U is upper bound; CohensD is the center, and L is lower bound
df <- data.frame(Scenario = 1:8,
                 U =       c(0.3, 0.65,  0.65, 0.2,   0.13, -0.1,  0.05, -0.35),
                 CohensD = c(0.0, 0.50,  0.30, 0.15, -0.04, -0.15,-0.30, -0.5),
                 L =      c(-0.3, 0.35, -0.05, 0.1,  -0.21, -0.2, -0.65, -0.65))
# Plots dots with geom_point and CIs with geom_errorbar
ggplot(df, aes(x = Scenario, y = CohensD)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymax = U, ymin = L), width=0.25) +
  scale_x_continuous(breaks=1:8) +
  scale_y_continuous(breaks=round(seq(from=-0.6, to=0.6, by=0.1),2))



# Conduct the analyses for the paper
require("readr") # Prepare to read_csv
sigDF <- read_csv("stressdata.csv")

table(sigDF$GENDER) # Slightly different group sizes

# Check the standard deviations by group
sd(sigDF$SIGPRES[sigDF$GENDER=="Male"])
sd(sigDF$SIGPRES[sigDF$GENDER=="Female"])
# The SDs are slightly different

sigTout <- t.test(sigDF$SIGPRES[sigDF$GENDER=="Male"], sigDF$SIGPRES[sigDF$GENDER=="Female"])
sigTout # Not significant
```

```
# Show the results calibrated as Cohen's d
require("effsize")
cohOut <- cohen.d(SIGPRES ~ GENDER, data=sigDF)
cohOut

# Analyze regular power and TOST power for this research scenario
require(pwr)
pwr.t.test(n=(min(table(sigDF$GENDER))), d=0.3, type="two.sample") # Power of 0.90
require(TOSTER)
powerTOSTtwo(alpha=0.05, N=min(table(sigDF$GENDER)), low_eqbound_d=-0.3, high_eqbound_d=0.3)

# Conduct TOST: Gender diffs in SIGPRES
sigTOST <- dataTOSTtwo(sigDF, "SIGPRES", "GENDER",
                       var_equal = F, low_eqbound = -0.3, high_eqbound = 0.3, desc=T, plots=T)
sigTOST # Shows that the difference is in fact equivalent to zero

# Now conduct the ROPE analysis
require(BEST)
bestOut <- BESTmcmc(sigDF$SIGPRES[sigDF$GENDER=="Male"], sigDF$SIGPRES[sigDF$GENDER=="Female"])
plot(bestOut)

# First, calculate a mean difference in the posterior distribution
meanDiff <- bestOut$mu1 - bestOut$mu2
# The plot shows that plus or minus 0.7 contains 95% of the posterior estimates
lowCt <- length(which(meanDiff < (-0.996))) # How many below the ROPE
highCt <- length(which(meanDiff > (0.996))) # How many above the ROPE
1 - (lowCt + highCt)/length(meanDiff) # Proportion within the ROPE

# And finally, the Bayes factor
require(BayesFactor)
tbfOut2 <- ttestBF(x=sigDF$SIGPRES[sigDF$GENDER=="Male"],
                   y=sigDF$SIGPRES[sigDF$GENDER=="Female"],
                   mu = 0, nullInterval = c(-0.3,0.3),
                   paired = FALSE, posterior = FALSE)
tbfOut2
```

```
# Produces two BF values, one representing the space between d=-0.3 and d=0.3
# And the other representing all other area. The ratio of these is 2295
# As an odds ratio it is very strong evidence in favor of the null interval

########################################################################
########################################################################
# Example null equivalence tests for Correlations, Regression
# Coefficients, and ANOVA Results. Each section contains inference
# by intervals, TOST, Bayesian estimation and Bayes factor.

