Package 'SE.EQ'

October 12, 2022

Type Package

Title SE-Test for Equivalence

Version 1.0

Date 2020-10-06

Author Thomas Hoffelder

Maintainer Thomas Hoffelder <thomas.hoffelder@boehringer-ingelheim.com>

Description Implements the SE-test for equivalence
according to Hoffelder et al. (2015) <DOI:10.1080/10543406.2014.920344>.
The SE-test for equivalence is a multivariate two-sample equivalence test.
Distance measure of the test is the sum of standardized differences
between the expected values or in other words: the sum of effect sizes (SE)
of all components of the two multivariate samples.
The test is an asymptotically valid test for normally distributed data
(see Hoffelder et al.,2015).
The function SE.EQ() implements the SE-test for equivalence
according to Hoffelder et al. (2015).
The function SE.EQ.dissolution.profiles() implements a variant
of the SE-test for equivalence for similarity analyses of dissolution
profiles as mentioned in Suarez-Sharp et al.(2020)
<DOI:10.1208/s12248-020-00458-9>). The equivalence margin used in
SE.EQ.dissolution.profiles() is analogically defined as for the T2EQ
approach according to Hoffelder (2019) <DOI:10.1002/bimj.201700257>)
by means of a systematic shift in location
of 10 [\% of label claim] of both dissolution profile populations.
SE.EQ.dissolution.profiles() checks whether the weighted mean of the
differences of the expected values of both dissolution profile populations
is statistically significantly smaller than 10 [\% of label claim]. The
weights are built up by the inverse variances.

Imports MASS

License GPL-3

NeedsCompilation no

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2020-10-13 14:10:05 UTC

R topics documented:

`SE.EQ-package` *SE-Test for Equivalence*

Description

Implements the SE-test for equivalence according to Hoffelder et al. (2015) <DOI:10.1080/10543406.2014.920344>. The SE-test for equivalence is a multivariate two-sample equivalence test. Distance measure of the test is the sum of standardized differences between the expected values or in other words: the sum of effect sizes (SE) of all components of the two multivariate samples. The test is an asymptotically valid test for normally distributed data (see Hoffelder et al.,2015). The function SE.EQ() implements the SE-test for equivalence according to Hoffelder et al. (2015). The function SE.EQ.dissolution.profiles() implements a variant of the SE-test for equivalence for similarity analyses of dissolution profiles as mentioned in Suarez-Sharp et al.(2020) <DOI:10.1208/s12248-020-00458-9>). The equivalence margin used in SE.EQ.dissolution.profiles() is analogically defined as for the T2EQ approach according to Hoffelder (2019) <DOI:10.1002/bimj.201700257>) by means of a systematic shift in location of 10 [% of label claim] of both dissolution profile populations. SE.EQ.dissolution.profiles() checks whether the weighted mean of the differences of the expected values of both dissolution profile populations is statistically significantly smaller than 10 [% of label claim]. The weights are built up by the inverse variances.

Details

The DESCRIPTION file:

Package:	SE.EQ
Type:	Package
Title:	SE-Test for Equivalence
Version:	1.0
Date:	2020-10-06
Author:	Thomas Hoffelder
Maintainer:	Thomas Hoffelder <thomas.hoffelder@boehringer-ingelheim.com>
Description:	Implements the SE-test for equivalence according to Hoffelder et al. (2015) <DOI:10.1080/10543406.2014.920
Imports:	MASS
License:	GPL-3

Index of help topics:

`SE.EQ` The SE-test for equivalence

```
SE.EQ-package           SE-Test for Equivalence
SE.EQ.dissolution.profiles
                        The SE-test for equivalence for dissolution
                        profile similarity analyses
ex_data_JoBS            Example dataset from Hoffelder et al. (2015)
```

Author(s)

Thomas Hoffelder

Maintainer: Thomas Hoffelder <thomas.hoffelder@boehringer-ingelheim.com>

References

EMA (2010). Guidance on the Investigation of Bioequivalence. European Medicines Agency, CHMP, London. Doc. Ref.: CPMP/EWP/QWP/1401/98 Rev. 1/ Corr **. URL: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-investigation-bioequivalence-rev1_en.pdf

