

# Package ‘reproducer’

May 9, 2026

**Title** Reproduce Statistical Analyses and Meta-Analyses

**Version** 0.5.3

**Date** 2023-10-18

**Maintainer** Lech Madeyski <lech.madeyski@gmail.com>

**Description** Includes data analysis and meta-analysis functions (e.g., to calculate effect sizes and 95% Confidence Intervals (CI) on Standardised Effect Sizes (d) for AB/BA cross-over repeated-measures experimental designs), data presentation functions (e.g., density curve overlaid on histogram), and the data sets analyzed in different research papers in software engineering (e.g., related to software defect prediction or multi-site experiment concerning the extent to which structured abstracts were clearer and more complete than conventional abstracts) to streamline reproducible research in software engineering.

**License** GPL (>= 2)

**URL** <https://madeyski.e-informatyka.pl/reproducible-research/>

**Depends** R (>= 3.5.0)

**Imports** dplyr(>= 0.8.0.1), GetoptLong(>= 0.1.7), ggplot2(>= 2.0.0), gridExtra(>= 0.9.1), httr(>= 1.4.0), jsonlite(>= 1.6), lme4(>= 1.1-10), MASS(>= 7.3-45), metafor(>= 1.9-2), nortest(>= 1.0-4), openxlsx(>= 2.4.0), readr(>= 1.3.1), reshape(>= 0.8.8), stats(>= 3.5.0), stringr(>= 1.4.0), tibble(>= 2.1.1), tidyr(>= 0.8.3), xtable(>= 1.7-4)

**Suggests** assertthat, testthat

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Lech Madeyski [cre, aut, ctb] (The main contributor and maintainer),  
Barbara Kitchenham [ctb] (Data and code contributor),  
Tomasz Lewowski [ctb] (Data and code contributor),  
Marian Jureczko [ctb] (Data contributor),

David Budgen [ctb] (Data contributor),  
 Pearl Brereton [ctb] (Data contributor),  
 Jacky Keung [ctb] (Data contributor),  
 Stuart Charters [ctb] (Data contributor),  
 Shirley Gibbs [ctb] (Data contributor),  
 Amnart Pohthong [ctb] (Data contributor),  
 Giuseppe Scanniello [ctb] (Data contributor),  
 Carmine Gravino [ctb] (Data contributor)

**Repository** CRAN

**Date/Publication** 2023-10-18 14:40:02 UTC

## Contents

aggregateIndividualDocumentStatistics . . . . .	5
AnalyseResiduals . . . . .	6
boxplotAndDensityCurveOnHistogram . . . . .	6
boxplotHV . . . . .	7
calc.a . . . . .	8
calc.b . . . . .	9
Calc4GroupNPStats . . . . .	9
calcCliffdConfidenceIntervals . . . . .	11
calcCliffdTestStatistics . . . . .	12
calcEffectSizeConfidenceIntervals . . . . .	13
calcPHatConfidenceIntervals . . . . .	15
calcPHatMATestStatistics . . . . .	16
calculate2GBias . . . . .	17
calculate2GType1Error . . . . .	19
calculate4GBias . . . . .	21
calculate4GType1Error . . . . .	23
calculateBasicStatistics . . . . .	25
calculateCliffd . . . . .	26
calculateGroupSummaryStatistics . . . . .	27
calculateHg . . . . .	29
calculateKendalltaupb . . . . .	30
calculateLargeSampleRandomizedBlockDesignEffectSizes . . . . .	31
calculateLargeSampleRandomizedDesignEffectSizes . . . . .	33
CalculateLevel2ExperimentRData . . . . .	34
calculateMABias . . . . .	36
calculateMAType1Error . . . . .	39
calculateNullIESAccuracy . . . . .	41
calculatePhat . . . . .	43
calculatePopulationStatistics . . . . .	44
CalculateRLevel1 . . . . .	45
calculateSmallSampleSizeAdjustment . . . . .	47
CatchError . . . . .	48
checkIfValidDummyVariable . . . . .	49
Ciolkowski09ESEM.MetaAnalysis.PBRvsCBRorAR . . . . .	49

Cliffd.test . . . . .	50
constructEffectSizes . . . . .	51
ConstructLevel1ExperimentRData . . . . .	53
crossoverResidualAnalysis . . . . .	55
densityCurveOnHistogram . . . . .	56
doLM . . . . .	57
effectSizeCI . . . . .	57
ExtractExperimentData . . . . .	59
ExtractGroupSizeData . . . . .	60
ExtractMAStatistics . . . . .	62
ExtractSummaryStatisticsRandomizedExp . . . . .	64
fmt . . . . .	65
getEffectSizesABBA . . . . .	66
getEffectSizesABBAAIIgnoringPeriodEffect . . . . .	67
getSimulationData . . . . .	68
getTheoreticalEffectSizeVariancesABBA . . . . .	69
KitchenhamEtAl.CorrelationsAmongParticipants.Abrahao13TSE . . . . .	70
KitchenhamEtAl.CorrelationsAmongParticipants.Gravino15JVLC . . . . .	71
KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10 . . . . .	72
KitchenhamEtAl.CorrelationsAmongParticipants.Reggio15SSM . . . . .	74
KitchenhamEtAl.CorrelationsAmongParticipants.Ricca10TSE . . . . .	75
KitchenhamEtAl.CorrelationsAmongParticipants.Ricca14TOSEM . . . . .	76
KitchenhamEtAl.CorrelationsAmongParticipants.Romano18ESEM . . . . .	77
KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14EASE . . . . .	78
KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14JVLC . . . . .	79
KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM . . . . .	80
KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE . . . . .	81
KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello17TOSEM . . . . .	83
KitchenhamEtAl.CorrelationsAmongParticipants.Torchiano17JVLC . . . . .	84
KitchenhamMadeyski.SimulatedCrossoverDataSets . . . . .	85
KitchenhamMadeyskiBrereton.ABBAMetaAnalysisReportedResults . . . . .	86
KitchenhamMadeyskiBrereton.ABBAReportedEffectSizes . . . . .	87
KitchenhamMadeyskiBrereton.DocData . . . . .	88
KitchenhamMadeyskiBrereton.ExpData . . . . .	90
KitchenhamMadeyskiBrereton.MetaAnalysisReportedResults . . . . .	91
KitchenhamMadeyskiBrereton.ReportedEffectSizes . . . . .	92
KitchenhamMadeyskiBudgen16.COCOMO . . . . .	93
KitchenhamMadeyskiBudgen16.DiffInDiffData . . . . .	95
KitchenhamMadeyskiBudgen16.FINNISH . . . . .	96
KitchenhamMadeyskiBudgen16.PolishData . . . . .	97
KitchenhamMadeyskiBudgen16.PolishSubjects . . . . .	98
KitchenhamMadeyskiBudgen16.SubjectData . . . . .	100
LaplaceDist . . . . .	102
Madeyski15EISEJ.OpenProjects . . . . .	103
Madeyski15EISEJ.PropProjects . . . . .	104
Madeyski15EISEJ.StudProjects . . . . .	105
Madeyski15SQJ.NDC . . . . .	106
MadeyskiKitchenham.EUBASdata . . . . .	107

MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR . . . . .	108
MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20190324 . . . . .	110
MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022 . . . . .	111
metaanalyse.Cliffd . . . . .	113
metaanalyse.PHat . . . . .	115
metaanalyse.SmallSampleSizeExperiments . . . . .	117
MetaAnalysisSimulations . . . . .	118
NP2GMetaAnalysisSimulation . . . . .	121
NP4GMetaAnalysisSimulation . . . . .	124
percentageInaccuracyOfLargeSampleVarianceApproximation . . . . .	126
PHat.test . . . . .	127
PHatonesidedTestStatistics . . . . .	128
PHattwosidedTestStatistics . . . . .	129
plotOutcomesForIndividualsInEachSequenceGroup . . . . .	131
PrepareForMetaAnalysisGtoR . . . . .	132
printXTable . . . . .	133
proportionOfSignificantTValuesUsingCorrectAnalysis . . . . .	134
proportionOfSignificantTValuesUsingIncorrectAnalysis . . . . .	135
RandomExperimentSimulations . . . . .	135
RandomizedBlockDesignEffectSizes . . . . .	138
RandomizedBlocksAnalysis . . . . .	139
RandomizedBlocksExperimentSimulations . . . . .	141
RandomizedDesignEffectSizes . . . . .	144
readExcelSheet . . . . .	145
reproduceForestPlotRandomEffects . . . . .	145
reproduceMixedEffectsAnalysisWithEstimatedVarianceAndExperimentalDesignModerator . . . . .	146
reproduceMixedEffectsAnalysisWithExperimentalDesignModerator . . . . .	146
reproduceMixedEffectsForestPlotWithExperimentalDesignModerator . . . . .	147
reproduceSimulationResultsBasedOn500Reps1000Obs . . . . .	147
reproduceTablesOfPaperMetaAnalysisForFamiliesOfExperiments . . . . .	148
reproduceTableWithEffectSizesBasedOnMeanDifferences . . . . .	149
reproduceTableWithPossibleModeratingFactors . . . . .	149
reproduceTableWithSourceDataByCiolkowski . . . . .	150
rSimulations . . . . .	150
searchForIndustryRelevantGitHubProjects . . . . .	152
simulate2GExperimentData . . . . .	153
simulate4GExperimentData . . . . .	155
simulateRandomizedBlockDesignEffectSizes . . . . .	157
simulateRandomizedDesignEffectSizes . . . . .	159
testfunctionParameterChecks . . . . .	161
transformHgtoR . . . . .	162
transformHgtoZr . . . . .	163
transformRtoHg . . . . .	164
transformRtoZr . . . . .	164
transformZrtoHg . . . . .	165
transformZrtoHgapprox . . . . .	166
transformZrtoR . . . . .	166
varStandardizedEffectSize . . . . .	167

---

aggregateIndividualDocumentStatistics  
*aggregateIndividualDocumentStatistics*

---

### Description

This function assumes an ABBA crossover experiment has reported means and variances for each technique in each time period. We calculate the weighted mean and pooled within group variance for the observations arising from the two different sets of materials for a specific technique.

### Usage

```
aggregateIndividualDocumentStatistics(D1.M, D1.SD, D1.N, D2.M, D2.SD, D2.N)
```

### Arguments

D1.M	is a vector of mean values from a set of experiments in a family reporting observations from participants using a specific document in the first time period with either the control or the treatment technique.
D1.SD	is a vector of results from the set of experiment in a family reporting the standard deviations of observations from participants using the same document in the first time period with the same technique.
D1.N	is a vector of the numbers of participants in each experiment in a family, using the same document for participants using either the same technique.
D2.M	is a vector of mean values of observations from participants using the alternative document in the second time period, but using the same technique.
D2.SD	is a vector of the standard deviations of observations from participants using the alternative document in the second time period with the same technique.
D2.N	is a vector of the numbers of participants using the same document in the second time period for participants using the same technique.

### Value

data frame incl. the overall weighted mean and pooled standard deviation

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
aggregateIndividualDocumentStatistics(10, 2, 20, 15, 2, 20)
#      M SD
# 1 12.5  2
```

---

AnalyseResiduals	<i>AnalyseResiduals</i>
------------------	-------------------------

---

**Description**

The function calculates sample statistics based on the residuals from a specified experiment

**Usage**

```
AnalyseResiduals(Residuals, ExperimentName = "ExpName")
```

**Arguments**

Residuals        a vector of residuals  
 ExperimentName a character string identifying the data set

**Value**

A dataframe identifying the ExperimentName and its associated sample parameter: Length, Mean, Median, Variance, Standard deviation, skewness, kurtosis, the outcome of the Shapiro and Anderson-Darling normality test and the number of outliers.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
ExpData=rnorm(30,0,1)
set.seed(123)
AnalyseResiduals(Residuals=ExpData,ExperimentName='ExpName')
# ExperimentName  Mean   Median Variance Skewness Kurtosis ShapiroTest AndersonDarling
#1      ExpName -0.1396192 -0.01943395 0.8424521 -0.1964175 4.559587 0.1608315 0.1316835
# NumOut
# 1
```

---

boxplotAndDensityCurveOnHistogram	<i>boxplotAndDensityCurveOnHistogram</i>
-----------------------------------	--

---

**Description**

Boxplot and density curve overlaid on histogram

**Usage**

```
boxplotAndDensityCurveOnHistogram(df, colName, limLow, limHigh)
```

**Arguments**

df	Data frame with data to be displayed
colName	Name of the selected column in a given data frame
limLow	the limit on the lower side of the displayed range
limHigh	the limit on the higher side of the displayed range

**Value**

A figure being a density curve overlaid on histogram

**Author(s)**

Lech Madeyski

**Examples**

```
library(ggplot2)
library(grid)
library(gridExtra)
boxplotAndDensityCurveOnHistogram(Madeyski15EISEJ.PropProjects, "STUD", 0, 100)
boxplotAndDensityCurveOnHistogram(Madeyski15SQJ.NDC, "simple", 0, 100)
```

---

boxplotHV

*boxplotHV*

---

**Description**

Box plot

**Usage**

```
boxplotHV(df, colName, limLow, limHigh, isHorizontal)
```

**Arguments**

df	Data frame with data to be displayed
colName	Name of the selected column in a given data frame
limLow	the limit on the lower side of the displayed range
limHigh	the limit on the higher side of the displayed range
isHorizontal	Boolean value to control whether the box plot should be horizontal or not (i.e., vertical)

**Value**

A box plot

**Author(s)**

Lech Madeyski

**Examples**

```
boxplotHV(Madeyski15EISEJ.PropProjects, "STUD", 0, 100, TRUE)
boxplotHV(Madeyski15EISEJ.PropProjects, "STUD", 0, 100, FALSE)
boxplotHV(Madeyski15SQJ.NDC, "simple", 0, 100, FALSE)
boxplotHV(Madeyski15SQJ.NDC, "simple", 0, 100, TRUE)
```

---

calc.a

*calc.a*

---

**Description**

This function is a helper function that calculates one element of the standardized mean difference effect size variance based on Hedges and Olkin p128-131.

**Usage**

```
calc.a(f, A)
```

**Arguments**

f	a vector defining the degrees of freedom for the effect sizes
A	a vector defining the constants that relate each StdMD to its related t-variable where $t^2=Ad^2$

**Value**

The value of a.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
reproducer:::calc.a(10,2/10)
# [1] 0.2128649
```

---

calc.b	<i>calc.b</i>
--------	---------------

---

**Description**

This function is a helper function that calculates one element of the standardized mean difference effect size variance based on Hedges and Olkin p128-131.

**Usage**

```
calc.b(f)
```

**Arguments**

f                    a vector defining the degrees of freedom for the effect sizes

**Value**

The value of b.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
reproducer:::calc.b(8)
#0.08649774
```

---

Calc4GroupNPStats	<i>Calc4GroupNPStats</i>
-------------------	--------------------------

---

**Description**

This function does a non-parametric analysis of a randomized blocks experiment assuming 2 blocks and 2 treatment conditions.

**Usage**

```
Calc4GroupNPStats(
  x1,
  x2,
  x3,
  x4,
  sigfig = -1,
  alpha = 0.05,
  alternative = "two.sided"
)
```

**Arguments**

x1	is the data associated with treatment A in one block 1
x2	is the data associated with treatment B in block 1
x3	is the data associated with treatment A in block 2
x4	is the data associated with treatment B in block 2
sigfig	is the number of significant digits in the data. If >0 the data will be appropriately truncated.
alpha	is the significance level for all statistical tests
alternative	The type of statistical test. Valid values are one of c('two.sided', 'greater', 'less')

**Value**

The function returns the Cliff's d and its variance, the probability of superiority, phat, and its variance for the 4 group experiment.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
set.seed(123)
x <- list()
x[[1]] <- rnorm(10, 0, 1)
x[[2]] <- rnorm(10, 0.8, 1)
x[[3]] <- rnorm(10, 0.5, 1)
x[[4]] <- rnorm(10, 1.3, 1)
as.data.frame(
  Calc4GroupNPStats(x[[1]], x[[2]], x[[3]], x[[4]], sigfig = -1, alpha = 0.05)
)
#   N phat   phat.var  phat.df phat.test  phat.pvalue phat.sig phat.ci.upper
# 1 40 0.17 0.004966667 31.00131 -4.682539 5.324252e-05  TRUE  0.3137336
#   phat.ci.lower   d      vard d.sig d.ci.lower d.ci.upper   cor      sqse
# 1  0.02626639 -0.66 0.02060121  TRUE -0.8545073 -0.3031667 -0.3473684 0.01315789
#           ctvar n1 n2 sigCVt
# 1 0.005990797 20 20  TRUE
as.data.frame(
  Calc4GroupNPStats(x[[1]], x[[2]], x[[3]], x[[4]], sigfig = -1, alpha = 0.05,
    alternative = "less")
)
#   N phat   phat.var  phat.df phat.test  phat.pvalue phat.sig phat.ci.upper
# 1 40 0.17 0.004966667 31.00131 -4.682539 2.662126e-05  TRUE  0.2894908
#   phat.ci.lower   d      vard d.sig d.ci.lower d.ci.upper   cor      sqse
# 1  0.02626639 -0.66 0.02060121  TRUE  -1 -0.3677704 -0.3473684 0.01315789
#           ctvar n1 n2 sigCVt
# 0.005990797 20 20  TRUE
as.data.frame(
  Calc4GroupNPStats(x[[2]], x[[1]], x[[4]], x[[3]], sigfig = -1, alpha = 0.05,
    alternative = "greater")
)
```

```

)
# N phat phat.var phat.df phat.test phat.pvalue phat.sig phat.ci.upper
# 1 40 0.83 0.004966667 31.00131 4.682539 2.662126e-05 TRUE 1
# phat.ci.lower d vard d.sig d.ci.lower d.ci.upper cor sqse
# 1 0.7105092 0.66 0.02060121 TRUE 0.3677704 1 0.3473684 0.01315789
# ctvar n1 n2 sigCVt
# 1 0.005990797 20 20 TRUE

#as.data.frame(
#Calc4GroupNPStats(x[[1]],x[[2]],x[[3]],x[[4]],sigfig=-1,alpha=0.00)
#Error in testfunctionParameterChecks(alternative = alternative, alpha = alpha, :
# Invalid alpha parameter, select alpha in range (0.0001,0.2)

```

---

```

calcCliffdConfidenceIntervals
      calcCliffdConfidenceIntervals

```

---

### Description

This functions is a helper function. It assesses the significance one-sided and two-sided statistical of Cliff's d based on its confidence interval. The type of test is determined by the parameter One.Sided.Tests, the direction of one-sided tests is determined by the parameter Positive.MD.

### Usage

```

calcCliffdConfidenceIntervals(
  d.value,
  d.variance,
  d.df,
  alpha = 0.05,
  alternative = "two.sided"
)

```

### Arguments

d.value	This is the value of Cliff's d.
d.variance	This is the estimated variance of Cliff's d
d.df	The degrees of freedom.
alpha	This is the alpha level required for the statistical tests (default 0.05)
alternative	This defines whether a one-sided test or a two-sided (default) test is required. For a one-sided test use parameter values greater' or 'less' to define whether the d-value should be greater or less than zero.

### Value

The function returns a Boolean variable identifying whether the effect size is significant and the confidence interval bounds.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
reproducer:::calcCliffdConfidenceIntervals(d.value=0.5, d.variance=0.04,d.df=18)
# A tibble: 1 x 5
#   d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl>    <dbl>    <dbl>    <dbl> <lgl>
#1     2.5    0.0223    0.0479    0.782 TRUE

reproducer:::calcCliffdConfidenceIntervals(
  d.value=0.5,d.variance=0.04,d.df=18,alternative='greater')
# A tibble: 1 x 5
#   d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl>    <dbl>    <dbl>    <dbl> <lgl>
#1     2.5    0.0112    0.123      1 TRUE

reproducer:::calcCliffdConfidenceIntervals(
  d.value=0.2,d.variance=0.04,d.df=18,alternative='greater')
# A tibble: 1 x 5
#   d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl>    <dbl>    <dbl>    <dbl> <lgl>
#1     1     0.165   -0.133     1 FALSE

reproducer:::calcCliffdConfidenceIntervals(
  d.value=-0.5,d.variance=0.04,d.df=18,alternative='less')
# A tibble: 1 x 5
#   d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl>    <dbl>    <dbl>    <dbl> <lgl>
#1    -2.5    0.0112    -1     -0.123 TRUE
```

---

calcCliffdTestStatistics

*calcCliffdTestStatistics*

---

**Description**

This function is a helper function for meta-analysis of experiments using Cliff's d as an effect size. It returns the  $100 \cdot (1 - \alpha/2)$

**Usage**

```
calcCliffdTestStatistics(
  d.value,
  d.variance,
  d.df = 0,
  alpha = 0.05,
  alternative = "two.sided"
)
```

**Arguments**

d.value	The overall estimate of Cliff's d from a group of effect sizes to be meta-analysed
d.variance	The estimate of the variance of the overall estimate of Cliff's d
d.df	The total degrees of freedom for the set of effect sizes. If d.df>0, the pvalues and significance test use the t-distribution probability values. If d.df=0 (default) the pvalues and significance test use the normal distribution probability values. The confidence intervals are always based on the normal probability values.
alpha	The significance level used to control the significance tests and calculation of confidence limits (default 0.05).
alternative	Specifies the type of significance test and can take the values "two.sided", "less" or "greater" (default "two.sided").