###################################################
# Test null interval for correlation coefficient r
# These two packages required for all sections.
# install.packages("TOSTER")
# install.packages("BayesFactor")
require("TOSTER")
require("BayesFactor")

sampSize = 250 # Set sample size

set.seed(1) # Control randomization of sampling
# Create a small data set of two random normal variables
testDF <- data.frame(x=rnorm(sampSize),y=rnorm(sampSize))

# Chosen equivalence interval
uleq = 0.2   # Upper limit of equivalence interval
lleq = -0.2  # Lower limit of equivalence interval

# Consider power of TOST
powerTOSTr(alpha=0.05, N=sampSize, low_eqbound_r=lleq, high_eqbound_r=uleq)

#----------------------
# Inference by intervals
corrCI <- cor.test(testDF$x, testDF$y, conf.level=0.90)$conf.int
( corrCI[2] < uleq ) && ( corrCI[1] > lleq )
# Results show TRUE if observed cor within interval
```

```
#----------------------
# TOST
TOSTr(n=250, r=cor(testDF$x,testDF$y), low_eqbound_r=lleq, high_eqbound_r=uleq, alpha=0.05)
# Results plot observed cor and CI within equivalence interval

#----------------------
# Bayes estimation
# Run the BayesFactor procedure for estimating correlations
# This command creates an MCMC posterior distribution.
corrOutMCMC <- correlationBF(testDF$y, testDF$x, rscale = "medium",
                             nullInterval = c(uleq,lleq), posterior = TRUE,
                             iterations=10000)
plot(corrOutMCMC[,"rho"]) # Trace and density plots of posterior estimates
ut <- length(which(corrOutMCMC[,"rho"] > uleq)) # Estimates above equiv zone
lt <- length(which(corrOutMCMC[,"rho"] < lleq)) # Estimate below equiv zone
max(corrOutMCMC[,"rho"]) # Confirms largest
min(corrOutMCMC[,"rho"]) # Confirms smallest
1 - (ut + lt)/10000 # Proportion of estimates within equivalence interval

#----------------------
# Bayes factor
# This command calculates Bayes factors
corrOutBF <- correlationBF(testDF$y, testDF$x, rscale = "medium",
                           nullInterval = c(lleq,uleq), posterior = FALSE)
corrOutBF # Overall output
corrOutBF[1]/corrOutBF[2] # Bayes factor in favor of null interval



##################################################
# Test null interval for a regression coefficient B
sampSize = 250

set.seed(1)
# Generate three random normal variables
testDF <- data.frame(x=rnorm(sampSize),y=rnorm(sampSize), z=rnorm(sampSize))
```

```
# Chosen equivalence interval:
uleq = 0.2   # Upper limit of equivalence interval
lleq = -0.2  # Lower limit of equivalence interval
# Note that these are expressed in the metric of the raw data.
# To calibrate by beta-weights, standardize first or use lm.beta()
# from the QuantPsyc package.

#----------------------
# Inference by intervals
lmOut <- lm(y ~ x + z, data=testDF)
summary(lmOut)
confint(lmOut, 'x', level=0.90) # Show the CI
#         5 %      95 %
#  x -0.06918416 0.1630529

( confint(lmOut, 'x', level=0.90)[1] > lleq ) &&
  ( confint(lmOut, 'x', level=0.90)[2] < uleq )
# Test will be TRUE if CI for the slope on X is within
# the equivalence interval.

#----------------------
# TOST using one sample t-test on B weight
TOSTone.raw(m=lmOut$coefficients['x'],
            mu=0, sd=summary(lmOut)$coefficients["x","Std. Error"]*sqrt(sampSize),
            n=sampSize, low_eqbound=lleq, high_eqbound=uleq,
            alpha=0.05, plot = TRUE, verbose = TRUE)
# Plot will show CI for slope on X within the equiv interval.

#----------------------
# Bayesian parameter estimation
lmOutMCMC <- lmBF(y ~ x + z, data=testDF, posterior=T, iterations=10000)
plot(lmOutMCMC[,'x']) # Posterior density and traceplots of the B weight
ut <- length(which(lmOutMCMC[,'x'] > 0.2)) # Estimates larger than uleq
lt <- length(which(lmOutMCMC[,'x'] < -0.2)) # Estimates smaller than lleq
max(lmOutMCMC[,'x']) # Confirms largest
```

```
min(lmOutMCMC[,'x']) # Confirms smallest
1 - (ut + lt)/10000 # Proportion of posterior estimates in the null interval
# Recommend that at least 90% of posterior estimates should fall within the null interval