FDA (1997). Guidance for Industry: Dissolution Testing of Immediate Release Solid Oral Dosage Forms. Food and Drug Administration FDA, CDER, Rockville. URL: https://www.fda.gov/media/70936/download

Hoffelder, T., Goessl, R., Wellek, S. (2015). Multivariate Equivalence Tests for Use in Pharmaceutical Development. *Journal of Biopharmaceutical Statistics*, 25:3, 417-437. URL: http://dx.doi.org/10.1080/10543406.2014.920344

Hoffelder, T. (2019) Equivalence analyses of dissolution profiles with the Mahalanobis distance. *Biometrical Journal*, 61:5, 1120-1137. URL: https://doi.org/10.1002/bimj.201700257

Suarez-Sharp, S., Abend, A., Hoffelder, T., Leblond, D., Delvadia, P., Kovacs, E., Diaz, D.A. (2020). In Vitro Dissolution Profiles Similarity Assessment in Support of Drug Product Quality: What, How, When - Workshop Summary Report. *The AAPS Journal*, 22:74. URL: http://dx.doi.org/10.1208/s12248-020-00458-9

Examples

```
# A reproduction of the three-dimensional SE example evaluation
# in Hoffelder et al. (2015) can be done with the following code:


data(ex_data_JoBS)
REF_JoBS <- cbind(ex_data_JoBS[ which(ex_data_JoBS$Group=='REF'), ]
                  [c("Diss_15_min","Diss_20_min","Diss_25_min")])
TEST_JoBS <- cbind(ex_data_JoBS[ which(ex_data_JoBS$Group=='TEST'), ]
                   [c("Diss_15_min","Diss_20_min","Diss_25_min")])
equivalence_margin_SE_JoBS <- 0.74^2
test_SE_JoBS <- SE.EQ(X=REF_JoBS
                      , Y=TEST_JoBS
                      , eq_margin=equivalence_margin_SE_JoBS
                      , print.results = TRUE)


# Apart from simulation errors, a recalculation of the SE results
```

```
# of some parts (normal distribution only) of the simulation study in
# Hoffelder et al. (2015) can be done with the following code. Please note that
# the simulation takes approximately 20 minutes for 50.000 simulation
# runs (number_of_simu_runs <- 50000). To shorten calculation time for
# test users, number_of_simu_runs is set to 100 here and can/should be adapted.
# In the result of the simulation the variable empirical.size.se presents the
# simulated size obtained by function \code{SE.EQ()} whereas variable
# empirical.size.se.disso shows the
# simulated size obtained by function \code{SE.EQ.dissolution.profiles()}.
# A detailed analysis of the operating characteristics of the SE variant
# implemented in \code{SE.EQ.dissolution.profiles()} is the content of
# a future paper.

library(MASS)
number_of_simu_runs <- 100
set.seed(2020)

mu1 <- c(41,76,97)
mu2 <- mu1 - c(10,10,10)
SIGMA_1 <- matrix(data = c(537.4 , 323.8 , 91.8 ,
                           323.8 , 207.5 , 61.7 ,
                           91.8  , 61.7  , 26.1) ,ncol = 3)
SIGMA_2 <- matrix(data = c(324.1 , 233.6 , 24.5 ,
                           233.6 , 263.5 , 61.4 ,
                           24.5  , 61.4  , 32.5) ,ncol = 3)
SIGMA   <- matrix(data = c(430.7 , 278.7 , 58.1 ,
                           278.7 , 235.5 , 61.6 ,
                           58.1  , 61.6  , 29.3) ,ncol = 3)


SIMULATION_SIZE_SE <- function(disttype , Hom , Var , mu_1 , mu_2
                               ,  n_per_group , n_simus ) {

  n_success_SE        <- 0
  n_success_SE_disso  <- 0
  if  ( Hom == "Yes" )  {
    COVMAT_1 <- SIGMA
    COVMAT_2 <- SIGMA
  }
  else     {
    COVMAT_1 <- SIGMA_1
    COVMAT_2 <- SIGMA_2
  }
  if  ( Var == "Low" )  {
    COVMAT_1 <- COVMAT_1 / 4
    COVMAT_2 <- COVMAT_2 / 4
  }

  d <- ncol(COVMAT_1)
  Mean_diff <- mu_1 - mu_2                  # Difference of both exp. values
  vars_X      <- diag(COVMAT_1)             #  variances of first sample
  vars_Y      <- diag(COVMAT_2)             #  variances of second sample
  dist_SE     <- sum( (Mean_diff * Mean_diff) / (0.5 * (vars_X + vars_Y) ) )
```

```
    # true SE distance and equivalence margin for SE.EQ


  if  ( n_per_group == 10 )  {
    cat("Expected value sample 1:",mu_1,"\n",
        "Expected value sample 2:",mu_2,"\n",
        "Covariance matrix sample 1:",COVMAT_1,"\n",
        "Covariance matrix sample 2:",COVMAT_2,"\n",
        "EM_SE:",dist_SE,"\n")
  }

  for (i in 1:n_simus)  {
    if  ( disttype == "Normal" )  {
      REF <- mvrnorm(n = n_per_group, mu=mu_1, Sigma=COVMAT_1)
      TEST<- mvrnorm(n = n_per_group, mu=mu_2, Sigma=COVMAT_2)
    }
    n_success_SE_disso  <- n_success_SE_disso +
                           SE.EQ.dissolution.profiles( X = REF ,
                                                       Y = TEST ,
                                                       print.results = FALSE
                                                      )$testresult.num
    n_success_SE        <- n_success_SE   +
                           SE.EQ( X=REF ,
                                  Y=TEST ,
                                  eq_margin = dist_SE ,
                                  print.results = FALSE
                                 )$testresult.num
  }
  empirical_succ_prob_SE        <- n_success_SE / n_simus
  empirical_succ_prob_SE_disso  <- n_success_SE_disso / n_simus
  simuresults  <- data.frame(dist = disttype , Hom = Hom , Var = Var
                     , dimension = d , em_se = dist_SE
                     , sample.size = n_per_group
                     , empirical.size.se = empirical_succ_prob_SE
                     , empirical.size.se.disso = empirical_succ_prob_SE_disso)
}

SIMULATION_LOOP_SAMPLE_SIZE <- function(disttype , Hom , Var
                                        , mu_1 , mu_2 ,  n_simus ) {

  run_10  <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 10 , n_simus = n_simus)
  run_30  <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 30 , n_simus = n_simus)
  run_50  <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 50 , n_simus = n_simus)
  run_100 <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 100 , n_simus = n_simus)
  RESULT_MATRIX <- rbind(run_10 , run_30 , run_50 , run_100)
```

```
  RESULT_MATRIX
}

simu_1 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "Yes"
                                      , Var = "High" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)
simu_2 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "Yes"
                                      , Var = "Low" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)
simu_3 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "No"
                                      , Var = "High" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)
simu_4 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "No"
                                      , Var = "Low" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)

FINAL_RESULT <- rbind(simu_1 , simu_2 , simu_3 , simu_4)

cat("******  Simu results n_simu_runs: ",number_of_simu_runs,"   ***** \n")
FINAL_RESULT
```