**Value**

d.tvalue	The value of the t-statistic
d.pvalue	The p-value of the t-test if the parameter d.df>0, or the normal probability value if d.df=0
d.ci.lower	The lower $100*(1-\alpha/2)$
d.ci.upper	The upper $100*(1-\alpha/2)$
d.sig	The significance of the statistical test of the d.tvalue return value at the alpha level for one sided tests and $\alpha/2$ for two sided tests as specified by the input parameter alternative

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
aveCliffd=mean(c(0.84,0.2,-0.04,0.44,0.76))
aveCliffdvar=sum(c(0.04,0.18,0.21,0.15,0.06))/25
df=45
calcCliffdTestStatistics(d.value=aveCliffd,d.variance=aveCliffdvar,d.df=df)
# A tibble: 1 x 5
#   d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl>    <dbl>    <dbl>    <dbl> <lgl>
# 1     2.75  0.00855    0.0923    0.692 TRUE
```

---

calcEffectSizeConfidenceIntervals

*calcEffectSizeConfidenceIntervals*

---

**Description**

This function provides single-sided and two-sided confidence interval of an effect size (assuming that the null hypothesis value is zero).

**Usage**

```
calcEffectSizeConfidenceIntervals(
  effectsize,
  effectsize.variance,
  effectsize.df = 0,
  alpha = 0.05,
  alternative = "two.sided",
  UpperValue = Inf,
  LowerValue = -Inf
)
```

**Arguments**

effectsize	The effect size
effectsize.variance	The effect size variance
effectsize.df	The degrees of freedom for confidence intervals based on the t- distribution. If df=0 (default), the confidence interval is based on the normal distribution
alpha	The significance level of the confidence interval (default 0.05).
alternative	This defines whether a one-sided test or a two-sided (default) test is required. For a one-sided test use parameter values greater' or 'less' to define whether the d-value should be greater or less than zero.
UpperValue	The maximum legal value of the effect size (default Inf). Used to ensure that confidence intervals of effect sizes such as correlation coefficients are restricted to sensible values.
LowerValue	The minimum legal value of the effect size (default -Inf). Used to ensure that confidence intervals of effect sizes such as correlation coefficients are restricted to sensible values

**Value**

The value of the test statistic, the p.value of test statistic, the upper and lower confidence interval of the effect size, a logical value specifying whether the effect size is significantly different from zero based on the confidence interval and the lower and upper confidence interval bounds.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
reproducer:::calcEffectSizeConfidenceIntervals(
  effectsize=0.37, effectsize.variance=0.00847, effectsize.df=11.1,
  alpha=0.05, alternative='two.sided', UpperValue=0.5, LowerValue=-0.5)
# A tibble: 1 x 5
#   ES.test ES.pvalue ES.sig ES.ci.lower ES.ci.upper
#   <dbl>   <dbl> <lgl>   <dbl>   <dbl>
#1     4.02   0.00198 TRUE     0.168     0.5
```

---

```
calcPHatConfidenceIntervals
  calcPHatConfidenceIntervals
```

---

## Description

This functions is a helper function. It assesses the significance one-sided and two-sided statistical of the probability of superiority based on its confidence interval. The type of test and the direction of the test is determined by the parameter alternative which takes one of the values 'two.sided', 'greater' or 'less'.

## Usage

```
calcPHatConfidenceIntervals(
  phat,
  phat.variance,
  phat.df,
  alpha = 0.05,
  alternative = "two.sided"
)
```

## Arguments

phat	This is the value of the probability of superiority.
phat.variance	This is the estimated variance of the probability of superiority.
phat.df	The degrees of freedom associated with phat value.
alpha	This is the alpha level required for the statistical tests (default 0.05)
alternative	This defines whether a one-sided test or a two-sided (i.e. default) test is required. For a one-sided test use parameter values 'greater' or 'less' to define whether the phat-value should be greater or less than 0.5.

## Value

The function returns a Boolean variable identifying whether the effect size is significant and the confidence interval bounds.

## Author(s)

Barbara Kitchenham and Lech Madeyski

## Examples

```
reproducer:::calcPHatConfidenceIntervals(.65,0.005,8)
# A tibble: 1 x 5
#   phat.test pvalue phat.sig phat.ci.lower phat.ci.upper
#   <dbl> <dbl> <lgl> <dbl> <dbl>
# 1      2.12 0.0667 FALSE      0.487      0.813
```

```

reproducer:::calcPHatConfidenceIntervals(.65,0.005,8,alternative='greater')
# A tibble: 1 x 5
#   phat.test pvalue phat.sig phat.ci.lower phat.ci.upper
#   <dbl> <dbl> <lgl> <dbl> <dbl>
# 1     2.12 0.0333 TRUE     0.519     1

```

---

calcPHatMATestStatistics

*calcPHatMATestStatistics*

---

## Description

This function is a helper function for meta-analysis of experiments using PHat as an effect size. It returns the  $100*(1-\alpha/2)$

## Usage

```

calcPHatMATestStatistics(
  effectsize,
  effectsize.variance,
  effectsize.df = 0,
  alpha = 0.05,
  alternative = "two.sided"
)

```

## Arguments

effectsize	The overall estimate of the centralized PHat (ie.Phat-0.5) from a group of effect sizes to be meta-analysed
effectsize.variance	The estimate of the variance of the overall estimate of PHat
effectsize.df	The total degrees of freedom for the set of effect sizes. If effectsize.df>0, the confidence intervals, pvalues and significance test use the t-distribution probability values. If effectsize.df=0 (default), the confidence intervals, the pvalues and significance test use the normal distribution probability values.
alpha	The significance level used to control the significance tests and calculation of confidence limits (default 0.05).
alternative	Specifies the type of significance test and can take the values "two.sided" (default), "less" or "greater"

## Value

ES.test The value of the t-statistic

ES.pvalue The p-value of the t-test if the parameter d.df>0, or the normal probability value if d.df=0

ES.sig The significance of the statistical test of the d.tvalue return value at the alpha level for one sided tests and alpha/2 for two sided tests as specified by the input parameter alternative.

ES.ci.lower The lower  $100*(1-\alpha/2)$

ES.ci.upper The upper  $100*(1-\alpha/2)$

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

avePHat=mean(c(0.92,0.6,0.48,0.72,0.88))
avePHatvar=sum(c(0.01,0.04,0.05,0.04,0.01))/25
PHatdf=sum(c(6.63,6.63,5.08,5.61,8))
calcPHatMATestStatistics(effectsize=avePHat-0.5,effectsize.variance=avePHatvar,effectsize.df=PHatdf)
# A tibble: 1 x 5
#   ES.test ES.pvalue ES.sig ES.ci.lower ES.ci.upper
#   <dbl>   <dbl> <lg1>   <dbl>   <dbl>
# 1     2.84  0.00778 TRUE     0.0622  0.378

```

---

calculate2GBias

*calculate2GBias*


---

**Description**

The function simulates two-group experiments and estimates the power, individual estimate error, and the small sample bias obtained from the set of simulated experiments. The set of simulations for a specific mean difference are repeated for three different values of the difference between the treatment and control groups specified by the parameter "diff". The power is estimated as the percentage of experiments for which the mean of the experiment was significantly different from zero. The experiment data may be one of four different type: Normal, Log-normal, Gamma or Laplace. The output is a table of values identifying the observed values of three effect sizes: Cliff's d, PHat and StdMD, estimate error and their related small sample bias and power for each set of simulated experiments. This function supports the production of the values reported in data tables in the paper "Recommendations for Analyzing Small Sample Size Software Engineering Experiments" and its Supplementary Material.

**Usage**

```

calculate2GBias(
  mean = 0,
  sd = 1,
  N,
  reps,
  diff = c(0.2, 0.5, 0.8),
  Expected.StdMD = c(0.2, 0.5, 0.8),
  Expected.PHat = c(0.556, 0.638, 0.714),
  type = "n",
  seed = 223,
  StdAdj = 0
)

```

**Arguments**

mean	This is the mean value of the control and treatment group(s) used in the simulations of each experiment for simulations of a specified sample size and mean difference (default 0).
sd	This is the standard deviation value of the control group(s) and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 1).
N	This specifies the sample size per group that will be used in each set of simulations.
reps	The number of experiments simulated for each mean difference.
diff	This specifies the mean difference between the control and treatment that will be used in each set of simulations. It must always have three values representing small, medium and large differences (default c(0.2, 0.5, 0.8)).
Expected.StdMD	This defines the theoretical value of the average StdMD obtained from the simulations for each mean difference. (default c(0.2, 0.5, 0.8))
Expected.PHat	This defines the expected population value of the average PHat obtained from the simulations for each mean difference (default c(0.556,0.638,0.714)).
type	This specifies the distribution of the data samples that will be simulated. Options are "n" for Normal, "l", for Log-normal, "g" for Gamma, "lap" for Laplace (default "n").
seed	A seed for the simulations (default 123).
StdAdj	Used to introduce variance heterogeneity for Laplace and Normal samples (default 0).

**Value**

**Design.** Specifies the type of experiment, the sample distribution (n,l,g,lap), and whether variance heterogeneity was added (het)

**GrpSize.** Specifies the size of each group in the simulated experiments.

**Diff.** The size of the difference between the control and treatment converted to an ordinal scale (Small, Medium, Large)

**NPBias** The relative difference between the average of the observed values of either Cliff's d or centralised PHat and the population value

**StdMDBias.** The relative difference between the average of the observed values of StdMDBias and the theoretical value

**NPMdMRE** The median of the absolute relative difference between the observed values of either Cliff's d or centralised PHat and the theoretical value for each experiment.

**StdMDMdmRE** The median of the relative difference between the observed values of StdMD and the population value for each experiment.

**ObsPHat.** The average of the PHat values found in the set of simulations.

**ObsCliffd.** The average of the Cliffd values found in the set of simulations.

**ObsStdES.** The average of StdMD values found in the set of simulations.

**PHatPower.** The percentage of the simulations, for a specific mean difference, for which the Phat estimate was significantly different from zero at the 0.05 alpha level based on one-sided tests.

**CliffdPower.** The percentage of the simulations, for a specific mean difference, for which the Cliff's d estimate was significantly different from zero at the 0.05 alpha level based on one-sided tests.

**StdMDPower.** The percentage of the simulations, for a specific mean difference, for which the StdMD estimate was significantly different from zero at the 0.05 alpha level based on one-sided tests.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
# as.data.frame(calculate2GBias(mean=0, sd=1, diff=c(0.2, 0.5, 0.8), Expected.StdMD=c(0.157, 0.392, 0.628),
# Expected.PHat=c(0.544, 0.609, 0.671), N=5, reps=50, type="n", seed=523, StdAdj =0.5 ))
# Results for reps=100 (due to NOTE "Examples with CPU (user + system) or elapsed time > 5s"):
# Design GrpSize Diff      NPBias StdMDBias NPMdMRE StdMDdMRE ObsPHat ObsCliffd ObsSt..
# 1 2G_n_het      5 Small -6.308085e-16 0.07088601 3.272727 3.2700082 0.5440 0.0880 0.168..
# 2 2G_n_het      5 Medium 3.486239e-02 0.09914637 1.385321 1.3502057 0.6128 0.2256 0.430..
# 3 2G_n_het      5 Large 2.222222e-02 0.10446123 0.754386 0.8626523 0.6748 0.3496 0.693..
as.data.frame(calculate2GBias(mean=0, sd=1, diff=c(0.283, 0.707104, 1.131374),
Expected.StdMD=c(0.157, 0.392, 0.628), Expected.PHat=c(0.556, 0.636, 0.705), N=10, reps=20,
type="lap", seed=1423, StdAdj=0.5 ))
#Parameter reps changed due to NOTE "Examples with CPU (user + system) or elapsed time > 5s"
#Results for reps=100:
# Design GrpSize Diff      NPBias StdMDBias NPMdMRE StdMDdMRE ObsPHat ObsCliffd Ob..
#1 2G_lap_het     10 Small -0.11071429 -0.080855612 1.8928571 2.1256888 0.5498 0.0996 0.1..
#2 2G_lap_het     10 Medium -0.07426471 0.003940804 0.6323529 0.8170856 0.6259 0.2518 0.3..
#3 2G_lap_het     10 Large -0.05756098 0.023696619 0.4146341 0.5447941 0.6932 0.3864 0.6..
```

---

calculate2GType1Error *calculate2GType1Error*

---

### Description

The function simulates multiple two-group experiments and estimates the Type1 Error rate obtained from the set of simulated experiments. The Type1 Error is estimated as the percentage of experiments for which the mean the experiment was significantly different from zero at the 0.05 significance level using two-sided tests. The experiment data may be one of four different type: Normal, Log-normal, Gamma or Laplace. The output is a set of values identifying three observed effect size estimates (Cliff's d, PHat and StdMD) and their related type 1 error rates. This function supports the production of the values reported in data tables in the paper "Recommendations for Analyzing Small Sample Size Software Engineering Experiments" and its Supplementary Material.

**Usage**

```
calculate2GType1Error(
  mean = 0,
  sd = 1,
  N = 10,
  reps,
  type = "n",
  seed = 123,
  StdAdj = 0
)
```

**Arguments**

mean	This is the mean value of the control and treatment group(s) used in the simulations of each experiment for simulations of a specified sample size (default 0).
sd	This is the standard deviation value of the control group(s) and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 1).
N	This specifies the sample size per group that will be used in each set of simulations (default 5).
reps	The number of experiments to simulated.
type	This specifies the distribution of the data samples that will be simulated. Options are "n" for Normal, "l", for Log-normal, "g" for Gamma, "lap" for Laplace (default "n").
seed	A seed for the simulations (default 123).
StdAdj	Used to introduce variance heterogeneity for Laplace and Normal samples (default 0).

**Value**

Design. Specifies the type of experiment 2G or 4G, the sample distribution (n,l,g,lap), and whether variance heterogeneity was added (het)

GrpSize. Specifies the size of each group in the individual experiments.

ObsPHat. The average Phat values found in the set of simulations.

ObsCliffd. The average Cliffd values found in the set of simulations.

ObsStdES. The average of StdMD values found in the set of simulations.

PHatType1ER. The proportion of the simulations for which the Phat estimate was significantly different from zero at the nominated alpha level.

CliffdType1ER. The proportion of the simulations for which the Cliff's d estimate was significantly different from zero at the nominated alpha level.

StdMDType1ER. The proportion of the simulations for which the StdMD estimate was significantly different from zero at the nominated 0.05 significance level.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

calculate2GType1Error(mean=1,sd=3,N=10, reps=100, type="g", seed=3256, StdAdj = 0)
# A tibble: 1 x 8
#   Design GrpSize ObsPHat ObsCliffd ObsStdES PHatType1ER CliffdType1ER StdESType1ER
#   <chr>   <chr>     <dbl>     <dbl>     <dbl>     <dbl>         <dbl>         <dbl>
# 1 2G_g    10         0.498     -0.0034  -0.00464    0.02          0.01          0.02

```

---

calculate4GBias	<i>calculate4GBias</i>
-----------------	------------------------

---

**Description**

The function simulates four-group experiments and estimates of the power, individual estimate error and small sample bias obtained from a set of simulated experiments. The function produces three set of simulations obtained using three different values of the mean difference between the treatment and control groups as specified by the parameter "diff". The power is estimated as the percentage of simulated experiments for which the mean of the experiment was significantly different from zero using one-sided tests. The experiment data may be one of four different type: Normal, Log-normal, Gamma or Laplace. The output is a table of values identifying the observed values of three effect sizes: Cliff's d, PHat and StdMD, their related estimate error, small sample bias and power for each set of simulated experiments. This function supports the production of the values reported in data tables in the paper "Recommendations for Analyzing Small Sample Size" and its Supplementary Material.

**Usage**

```

calculate4GBias(
  mean = 0,
  sd = 1,
  N,
  reps,
  diff = c(0.2, 0.5, 0.8),
  Expected.StdMD = c(0.2, 0.5, 0.8),
  Expected.PHat = c(0.556, 0.638, 0.714),
  type = "n",
  seed = 223,
  StdAdj = 0,
  Blockmean = 0
)

```

**Arguments**

mean	This is the mean value of the control group(s) used in the simulations of each experiment for simulations of a specified mean difference (default 0).
sd	This is the standard deviation value of the control group(s) and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 1).
N	This specifies the sample size per group that will be used in each set of simulations.
reps	The number of families simulated for each sample size.
diff	This specifies the difference between the control and treatment that will be used in each set of simulations. It must always have three values representing small, medium and large mean differences (default c(0.2, 0.5, 0.8)).
Expected.StdMD	This defines the expected value of the overall average StdMD for each mean difference (default c(0.2, 0.5, 0.8)).
Expected.PHat	This defines the expected population value of the overall average PHat for each mean difference (default c(0.556,0.638,0.714)).
type	This specifies the distribution of the data samples that will be simulated. Options are "n" for Normal, "l", for Log-normal, "g" for Gamma, "lap" for LaPlace (default "n").
seed	A seed for the simulations (default 123).
StdAdj	Used to introduce variance heterogeneity for Laplace and Normal samples (default 0).
Blockmean	Specifies the value of the block effect (default 0).

**Value**

Design. Specifies the type of experiment 2G, the sample distribution (n,l,g,lap), and whether variance heterogeneity was added (het)

BEIncluded. Specifies whether or not a block effect was introduced.

GrpSize. Specifies the size of each group in the individual experiments.

Diff. The size of the difference between the control and treatment converted to an ordinal scale (Small, Medium, Large)

NPBias The relative difference between the average of the observed values of either Cliff's d or centralised PHat and the population value

StdMDBias. The relative difference between the average of the observed values of StdMDBias and the theoretical value

NPMdMRE The median of the absolute relative difference between the observed values of either Cliff's d or centralised PHat and the theoretical value for each experiment.

StdMDMdmRE The median of the relative difference between the observed values of StdMD and the population value for each experiment.

ObsPHat. The average PHat value found for each simulation.

ObsCliffd. The average Cliffd value found for each simulation.

ObsStdES. The average of StdMD calculated for each simulation.

PHatPower. The proportion of the simulations, for a given mean difference, for which the Phat estimate was significantly different from zero at the 0.05 alpha level based on one-sided tests.

CliffdPower. The proportion of the simulations, for a given mean difference, for which the Cliff's d estimate was significantly different from zero at the 0.05 alpha level based on one-sided tests.

StdMDPower. The proportion of the simulations, for a given mean difference, for which the StdMD estimate was significantly different from zero at the 0.05 alpha level based on one-sided tests.

## Author(s)

Barbara Kitchenham and Lech Madeyski

## Examples

```
#as.data.frame(calculate4GBias(mean=0,sd=1,diff=c(0.266,0.72375,1.43633),
# Expected.StdMD=c(0.2,0.5,0.8),Expected.PHat=c(0.575,0.696,0.845),N=10, reps=200, type="l",
# seed=17+1823,StdAdj=0,Blockmean=0))
# Design BEIncluded GrpSize Diff NPBias StdMDBias NPMdMRE StdMDdMRE ObsPHat ObsCliffd.
# 1 4G_l No 10 Small -0.05933333 0.1247408 0.8666667 1.2047848 0.570550 0.1411..
# 2 4G_l No 10 Medium -0.01760204 0.1565643 0.3112245 0.4426859 0.692550 0.3851..
# 3 4G_l No 10 Large -0.00326087 0.2273638 0.1594203 0.2924361 0.843875 0.6877..
as.data.frame(calculate4GBias(mean=1,sd=3,diff=c(0.1225,0.3415,0.6224),
Expected.StdMD=c(-0.208,-0.52,-0.833),Expected.PHat=c(0.444,0.360,0.277),N=20, reps=30, type="g",
seed=17+977,StdAdj=0,Blockmean=0.5))
# Results for reps=200:
# Design BEIncluded GrpSize Diff NPBias StdMDBias NPMdMRE StdMDdMRE ObsPHat ObsCli..
#1 4G_g Yes 20 Small 0.04274554 0.02242895 0.8370536 0.7960052 0.4416062 -0.1167..
#2 4G_g Yes 20 Medium 0.01959821 0.01585829 0.3348214 0.3210435 0.3572562 -0.2854..
#3 4G_g Yes 20 Large 0.01303251 0.01515967 0.1905830 0.1871956 0.2740938 -0.4518..
```

---

calculate4GType1Error *calculate4GType1Error*

---

## Description

The function simulates multiple four-group experiments and estimates the Type1 Error rate obtained from the set of simulated experiments. The Type1 Error is estimated as the percentage of experiments for which the mean the experiment was significantly different from zero at the 0.05 significance level using two-sided tests. The experiment data may be one of four different type: Normal, Log-normal, Gamma or Laplace. The output is a set of values identifying three observed effect size estimates (Cliff's d, PHat and StdMD) and their related type 1 error rates. This function supports the production of the values reported in data tables in the paper "Recommendations for Analyzing Small Sample Size Software Engineering Experiments" and its Supplementary Material.

**Usage**