#----------------------
# Bayes factor
# Calculate a Bayes factor by residualizing on the other predictor
# Note that the null interval is here expressed in standardized terms
# because the statistic being tested is the correlation coefficient
lmOut2 <- lm(y ~ z, data=testDF)
corrOutBF <- correlationBF(residuals(lmOut2), testDF$x, rscale = "medium",
                           nullInterval = c(lleq,uleq), posterior = FALSE)
corrOutBF
corrOutBF[1]/corrOutBF[2] # Bayes factor in favor of null interval
# Recommend that a Bayes factor of 50 or higher is observed to support the null interval

#####################################################
# ANOVA test of a factor in a two-factor model
sampSize = 250 # Set sample size

set.seed(1)
# Set up a data set with two factors.
# The two factors use standard effect coding, so they
# are centered on 0 and have an sd of 1. This
# simplifies the choice of upper and lower
# equivalence limits,
testDF <- data.frame(y=rnorm(sampSize),
                     x=scale(sample.int(n=2,size=sampSize,replace=T)),
                     z=scale(sample.int(n=2,size=sampSize,replace=T)))

# Diagnostics on the data set:
table(testDF$x, testDF$z) # Look at how many per cell
# Note that the "codes" for the levels of x and z are fractional
# because of unequal cell sizes.

boxplot(y ~ z + x, data=testDF) # Check the cell means
```

```
aovOut <- aov(y ~ z * x, data=testDF) # Examine a conventional ANOVA
summary(aovOut) # z, x, and z:x all are not significant

# Takes advantage of the fact that ANOVA and Regression
# are mathematically interchangeable. The y is random normal
# and x and z are scaled so no need for centering, but with a
# regular data set it would be wise to standardize first.
# Gives a t-test on each coefficient that we can test
# using the regression techniques shown above.
lmOut <- lm(y ~ z * x, data=testDF)
summary(lmOut) # Note that probabilities are identical to aov() output

# Chosen equivalence interval
uleq = 0.2
lleq = -0.2

#----------------------
# Inference by intervals
confint(lmOut, 'x', level=0.90)
#          5 %      95 %
#  x -0.07782648 0.1238917
( confint(lmOut, 'x', level=0.90)[1] > lleq ) &&
  ( confint(lmOut, 'x', level=0.90)[2] < uleq )
# Note that if result is true, the B on the x factor is
# contained within the null interval.

#----------------------
# TOST results
TOSTone.raw(m=lmOut$coefficients['x'],
            mu=0, sd=summary(lmOut)$coefficients["x","Std. Error"]*sqrt(sampSize),
            n=250, low_eqbound=lleq, high_eqbound=uleq,
            alpha=0.05, plot = TRUE, verbose = TRUE)
# TOST results confirm whether the B weight is within the null interval

#----------------------
# Bayesian parameter estimation
```

```
lmOutMCMC <- lmBF(y ~ x * z, data=testDF, posterior=T, iterations=10000)
plot(lmOutMCMC[,'x']) # Posterior density and traceplots of the B weight
ut <- length(which(lmOutMCMC[,'x'] > uleq)) # Number of estimates larger than uleq
lt <- length(which(lmOutMCMC[,'x'] < lleq)) # Number of estimates smaller than lleq
max(lmOutMCMC[,'x']) # Confirms largest
min(lmOutMCMC[,'x']) # Confirms smallest
1 - (ut + lt)/10000 # Proportion of posterior estimates in the null interval
# Recommend that at least 90% of posterior estimates should fall within the null interval

#----------------------
# Bayes factor
# Calculate a Bayes factor by residualizing on the other predictor
# plus the interaction term.
# Note that the null interval is here expressed in standardized terms
# because the statistic being tested is the correlation coefficient.
lmOut2 <- lm(y ~ z + z:x, data=testDF)
corrOutBF <- correlationBF(residuals(lmOut2), testDF$x, rscale = "medium",
                           nullInterval = c(lleq,uleq), posterior = FALSE)
corrOutBF
corrOutBF[1]/corrOutBF[2] # Bayes factor in favor of null interval
# Recommend that a Bayes factor of 50 or higher is observed to support the null interval
```