`ex_data_JoBS` *Example dataset from Hoffelder et al. (2015)*

Description

Multivariate example dataset of dissolution profiles. Dataset consists of two three-dimensional samples. The names of the three variables are "Diss_15_min","Diss_20_min" and "Diss_25_min". Variable "Group" discriminates between first sample (`Group == "REF"`) and second sample (`Group == "Test"`). Sample size is 12 per group.

Usage

```
data("ex_data_JoBS")
```

Format

A data frame with 24 observations on the following 4 variables.

`Group` a factor with levels `REF` `TEST`

`Diss_15_min` a numeric vector

`Diss_20_min` a numeric vector

`Diss_25_min` a numeric vector

Details

Example dataset from Hoffelder et al. (2015).

Source

Hoffelder, T., Goessl, R., Wellek, S. (2015), "Multivariate Equivalence Tests for Use in Pharmaceutical Development", *Journal of Biopharmaceutical Statistics*, 25:3, 417-437.

References

URL: http://dx.doi.org/10.1080/10543406.2014.920344

Examples

```
data(ex_data_JoBS)
```

SE.EQ *The SE-test for equivalence*

Description

The function SE.EQ() implements the SE-test for equivalence according to Hoffelder et al. (2015). It is a multivariate two-sample equivalence procedure. Distance measure of the test is the sum of standardized differences between the expected values or in other words: the sum of effect sizes of all components of the two multivariate samples.