```
calculate4GType1Error(
  mean = 0,
  sd = 1,
  N = 10,
  reps = 10,
  type = "n",
  seed = 123,
  StdAdj = 0,
  Blockmean = 0
)
```

**Arguments**

mean	This is the mean value of the control and treatment group(s) used in the simulations (default 0).
sd	This is the standard deviation value of the control group(s) and treatment group(s) used in the simulations (default 1).
N	This specifies the sample size per group that will be used in each simulation (default 5).
reps	The number of experiments to simulated.
type	This specifies the distribution of the data samples that will be simulated. Options are "n" for Normal, "l", for Log-normal, "g" for Gamma, "lap" for LaPlace (default "n").
seed	A seed for the simulations (default 123).
StdAdj	Used to introduce variance heterogeneity for Laplace and Normal samples (default 0).
Blockmean	Used to specify the block effect (default 0).

**Value**

Design. Specifies the type of experiment 2G or 4G, the sample distribution (n,l,g,lap), and whether variance heterogeneity was added (het)

GrpSize. Specifies the size of each group in the simulations.

BEIncluded. Specifies whether or not a block effect was introduced.

ObsPHat. The average of the average Phat values found in the set of simulations.

ObsCliffd. The average of the average Cliffd values found in the set of simulations.

ObsStdES. The average of StdMD values found in the set of simulations.

PHatType1ER. The percentage of the simulations for which the Phat estimate was significantly different from zero at the 0.05 alpha level.

CliffdType1ER. The percentage of the simulations for which the overall Cliff's d estimate was significantly different from zero at the 0.05 alpha level.

StdMDType1ER. The percentage of the simulations for which the overall StdMD estimate was significantly different from zero at the 0.05 significance level.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
as.data.frame(calculate4GType1Error(mean=0, sd=1, N=40, reps=100, type="n", seed=17+1056, StdAdj = 0.5,
  Blockmean=0.5))
# Results for reps=300
# Design GrpSize BEIncluded ObsPHat ObsCliffd ObsStdES PHatType1ER CliffdType1ER StdES..
#1 4G_n_het 40 Yes 0.5034729 0.006945833 0.01316457 0.03 0.02333333 0.046..

#as.data.frame(calculate4GType1Error(mean=0, sd=1, N=40, reps=300, type="lap", seed=17+2056,
# StdAdj = 0.5, Blockmean=0.5))
# Design GrpSize BEIncluded ObsPHat ObsCliffd ObsStdES PHatType1ER CliffdType1ER Std..
#1 4G_lap_het 40 Yes 0.4992708 -0.001458333 0.0014446 0.04333333 0.04 0.06
```

---

calculateBasicStatistics

*calculateBasicStatistics*

---

**Description**

This function calculates the following statistics for a set of data: length, mean, median, variance, standard error of the mean, and confidence interval bounds. The input data must be a vector of 2 or more numerical values.

**Usage**

```
calculateBasicStatistics(x, alpha = 0.05)
```

**Arguments**

x	The data to be summarized
alpha	The probability level to be used when constructing the confidence interval bounds.

**Value**

A dataframe comprising the length, mean, variance, standard error and confidence limit bounds of the input data x.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

ShortExperimentNames <- c("E1", "E2", "E3", "E4")
FullExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
Metrics <- c("Comprehension", "Modification")
Groups <- c("A", "B", "C", "D")
Type <- c(rep("4G", 4))
StudyID <- "S2"
Control <- "SC"
ReshapedData <- ExtractExperimentData(
  KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM,
  ExperimentNames = FullExperimentNames, idvar = "ParticipantID",
  timevar = "Period", ConvertToWide = TRUE
)
NewTable <- ConstructLevel1ExperimentRData(
  ReshapedData, StudyID, ShortExperimentNames, Groups,
  Metrics, Type, Control
)
calculateBasicStatistics(NewTable$r)
#   N   Mean Median Variance      SE LowerBound UpperBound
# 1 32 0.06175 0.1688   0.2482 0.08808   -0.1109   0.2344

```

---

 calculateCliffd

*calculateCliffd*


---

**Description**

This function implements finds Cliff's d and its confidence intervals. The null hypothesis is that for two independent group,  $P(X < Y) = P(X > Y)$ . The function reports a 1-alpha confidence interval for  $P(X > Y) - P(X < Y)$ . The algorithm computes a confidence interval for Cliff's d using the method in Cliff, 1996, p. 140, eq 5.12. The function is based on code produce by Rand Wilcox but has been amended. The plotting function has been removed and the dependency on Wilcox's binomci function has been removed. Construction of confidence intervals if values in one group are all larger than values in the other group has been amended to use the smallest non-zero variance method. Upper and lower confidence interval bounds cannot assume invalid values, i.e. values  $< -1$  or  $> 1$ .

**Usage**

```
calculateCliffd(x, y, alpha = 0.05, sigfig = -1)
```

**Arguments**

x	is a vector of values from group 1
y	is a vector of values from group 2
alpha	is the Type 1 error level for statistical tests
sigfig	is the number of significant digit. If sigfig>0 the data in x and y is truncated to the specified value.

**Value**

list including the value of Cliffs d its consistent variance and confidence intervals and the equivalent probability of superiority value and its confidence intervals.

**Author(s)**

Rand Wilcox, amendments Barbara Kitchenham and Lech Madeyski

**Examples**

```
x=c(1.2,3,2.2,4,2.5,3)
y=c(3,4.2,4,6,7,5.9)
calculateCliffd(x,y)
# $n1
# [1] 6
# $n2
# [1] 6
# $d
# [1] -0.8611111
# $sqse.d
# [1] 0.02017931
# $phat
# [1] 0.06944444
z=c(1,2,3,4)
y=c(5,6,7,8)
calculateCliffd(z,y)
# $n1
# [1] 4
# $n2
# [1] 4
# $d
# [1] -1
# $sqse.d
# [1] 0.009765625
# $phat
# [1] 0
```

---

calculateGroupSummaryStatistics

*calculateGroupSummaryStatistics*

---

**Description**

This function calculates the following statistics data within groups: length, mean, median, variance, standard error of the mean, and confidence interval bounds.

**Usage**

```
calculateGroupSummaryStatistics(x, Group)
```

**Arguments**

x	The data to be summarized. This must be a vector of 2 or more numerical values
Group	The categorical data data defining the groups. This must vector of the same length as x containing factors specifying the data groups

**Value**

A dataframe comprising the number, mean, variance, standard error and confidence limit bounds of the data in each category

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

ShortExperimentNames <- c("E1", "E2", "E3", "E4")
FullExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
Metrics <- c("Comprehension", "Modification")
Groups <- c("A", "B", "C", "D")
Type <- c(rep("4G", 4))
StudyID <- "S2"
Control <- "SC"
ReshapedData <- ExtractExperimentData(
  KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM,
  ExperimentNames = FullExperimentNames, idvar = "ParticipantID", timevar = "Period",
  ConvertToWide = TRUE
)
NewTable <- ConstructLevel1ExperimentRData(
  ReshapedData, StudyID,
  ShortExperimentNames, Groups, Metrics, Type, Control
)
SeqGroupLev <- NULL
N.NT <- length(NewTable$r)
for (i in 1:N.NT) {
  if (NewTable$n[i] <= 8) SeqGroupLev[i] <- as.character(NewTable$n[i])
  if (NewTable$n[i] > 8) SeqGroupLev[i] <- as.character(9)
}
calculateGroupSummaryStatistics(NewTable$r, Group = SeqGroupLev)
#   N   Mean  Median Variance StDev  SE
#  1  4 -0.0833 -0.1699  0.2314 0.4810 0.2405
#  2 12  0.3658  0.4477  0.2109 0.4592 0.1326
#  3 16 -0.1300 -0.2214  0.1933 0.4397 0.1099

```

---

`calculateHg`*calculateHg*

---

**Description**

This function calculates Hedges g and Hedges g adjusted given the basic experimental statistics - the mean values for participants, number of observations (participants), and standard deviation in both the control group and the treatment group. . Hence, the function assumes the data is held as summary statistics including the control group mean, standard deviation and sample size and equivalent values for treatment group

**Usage**

```
calculateHg(Mc, Mt, Nc, Nt, SDc, SDt)
```

**Arguments**

Mc	is a vector containing the mean value of the control group for each experiment.
Mt	is a vector containing the mean value of the treatment group for each experiment.
Nc	is a vector containing the the number of observations (participants) in the control group for each experiment.
Nt	is a vector of the number of observations (participants) in the treatment group for each experiment.
SDc	is a vector of the standard deviations of the control group for each experiment.
SDt	is a vector of the standard deviations of the the treatment group for each experiment.

**Value**

data frame composed of Hedges' g and Hedges' g adjusted effect sizes

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
calculateHg(10, 15, 20, 20, 2, 2)
#   Hg   HgAdjusted
# 1  2.5  2.450276
```

---

```
calculateKendalltaupb @title calculateKendalltaupb @description Computes point bi-serial
version of Kendall's tau plus a 1-alpha confidence interval using the
method recommended by Long and Cliff (1997). The algorithm is
based on Wilcox's code but was extended to return the consistent vari-
ance and the confidence intervals based on the t-distribution. Also
added a Diagnostic parameter to output internal calculations.
```

---

## Description

@title calculateKendalltaupb @description Computes point bi-serial version of Kendall's tau plus a 1-alpha confidence interval using the method recommended by Long and Cliff (1997). The algorithm is based on Wilcox's code but was extended to return the consistent variance and the confidence intervals based on the t-distribution. Also added a Diagnostic parameter to output internal calculations.

## Usage

```
calculateKendalltaupb(x, y = NULL, alpha = 0.05, alternative = "two.sided")
```

## Arguments

x	either a matrix with two columns containing two correlated variables or a vector of variables
y	if y=NULL, assume x is a matrix with two columns, otherwise y is a vector of variables with x[i] and y[i] being from the same experimental unit
alpha	the Type 1 error level used for statistical tests (default 0.05)
alternative	The type of statistical test. Valid values are one of c('two.sided', 'greater', 'less')

## Value

list containing the estimate of Kendall's tau, the consistent variance of tau and its confidence intervals based on the t-test (recommended by Long and Cliff)

## Author(s)

Rand Wilcox, Barbara Kitchenham and Lech Madeyski

## Examples

```
set.seed(123)
a <- c(1.2, 3, 1.8, 2, 2, 0.5, 0.5, 1, 3, 1)
b <- c(1, 1, 1, 1, 1, 0, 0, 0, 0, 0)
calculateKendalltaupb(a,b,alpha=.05)
#$cor
#[1] 0.3555556
#$cit
```

```

#[1] -0.1240567  0.5555556
#n
#[1] 10
#df
#[1] 7
#$consistentvar
#[1] 0.04113925
#$sig
#[1] FALSE
set.seed(234)
a2=c(rnorm(10,0,1),rnorm(10,0.5,1))
b2=c(rep(0,10),rep(1,10))
calculateKendalltau(a2,b2,alpha=.05,alternative='greater')
#$cor
#[1] 0.2842105
#$cit
#[1] 0.06517342  0.52631579
#n
#[1] 20
#df
#[1] 17
#consistentvar
#[1] 0.01585379
#$sig
#[1] TRUE
calculateKendalltau(a2,b2,alpha=.05,alternative='less')
#$cor
#[1] 0.2842105
#$cit
#[1] -0.5263158  0.5032476
#n
#[1] 20
#df
#[1] 17
#$consistentvar
#[1] 0.01585379
#$sig
#[1] FALSE

```

---

calculateLargeSampleRandomizedBlockDesignEffectSizes

*calculateLargeSampleRandomizedBlockDesignEffectSizes*

---

## Description

The function uses a simulates a large experiment to estimate the asymptotic values of the probability of superiority, Cliff's d and the standardized mean difference data for a four group randomized blocks experiment for four different distributions: Normal (i.e. type='n'), log-normal (i.e. type='l'), gama (i.e. type='g') and Laplace (i.e., type='lap').

**Usage**

```
calculateLargeSampleRandomizedBlockDesignEffectSizes(
  meanC = 0,
  sdC = 1,
  diff,
  N = 5e+06,
  type = "n",
  Blockmean = 0,
  StdAdj = 0
)
```

**Arguments**

meanC	to act as the mean of the distribution (default 0) used to generate the control group data (note for the gamma distribution this is the rate parameter and must not be zero)
sdC	the variance/spread of the distribution (default 1) used to generate the control group data.
diff	a value added to meanC to generate the treatment group data (default 0).
N	the size of each group (default 5000000)
type	the distribution of the data to be generated. One of: 'n' for normal (default), 'l' for log-normal, 'g' for gamma, and 'lap' for Laplace.
Blockmean	a value that can be added one of the blocks to represent a fixed block effect (default 0).
StdAdj	a value that can be added to sdC to introduce heterogeneity into the treatment group (default 0).

**Value**

A tibble identifying the sample statistics and the values of the probability of superiority, Cliff's d and StdMD (labelled StdES)

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
set.seed=400
calculateLargeSampleRandomizedBlockDesignEffectSizes(
  meanC=0, sdC=1, diff=.5, N=100000, type='n',Blockmean=0.5,StdAdj = 0)
# MeanC SdC MeanT SdT BE Phat Cliffd UES Var StdES
# <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1 0 1 0.5 1 0.5 0.638 0.277 0.501 0.998 0.502
```

---

```
calculateLargeSampleRandomizedDesignEffectSizes
      calculateLargeSampleRandomizedDesignEffectSizes
```

---

### Description

The function simulates a large experiment to estimate the asymptotic values of the probability of superiority, Cliff's  $d$  and the standardized mean difference data for a two group randomized experiment for four different distributions: Normal (i.e. `type="n"`), log-normal (i.e. `type="l"`), gama (i.e. `type="g"`) and Laplace (i.e., `type="lap"`).

### Usage

```
calculateLargeSampleRandomizedDesignEffectSizes(
  meanC = 0,
  sdC = 1,
  diff = 0,
  N = 5e+06,
  type = "n",
  StdAdj = 0,
  reporttrans = "No"
)
```

### Arguments

<code>meanC</code>	to act as the mean of the distribution used to generate the control group data (default 0) (note for the gamma distribution this is the rate parameter and must not be zero)
<code>sdC</code>	the variance/spread of the distribution used to generate the control group data (default 1).
<code>diff</code>	a value added to <code>meanC</code> to generate the treatment group data (default 0).
<code>N</code>	the size of each group (default 5000000)
<code>type</code>	the distribution of the data to be generated (default "n").
<code>StdAdj</code>	a value that can be added to <code>sdC</code> to introduce heterogeneity into the treatment group (default 0).
<code>reporttrans</code>	If set to "Yes" AND <code>type="l"</code> the algorithm returns the values obtained by analysing applying the logarithmic transformation to the simulated data (default "No").

### Value

A tibble identifying the sample statistics and the values of the probability of superiority, Cliff's  $d$  and StdMD (labelled StdES)

### Author(s)

Barbara Kitchenham and Lech Madeyski

**Examples**

```

set.seed=400
calculateLargeSampleRandomizedDesignEffectSizes(meanC=0, sdC=1, diff=.5,
N=10000, type="n",StdAdj = 0) #N=100000, type="n",StdAdj = 0)
# A tibble: 1 x 9
#   MeanC  SdC MeanT  SdT  Phat Cliffd  UES  Var StdES
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#1  0.00642  1.00 0.519 0.995 0.642  0.284 0.513 0.996 0.514
#1    0    1  0.5    1 0.637  0.275 0.499  1.01 0.497
as.data.frame(calculateLargeSampleRandomizedDesignEffectSizes(meanC=0, sdC=1, diff=0.707104,
N=100000, type="1",StdAdj = 0,reporttrans="Yes"))
#N=100000, type="1",StdAdj = 0,reporttrans="Yes"))
#   MeanC  StdC  MeanT  StdT  Phat  Cliffd  UES  Var
#1  1.647446  2.219114  3.33124  4.404537  0.6926779  0.3853558  1.683795  12.1622
#   StdES  MeanCTrans  MeanTTrans  StdCTrans  StdTTrans  PhatTrans  CliffdTrans
#1  0.4828175 -0.004298487  0.7066049  1.001199  0.9963736  0.6926779  0.3853558
#   UESTrans  VarTrans  StdESTrans
#1  0.7109034  0.99758  0.7117651

```

---

CalculateLevel2ExperimentRData

*CalculateLevel2ExperimentRData*

---

**Description**

This function analyses data on r values obtained in the format obtained from the ConstructLevel1ExperimentRData function and finds the r-value for each metric for each experiment.

**Usage**

```

CalculateLevel2ExperimentRData(
  Level1Data,
  Groups,
  StudyID,
  ExperimentNames,
  Metrics,
  Type
)

```

**Arguments**

Level1Data	a tibble in the format produced by the ConstructLevel1ExperimentRData function which has r-values for each sequence group in a crossover experiment
Groups	This is a list that defines the sequence group labels used in the dataset.
StudyID	This holds an identifier used to identify the origin of the experimental data in the output from this function.

ExperimentNames	This a list of identifiers used to define each experiment in the output from this function.
Metrics	This is a list of of character strings identifying each outcome metric reported in each of the experiments in the set of replicated experiments.
Type	this is a list of character strings specifying for each experiment whether the experiment is a two sequence group '2G' or four sequence group '4G' experiment. return RExp.Table This is a table containing the pooled data variance and the pooled difference variance for the experiment and the value r for the experiment for each metric

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

ShortExperimentNames <- c("E1", "E2", "E3", "E4")
FullExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
Metrics <- c("Comprehension", "Modification")
Groups <- c("A", "B", "C", "D")
Type <- c(rep("4G", 4))
StudyID <- "S2"
Control <- "SC"
# Obtain experimental data from each file and put in wide format
ReshapedData <- ExtractExperimentData(
  KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM,
  ExperimentNames = FullExperimentNames, idvar = "ParticipantID",
  timevar = "Period", ConvertToWide = TRUE
)
Lev1Data <- ConstructLevel1ExperimentRData(
  ReshapedData, StudyID, ShortExperimentNames, Groups,
  Metrics, Type, Control
)
CalculateLevel2ExperimentRData(Lev1Data,
  Groups = Groups, StudyID = StudyID,
  ExperimentNames = ShortExperimentNames, Metrics = Metrics, Type = Type
)
# A tibble: 8 x 10
#   StudyID ExpID   N Metric      PooledVar1 PooledVar2 VarProp PooledVar PooledDiffVar   r.Exp
#   <chr>   <chr> <int> <chr>          <dbl>      <dbl> <dbl>    <dbl>      <dbl> <dbl>
# 1 S2     S2E1    24 Comprehension  0.0148     0.0212  0.412   0.0180     0.0248  0.311
# 3 S2     S2E2    22 Comprehension  0.0487     0.0224  0.684   0.0356     0.0534  0.250
# 4 S2     S2E2    22 Modification  0.0445     0.0266  0.626   0.0356     0.0628  0.117
# 5 S2     S2E3    22 Comprehension  0.0353     0.0402  0.467   0.0377     0.105   -0.391
# 6 S2     S2E3    22 Modification  0.0433     0.0414  0.511   0.0424     0.0997  -0.176
# 7 S2     S2E4    18 Comprehension  0.0439     0.0237  0.649   0.0338     0.0355  0.475
# 8 S2     S2E4    18 Modification  0.0322     0.0592  0.353   0.0457     0.0894  0.0222

```

---

 calculateMABias

*calculateMABias*


---

### Description

The function simulates multiple five group families of either two-group or four-group experiments and estimates the power, individual estimate error, and the small sample bias obtained after synthesizing the analysis results obtained from the experiments in each family. The power is estimated as the percentage of families for which the overall mean of the five experiments was significantly different from zero. The experiment data may be one of four different type: Normal, Log-normal, Gamma or Laplace. The simulations can be repeated for different mean differences between the control mean and treatment mean depending on the parameter diff. The output is a table of values identifying the observed values of three effect sizes: Cliff's d, PHat and StdMD, estimate error and their related small sample bias and power for each set of simulated families. The synthesis method for all the effect sizes is based on calculating the overall mean and variance for experiments in each family and then using those values to calculate the overall effect size and its variance. This function supports the production of the values reported in data tables in the paper "Recommendations for Analyzing Small Sample Size Software Engineering Experiments" and its Supplementary Material.

### Usage

```
calculateMABias(
  mean = 0,
  sd = 1,
  N,
  reps,
  diff = c(0.2, 0.5, 0.8),
  Experiments = 5,
  Expected.StdMD = c(0.2, 0.5, 0.8),
  Expected.PHat = c(0.556, 0.638, 0.714),
  type = "n",
  FourG = FALSE,
  seed = 223,
  StdAdj = 0,
  Blockmean = 0,
  StdExp = 0,
  MAMethod = "PM",
  alpha = 0.05
)
```

### Arguments

mean	This is the mean value of the control and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 0).
------	--

sd	This is the standard deviation value of the control group(s) and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 1).
N	This specifies the sample size per group that will be used in each set of simulations.
reps	The number of families simulated for each sample size.
diff	This specifies the difference between the control and treatment that will be used in each set of simulations. It must always have three values representing small, medium and large values (default c(0.2, 0.5, 0.8)).
Experiments	The number of experiments in each family (default 5).
Expected.StdMD	This defines the expected value of the overall average StdMD for each mean difference (default c(0.2,0.5,0.8)).
Expected.PHat	This defines the expected population value of the overall average Phat for each mean difference (default c(0.556,0.638,0.714)).
type	This specifies the distribution of the data samples that will be simulated. Options are "n" for Normal, "l", for Log-normal, "g" for Gamma, "lap" for LaPlace (default "n").
FourG	If FourG is FALSE (default) the individual experiments in each family will be two-group experiments, otherwise the individual experiments will be four-group families.
seed	A seed for the simulations (default 123).
StdAdj	Used to introduce variance heterogeneity for Laplace and Normal samples (default 0).
Blockmean	Used to set a fixed block effect for four-group experiments (default 0).
StdExp	Used to introduce heterogeneity among families of experiments (default 0).
MAMethod	Not used (default "PM").
alpha	The significance level for statistical tests (default 0.05).

### Value

Design. Specifies the type of experiment 2G or 4G, the sample distribution (n,l,g,lap), and whether variance heterogeneity was added (het)

BEIncluded. Specifies whether or not a block effect was introduced. Always set to "No" for two-group experiments.

GrpSize. Specifies the size of each group in the individual experiments.

Diff. The size of the difference between the control and treatment converted to an ordinal scale (Small, Medium, Large)

NPBias The relative difference between the average of the observed values of either Cliff's d or centralised PHat and the population value

StdMDBias. The relative difference between the average of the observed values of StdMDBias and the theoretical value

NPMdMRE The median of the absolute relative difference between the observed values of either Cliff's d or centralised PHat and the theoretical value for each experiment.

StdMDMdMRE The median of the absolute relative difference between the observed values of StdMD and the population value for each experiment.

ObsPHat. The average of the average Phat value found for each family in the set of simulations.

ObsCliffd. The average of the average Cliffd value found for each family in the set of simulations.

ObsStdES. The average of StdMD calculated for each family in the set of simulations.

PHatPower. The percentage of the simulations, for a specific mean difference, for which the overall PHat estimate was significantly different from zero at the nominated alpha level using one-sided tests.

CliffdPower. The percentage of the simulations, for a specific mean difference, for which the overall Cliff's d estimate was significantly different from zero at the nominated alpha level using one-sided tests.

StdMDPower. The percentage of the simulations, for a specific mean difference, for which the overall StdMD estimate was significantly different from zero at the nominated alpha level using one-sided tests.

## Author(s)

Barbara Kitchenham and Lech Madeyski

## Examples

```
# as.data.frame(calculateMABias(mean=0,sd=1,N=10,diff=c(0.2,0.5,0.8), Experiments=5, reps=10,
# Expected.StdMD=c(0.2,0.5,0.8), Expected.PHat=c(0.556,0.638,0.714), type="n", FourG=FALSE,
# seed= 123, StdAdj = 0, Blockmean=0, StdExp=0))
# Design Blockmean GrpSize Diff NPBias StdMDBias NPMdMRE StdMDMdMRE ObsPHat ObsCliffd..
#1 2G_n No 10 Small 0.09285714 0.02606704 0.8928571 1.0741432 0.5612 0.1224..
#2 2G_n No 10 Medium 0.03768116 0.01740262 0.2391304 0.4171896 0.6432 0.2864..
#3 2G_n No 10 Large 0.03738318 0.01523651 0.2009346 0.2490287 0.7220 0.4440..
# PHatPower CliffdPower StdESPower
#1 0.2 0.2 0.3
#2 0.7 0.7 0.7
#3 1.0 1.0 1.0
as.data.frame(calculateMABias(mean=0,sd=1,N=10,diff=c(0.2,0.5,0.8), Experiments=5, reps=4,
Expected.StdMD=c(0.2,0.5,0.8), Expected.PHat=c(0.556,0.638,0.714), type="n", FourG=TRUE,
seed= 123, StdAdj = 0.5, Blockmean=0.5, StdExp=0))
#Results for reps=10
# Design Blockmean GrpSize Diff NPBias StdMDBias NPMdMRE StdMDMdMRE ObsPHat ObsClif..
#1 4G_n_het Yes 10 Small -0.1321429 -0.1372277 0.6696429 0.4698935 0.5486 0.0972..
#2 4G_n_het Yes 10 Medium -0.1869565 -0.1882479 0.2318841 0.1472392 0.6122 0.2244..
#3 4G_n_het Yes 10 Large -0.1864486 -0.2010029 0.1612150 0.1531253 0.6741 0.3482..
# PHatPower CliffdPower StdESPower
#1 0.4 0.4 0.4
#2 0.9 0.9 0.8
#3 1.0 1.0 1.0
```

---

`calculateMAType1Error calculateMAType1Error`

---

### Description

The function simulates multiple five group families of either two-group or four-group experiments and estimates the Type1 Error rate obtained after synthesizing the analysis results obtained from the experiments in each family. The Type1 Error is estimated as the percentage of families for which the overall mean of the five experiments was significantly different from zero. The experiment data may be one of four different type: Normal, Log-normal, Gamma or Laplace. The simulations can be repeated for different sample sizes depending on the parameter N. The output is a table of values identifying three observed effect size estimates (Cliff's d, PHat and StdMD) and their related type 1 error rates for each set of simulated families. The synthesis method for all three effect sizes is based on calculating the overall mean and variance for the family of experiments, and then using those values to calculate the effect size variance and its variance. This function supports the production of the values reported in data tables in the paper "Recommendations for Analyzing Small Sample Size Software Engineering Experiments" and its Supplementary Material.

### Usage

```
calculateMAType1Error(  
  mean = 0,  
  sd = 1,  
  N = c(5, 10, 15, 20, 30, 40),  
  reps,  
  type = "n",  
  seed = 123,  
  Experiments = 5,  
  FourG = FALSE,  
  StdAdj = 0,  
  Blockmean = 0,  
  BlockStdAdj = 0,  
  StdExp = 0,  
  MAMethod = "PM",  
  alpha = 0.05  
)
```

### Arguments

mean	This is the mean value of the control and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 0).
sd	This is the standard deviation value of the control group(s) and treatment group(s) used in the simulations of each experiment of each family for simulations of a specified sample size (default 1).
N	This specifies the sample sizes per group that will be used in each set of simulations (default c(5,10,15,20,30,40)).

reps	The number of families simulated for each sample size.
type	This specifies the distribution of the data samples that will be simulated. Options are "n" for Normal, "l", for Log-normal, "g" for Gamma, "lap" for Laplace (default "n").
seed	A seed for the simulations (default 123).
Experiments	The number of experiments in each family (default 5).
FourG	If FourG is FALSE the individual experiments in each family will be two-group experiments, otherwise the individual experiments will be four-group families (default FALSE).
StdAdj	Used to introduce variance heterogeneity for Laplace and Normal samples (default 0).
Blockmean	Used to set a fixed block effect for four-group experiments (default 0).
BlockStdAdj	Not used (default 0).
StdExp	Used to introduce heterogeneity among families of experiments (default 0).
MAMethod	Not used (default "PM").
alpha	The significance level for statistical tests (default 0.05).

### Value

Design. Specifies the type of experiment 2G or 4G, the sample distribution (n,l,g,lap), and whether variance heterogeneity was added (het)

BEIncluded. Specifies whether or not a block effect was introduced. Always set to "No" for two-group experiments.

GrpSize. Specifies the size of each group in the individual experiments.

ObsPHat. The average of the average Phat value found for each family in the set of simulations.

ObsCliffd. The average of the average Cliffd value found for each family in the set of simulations.

ObsStdES. The average of StdMD calculated for each family in the set of simulations.

PHatType1ER. The percentage of the simulations, for a specific group size, for which the overall Phat estimate was significantly different from zero at the nominated alpha level.

CliffdType1ER. The percentage of the simulations, for a specific group size, for which the overall Cliff's d estimate was significantly different from zero at the nominated alpha level.

StdMDType1ER. The percentage of the simulations, for a specific group size, for which the overall StdMD estimate was significantly different from zero at the nominated alpha level.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
# as.data.frame(calculateMAType1Error(mean=0, sd=1, N=c(5, 10), reps=10, type="n", Experiments=5,
# FourG=FALSE, StdAdj=0, Blockmean=0, seed=123))
# Design BEIncluded GrpSize ObsPHat ObsCliffd ObsStdES PHatType1ER CliffdType1ER StdMDType1ER
#1 2G_n No 5 0.4848 -0.0304 -0.054156883 0 0 0.0
```

```

#2  2G_n      No      10  0.5036  0.0072  0.002888142      0      0      0.1

#as.data.frame(calculateMAType1Error(mean=0,sd=1,N=c(5,10),reps=10,type="l",Experiments=5,
# FourG=FALSE,StdAdj=0,Blockmean=0,seed=123))
# Design BEIncluded GrpSize ObsPHat ObsCliffd  ObsStdES PHatType1ER CliffdType1ER StdMDType1ER
#1  2G_l      No       5  0.4848  -0.0304 -0.02789656      0      0      0.0
#2  2G_l      No      10  0.5036  0.0072  0.06473696      0      0      0.2

#as.data.frame(calculateMAType1Error(mean=0,sd=1,N=c(5,10),reps=10,type="n",Experiments=5,
# FourG=TRUE,StdAdj=0.5,Blockmean=0.5,seed=123))
# Design BEIncluded GrpSize ObsPHat ObsCliffd  ObsStdES PHatType1ER CliffdType1ER StdMDType1ER
# 1 4G_n_het  Yes       5  0.5108  0.0216  0.01361820      0      0      0.1
# 2 4G_n_het  Yes      10  0.5069  0.0138  0.01700672      0      0      0.0

as.data.frame(calculateMAType1Error(mean=0,sd=1,N=c(5,10),reps=5,type="l",Experiments=5,
  FourG=TRUE,StdAdj=0,Blockmean=0.5,seed=123))
#Results for reps=10
# Design BEIncluded GrpSize ObsPHat ObsCliffd  ObsStdES PHatType1ER CliffdType1ER StdMDType1ER
#1  4G_l      Yes       5  0.5108  0.0216  0.07578257      0      0      0.2
#2  4G_l      Yes      10  0.5072  0.0144  0.04839936      0      0      0.0

```

---

calculateNullIESAccuracy

*calculateNullIESAccuracy*

---

## Description

The function uses simulation to assess the accuracy when the mean difference is zero, and the type 1 error rates of parametric and non-parametric effect sizes for both two group randomized designs and four group randomized block designs, for each of four different distributions.

## Usage

```

calculateNullIESAccuracy(
  mean = 0,
  sd = 1,
  N = 10,
  reps = 10,
  type = "n",
  seed = 123,
  StdAdj = 0,
  Blockmean = 0.5
)

```

## Arguments

mean	The mean of the baseline distribution.
sd	The standard deviation or shape of the baseline distribution

N	The number of observations per group for two group experiments and N/2 the sample sizes for four group experiments. N must be even to ensure equal N/2 defines appropriate sample sizes per group for 4 group experiments
reps	The number of replications (i.e. two-group and four group experiments) to be simulated
type	A string parameter defining the distribution being simulated i.e. 'n' for normal data, 'l' for log-normal data, 'g' for gamma data and 'lap' for LaPlace data.
seed	A starting value for the simulations
StdAdj	A numerical parameter that can be used to add additional variance for normal, lognormal and Laplace data and to change the shape parameter for gamma data.
Blockmean	A numerical parameter used to introduce a fixed Block effect for four group experiments

### Value

A tibble identifying the median absolute error for the effect sizes Cliff's d, phat and StdMD and the Type 1 error rate, estimated from the proportion of significant effect sizes in the simulated experiments.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
as.data.frame(
  calculateNullIESAccuracy(
    mean=0,sd=1,N=10,reps=30,type='n',seed=123,StdAdj = 0,Blockmean = 0.5))
# Design Obs CliffdAbsError PHatAbsError StdESdAbsError varCliffd varPHat
# 1 2G_n 20 0.20 0.10 0.2624447 0.05530851 0.01382713
# 2 4G_n 20 0.16 0.08 0.1848894 0.05447540 0.01361885
# varStdES ObsCliffd ObsPHat ObsStdES CliffdType1ER PHatType1ER
# 1 0.1425374 0.021333333 0.5106667 0.0001190251 0 0
# 2 0.1484728 -0.009333333 0.4953333 0.0295002335 0 0
# StdESType1ER
# 1 0.03333333
# 2 0.03333333
#as.data.frame(
# calculateNullIESAccuracy(
# mean=0,sd=1,N=10,reps=100,type='n',seed=123,StdAdj = 0,Blockmean = 0.5))
# Design Obs CliffdAbsError PHatAbsError StdESdAbsError varCliffd varPHat varStdES ObsCliffd
#1 2G_n 20 0.21 0.105 0.3303331 0.08064949 0.02016237 0.2488365 -0.0010
#2 4G_n 20 0.16 0.080 0.2565372 0.05933430 0.01483358 0.1769521 0.0052
# ObsPHat ObsStdES CliffdType1ER PHatType1ER StdESType1ER
#1 0.4995 -0.02395895 0.07 0.08 0.08
#2 0.5026 0.03769940 0.01 0.01 0.02
```

---

calculatePhat	<i>calculatePhat</i>
---------------	----------------------

---

### Description

This function calculates the probability of superiority (i.e., Phat) and its confidence interval based on Brunner and Munzel (2000) heteroscedastic analog of WMW test. It is based on Wilcox's `bmp` function with some amendments. It does not include a plotit facility. It uses the smallest non-zero variance to identify confidence intervals and statistical significance for values of Phat=0 and Phat=1. It ensure that confidence intervals do not take on invalid values such as values <0 or >1.

### Usage

```
calculatePhat(x, y, alpha = 0.05, sigfig = -1)
```

### Arguments

<code>x</code>	is a vector of values from group 1
<code>y</code>	is a vector of values from group 2
<code>alpha</code>	is the Type 1 error level for statistical tests
<code>sigfig</code>	is the number of significant digits. If <code>sigfig&gt;0</code> the data in <code>x</code> and <code>y</code> is truncated to the specified number of significant digits.

### Value

list including the value of the t-test for PHat, the estimate of PHat and Cliff's d, and the confidence intervals for PHat.

### Author(s)

Rand Wilcox amendments by Barbara Kitchenham and Lech Madeyski

### Examples

```
x <- c(1.2, 3.0, 2.2, 4.0, 2.5, 3.0)
y <- c(3, 4.2, 4, 6, 7, 5.9)
reproducer:::calculatePhat(x, y)
# $test.stat
# [1] 6.381249
# $phat
# [1] 0.9305556
# $dhat
# [1] 0.8611111
# $sig.level
# [1] 0.0001191725
# $s.e.
# [1] 0.06747199
# $ci.p
```

```

# [1] 0.7783001 1.0000000
# $df
# [1] 9.148489
# Another example:
z <- c(1, 2, 3, 4)
y <- c(5, 6, 7, 8)
reproducer::calculatePhat(z, y)
# $test.stat
# [1] 10.6066
# $phat
# [1] 1
# $dhat
# [1] 1
# $sig.level
# [1] 4.135921e-05
# $s.e.
# [1] 0.04419417
# $ci.p
# [1] 0.8918608 1.0000000
# $df
# [1] 6

```

---

calculatePopulationStatistics

*calculatePopulationStatistics*

---

## Description

This helper function constructs the theoretical effect sizes and distribution statistics four (normal, lognormal, Laplace & gamma) given specific parameter values for the distributions and is used to support the calculation of population statistics for two and four group experiments.

## Usage

```
calculatePopulationStatistics(mean, std, type = "n")
```

## Arguments

mean	The theoretical central location parameter for the distribution specified by the type parameter.
std	The theoretical spread parameter for the distribution specified by the type parameter.
type	String identifying the distribution, 'n' for normal, 'ln' for lognormal, 'lap' for Laplace, 'g' for Gamm

## Value

dataframe containing the expected standardized effect size, mean, variance, skewness and kurtosis statistics for samples from the specific distribution

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
reproducer:::calculatePopulationStatistics(mean=0, std=1, type='l')
# RawMean RawVariance RawEffectSize RawSkewness RawKurtosis
#1 1.648721 4.670774 0.762874 6.184877 88.54343
reproducer:::calculatePopulationStatistics(mean=0, std=1, type='n')
# RawMean RawVariance RawEffectSize RawSkewness RawKurtosis
# 1 0 1 0 0 3
```

---

CalculateRLevel1

*CalculateRLevel1*

---

**Description**

This function calculates the r value for a 2-group (2G) or 4-Group (4G) Crossover experiment for each sequence group and each outcome metric. The function returns both the exact r value and the r value based on pooled variances for each sequence group and outcome metric

**Usage**

```
CalculateRLevel1(
  Dataset,
  StudyID,
  Groups = c("A", "B", "C", "D"),
  ExperimentName,
  Metrics,
  Type,
  Control
)
```

**Arguments**

Dataset	This holds the data for each participant in a 2-group or 4-group crossover experiment in the 'wide' format. I.e., there is only one entry per participant. The data set should have been generated from a long version of the data based on a variable labelled 'Period' which is used to define which participant data was collected in the first period of the experiment - see function <code>ExtractLevel1ExperimentRData</code> .
StudyID	This holds an identifier used to identify the origin of the experimental data in the output from this function.
Groups	This is a list that defined the sequence group identifiers used in the dataset.
ExperimentName	This an identifiers used to define the specific experiment in the output from this function.
Metrics	This is a list of metrics, e.g., ('Correctness','Time','Efficiency').

Type            this is a character string specifying whether the experiment is a two sequence group of four sequence group experiment.

Control         this is a character string that defines the control treatment in the experiment.