Usage

```
SE.EQ(X, Y, eq_margin, alpha = 0.05, print.results = TRUE)
```

Arguments

X	numeric data matrix of the first sample (REF). The rows of X contain the individual observations of the REF sample, the columns contain the variables/components of the multivariate sample.
Y	numeric data matrix of the second sample (TEST). The rows of Y contain the individual observations of the TEST sample, the columns contain the variables/components of the multivariate sample.
eq_margin	numeric (>0). The equivalence margin of the test.
alpha	numeric (0<alpha<1). The significance level of the SE-test for equivalence. Usually set to 0.05 which is the default.
print.results	logical; if TRUE (default) summary statistics and test results are printed in the output. If FALSE no output is created

Details

This function implements the SE-test for equivalence. Distance measure of the test is the sum of standardized differences between the expected values or in other words: the sum of effect sizes of all components of the two multivariate samples. The test is an asymptotically valid test for normally distributed data (see Hoffelder et al.,2015).

Value

a data frame; three columns containing the results of the test

`p.value`	numeric; the p-value of the SE test for equivalence
`testresult.num`	numeric; 0 (null hypothesis of nonequivalence not rejected) or 1 (null hypothesis of nonequivalence rejected, decision in favor of equivalence)
`testresult.text`	character; test result of the test in text mode

Author(s)

Thomas Hoffelder <thomas.hoffelder at boehringer-ingelheim.com>

References

Hoffelder, T., Goessl, R., Wellek, S. (2015). Multivariate Equivalence Tests for Use in Pharmaceutical Development. *Journal of Biopharmaceutical Statistics*, 25:3, 417-437. URL: http://dx.doi.org/10.1080/10543406.2014.920344

Examples

```
# A reproduction of the three-dimensional SE example evaluation
# in Hoffelder et al. (2015) can be done with the following code:

data(ex_data_JoBS)
REF_JoBS <- cbind(ex_data_JoBS[ which(ex_data_JoBS$Group=='REF'), ]
                  [c("Diss_15_min","Diss_20_min","Diss_25_min")])
TEST_JoBS <- cbind(ex_data_JoBS[ which(ex_data_JoBS$Group=='TEST'), ]
                   [c("Diss_15_min","Diss_20_min","Diss_25_min")])
equivalence_margin_SE_JoBS <- 0.74^2
test_SE_JoBS <- SE.EQ(X=REF_JoBS
                      , Y=TEST_JoBS
                      , eq_margin=equivalence_margin_SE_JoBS
                      , print.results = TRUE)
```

SE.EQ.dissolution.profiles

The SE-test for equivalence for dissolution profile similarity analyses

---

Description

The function `SE.EQ.dissolution.profiles()` implements a variant of the SE-test for equivalence with a concrete equivalence margin for analyses of dissolution profiles. It is a multivariate two-sample equivalence procedure. Distance measure of the test is the sum of standardized differences between the expected values or in other words: the sum of effect sizes of all components of the two multivariate samples.

Usage

```
SE.EQ.dissolution.profiles(X, Y, alpha = 0.05, print.results = TRUE)
```

Arguments

X	numeric data matrix of the first sample (REF). The rows of X contain the individual observations of the REF sample, the columns contain the variables/components of the multivariate sample.
Y	numeric data matrix of the second sample (TEST). The rows of Y contain the individual observations of the TEST sample, the columns contain the variables/components of the multivariate sample.
alpha	numeric (0<alpha<1). The significance level of the SE-test for equivalence. Usually set to 0.05 which is the default.
print.results	logical; if TRUE (default) summary statistics and test results are printed in the output. If FALSE no output is created

Details

The function SE.EQ.dissolution.profiles() implements a variant of the SE-test for equivalence for similarity analyses of dissolution profiles as mentioned in Suarez-Sharp et al.(2020) <DOI:10.1208/s12248-020-00458-9>). The equivalence margin is analogically defined as for the T2EQ approach according to Hoffelder (2019) <DOI:10.1002/bimj.201700257>) by means of a systematic shift in location of 10 [% of label claim] of both dissolution profile populations. SE.EQ.dissolution.profiles() checks whether the weighted mean of the differences between the expected values of both dissolution profile populations is statistically significantly smaller than 10 [% of label claim]. The weights are built up by the inverse variances.