### Details

script to obtain correlation coefficients

### Value

table this is a tibble holding information identifying for each metric and sequence group the first time period and second time period variance, the pooled variance, the variance of the difference values and the exact r and pooled r. # importFrom stats # importFrom var # importFrom tibble

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
ExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
ShortExperimentNames <- c("E1", "E2", "E3", "E4")
Metrics <- c("Comprehension", "Modification")
Type <- c("4G", "4G", "4G", "4G")
Groups <- c("A", "B", "C", "D")
StudyID <- "S2"
Control <- "SC"
# Obtain experimental data from a file and put in wide format
dataset2 <- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM
ReshapedData <- ExtractExperimentData(dataset2,
  ExperimentNames = ExperimentNames,
  idvar = "ParticipantID", timevar = "Period", ConvertToWide = TRUE
)
# Calculate the correlations for each sequence group and each metric.
CalculateRLevel1(
  Dataset = ReshapedData[[1]], StudyID, Groups = c("A", "B", "C", "D"),
  ExperimentName = ShortExperimentNames[1], Metrics, Type = Type[1], Control
)
# A tibble: 8 x 15
# # A tibble: 8 x 15
# Study Exp Group Metric Id n ControlFirst var1 var2
# <chr> <chr> <chr> <chr> <chr> <int> <lg1> <dbl> <dbl>
# 1 S2 E1 A Compr... S2E1A 6 FALSE 0.0183 0.0163
# 2 S2 E1 B Compr... S2E1B 6 TRUE 0.0201 0.0326
# 3 S2 E1 C Compr... S2E1C 6 FALSE 0.00370 0.0155
# 4 S2 E1 D Compr... S2E1D 6 TRUE 0.0173 0.0201
# 5 S2 E1 A Modif... S2E1A 6 FALSE 0.0527 0.0383
# 6 S2 E1 B Modif... S2E1B 6 TRUE 0.0185 0.0482
# 7 S2 E1 C Modif... S2E1C 6 FALSE 0.00655 0.0244
# 8 S2 E1 D Modif... S2E1D 6 TRUE 0.0222 0.0266
# # ... with 6 more variables: varp <dbl>, ControlVarProp <dbl>,
# # VarProp <dbl>, vardiff <dbl>, r <dbl>, r.p <dbl>
```

---

```
calculateSmallSampleSizeAdjustment
      calculateSmallSampleSizeAdjustment
```

---

### Description

Function calculates the Hedges small sample size adjustment for standardized mean effect sizes. It calculates the exact value unless the caller sets the parameter exact to FALSE, or the degrees of freedom is too large.

Function calculates the small sample size adjustment for standardized mean effect sizes

### Usage

```
calculateSmallSampleSizeAdjustment(df, exact = TRUE)
calculateSmallSampleSizeAdjustment(df, exact = TRUE)
```

### Arguments

df	A vector of degrees of freedom
exact	Default value=TRUE, if exact==TRUE the function returns the exact value of the adjustment(s) which is suitable for small values of df, if exact==FALSE the function returns the approximate version of the adjustment(s). See Hedges and Olkin 'Statistical methods for Meta-Analysis' Academic Press 1985.

### Value

small sample size adjustment value  
small sample size adjustment value

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
df <- 2
c <- calculateSmallSampleSizeAdjustment(df)

df <- c(5, 10, 17)
adjexact <- calculateSmallSampleSizeAdjustment(df)
# adjexact=0.8407487 0.9227456 0.9551115
# Hedges and Olkin values 0.8408, 0.9228,0.9551
adjapprox <- calculateSmallSampleSizeAdjustment(df, FALSE)
# adjapprox=0.8421053 0.9230769 0.9552239
df <- 2
```

```

a <- calculateSmallSampleSizeAdjustment(df)
# > a
# [1] 0.5641896

df <- c(5, 10, 17)
adjexact <- calculateSmallSampleSizeAdjustment(df)
# > adjexact
# [1] 0.8407487 0.9227456 0.9551115
# Hedges and Olkin values 0.8408, 0.9228, 0.9551
adjapprox <- calculateSmallSampleSizeAdjustment(df, FALSE)
# > adjapprox
# [1] 0.8421053 0.9230769 0.9552239
# Another example:
df <- c(10, 25, 50)
calculateSmallSampleSizeAdjustment(df, exact = TRUE)
# [1] 0.9227456 0.9696456 0.9849119
calculateSmallSampleSizeAdjustment(df, exact = FALSE)
# [1] 0.9230769 0.9696970 0.9849246

```

---

CatchError

*CatchError*


---

### Description

This is a helper function to stop simulations failing if the metafor function `rma` fails for example cannot converge properly for a specific dataset.

### Usage

```
CatchError(expr)
```

### Arguments

`expr`                    The expression that is being monitored

### Value

A message confirming whether the expression has performed successfully

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```

ES=c(0.2,0.3)
ESvar=c(0.04,0.03)
outcome=reproducer:::CatchError(metafor:::rma(ES,ESvar,method='Meth'))
outcome
# [1] 'Failure'

```

---

```
checkIfValidDummyVariable  
    checkIfValidDummyVariable
```

---

**Description**

This helper function checks whether a vector variable comprises only zeros and 1's.

**Usage**

```
checkIfValidDummyVariable(vector)
```

**Arguments**

vector	Vector variable
--------	-----------------

**Value**

The logical value: TRUE - if a vector variable passed as the function's parameter represents a valid dummy variable, i.e., comprises only zeros and 1's. FALSE - if a vector variable passed as the function's parameter does not represent a valid dummy variable.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
print(reproducer:::checkIfValidDummyVariable(c(0, 1, 0, 0)))  
# [1] TRUE  
print(reproducer:::checkIfValidDummyVariable(c(0, 1, 2, 0)))  
# [1] FALSE
```

---

```
Ciolkowski09ESEM.MetaAnalysis.PBRvsCBRorAR  
    Ciolkowski09ESEM.MetaAnalysis.PBRvsCBRorAR data
```

---

**Description**

Data form a set of primary studies on reading methods for software inspections. They were reported and analysed by M. Ciolkowski ('What do we know about perspective-based reading? an approach for quantitative aggregation in software engineering', in Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM'09, pp. 133-144, IEEE Computer Society, 2009), corrected and re-analysed by Madeyski and Kitchenham ('How variations in experimental designs impact the construction of comparable effect sizes for meta-analysis' (to be submitted)).

**Usage**

```
Ciolkowski09ESEM.MetaAnalysis.PBRvsCBRorAR
```

**Format**

A data frame with 21 rows and 7 variables:

**Study** Name of empirical study

**Ref.** Reference to the paper reporting primary study or experimental run where data were originally reported

**Control** Control treatment: Check-Based Reading (CBR) or Ad-hoc Reading (AR)

**Within-subjects** Yes - if the primary study used the within-subjects experimental design, No - if the primary study did not use the within-subjects experimental design

**Cross-over** Yes - if the primary study used the cross-over experimental design, No - if the primary study did not use the cross-over experimental design

**d\_ByCiolkowski** d effect size calculated by Ciolkowski

**d\_ByOriginalAuthors** d effect size as reported by the original authors

**Details**

If you use this data set please cite: Lech Madeyski and Barbara Kitchenham, 'How variations in experimental designs impact the construction of comparable effect sizes for meta-analysis', 2015.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

```
Ciolkowski09ESEM.MetaAnalysis.PBRvsCBRorAR
```

---

Cliffd.test

*Cliffd.test*

---

**Description**

This function provides single-sided and two-sided tests of Cliff's d

**Usage**

```
Cliffd.test(x, y, alpha = 0.05, alternative = "two.sided", sigfig = -1)
```

**Arguments**

x	The data from one group
y	The data from the alternative group
alpha	The significance level of tests which also controls the values of the confidence interval (default 0.05)
alternative	This defines whether a one-sided test or a two-sided test is required (default "two.sided"). For a one-sided test use parameter values 'greater' or 'less' to define whether the d-value should be greater or less than zero.
sigfig	is the number of significant digits. If sigfig>0 the data in x and y is truncated to the specified number of significant digits.

**Value**

The values of Cliff's d and its standard error, the t-value, its pvalue and the upper and lower confidence interval.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
a=c(1.2,3,2.2,4,2.5,3)
b=c(3,4.2,4,6,7,5.9)
Cliffd.test(a,b,alpha = .05,alternative='two.sided',sigfig = -1)
# A tibble: 1 x 7
#   d sqse.d d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <lgl>
#1 -0.861 0.0202 -42.7 1.20e-12 -0.896 -0.816 TRUE

Cliffd.test(b,a,alpha = .05,alternative='greater',sigfig = -1)
# A tibble: 1 x 7
#   d sqse.d d.tvalue d.pvalue d.ci.lower d.ci.upper d.sig
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <lgl>
#1 0.861 0.0202 42.7 5.99e-13 0.824 1 TRUE
```

---

constructEffectSizes *constructEffectSizes*

---

**Description**

The function constructs various different d-style effect sizes for a set of different experiments given basic statistics from each experiment ( the mean value of the control group  $M_c$ , the mean value of the treatment group  $M_t$ , the standard deviation of the control group  $SD_c$ , standard deviation of the the treatment group  $SD_t$ , the number of observations (participants) in the control group  $N_c$ , and the number of observations (participants) in the treatment group  $N_t$ ). The input variables can be

vectors or individual numbers but all input vectors must be of the same length. The function returns Glass's Delta, Cohen's D, point bi-serial r (based on Hedges' g unadjusted), Hedges' g and Hedges' g adjusted for small sample size.

### Usage

```
constructEffectSizes(Mc, Mt, SDc, SDt, Nc, Nt)
```

### Arguments

Mc is a vector containing the mean value of the control group for each experiment.  
 Mt is a vector containing the mean value of the treatment group for each experiment.  
 SDc is a vector of the standard deviations of the control group for each experiment.  
 SDt is a vector of the standard deviations of the the treatment group for each experiment.  
 Nc is a vector containing the the number of observations (participants) in the control group for each experiment.  
 Nt is a vector of the number of observations (participants) in the treatment group for each experiment.

### Value

data frame composed of five effect sizes (Glass delta, Cohen's d, Hedges' g, r, Hedges' g adjusted)

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
constructEffectSizes(10, 15, 0.3, 0.2, 15, 15)

Mt <- c(0.633, 0.673, 0.423, 0.727, 0.631)
Mc <- c(0.612, 0.526, 0.356, 0.618, 0.534)
SDt <- c(0.198, 0.115, 0.172, 0.088, 0.122)
SDc <- c(0.159, 0.089, 0.111, 0.166, 0.119)
Nt <- c(12, 12, 14, 10, 8)
Nc <- c(12, 12, 14, 10, 8)
EffectSizes <- constructEffectSizes(Mc, Mt, SDc, SDt, Nt, Nc)
EffectSizes
# GlassDelta Cohend Hedgesg r HedgesgAdjusted
# 1 0.1320755 0.1221516 0.1169513 0.05837591 0.1129107
# 2 1.6516854 1.4931812 1.4296121 0.58151846 1.3802200
# 3 0.6036036 0.4803405 0.4628677 0.22547423 0.4493641
# 4 0.6566265 0.8648343 0.8204538 0.37953300 0.7857047
# 5 0.8151261 0.8604924 0.8049169 0.37335594 0.7608781
```

---

ConstructLevel1ExperimentRData  
*ConstructLevel1ExperimentRData*

---

### Description

This function returns the r value for a 2-group (2G) or 4-Group (4G) Crossover experiment for a group of 1 or more experiments for each sequence group and each outcome metric. For sets of 2 or more experiments, the experiments are assumed to be replicates and to report the same sets of Metrics and have the same Control treatment and use the same sequence Group identifiers, but are not necessarily the same Type. We return both the exact r value and the r value based on pooled variances for each sequence group and outcome metric.

### Usage

```
ConstructLevel1ExperimentRData(  
  Data,  
  StudyID,  
  ExperimentNames,  
  Groups,  
  Metrics,  
  Type,  
  Control  
)
```

### Arguments

Data	This is a list parameter each entry in the list holds the data for each participant in a 2-group or 4-group crossover experiment in the 'wide' format. I.e., there is only one entry per participant. The data should have been generated from a long version of the data based on a variable labelled 'Period' which is used to define which participant data was collected in the first period of the experiment - see function ExtractLevel1ExperimentRData.
StudyID	This holds an identifier used to identify the origin of the experimental data in the output from this function.
ExperimentNames	This a list of identifiers used to define each experiment in the output from this function.
Groups	This is a list that defined the sequence group identifiers used in the dataset.
Metrics	This is a list of of character strings identifying each outcome metric reported in each of the experiments in the set of replicated experiments.
Type	this is a list of character strings specifying for each experiment whether the experiment is a 2-group or 4-group experiment
Control	this is a character string that defines the control treatment in the experiment.

**Value**

R.Data.Table this is a tibble holding information identifying for each metric and sequence group the first time period and second time period variance, the pooled variance, the variance of the difference values and the exact r and pooled r.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
#
ShortExperimentNames <- c("E1", "E2", "E3", "E4")
FullExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
Metrics <- c("Comprehension", "Modification")
Groups <- Groups <- c("A", "B", "C", "D")
Type <- c(rep("4G", 4))
StudyID <- "S2"
Control <- "SC"
# Obtain experimental data from each file and put in wide format
dataset2 <- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM
ReshapedData <- ExtractExperimentData(dataset2,
  ExperimentNames = FullExperimentNames,
  idvar = "ParticipantID", timevar = "Period", ConvertToWide = TRUE
)
# Calculate the correlations for each sequence group and each metric in each experiment
ConstructLevel1ExperimentRData(
  Data = ReshapedData, StudyID = StudyID,
  ExperimentNames = ShortExperimentNames, Groups = Groups, Metrics = Metrics, Type = Type,
  Control = Control
)
# # A tibble: 32 x 15
# Study Exp Group Metric Id n ControlFirst var1 var2 varp
# <chr> <chr> <chr> <chr> <chr> <int> <lgl> <dbl> <dbl> <dbl>
# 1 S2 E1 A Compr... S2E1A 6 FALSE 0.0183 0.0163 0.0173
# 2 S2 E1 B Compr... S2E1B 6 TRUE 0.0201 0.0326 0.0263
# 3 S2 E1 C Compr... S2E1C 6 FALSE 0.00370 0.0155 0.00962
# 4 S2 E1 D Compr... S2E1D 6 TRUE 0.0173 0.0201 0.0187
# 5 S2 E1 A Modif... S2E1A 6 FALSE 0.0527 0.0383 0.0455
# 6 S2 E1 B Modif... S2E1B 6 TRUE 0.0185 0.0482 0.0333
# 7 S2 E1 C Modif... S2E1C 6 FALSE 0.00655 0.0244 0.0155
# 8 S2 E1 D Modif... S2E1D 6 TRUE 0.0222 0.0266 0.0244
# 9 S2 E2 A Compr... S2E2A 6 FALSE 0.0194 0.0425 0.0309
# 10 S2 E2 B Compr... S2E2B 6 TRUE 0.0198 0.0192 0.0195
# # ... with 22 more rows, and 5 more variables: ControlVarProp <dbl>,
# # VarProp <dbl>, vardiff <dbl>, r <dbl>, r.p <dbl>
```

---

`crossoverResidualAnalysis`*crossoverResidualAnalysis*

---

## Description

This function analyses one or more crossover experiments where each experiment can be either a two group or four group experiment as specified by `Type` parameter. The `file` parameter includes a dataset comprising one or more experiments as defined by the `ExperimentNames` parameter and each experiment includes values for every output variable defined in the `Metrics` parameter. After being analysed using the linear modeling `lmer` function of the `lme4` package, the residuals are assessed for normality based on the Anderson-Darling test. Warning 1. This function should only be used with data sets that include one or more individual crossover experiments in the format used in the datasets reported in the `reproducer` package that were used in the paper B. Kitchenham, L. Madeyski, G. Scanniello, and C. Gravino, “The importance of the correlation between results from individual participants in crossover studies,” *IEEE Transactions in SoftwareEngineering*, 2021 (Accepted for Publication). [Online]. Available: <https://doi.org/10.1109/TSE.2021.30>. Warning 2. The `lmer` function assumes that when experiments include multiple data from the same participant, the correlation between measures for the same participant will be positive. If there is no positive correlation, the function will deliver the warning: `boundary (singular) fit: see ?isSingular`. This does not mean the analysis has failed. It means that the within participant and between participant variance are set to the same value.

## Usage

```
crossoverResidualAnalysis(file, StudyID, ExperimentNames, Type, Metrics)
```

## Arguments

<code>file</code>	The dataset to be analysed.
<code>StudyID</code>	A character string used to identify the origin of the dataset
<code>ExperimentNames</code>	A vector of one or more strings variables identifying each experiment in the file.
<code>Type</code>	A vector of string variables identifying the design for each experiment. Each element should have the value <code>'2G'</code> or <code>'4G'</code>
<code>Metrics</code>	A vector of string variables identifying the variables to be analysed.

## Value

The results of analysing the residuals for each experiment and metric using the Anderson-Darling test (`ADpval`) and the number of outliers (`NUMOut`).

## Author(s)

Barbara Kitchenham and Lech Madeyski

**Examples**

```
File=reproducer::KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE
crossoverResidualAnalysis(
  File,StudyID='S1',ExperimentNames=c('USB2'),Type=c('4G'),
  Metrics=c('Correctness','Time','Efficiency'))
# Study Exp Metrics N ADpval NumOut
#1 S1 USB2 Correctness 24 0.0846 2
#2 S1 USB2 Time 24 0.0448 1
#3 S1 USB2 Efficiency 24 0.365 4
```

---

```
densityCurveOnHistogram
```

```
densityCurveOnHistogram
```

---

**Description**

Density curve overlaid on histogram

**Usage**

```
densityCurveOnHistogram(df, colName, limLow, limHigh)
```

**Arguments**

df	Data frame with data to be displayed
colName	Name of the selected column in a given data frame
limLow	the limit on the lower side of the displayed range
limHigh	the limit on the higher side of the displayed range

**Value**

A figure being a density curve overlaid on histogram

**Author(s)**

Lech Madeyski

**Examples**

```
densityCurveOnHistogram(Madeyski15EISEJ.PropProjects, "STUD", 0, 100)
# densityCurveOnHistogram(data.frame(x<-rnorm(50, mean=50, sd=5)), 'x', 0, 100)
```

---

doLM	<i>doLM</i>
------	-------------

---

### Description

This helper function is called by the function 'crossoverResidualAnalysis' to perform either an AB/BA crossover analysis or a four-group crossover on the data set defined by the parameter DataSet depending on the value of the Type parameter.

### Usage

```
doLM(DataSet, Metric, Type)
```

### Arguments

DataSet	The dataset to be analysed.
Metric	The name of the variable to be analysed
Type	Defines the experimental design

### Value

The data analysis results provided by the lmer function of the lme4 package

### Author(s)

Barbara Kitchenham and Lech Madeyski

---

effectSizeCI	<i>effectSizeCI</i>
--------------	---------------------

---

### Description

95 The procedure is based on finding the upper and lower 0.025 bounds for the related t-variable. The t-variable needs to be adjusted for bias by multiplying by c. The upper and lower bounds on the t-variable are then used to calculate the upper and lower bounds on the repeated measures effect size (d\_RM) by multiplying the upper and lower bound of the t-variable by  $\sqrt{(n1+n2)/(2*(n1*n2))}$ . Upper and lower bounds on the equivalent independent groups effect size (d\_IG) are found by multiplying the upper and lower bounds on d\_RM by  $\sqrt{1-r}$ .

**Usage**

```
effectSizeCI(
  expDesign,
  t,
  n1,
  n2,
  r = 0,
  epsilon = 1e-10,
  maxsteps = 1000,
  stepsize = 3
)
```

**Arguments**

expDesign	Experimental design: 1) crossover repeated measures ('CrossOverRM'), 2) before-after repeated measures (expDesign=='BeforeAfterRM'), 3) independent groups ('IG')
t	t-statistics (t must be less than or equal to 37.62, the limit from the R function documentation)
n1	The number of observations in sequence group 1 (expDesign=='CrossOverRM'), the number of observations in group 1 (expDesign=='IG'), or the total number of observations (expDesign=='BeforeAfterRM')
n2	The number of observations in sequence group 2 (expDesign=='CrossOverRM') or the number of observations in group 2 (expDesign=='IG')
r	The correlation between outcomes for individual subject (the within subject correlation)
epsilon	The precision of the iterative procedure
maxsteps	The maximum number of steps of the iterative procedure (the procedure terminates at maxsteps or earlier if CI with enough precision have been calculated)
stepsize	The size of steps (influences the convergence of the calculations, i.e., the number of steps required to obtain the final result of precision defined by the epsilon)

**Value**

A list of Confidence Intervals for: t-statistic (t\_LB and t\_UB), repeated-measures effect size d\_RM (d\_RM\_LB, d\_RM\_UB), independent groups effect size (d\_IG\_LB, d\_IG\_UB)

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
effectSizeCI(expDesign = "CrossOverRM", t = 14.4, n1 = 15, n2 = 15, r = 0.6401)
effectSizeCI(expDesign = "BeforeAfterRM", t = 14.16536, n1 = 15, n2 = 0, r = 0.6146771)
effectSizeCI(expDesign = "IG", t = -6.344175, n1 = 15, n2 = 15)
effectSizeCI(expDesign = "CrossOverRM", t = 0.5581, n1 = 6, n2 = 6, r = 0.36135)
effectSizeCI(expDesign = "CrossOverRM", r = 0.855, t = 4.33, n1 = 7, n2 = 6)
```

---

ExtractExperimentData *ExtractExperimentData*

---

## Description

This function reads datasets from a defined directory in the reproducer package that hold the results of a family crossover experiments in the long format. It converts the data to the wide format if required.