The current regulatory standard approach for comparing dissolution profiles is the similarity factor f2 (see FDA, 1997, EMA, 2010, among others) with which the type I error cannot be controlled. According to EMA (2010) "similarity acceptance limits should be pre-defined and justified and not be greater than a 10% difference". The functions

- SE.EQ.dissolution.profiles
- EDNE.EQ.dissolution.profiles
- T2EQ.dissolution.profiles.hoffelder

and f2 have in common that they all check wether a kind of average difference between the expected values is smaller than 10 [% of label claim] (see Suarez-Sharp et al., 2020). Thus, all three methods

- SE.EQ.dissolution.profiles
- EDNE.EQ.dissolution.profiles
- T2EQ.dissolution.profiles.hoffelder

are compliant with current regulatory requirements. In contrast to the standard approach f2 they all allow (at least approximate) type I error control.

Value

a data frame; three columns containing the results of the test

`p.value`	numeric; the p-value of the SE test for equivalence
`testresult.num`	numeric; 0 (null hypothesis of nonequivalence not rejected) or 1 (null hypothesis of nonequivalence rejected, decision in favor of equivalence)
`testresult.text`	character; test result of the test in text mode

Author(s)

Thomas Hoffelder <thomas.hoffelder at boehringer-ingelheim.com>

References

EMA (2010). Guidance on the Investigation of Bioequivalence. European Medicines Agency, CHMP, London. Doc. Ref.: CPMP/EWP/QWP/1401/98 Rev. 1/ Corr **. URL: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-investigation-bioequivalence-rev1_en.pdf

FDA (1997). Guidance for Industry: Dissolution Testing of Immediate Release Solid Oral Dosage Forms. Food and Drug Administration FDA, CDER, Rockville. URL: https://www.fda.gov/media/70936/download

Hoffelder, T., Goessl, R., Wellek, S. (2015). Multivariate Equivalence Tests for Use in Pharmaceutical Development. *Journal of Biopharmaceutical Statistics*, 25:3, 417-437. URL: http://dx.doi.org/10.1080/10543406.2014.920344

Hoffelder, T. (2019) Equivalence analyses of dissolution profiles with the Mahalanobis distance. *Biometrical Journal*, 61:5, 1120-1137. URL: https://doi.org/10.1002/bimj.201700257

Suarez-Sharp, S., Abend, A., Hoffelder, T., Leblond, D., Delvadia, P., Kovacs, E., Diaz, D.A. (2020). In Vitro Dissolution Profiles Similarity Assessment in Support of Drug Product Quality: What, How, When - Workshop Summary Report. *The AAPS Journal*, 22:74. URL: http://dx.doi.org/10.1208/s12248-020-00458-9