## Usage

```
ExtractExperimentData(  
  DataSet,  
  ExperimentNames,  
  idvar = "ParticipantID",  
  timevar = "Period",  
  ConvertToWide = TRUE  
)
```

## Arguments

DataSet	This is a tibble holding the data for each crossover experiment in a family (a family can include only one experiment).
ExperimentNames	This is a list with the full names of each experiment.
idvar	This is the name of the column that contains the data for specific participants. It is only assumed to be unique within an experiment (default idvar='ParticipantID').
timevar	This is the name of the table column that defines which data was collected in a specific time period. This function assumes that there are only two time periods (default timevar='Period').
ConvertToWide	This determine whether the function converts the data to the wide format (default ConvertToWide=TRUE).

## Value

A list with an entry for the data for each experiment. If ConvertToWide is TRUE, it returns the data in the wide format otherwise it returns the data as it was read. Within each list item the data is returned as a tibble #importFrom stats # importFrom tibble # importFrom base

## Author(s)

Barbara Kitchenham and Lech Madeyski

**Examples**

```

ExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
Metrics <- c("Comprehension", "Modification")
Groups <- c("A", "B", "C", "D")
Type <- c(rep("4G", 4))
StudyID <- "S2"
Control <- "SC"
# Obtain experimental data from each file and put in wide format
dataset2 <- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM
ReshapedData <- ExtractExperimentData(dataset2,
  ExperimentNames = ExperimentNames,
  idvar = "ParticipantID", timevar = "Period", ConvertToWide = TRUE
)
ReshapedData[[1]]

# A tibble: 24 x 15
# ParticipantID ExperimentID.1 SequenceGroup.1 System.1 Treatment.1 Comprehension.1
# <fct> <fct> <fct> <fct> <fct> <dbl>
# 1 1 EUBAS A S1 AM 0.77
# 2 5 EUBAS A S1 AM 0.61
# 3 9 EUBAS A S1 AM 0.61
# 4 13 EUBAS A S1 AM 0.52
# 5 17 EUBAS A S1 AM 0.43
# 6 21 EUBAS A S1 AM 0.77
# 7 2 EUBAS B S1 SC 0.92
# 8 6 EUBAS B S1 SC 0.63
# 9 10 EUBAS B S1 SC 0.51
# 10 14 EUBAS B S1 SC 0.64
# ... with 14 more rows, and 9 more variables: Modification.1 <dbl>, CrossOverID.1 <fct>,
# ExperimentID.2 <fct>, SequenceGroup.2 <fct>, System.2 <fct>, Treatment.2 <fct>,
# Comprehension.2 <dbl>, Modification.2 <dbl>, CrossOverID.2 <fct>

```

---

ExtractGroupSizeData *ExtractGroupSizeData*

---

**Description**

This function constructs a table identifying the number of participants in each sequence group for a set of experiments each of which used a crossover design.

**Usage**

```

ExtractGroupSizeData(
  ExpDataWide,
  StudyID,
  ShortExperimentNames,
  Type,
  Groups = c("A", "B", "C", "D")
)

```

**Arguments**

ExpDataWide	this is a list of tibbles each comprising data from one experiment in its wide format
StudyID	an identifier for the group of related experiments (i.e., a family).
ShortExperimentNames	a list of character strings identifying each experiment.
Type	A list identifying the type of crossover '2G' or '4G' for each experiment in the family
Groups	a list of the terms used to specify sequence groups in the experiments.

**Value**

A tibble containing the number of participants in each sequence group in each experiment.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

ExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
ShortExperimentNames <- c("E1", "E2", "E3", "E4")
Metrics <- c("Comprehension", "Modification")
Type <- c("4G", "4G", "4G", "4G")
Groups <- c("A", "B", "C", "D")
StudyID <- "S2"
Control <- "SC"
# Obtain experimental data from a file and put in wide format
dataset2 <- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM
ReshapedData <- ExtractExperimentData(dataset2,
  ExperimentNames = ExperimentNames,
  idvar = "ParticipantID", timevar = "Period", ConvertToWide = TRUE
)
ExtractGroupSizeData(ReshapedData, StudyID, ShortExperimentNames, Type, Groups = Groups)
# A tibble: 16 x 4
#   Study Exp  Group    n
#   <chr> <chr> <chr> <int>
# 1 S2   Exp1  A         6
# 2 S2   Exp1  B         6
# 3 S2   Exp1  C         6
# 4 S2   Exp1  D         6
# 5 S2   Exp2  A         6
# 6 S2   Exp2  B         6
# 7 S2   Exp2  C         5
# 8 S2   Exp2  D         5
# 9 S2   Exp3  A         5
# 10 S2  Exp3  B         5
# 11 S2  Exp3  C         6
# 12 S2  Exp3  D         6
# 13 S2  Exp4  A         5

```

```
# 14 S2   Exp4 B      5
# 15 S2   Exp4 C      4
# 16 S2   Exp4 D      4
```

---

ExtractMAStatistics    *ExtractMAStatistics*

---

### Description

This function extracts summary statistics from meta-analysis results obtained from the `rma` function of the `metafor` R package. If required the function transform back to standardized mean difference (effect size type 'd' i.e. Hg) or point biserial correlations (effect size type 'r'). Warning: the 'ExtractMAStatistics' function works with 'metafor' version 2.0-0, but changes to metafor's method of providing access to its individual results may introduce errors into the function.

This function extracts summary statistics from meta-analysis results obtained from the `rma` function of the `metafor` R package. If required the function transform back to standardized mean difference (effect size type 'd' i.e. Hg) or point biserial correlations (effect size type 'r'). Warning: the 'ExtractMAStatistics' function works with 'metafor' version 2.0-0, but changes to metafor's method of providing access to its individual results may introduce errors into the function.

### Usage

```
ExtractMAStatistics(
  maresults,
  Nc,
  Nt,
  Transform = TRUE,
  type = "d",
  sig = 4,
  returnse = FALSE
)
```

```
ExtractMAStatistics(
  maresults,
  Nc,
  Nt,
  Transform = TRUE,
  type = "d",
  sig = 4,
  returnse = FALSE
)
```

### Arguments

`maresults`        is the output from the `rma` function.  
`Nc`                is the number of participants in the control condition group.

Nt	is the number of participants in the treatment condition group.
Transform	is a boolean value indicating whether the outcome values need to be transformed back to standardized mean difference ('d' i.e. Hg or d) or point biserial correlations ('r'). It is defaulted to TRUE. If this parameter is set to FALSE, no transformation will be applied.
type	this indicates the type of transformation required - it defaults to 'd' which requests transformation from Zr to Hg, using 'r' requests transformation from Zr to r.
sig	indicates the number of significant digits requested in the output, the default is 4; it rounds the values of mean, pvalue, upper and lower bound to the specified number of significant digits.
returnse	if set to TRUE returns the standard error of the effect size (default: returnse=FALSE)

### Value

data frame incl. summary statistics from meta-analysis results: overall mean value for the effect sizes, the p-value of the mean, the upper and lower confidence interval bounds (UB and LB), QE which is the heterogeneity test statistic and QEp which the the p-value of the heterogeneity statistic

data frame incl. summary statistics from meta-analysis results: overall mean value for the effect sizes, the p-value of the mean, the upper and lower confidence interval bounds (UB and LB), QE which is the heterogeneity test statistic and QEp which the the p-value of the heterogeneity statistic

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
ExpData <- reproducer::KitchenhamMadeyskiBrereton.ExpData
# Extract the experiment basic statics
S1data <- subset(ExpData, ExpData == "S1")
# Use the descriptive data to construct effect size
S1EffectSizes <- reproducer::PrepareForMetaAnalysisGtoR(
  S1data$Mc, S1data$Mt, S1data$SDc, S1data$SDt, S1data$Nc, S1data$Nt
)
# Do a random effect meta-analysis of the transformed r_pbs effect size
S1MA <- metafor::rma(S1EffectSizes$zr, S1EffectSizes$vi)
# Extract summary statistics from meta-analysis results and transform back to Hg scale
S1MAStats <- reproducer::ExtractMAStatistics(S1MA, sum(S1data$Nc), sum(S1data$Nt), TRUE, "d", 4)
#   mean  pvalue  UB   LB  QE  QEp
# 1 0.6658 0.002069 1.122 0.2384 4 0.41
ExpData <- reproducer::KitchenhamMadeyskiBrereton.ExpData
# Extract the experiment basic statics
S1data <- subset(ExpData, ExpData == "S1")
# Use the descriptive data to construct effect size
S1EffectSizes <- reproducer::PrepareForMetaAnalysisGtoR(
  S1data$Mc, S1data$Mt, S1data$SDc, S1data$SDt, S1data$Nc, S1data$Nt
)
# Do a random effect meta-analysis of the transformed r_pbs effect size
```

```

S1MA <- metafor::rma(S1EffectSizes$zr, S1EffectSizes$vi)
# Extract summary statistics from meta-analysis results and transform back to Hg scale
ExtractMAStatistics(S1MA, sum(S1data$Nc), sum(S1data$Nt), TRUE, "d", 4)
#   mean  pvalue  UB   LB  QE  QEp
# 1 0.6658 0.002069 1.122 0.2384 4 0.41
ExtractMAStatistics(S1MA, sum(S1data$Nc), sum(S1data$Nt), FALSE, "d", 4)
# A tibble: 1 x 6
#   mean  pvalue  UB   LB   QE  QEp
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
# 1 0.327 0.00207 0.535 0.119     4 0.41

```

---

ExtractSummaryStatisticsRandomizedExp

*ExtractSummaryStatisticsRandomizedExp*

---

## Description

This function extracts data obtained from the lme4 package lmer function. It assumes a simple randomized experiment with each element having one or more repeated measures. It outputs the mean together with its standard error and confidence interval bounds.

## Usage

```
ExtractSummaryStatisticsRandomizedExp(lmeRA, N, alpha = 0.05)
```

## Arguments

lmeRA	The output from the lmer function
N	The total number of observations
alpha	the probability level to be used when constructing the confidence interval bounds.

## Value

REA.Summary A dataframe holding the number of observations N, the overall mean value as its standard error reported as by the lmer function, and its confidence interval bounds.

## Author(s)

Barbara Kitchenham and Lech Madeyski

## Examples

```

ShortExperimentNames <- c("E1", "E2", "E3", "E4")
FullExperimentNames <- c("EUBAS", "R1UCLM", "R2UCLM", "R3UCLM")
Metrics <- c("Comprehension", "Modification")
Groups <- c("A", "B", "C", "D")
Type <- c(rep("4G", 4))

```

```

StudyID <- "S2"
Control <- "SC"
ReshapedData <- ExtractExperimentData(
  KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM,
  ExperimentNames = FullExperimentNames, idvar = "ParticipantID", timevar = "Period",
  ConvertToWide = TRUE
)
NewTable <- ConstructLevel1ExperimentRData(
  ReshapedData, StudyID, ShortExperimentNames, Groups,
  Metrics, Type, Control
)
resRe <- lme4::lmer(r ~ (1 | Id), data = NewTable)
summary(resRe)
# Linear mixed model fit by REML ['lmerMod']
# Formula: r ~ (1 | Id)
# REML criterion at convergence: 47.8
# Scaled residuals:
#   Min       1Q   Median       3Q      Max
# -1.4382 -0.9691  0.2190  0.8649  1.4761
#
# Random effects:
#   Groups      Name                Variance Std.Dev.
#   Id          (Intercept) 0.03978  0.1994
#   Residual                    0.20974  0.4580
# Number of obs: 32, groups: Id, 16
#
# Fixed effects:
#              Estimate Std. Error t value
# (Intercept)  0.06175    0.09508    0.649
# N=length(NewTable$r)
ExtractSummaryStatisticsRandomizedExp(lmeRA = resRe, N = 32, alpha = 0.05)
#      N      Mean      SE LowerBound UpperBound
#  1 32 0.06175 0.09508   -0.1319    0.2554

```

---

 fmt

*fmt*


---

## Description

Formatting function to set decimal precision in labels

## Usage

```
fmt()
```

## Author(s)

Lech Madeyski

---

```
getEffectSizesABBA    getEffectSizesABBA
```

---

### Description

Function to calculate both effect sizes (dIG, dRM), i.e., independent groups and repeated measures standardized effect sizes and variances, for AB/BA crossover design studies. Function is used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

### Usage

```
getEffectSizesABBA(simulationData)
```

### Arguments

simulationData - data set in a form required to calculate effect sizes in AB/BA crossover experimental designs

### Value

data frame incl. calculated effect sizes and variances: # dIG - independent groups standardized effect size # var.dIG - variance of independent groups standardized effect size # dRM - repeated measures (within-subjects) standardized effect size # var.dRM - variance of repeated measures (within-subjects) standardized effect size # dIG.Fromt - independent groups standardized effect size calculated from t:  $dIG.Fromt = t * \sqrt{(1-r) * \sqrt{(N1+N2)/(2*N1*N2)}}$  # var.dIG.Fromt - variance of independent groups standardized effect size calculated from t:  $var.dIG.Fromt = var.t * (1-r) * (N1+N2)/(2*N1*N2)$  # dRM.Fromt - dRM calculated from t:  $dRM.Fromt = t * \sqrt{(N1+N2)/(2*N1*N2)}$  # var.dRM.Fromt - var.dRM calculated from t:  $var.dRM.Fromt = var.t * (N1+N2)/(2*N1*N2)$  # var.dRM.Fromt2 - var.dRM calculated from t or rather dRM.Fromt:  $var.dRM.Fromt2 = (df/(df-2)) * ((N1+N2)/(2*N1*N2) + dRM.Fromt^2/c^2)$  # var.dRM.Approx - var.dRM calculated on a basis of Johnson and Welch (1940) report an approximate formulate for the variance of a t variable:  $var.dRM.Approx = ((N1+N2)/(2*N1*N2)) + (dRM^2)/(2*(N1+N2-2))$  #see paper and Equation 49 # var.dIG.Approx - var.dIG calculated on a basis of Johnson and Welch (1940) report an approximate formulate for the variance of a t variable:  $var.dIG.Approx = (((N1+N2)*(1-r))/(2*N1*N2)) + (dIG^2)/(2*(N1+N2-2))$  #see paper and Equation 50 # unstandardizedES - estimated unstandardized technique effect size # periodES - estimated period effect # var.sig - sum of within-subjects variance and between-subjects variance # var.within - within-subjects variance # var.between - between-subjects variance # t - t-value # var.t - variance of t-variable # gRM - Hedges and Olkin (1985) unbiased estimator of the repeated measures effect size  $gRM = dRM * c$  # var.gRM - variance of gRM calculated as follows:  $var.gRM = (df/(df-2)) * (((N1+N2)/(2*N1*N2)) * c^2 + gRM^2) - gRM^2/c^2$  #Equation 56 # var.gRM2 - variance of gRM calculated as follows:  $var.gRM2 = var.dRM * c^2$  # gIG - Hedges and Olkin (1985) unbiased estimator of the independent groups effect size  $gIG = dIG * c$  # var.gIG - variance of gIG calculated as follows:  $var.gIG = (df/(df-2)) * (((N1+N2)/(2*N1*N2)) * c^2 + gIG^2) - gIG^2/c^2$  #Equation 57 # var.gIG2 - variance of gRM calculated as follows:  $var.gIG2 = var.dIG * c^2$  # r - the correlation between the values observed for the same subject

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
simulationData <- getSimulationData(25, 18.75, 50, 10, 5, 500) # generate simulated data set
es <- getEffectSizesABBA(simulationData) # return effect sizes and variances
# OR
simulationData <- getSimulationData(25, 18.75, 50, 10, 5, 15)
es <- getEffectSizesABBA(simulationData) # return effect sizes and variances
```

---

```
getEffectSizesABBAIgnoringPeriodEffect
      getEffectSizesABBAIgnoringPeriodEffect
```

---

**Description**

Function to calculate both effect sizes (dIG.ipe, dRM.ipe), i.e., independent groups and repeated measures standardized effect sizes and variances, for AB/BA crossover design studies ignoring period effect (thus wrong). Function was removed in the revision of the paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

**Usage**

```
getEffectSizesABBAIgnoringPeriodEffect(simulationData)
```

**Arguments**

simulationData - data set in a form required to calculate effect sizes in AB/BA crossover experimental designs

**Value**

data frame incl. calculated effect sizes and variances: # dIG.ipe - independent groups standardized effect size # var.dIG.ipe - variance of independent groups standardized effect size # dRM.ipe - repeated measures (within-subjects) standardized effect size # var.dRM.ipe - variance of repeated measures (within-subjects) standardized effect size # dIG.Fromt.ipe - independent groups standardized effect size calculated from t:  $dIG.Fromt = t * \sqrt{(1-r) * \sqrt{(N1+N2)/(2*N1*N2)}}$  # var.dIG.Fromt.ipe - variance of independent groups standardized effect size calculated from t:  $var.dIG.Fromt = var.t * (1-r) * (N1+N2)/(2*N1*N2)$  # dRM.Fromt.ipe - dRM calculated from t:  $dRM.Fromt = t * \sqrt{(N1+N2)/(2*N1*N2)}$  # var.dRM.Fromt.ipe - var.dRM calculated from t:  $var.dRM.Fromt = var.t * ((N1+N2)/(2*N1*N2))$  # var.dRM.Fromt2.ipe - var.dRM calculated from t or rather dRM.Fromt:  $var.dRM.Fromt2 = (df/(df-2)) * ((N1+N2)/(2*N1*N2) + dRM.Fromt^2) - dRM.Fromt^2/c^2$  # unstandardizedES.ipe - estimated unstandardized technique effect size # var.sig.ipe - sum of within-subjects variance and between-subjects variance # var.within.ipe - within-subjects variance # var.between.ipe - between-subjects variance # t.ipe - t-value # var.t.ipe - variance of t-variable

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
simulationData <- getSimulationData(25, 18.75, 50, 10, 5, 500) # generate simulated data set
es.ipe <- getEffectSizesABBAIgnoringPeriodEffect(simulationData) # return effect sizes and variances
```

---

```
getSimulationData      getSimulationData
```

---

**Description**

Function to generate the simulated data set used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham

**Usage**

```
getSimulationData(
  var,
  covar,
  meanA1,
  treatmentDiff,
  periodEffect,
  numOfSamples
)
```

**Arguments**

var	Variance among subjects is a sum of the between subjects variance and the within subjects variance
covar	Covariance equal to the between subjects variance
meanA1	Mean for treatment sequence A1
treatmentDiff	technique effect which is the difference between the effect of technique A and technique B
periodEffect	Period effect which is the difference between period 1 and period 2
numOfSamples	Number of samples ('rows' of data) required for each technique and period

**Details**


---

----- Functions related to a paper 'Effect sizes and their variance for AB/BA crossover design studies' by Lech Madeyski and Barbara Kitchenham -----

---

**Value**

Data frame: 'data.frame': 4\*numOfSamples obs. of 5 variables: \$ pid : int 1 2 3 4 5 6 7 8 9 10 ... \$ technique: Factor w/ 2 levels 'T1','T2': ... \$ period : Factor w/ 2 levels 'P1','P2': ... \$ sequence : Factor w/ 2 levels 'S1','S2': ... \$ result : num ...

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
# generate the simulated data set from the paper
data <- getSimulationData(25, 18.75, 50, 10, 5, 500)
data <- getSimulationData(25, 18.75, 50, 10, 5, 15)
```

---

```
getTheoreticalEffectSizeVariancesABBA
  getTheoreticalEffectSizeVariancesABBA
```

---

**Description**

Function provides the theoretical value of the t-statistic, variance of t, and variance of the effect sizes based on the parameters built into crossover model data simulated by the getSimulationData() function. Function is used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

**Usage**

```
getTheoreticalEffectSizeVariancesABBA(
  theoreticalvarW,
  theoreticalTechniqueEffect,
  theoreticalrho,
  N1,
  N2
)
```

**Arguments**

theoreticalvarW - The within subject variance used to construct the simulation, i.e., the built-in Variance - the built-in Covariance

theoreticalTechniqueEffect - The technique effect built into the crossover model data

theoreticalrho - The between subject correlation built into the crossover model simulation data

N1 - The number of subjects in sequence group 1 in the crossover model simulation

N2 - The number of subjects in sequence group 2 in the crossover model simulation

**Value**

data frame incl. calculated: theoreticalt - the theoretical value of the t-statistic theoreticalvar - variance of t theoreticalvardIG - variance of the effect size dIG based on the parameters built into crossover model data simulated by the getSimulationData function theoreticalvardRM - variance of the effect size dRM based on the parameters built into crossover model data simulated by the getSimulationData function

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
# Generates data used in Table 15 of the paper
theoreticalEffectSizeVariances <- getTheoreticalEffectSizeVariancesABBA(6.25, -10, 0.75, 15, 15)
```

---

KitchenhamEtAl.CorrelationsAmongParticipants.Abrahao13TSE

*KitchenhamEtAl.CorrelationsAmongParticipants.Abrahao13TSE*  
*data*

---

**Description**

Data illustrate correlations between results from individual participants in a family of five crossover experiments conducted by Abrahao et al: [1] S. Abrahao, C. Gravino, E. Insfran Pelozo, G. Scanniello, and G. Tortora, 'Assessing the effectiveness of sequence diagrams in the comprehension of functional requirements: Results from a family of five experiments,' IEEE Transactions on Software Engineering, vol. 39, no. 3, pp. 327–342, March 2013 The five experiments assess whether the comprehensibility of function requirements improve when software models include UML sequence diagrams. If you use this data set please cite: [1] S. Abrahao, C. Gravino, E. Insfran Pelozo, G. Scanniello, and G. Tortora, 'Assessing the effectiveness of sequence diagrams in the comprehension of functional requirements: Results from a family of five experiments,' IEEE Transactions on Software Engineering, vol. 39, no. 3, pp. 327–342, March 2013 [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

**Usage**

KitchenhamEtAl.CorrelationsAmongParticipants.Abrahao13TSE

**Format**

A data frame with 224 rows and 8 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the five experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A (DM-NODM,ECP-EPlat or MShop-Theatre ), B (NODM-DM,ECP-EPlat or MShop-Theatre ), C(DM-NODM,EPlat-ECP or Theatre-MShop), D(NODM-DM,EPlat-ECP or Theatre-MShop)

**System** <fct>|Software systems used in the experiment: ECP an e-commerce platform from which CDs and books can be bought, EPlat a system for the management of courses, lectures and students of a university, M-Shop a system for managing sales at a music shop, Theatre a system for managing bookings for a theatre.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: A Dynamic Model (DM) vs No Dynamic Model (NODM)

**Comprehension** <dbl>|Dependent variable: The comprehension level the software engineer achieved based on the F-measure

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For a 2 group crossover, the category is set to CO1 only

**Ability** <fct>|Ability: An assessment of the ability of participants: Low, High, NA (not available)

## Examples

KitchenhamEtAl.CorrelationsAmongParticipants.Abrahao13TSE

---

KitchenhamEtAl.CorrelationsAmongParticipants.Gravino15JVLC

*KitchenhamEtAl.CorrelationsAmongParticipants.Gravino15JVLC  
data*

---

## Description

Data illustrate correlations between results from individual participants in a family of two cross-over experiments conducted by Gravino et al.: [1] C. Gravino, G. Scanniello, and G. Tortora, 'Source-code comprehension tasks supported by UML design models: Results from a controlled experiment and a differentiated replication,' *Journal of Visual Languages and Computing*, vol. 28, pp. 23–38, 2015. The experiments assess whether the comprehension of object oriented source-code increases used with UML class and sequence diagrams produced in the software design phase. If you use this data set please cite: [1] C. Gravino, G. Scanniello, and G. Tortora, 'Source-code comprehension tasks supported by UML design models: Results from a controlled experiment and a differentiated replication,' *Journal of Visual Languages and Computing*, vol. 28, pp. 23–38, 2015. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

## Usage

KitchenhamEtAl.CorrelationsAmongParticipants.Gravino15JVLC

**Format**

A data frame with 64 rows and 9 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the three experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B , C, D

**System** <fct>|Software systems used in the experiment: Music shop, a system for handling the sales of a music shop. Theater ticket, a system for managing theatre reservations.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: Mo, design models were available, NOMo design models were not available

**Comprehension** <dbl>|Dependent variable: The level of comprehension achieved by the software engineer.

**Time** <dbl>|Dependent variable: The time [min] taken to complete the comprehension task.

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Gravino15JVLC

---

KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10

*KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10 data*

---

**Description**

Data illustrate correlations between results from individual participants in cross-over experiment P2007 (Smell and Library) conducted by Madeyski, see: [1] Lech Madeyski, Test-Driven Development: An Empirical Evaluation of Agile Practice. (Heidelberg, London, New York): Springer, 2010. Foreword by Prof. Claes Wohlin. If you use this data set please cite: [1] Lech Madeyski, Test-Driven Development: An Empirical Evaluation of Agile Practice. (Heidelberg, London, New York): Springer, 2010. Foreword by Prof. Claes Wohlin. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

**Usage**

KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10

**Format**

'KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10': a data frame with 45 rows and 10 variables:

**ExperimentID** <fct> | ExperimentID: This experiment is the only cross-over experiment in the family of TDD and Pair-Programming experiments conducted by Madeyski, so all values in this column are set to 'P2007'.

**ParticipantID** <fct> | Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct> | Experimental Sequence Group: A (TLSP-TFSP), B (TFSP-TLSP)

**System** <fct> | Software system to develop: Smell (a tool for identifying bad code smells in Java source code through the use of a set of software metrics) or Library (a library application)

**Period** <fct> | Time period of the cross-over experiment: 1 or 2

**Treatment** <fct> | Experimental Treatment: Test-First Solo Programming (TFSP) vs Test-Last Solo Programming (TLSP)

**PATP** <dbl> | Dependent variable: Percentage of Acceptance Tests Passed

**NATPPH** <dbl> | Dependent variable: Number of Acceptance Tests Passed Per Hour

**CBO** <dbl> | Dependent variable: Mean value of Coupling Between Objects (CBO), see CK set of metrics

**WMC** <dbl> | Dependent variable: Mean value of Weighted Number of Methods in Class (WMC), see CK set of metrics

**RFC** <dbl> | Dependent variable: Mean value of Response For a Class (RFC), see CK set of metrics

**CrossOverID** <fct> | Cross-Over Code. This experiment is a simple two-group cross-over experiment with one cross-over code, so all values in this column are set to 'CO1'. However, four-group experiments require a code to identify the linked sequence groups (although that can be deduced from the system used in the first time period). A crossover code is also essential for non-parametric analysis.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10

---

KitchenhamEtAl.CorrelationsAmongParticipants.Reggio15SSM

*KitchenhamEtAl.CorrelationsAmongParticipants.Reggio15SSM data*

---

### Description

Data illustrate correlations between results from individual participants in a family of two cross-over experiments conducted by Reggio et al: [1] G. Reggio, F. Ricca, G. Scanniello, F. D. Cerbo, and G. Doderò, 'On the comprehension of workflows modeled with a precise style: results from a family of controlled experiments'. *Software and Systems Modeling*, vol. 14, pp. 1481–1504, 2015. The experiments assess whether the level of formality/precision in workflow model influences comprehension. If you use this data set please cite: [1] G. Reggio, F. Ricca, G. Scanniello, F. D. Cerbo, and G. Doderò, 'On the comprehension of workflows modeled with a precise style: results from a family of controlled experiments'. *Software and Systems Modeling*, vol. 14, pp. 1481–1504, 2015. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The Importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

### Usage

KitchenhamEtAl.CorrelationsAmongParticipants.Reggio15SSM

### Format

A data frame with 78 rows and 9 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the three experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B , C, D

**System** <fct>|Software systems used in the experiment: PO, a system to process orders for an online shop. DM, a system to manage an online document review process.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment:

**Comprehension** <dbl>|Dependent variable: The comprehension level obtained by each participant.

**Time** <dbl>|Dependent variable: The time [min] taken by each participant to complete the comprehension task.

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For a 2 group crossover, the category is set to CO1 only

### Examples

KitchenhamEtAl.CorrelationsAmongParticipants.Reggio15SSM

---

 KitchenhamEtAl.CorrelationsAmongParticipants.Ricca10TSE

*KitchenhamEtAl.CorrelationsAmongParticipants.Ricca10TSE data*


---

## Description

Data illustrate correlations between results from individual participants in a family of four cross-over experiments conducted by Ricca et al.: [1] F. Ricca, M. D. Penta, M. Torchiano, P. Tonella, and M. Ceccato 'How developers' experience and ability influence web application comprehension tasks supported by uml stereotypes: A series of four experiments', IEEE Transactions on Software Engineering, vol. 36, no. 1, pp. 96-118, 2010. Although we present the full data set, only the first two experiments were used in the correlation study, because many of the observations in the final two studies were unpaired. The experiments assess whether participants performance comprehension tasks better when using source code complemented by standard UML diagrams (UML) or by diagrams stereotyped using the Conallen notation (Conallen). If you use this data set please cite: [1] F. Ricca, M. D. Penta, M. Torchiano, P. Tonella, and M. Ceccato 'How developers' experience and ability influence web application comprehension tasks supported by uml stereotypes: A series of four experiments', IEEE Transactions on Software Engineering, vol. 36, no. 1, pp. 96—118, 2010. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The Importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

## Usage

KitchenhamEtAl.CorrelationsAmongParticipants.Ricca10TSE

## Format

A data frame with 176 rows and 10 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the four experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B , C, D

**System** <fct>|Software systems used in the experiment: Two Java-based Web applications, Claros and WfMS

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: UML or Conallon

**FMeasure** <dbl>|Dependent variable: The comprehension level achieved by the participant.

**Time** <dbl>|Dependent variable: The time [min] to complete the experimental task

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only

**Ability** <fct>| h: High l: Low, NA: Not available

**Experience** <fct>| G: Master students, U: undergraduates, P: researchers

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Ricca10TSE

---

KitchenhamEtAl.CorrelationsAmongParticipants.Ricca14TOSEM

*KitchenhamEtAl.CorrelationsAmongParticipants.Ricca14TOSEM  
data*

---

**Description**

Data illustrate correlations between results from individual participants in a family of three of four cross-over experiments conducted by Ricca et al: [1] F. Ricca, G. Scanniello, M. Torchiano, G. Reggio, and E. Astesiano, 'Assessing the effect of screen mockups on the comprehension of functional requirements,' ACM Transactions on Software Engineering and Methodology, vol. 24, no. 1, pp. 1:1–1:38, Oct. 2014. The goal of the study was to assess whether stakeholders benefit from the presence of screen mock-ups in the comprehension of functional requirements represented with use cases. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

**Usage**

KitchenhamEtAl.CorrelationsAmongParticipants.Ricca14TOSEM

**Format**

A data frame with 176 rows and 10 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the three experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B , C, D

**System** <fct>|Software systems used in the experiment: AMICO, a system for management of condominiums. EasyCoin, a system for cataloguing collections of coins.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: Screen mockup available (S) vs Text only (T)

**Time** <dbl>|Dependent variable: The time [min] taken to perform the software engineering task.

**Comprehension** <dbl>|Dependent variable: The comprehension level the software engineers.

**Efficiency** <dbl>|Dependent variable: The ratio of comprehension to time.

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For a 2 group crossover, the category is set to CO1 only.

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Ricca14TOSEM

---

KitchenhamEtAl.CorrelationsAmongParticipants.Romano18ESEM

*KitchenhamEtAl.CorrelationsAmongParticipants.Romano18ESEM  
data*

---

**Description**

Data illustrate correlations between results from individual participants in a cross-over experiment conducted by Romano et al.: [1] S. Romano, G. Scanniello, D. Fucci, N. Juristo, and B. Turhan, 'The effect of noise on software engineers' performance', in Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ser. ESEM' 18, 2018. The experiments assess whether noise has an impact on the performance of software engineers. If you use this data set please cite: [1] S. Romano, G. Scanniello, D. Fucci, N. Juristo, and B. Turhan, 'The effect of noise on software engineers' performance', in Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ser. ESEM' 18, 2018. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The Importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020). The experiment had two parts but Kitchenham et al. only use the data from the first part of the experiment.

**Usage**

KitchenhamEtAl.CorrelationsAmongParticipants.Romano18ESEM

**Format**

A data frame with 194 and 10 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each part of the experiment. Exp.1 identifies data from the first part of the experiment, Exp.2 identifies data from the second part of the experiment.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for both parts of the experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B

**System** <fct>|Software systems used in the experiment: For the first part of the experiment, M-Shop (a system for managing a music shop) and Theater (a system for managing theatre reservations). For the second part of the experiment: AveCalc (a system that manages as electronic register and LaTazza (a system for a drinks vending machine)

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: NOISE, participants were asked to perform a comprehension task in a noisy environment. NORMAL, participants were asked to perform a comprehension task under normal working conditions.

**Fc** <dbl>|Dependent variable: the balanced F-measure which represents the trade-off between precision and recall, measured in the first part of the experiment.

**Avg** <dbl>|Dependent variable: The average number of fully correct answers, measured in the first part of the experiment.

**Ff** <dbl>|Dependent variable: Effectiveness of fault correction. Measured in the second part of the experiment.

**CrossOverID** <fct>|CrossOver category: For 4 group crossovers, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only.

## Examples

KitchenhamEtAl.CorrelationsAmongParticipants.Romano18ESEM

---

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14EASE

*KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14EASE  
data*

---

## Description

Data illustrate correlations between results from individual participants in a family of two cross-over experiments conducted by Scanniello et al: [1] G. Scanniello, M. Staron, H. Burden, and R. Heldal, 'On the effect of using SysML requirement diagrams to comprehend requirements: results from two controlled experiments,' in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE. ACM, 2014. The two experiments investigate whether requirements specified as SysML requirement diagrams improve the comprehensibility of requirements. If you use this data set please cite: [1] G. Scanniello, M. Staron, H. Burden, and R. Heldal, 'On the effect of using SysML requirement diagrams to comprehend requirements: results from two controlled experiments', in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE. ACM, 2014. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

## Usage

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14EASE

## Format

A data frame with 174 rows and 9 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each experiment in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A (RD-NORD,Automobile-ESS), B (NORD-RD,ESS-Automobile), C(NORD-RD,Automobile-ESS), D(RD-NORD,ESS-Automobile).

**System** <fct>|Software systems used in the experiment: Automobile: A system for controlling car behavior with use cases about entering the car, anti-lock breaking or operating the climate control of a car. ESS (Enhanced Security System) a system designed to detect potential home intruders.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: RD availability of a SysML requirements diagram vs No requirements diagram (NORD)

**Time** <dbl>|Dependent variable: The time [min] required for the comprehension task.

**Comprehension** <dbl>|Dependent variable: The comprehension level the software engineer achieved.

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only

## Examples

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14EASE

---

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14JVLC

*KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14JVLC  
data*

---

## Description

Data illustrate correlations between results from individual participants in a cross-over experiment conducted by Scanniello and Erra: [1] G. Scanniello and U. Erra, 'Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments', *Journal of Visual Languages and Computing*, vol. 25, no. 4, pp. 494–517, 2014. The experiment investigated whether a new method based on think-pair-square and its implementation in an integrated communication/modeling environment (TPS approach) is as effective as traditional face-to-face (F2F approach) for requirements elicitation. The experiment was performed in two stages using different software systems. If you use this data set please cite: [1] G. Scanniello and U. Erra, 'Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments,' *Journal of Visual Languages and Computing*, vol. 25, no. 4, pp. 494–517, 2014. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The Importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

## Usage

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14JVLC

**Format**

A data frame with 36 rows and 12 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each experiment in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each team of four participants, unique for the specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B

**System** <fct>|Software systems used in the experiment: Library (a software system to manage books and users of a library) and FilmCollection (a software system for the selling and the rental of films in a shop) in ExperimentStage1 and Rent (a car rental software to manage cars, customers, and reservations) and ECP (an E-Commerce Platform to order CDs and books via the Internet from an on line catalogue), in ExperimentStage2.

**Treatment** <fct>|Experimental Treatment: TPS vs F2F.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2 within each stage of the experiment

**Time** <dbl>|Dependent variable: The total time [min] to accomplish the requirement engineering task.

**Quality** <dbl>|Dependent variable: The quality of the requirements engineering task.

**CrossOverID** <fct>|Crossover category: For a single 2 group crossover experiment, the value is set to CO1 for each experiment stage.

**ExperimentPeriod** <fct>|ExperimentPeriod: The time period across both stages of the experiment.

**ExperimentStage** <fct>|ExperimentStage: 1 first stage, 2 second stage.

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14JVLC

---

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM

*KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM  
data*

---

**Description**

Data illustrate correlations between results from individual participants in a family of four cross-over experiments conducted by Scanniello et al: [1] G. Scanniello, C. Gravino, M. Genero, J.A. Cruz-Lemus, and G. Tortora, 'On the Impact of UML Analysis Models on Source-Code Comprehensibility and Modifiability', ACM Transactions on Software Engineering and Methodology, vol. 23, no. 2, pp. 13:1-13:26, 2014 The family of experiments investigated whether the availability of analysis models in addition to the source code made the code easier to understand and modify. If you use this data set please cite: [1] G. G. Scanniello, C. Gravino, M. Genero, J.A. Cruz-Lemus, and G. Tortora, 'On the Impact of UML Analysis Models on Source-Code Comprehensibility and Modifiability', ACM Transactions on Software Engineering and Methodology, vol. 23, no. 2, pp. 13:1-13:26, 2014 [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

**Usage**

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM

**Format**

'KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM': a data frame with 172 rows and 9 variables:

**ExperimentID** <fct> | ExperimentID: A unique identifier for each experiment in the data set.

**ParticipantID** <fct> | Participant ID: An identifier for each participant, unique for a specific experiment.

**Treatment** <fct> | Experimental Treatment: AM an Analysis Model with source code (AM) vs Source Code only (SC)

**SequenceGroup** <fct> | Experimental Sequence Group: A (AM-SC,S1-S2), B (SC-AM,S1-S2), C(AM-SC,S2-S1), D(SC-AM,S2-S1)

**System** <fct> | Software systems used in the experiment: S1 A system to sell and manage CDs/DVDs in a music shop, S2 A system to book and by theater tickets.

**Comprehension** <dbl> | Dependent variable: The comprehension level the software engineer achieved based on the F-measure

**Modification** <dbl> | Dependent variable: The modifiability level the software engineer achieved based on the F-measure

**Period** <fct> | Time period of the cross-over experiment: 1 or 2

**CrossOverID** <fct> | CrossOver category: For 4 group the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14TOSEM

---

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE

*KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE  
data*

---

**Description**

Data illustrate correlations between results from individual participants in cross-over experiment usb2 conducted by Scanniello et al: [1] G. Scanniello, A. Marcus, and D. Pascale, 'Link analysis algorithms for static concept location: an empirical assessment', Empirical Software Engineering, vol. 20, no. 6, pp. 1666–1720, 2015. The goal of the experiment is to assess whether a new technique (implemented as an Eclipse plug-in) for static concept location (proposed by the authors) supports users in identifying the places in the code where changes are to be made.

**Usage**

`KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE`

**Format**

‘`KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE`’: a data frame with 48 rows and 10 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each experiment in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A (CL-NOCL,Jedit-Atunes), B (NOCL-CL,Atunes-Jedit), C(NOCL-CL,Jedit-Atunes), D(CL-NOCL,Atunes-Jedit)

**System** <fct>|Software systems used in the experiment: Jedit and Atunes

**Treatment** <fct>|Experimental Treatment: Use of Concept Location plug-in (CL) vs no Concept Location plug-in (NOCL)

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Correctness** <int>|Dependent variable: 0, 1, 2, 3, 4. The participants are asked to indicate a single change method for each of 4 bug reports. A change method is correctly identified if that method is in the change set of the bug report.

**Time** <dbl>|Dependent variable: The total time [min] to accomplish concept location tasks, i.e.,to identify (four) bugs given their reports

**Efficiency** <dbl>|Dependent variable: The participants’ efficiency in the execution of concept location tasks. It is computed dividing correctness by time.

**CrossOverID** <fct>|Crossover category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2.

**Details**

If you use this data set please cite: [1] G. Scanniello, A. Marcus, and D. Pascale, ‘Link analysis algorithms for static concept location: an empirical assessment’, *Empirical Software Engineering*, vol. 20, no. 6, pp. 1666–1720, 2015. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, ‘The importance of the Correlation between Results from Individual Participants in Crossover Experiments’ (to be submitted as of 2020).

**Examples**

`KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15EMSE`

---

 KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello17TOSEM

*KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello17TOSEM  
data*


---

## Description

Data illustrate correlations between results from individual participants in a family of four cross-over experiments conducted by Scanniello et al.: [1] G. Scanniello, M. Risi, P. Tramontana, and S. Romano, 'Fixing faults in C and Java source code: Abbreviated vs. full-word identifier names', ACM Transactions on Software Engineering Methodology, vol. 26, no. 2, 2017. The experiments assess whether the use of abbreviated identifier names (ABBR), impacts the effectiveness of fault fixing in C and Java source code in comparison with full-word identifier names (FULL). If you use this data set please cite: [1] G. Scanniello, M. Risi, P. Tramontana, and S. Romano, "Fixing faults in C and Java source code: Abbreviated vs. full-word identifier names", ACM Transactions on Software Engineering Methodology, vol. 26, no. 2, 2017. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'On the Importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

## Usage

KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello17TOSEM

## Format

A data frame with 200 rows and 17 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B , C, D

**System** <fct>|Software systems used in the experiments: The Unibas experiment used Agenda (a system for tracking personal contacts) and Gas-Station (a system for managing a petrol station). The UniNa experiment used Financial (a system which is a command line option price calculator) and Hotel-Reservation. The POLINA and PROF experiments used AveCalc (a system that manages as electronic register and LaTazza (a system for a drinks vending machine).