Examples

```
# Apart from simulation errors, a recalculation of the SE results
# of some parts (normal distribution only) of the simulation study in
# Hoffelder et al. (2015) can be done with the following code. Please note that
# the simulation takes approximately 20 minutes for 50.000 simulation
# runs (number_of_simu_runs <- 50000). To shorten calculation time for
# test users, number_of_simu_runs is set to 100 here and can/should be adapted.
# In the result of the simulation the variable empirical.size.se presents the
# simulated size obtained by function \code{SE.EQ()} whereas variable
# empirical.size.se.disso shows the
# simulated size obtained by function \code{SE.EQ.dissolution.profiles()}.
# A detailed analysis of the operating characteristics of the SE variant
# implemented in \code{SE.EQ.dissolution.profiles()} is the content of
# a future paper.
```

```
library(MASS)
number_of_simu_runs <- 100
set.seed(2020)

mu1 <- c(41,76,97)
mu2 <- mu1 - c(10,10,10)
SIGMA_1 <- matrix(data = c(537.4 , 323.8 , 91.8 ,
                           323.8 , 207.5 , 61.7 ,
                           91.8  , 61.7  , 26.1) ,ncol = 3)
SIGMA_2 <- matrix(data = c(324.1 , 233.6 , 24.5 ,
                           233.6 , 263.5 , 61.4 ,
                           24.5  , 61.4  , 32.5) ,ncol = 3)
SIGMA   <- matrix(data = c(430.7 , 278.7 , 58.1 ,
                           278.7 , 235.5 , 61.6 ,
                           58.1  , 61.6  , 29.3) ,ncol = 3)


SIMULATION_SIZE_SE <- function(disttype , Hom , Var , mu_1 , mu_2
                               ,  n_per_group , n_simus ) {

  n_success_SE        <- 0
  n_success_SE_disso  <- 0
  if  ( Hom == "Yes" )  {
    COVMAT_1 <- SIGMA
    COVMAT_2 <- SIGMA
  }
  else     {
    COVMAT_1 <- SIGMA_1
    COVMAT_2 <- SIGMA_2
  }
  if  ( Var == "Low" )  {
    COVMAT_1 <- COVMAT_1 / 4
    COVMAT_2 <- COVMAT_2 / 4
  }

  d <- ncol(COVMAT_1)
  Mean_diff <- mu_1 - mu_2                    # Difference of both exp. values
  vars_X      <- diag(COVMAT_1)               #  variances of first sample
  vars_Y      <- diag(COVMAT_2)               #  variances of second sample
  dist_SE     <- sum( (Mean_diff * Mean_diff) / (0.5 * (vars_X + vars_Y) ) )
     # true SE distance and equivalence margin for SE.EQ


  if  ( n_per_group == 10 )  {
    cat("Expected value sample 1:",mu_1,"\n",
        "Expected value sample 2:",mu_2,"\n",
        "Covariance matrix sample 1:",COVMAT_1,"\n",
        "Covariance matrix sample 2:",COVMAT_2,"\n",
        "EM_SE:",dist_SE,"\n")
  }

  for (i in 1:n_simus)  {
```

```
    if  ( disttype == "Normal" )  {
      REF <- mvrnorm(n = n_per_group, mu=mu_1, Sigma=COVMAT_1)
      TEST<- mvrnorm(n = n_per_group, mu=mu_2, Sigma=COVMAT_2)
    }
    n_success_SE_disso  <- n_success_SE_disso +
                           SE.EQ.dissolution.profiles( X = REF ,
                                                       Y = TEST ,
                                                       print.results = FALSE
                                                      )$testresult.num
    n_success_SE        <- n_success_SE   +
                           SE.EQ( X=REF ,
                                  Y=TEST ,
                                  eq_margin = dist_SE ,
                                  print.results = FALSE
                                 )$testresult.num
  }
  empirical_succ_prob_SE        <- n_success_SE / n_simus
  empirical_succ_prob_SE_disso  <- n_success_SE_disso / n_simus
  simuresults  <- data.frame(dist = disttype , Hom = Hom , Var = Var
                    , dimension = d , em_se = dist_SE
                    , sample.size = n_per_group
                    , empirical.size.se = empirical_succ_prob_SE
                    , empirical.size.se.disso = empirical_succ_prob_SE_disso)
}

SIMULATION_LOOP_SAMPLE_SIZE <- function(disttype , Hom , Var
                                        , mu_1 , mu_2 ,  n_simus ) {

  run_10  <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 10 , n_simus = n_simus)
  run_30  <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 30 , n_simus = n_simus)
  run_50  <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 50 , n_simus = n_simus)
  run_100 <- SIMULATION_SIZE_SE(disttype = disttype , Hom = Hom , Var = Var
                                , mu_1 = mu_1 , mu_2 = mu_2
                                , n_per_group = 100 , n_simus = n_simus)
  RESULT_MATRIX <- rbind(run_10 , run_30 , run_50 , run_100)
  RESULT_MATRIX
}

simu_1 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "Yes"
                                      , Var = "High" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)
simu_2 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "Yes"
                                      , Var = "Low" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)
simu_3 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "No"
                                      , Var = "High" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)
```

```
simu_4 <- SIMULATION_LOOP_SAMPLE_SIZE(disttype = "Normal", Hom = "No"
                                      , Var = "Low" , mu_1 = mu1  , mu_2 = mu2
                                      , n_simus = number_of_simu_runs)

FINAL_RESULT <- rbind(simu_1 , simu_2 , simu_3 , simu_4)

cat("******  Simu results n_simu_runs: ",number_of_simu_runs,"   ***** \n")
FINAL_RESULT
```

Index