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment:ABBR, abbreviated names. FULL, full names

**Time** <dbl>|Dependent variable: The time each participant spent performing the SE task.

**FMeasure** <dbl>|Dependent variable: The effectiveness of the participants taking into account correctness and completeness of the fault fixing tasks

**Efficiency** <dbl>|Dependent variable: The ratio of effectiveness to time.

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only.

**Language** <fct>|Java or C. The language was the same for all participants in a specific experiment. POLINA and PROF used Java, UNIBAS and UNINA used C.

**Ident** <dbl>|Dependent variable: The number of faults identified.

**Fixed** <dbl>|Dependent variable: The number of faults identified.

**WrongIdent** <dbl>|Dependent variable: The number of faults incorrectly identified

**WronglyFixed** <dbl>|Dependent variable: The number of faults incorrectly fixed.

**precision** <dbl>|Dependent variable: The ratio of number of faults correctly fixed to the number of faults correctly identified.

**recall** <dbl>|Dependent variable: The ratio of number of faults correctly fixed to the total number of fault.

## Examples

`KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello17T0SEM`

---

`KitchenhamEtAl.CorrelationsAmongParticipants.Torchiano17JVLC`

*KitchenhamEtAl.CorrelationsAmongParticipants.Torchiano17JVLC*  
*data*

---

## Description

Data illustrate correlations between results from individual participants in a family of three crossover experiments conducted by Torchiano et al: [1] M. Torchiano, G. Scanniello, F. Ricca, G. Reggio, and M. Leotta, 'Do UML object diagrams affect design comprehensibility? Results from a family of four controlled experiments.' *Journal of Visual Languages and Computing*, vol. 41, pp. 10–21, 2017. Although the paper reports four experiment, we only have data from three of those experiments. The experiments assess whether the comprehensibility of UML specifications improve when the software documents include UML object diagrams as well as the standard UML class diagrams. If you use this data set please cite: [1] M. Torchiano, G. Scanniello, F. Ricca, G. Reggio, and M. Leotta, 'Do UML object diagrams affect design comprehensibility? Results from a family of four controlled experiments.' *Journal of Visual Languages and Computing*, vol. 41, pp. 10–21, 2017. [2] Barbara Kitchenham, Lech Madeyski, Giuseppe Scanniello and Carmine Gravino, 'The importance of the Correlation between Results from Individual Participants in Crossover Experiments' (to be submitted as of 2020).

## Usage

`KitchenhamEtAl.CorrelationsAmongParticipants.Torchiano17JVLC`

**Format**

A data frame with 214 rows and 8 variables:

**ExperimentID** <fct>|ExperimentID: A unique identifier for each of the three experiments in the data set.

**ParticipantID** <fct>|Participant ID: An identifier for each participant, unique for a specific experiment.

**SequenceGroup** <fct>|Experimental Sequence Group: A , B , C, D

**System** <fct>|Software systems used in the experiment: File System manager (FS) for folders, files, links. Roads system (R) handles maps made up of cities connected by means of roads. Train (T) a system to manage timetables, trains, and paths. Catalogue system (C). It collects category of items (e.g., cars) and items (e.g., car models) based on a set of features (e.g., number of doors). In PoliTo2, only FS and T were administered to the participants, while in UniBas1 and UniGe1 all the four experimental objects were used.

**Period** <fct>|Time period of the cross-over experiment: 1 or 2

**Treatment** <fct>|Experimental Treatment: Object Diagram (OD) vs No Object Diagram (NoOD)

**Comprehension** <dbl>|Dependent variable: The comprehension level the software engineers. For PoliTo2 Comprehension was based on answering a set of 4 questions, for UniBas and UniGe comprehension was measured using the F metric.

**CrossOverID** <fct>|CrossOver category: For 4 group crossover designs, the crossover category specifies the matching pairs of sequence groups, CO1 and CO2. For 2 group crossover, the category is set to CO1 only

**Examples**

KitchenhamEtAl.CorrelationsAmongParticipants.Torchiano17JVL

---

KitchenhamMadeyski.SimulatedCrossoverDataSets

*KitchenhamMadeyski.SimulatedCrossoverDataSets data*

---

**Description**

If you use this data set please cite this R package and the following paper: Lech Madeyski and Barbara Kitchenham, 'Effect Sizes and their Variance for AB/BA Crossover Design Studies', Empirical Software Engineering, vol. 24, no.4, p. 1982-2017, 2018. DOI: 10.1007/s10664-017-9574-5

**Usage**

KitchenhamMadeyski.SimulatedCrossoverDataSets

**Format**

A data frame with variables:

**actualSampleSize** Sample size

**SSFull** Sample Size

**CFull** Correlation

**ESFull** Effect Size

**Accuracy** Accuracy

**PropSig** ...

**WrongTSig** ...

**Details**

This is simulated normally distributed data from 30 subjects, with technique A being 10 units more effective than technique B, and there is a period effect equaling 5 units. Subject 1 to 15 used technique B first while subjects 16 to 30 used technique A first.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

KitchenhamMadeyski.SimulatedCrossoverDataSets

---

KitchenhamMadeyskiBrereton.ABBAMetaAnalysisReportedResults

*KitchenhamMadeyskiBrereton.ABBAMetaAnalysisReportedResults  
data*

---

**Description**

This data is used in the paper: Barbara Kitchenham, Lech Madeyski and Pearl Brereton. Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment, Empirical Software Engineering (2019) doi:10.1007/s10664-019-09747-0. This data set reports the meta-analysis results reported by the authors of the primary studies included in the systematic review that reported results on a per document basis which for S7 and S11 was equivalent to reporting the results for each time period.

**Usage**

KitchenhamMadeyskiBrereton.ABBAMetaAnalysisReportedResults

**Format**

A text file with variables:

**Study** This field includes the study identifier of each of the the 3 primary studies which reported results per document.

**Type** This identifies the type of effect size used by the study authors. d or g refer to d\_IG and g\_IG, P is the aggregated p values, if the repeated measures (RM) estimate was obtained it is appropriately specified.

**Source** Always set to Rep. This identifies that the data was as reported by the primary study authors.

**mean** The overall mean effect size reported by the study authors

**pvalue** The one-sided p-value associated with the overall mean reported by the study authors. NA means the authors did not report this statistic.

**UB** The upper bound of the confidence interval of the overall mean as reported by the primary study authors. NA means the authors did not report this statistic.

**LB** The lower bound of the confidence interval of the overall mean as reported by the primary study authors. NA means the authors did not report this statistic.

**QE** The heterogeneity statistic associated with the meta-analysis as reported by the study authors. NA means the authors did not report this statistic.

**Qep** The p-value of the heterogeneity statistic associated with the meta-analysis as reported by the study authors. NA means the authors did not report this statistic.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

KitchenhamMadeyskiBrereton.ABBAMetaAnalysisReportedResults

---

KitchenhamMadeyskiBrereton.ABBAReportedEffectSizes

*KitchenhamMadeyskiBrereton.ABBAReportedEffectSizes data*

---

**Description**

This data is used in the paper: Barbara Kitchenham, Lech Madeyski and Pearl Brereton. Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment, Empirical Software Engineering (2019) doi:10.1007/s10664-019-09747-0. This file holds the individual effect sizes for the first time period (or equivalently the first document), as reported by the 3 primary studies in the systematic review that reported results for each document/time period separately.

**Usage**

KitchenhamMadeyskiBrereton.ABBAReportedEffectSizes

**Format**

A text file with variables:

**Study** This field includes the study identifier of each of the 3 primary studies which were included in the systematic review. The studies are S3, S7 and S11.

**Type** This identifies the type of effect size used by the study authors. d or g refer to dIG and gIG.

**Source** Always set to Rep. This identifies that the data was as reported by the primary study authors.

**Design** Mixed means different experiments in a particular family used different methods (only S3 used mixed methods and 4 experiments used the 4 group crossover and one used an independent groups design). ABBACO is the standard 2-group crossover design.

**Exp1** This is the reported standardised effect size for the first time period and the first experiment in the family.

**Exp2** This is the reported standardised effect size for the first time period and second experiment in the family.

**Exp3** This is the reported standardised effect size for the first time period and the third experiment in the family.

**Exp4** This is the reported standardised effect size for the first time period and the fourth experiment in the family. NA means there was no fourth experiment in the family.

**Exp5** This is the reported standardised effect size for the first time period and the fifth experiment in the family. NA means there was no fifth experiment in the family.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

KitchenhamMadeyskiBrereton.ABBAReportedEffectSizes

---

KitchenhamMadeyskiBrereton.DocData

*KitchenhamMadeyskiBrereton.DocData data*

---

**Description**

This data is used in the paper: Barbara Kitchenham, Lech Madeyski and Pearl Brereton. Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment, Empirical Software Engineering (2019) doi:10.1007/s10664-019-09747-0. This file holds the descriptive data for each document and each experiment for studies 3, 7 and 11 which include the mean, standard deviation and sample size for the control and treatment techniques. These studies performed ABBA crossover experiments and reported data for each document separately. Note Study 3 also undertook an independent groups study but data from that experiment is held in the ExpData file.

**Usage**

KitchenhamMadeyskiBrereton.DocData

**Format**

A text file with variables:

**Study** This field includes the study identifier of each of the 3 primary studies which reported their basic statistics on a time period & document basis.

**Exp** This identifies the experiment to which the descriptive data belongs.

**Doc** This identifies whether the data arose from the document used in the first or second time period. The value 'Doc1' identifies the data as coming from the first document or first time period. The value 'Doc2' identifies the data as coming from the second time period or document. Note for Study 3 we used the analysis of a specific document that was used in all 4 ABBA experiments. For studies 7 and 11, the authors identified which we used in r=each time period and Doc1 refers to data from the first time period.

**Mc** The mean value of the observations obtained using the control technique for the identified document.

**SDc** The standard deviation of the observations obtained using the control technique for the identified document.

**Nc** The number of participants using the control technique in the first time period for the identified document.

**Mt** The mean value of the observations obtained using the treatment technique for the identified document.

**SDt** The standard deviation of the observations obtained using the treatment technique for the identified document.

**Nt** The number of participants using the treatment technique in the first time period for the identified document.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

KitchenhamMadeyskiBrereton.DocData

---

KitchenhamMadeyskiBrereton.ExpData

*KitchenhamMadeyskiBrereton.ExpData data*

---

## Description

This data is used in the paper: Barbara Kitchenham, Lech Madeyski and Pearl Brereton. Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment, Empirical Software Engineering (2019) doi:10.1007/s10664-019-09747-0. This file holds the descriptive data for each experiment which include the mean, standard deviation and sample size for the control and treatment techniques. Note in the case of studies 3, 7 and 11, which reported descriptive data for each time period (or equivalently each document) separately, the values for of the descriptive data were obtained by analysing the data reported in the DocData file.

## Usage

KitchenhamMadeyskiBrereton.ExpData

## Format

A text file with variables:

**Study** This field includes the study identifier of each of the 13 primary studies which were included in the systematic review.

**Exp** This identifies the experiment to which the descriptive data belongs.

**Source** Always set to Rep. This identifies that the data was as reported by the primary study authors.

**Mc** The mean value of the observations obtained using the control technique.

**SDc** The standard deviation of the observations obtained using the control technique.

**Nc** The number of participants using the control technique in the first time period.

**Mt** The mean value of the observations obtained using the treatment technique.

**SDt** The standard deviation of the observations obtained using the treatment technique.

**Nt** The number of participants using the treatment technique in the first time period.

**r** The correlation between repeated measures. NA if not reported. Note only study 13 reported this correlation.

## Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

## Examples

KitchenhamMadeyskiBrereton.ExpData

---

KitchenhamMadeyskiBrereton.MetaAnalysisReportedResults

*KitchenhamMadeyskiBrereton.MetaAnalysisReportedResults data*

---

### Description

This data is used in the paper: Barbara Kitchenham, Lech Madeyski and Pearl Brereton. Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment (to be submitted). This data set reports the meta-analysis results reported by the authors of the 13 primary studies included in the systematic review.

### Usage

KitchenhamMadeyskiBrereton.MetaAnalysisReportedResults

### Format

A text file file with variables:

**Study** This field includes the study identifier of each of the 13 primary studies which were included in the systematic review.

**Type** This identifies the type of effect size used by the study authors. d or g refer to d<sub>IG</sub> and g<sub>IG</sub>, P is the aggregated p values, if the repeated measures estimate was obtained it is appropriately specified, r refers to the point bi-serial correlation.

**Source** Always set to Rep. This identifies that the data was as reported by the primary study authors.

**mean** The overall mean effect size reported by the study authors

**pvalue** The one-sided p-value associated with the overall mean reported by the study authors. NA means the authors did not report this statistic.

**UB** The upper bound of the confidence interval of the overall mean as reported by the primary study authors. NA means the authors did not report this statistic.

**LB** The lower bound of the confidence interval of the overall mean as reported by the primary study authors. NA means the authors did not report this statistic.

**QE** The heterogeneity statistic associated with the meta-analysis as reported by the study authors. NA means the authors did not report this statistic.

**Qep** The p-value of the heterogeneity statistic associated with the meta-analysis as reported by the study authors. NA means the authors did not report this statistic.

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

KitchenhamMadeyskiBrereton.MetaAnalysisReportedResults

---

KitchenhamMadeyskiBrereton.ReportedEffectSizes

*KitchenhamMadeyskiBrereton.ReportedEffectSizes data*

---

### Description

This data is used in the paper: Barbara Kitchenham, Lech Madeyski and Pearl Brereton. Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment, Empirical Software Engineering (2019) doi:10.1007/s10664-019-09747-0. This file holds the individual effect sizes for each experiment, as reported by 13 primary studies in the systematic review.

### Usage

KitchenhamMadeyskiBrereton.ReportedEffectSizes

### Format

A text file with variables:

**Study** This field includes the study identifier of each of the 13 primary studies which were included in the systematic review.

**Type** This identifies the type of effect size used by the study authors. d or g refer to dIG and gIG, p is the p-value used for aggregation, if the repeated measures estimate was obtained it is appropriately specified as gRM, r refers to the point bi-serial correlation.

**Source** Always set to Rep. This identifies that the data was as reported by the primary study authors.

**Design** This refers to the design method used by the study author. 4GroupCO is a 4-group crossover design. Mixed means different experiments in a particular family used different methods (only S3 used mixed methods and 4 experiments used the 4 group crossover and one used an independent groups design). ABBACO is the standard 2-group crossover design. IndGroups is the independent groups design also called between groups design or a randomised design. PrePost is pretest and posttest design with a post test control.

**Exp1** This is the reported standardized effect size for the first experiment in the family.

**Exp2** This is the reported standardized effect size for the second experiment in the family.

**Exp3** This is the reported standardized effect size for the third experiment in the family.

**Exp4** This is the reported standardized effect size for the fourth experiment in the family. NA means there was no fourth experiment in the family.

**Exp5** This is the reported standardized effect size for the fifth experiment in the family. NA means there was no fifth experiment in the family.

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

KitchenhamMadeyskiBrereton.ReportedEffectSizes

---

KitchenhamMadeyskiBudgen16.COCOMO

*KitchenhamMadeyskiBudgen16.COCOMO data*

---

**Description**

If you use this data set please cite this R package and the following paper when accepted: Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong, 'Robust Statistical Methods for Empirical Software Engineering', Empirical Software Engineering, vol. 22, no.2, p. 579-630, 2017. DOI: 10.1007/s10664-016-9437-5 (<https://dx.doi.org/10.1007/s10664-016-9437-5>), URL: <https://madeyski.e-informatyka.pl/download/KitchenhamMadeyski>

**Usage**

KitchenhamMadeyskiBudgen16.COCOMO

**Format**

A data frame with variables:

**Project** Project ID

**Type** A categorical variable describing the type of the project

**Year** The year the project was completed

**Lang** A categorical variable describing the development language used

**Rely** Ordinal value defining the required software reliability

**Data** Ordinal value defining the data complexity / Data base size

**Cplx** Ordinal value defining the complexity of the software / Process complexity

**Aaf** ??

**Time** Ordinal value defining the stringency of timing constraints / Time constraint for cpu

**Stor** Ordinal value defining the stringency of the data storage requirements / Main memory constraint

**Virt** Virtual Machine volatility

**Turn** Turnaround time

**Type2** A categorical variable defining the hardware type: mini, max=mainframe, midi

**Acap** Ordinal value defining the analyst capability

**Aexp** Ordinal value defining the analyst experience / application experience

**Pcap** Ordinal value defining the programming capability of the team / Programmers capability

**Vexp** Ordinal value defining the virtual machine experience of the team

- Lexp** Ordinal value defining the programming language experience of the team
- Cont** ??
- Modp** / Modern programming practices
- Tool** Ordinal value defining the extent of tool use / Use of software tools
- ToolCat** Recoding of Tool to labelled ordinal scale
- Sced** Ordinal value defining the stringency of the schedule requirements / Schedule constraint
- Rvol** Ordinal value defining the requirements volatility of the project
- Select** Categorical value calculated by BAK for an analysis example
- Rvolcat** Recoding of Rvol to a labelled ordinal scale
- Modecat** Mode of the projects: O=Organic, E=Embedded, SD=Semi-Detached
- Mode1** Dummy variable calculated by BAK: 1 if the project is Organic, 0 otherwise
- Mode2** Dummy variable calculated by BAK: 1 if the project is Semi-detached, 0 otherwise
- Mode3** Dummy variable calculated by BAK: 1 if the project is Embedded, 0 otherwise
- KDSI** Product Size Thousand of Source Instructions
- AKDSI** Adjusted Product Size for Project in Thousand Source Instructions - differs from KDSI for enhancement projects
- Effort** Project Effort in Man months
- Duration** Duration in months
- Productivity** Productivity of project calculated by BAK as AKDSI/Effort, so the the larger the value the better the productivity

### Details

Data set collected at TRW by Barry Boehm see: B.W. Boehm. 1981. Software Engineering Economics. Prentice-Hall.

Explanations by Barbara Kitchenham / <https://terapromise.csc.ncsu.edu:8443/#!/repo/view/head/effort/cocomo/cocomo1/nas>

COCOMO.txt: pro type year Lang Rely Data CPLX aaf time store virt turn type2 acap aexp pcap  
vexp lexp cont modp TOOL TOOLcat SCED RVOL Select rvolcat Modecat Mode1 Mode2 Mode3  
KDSI AKDSI Effort Dur Productivity

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

KitchenhamMadeyskiBudgen16.COCOMO

---

 KitchenhamMadeyskiBudgen16.DiffInDiffData

*KitchenhamMadeyskiBudgen16.DiffInDiffData data*


---

## Description

If you use this data set please cite this R package and the following paper when accepted: Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong, 'Robust Statistical Methods for Empirical Software Engineering', Empirical Software Engineering, vol. 22, no.2, p. 579-630, 2017. DOI: 10.1007/s10664-016-9437-5 (<https://dx.doi.org/10.1007/s10664-016-9437-5>), URL: <https://madeyski.e-informatyka.pl/download/KitchenhamMadeyski>

## Usage

```
KitchenhamMadeyskiBudgen16.DiffInDiffData
```

## Format

A data frame with variables:

**Abstract** The abstract identifier

**Site** A numeric identifier of the site

**Treatment** A three character alphanumeric identifying the journal and time period of the abstract

**Journal** The journal in which the abstract was published: IST or JSS

**Timeperiod** The time period in which the abstract: 1 or 2

**J1** The identifier for the judge who made the next 2 assessments

**J1Completeness** The average completeness made by judge J1 based on the 8 completeness questions

**J1Clarity** The clarity assessment made by judge J1

**J2** The identifier for the judge who made the next 2 assessments

**J2Completeness** The average completeness made by judge J2 based on the 8 completeness questions

**J2Clarity** The clarity assessment made by judge J2

**J3** The identifier for the judge who made the next 2 assessments

**J3Completeness** The average completeness made by judge J3 based on the 8 completeness questions

**J3Clarity** The clarity assessment made by judge J3

**J4** The identifier for the judge who made the next 2 assessments

**J4Completeness** The average completeness made by judge J4 based on the 8 completeness questions

**J4Clarity** The clarity assessment made by judge J4

**MeanCompleteness** The mean of J1Completeness, J2Completeness, J3Completeness, J4Completeness

**MedianCompleteness** The median of J1Completeness, J2Completeness, J3Completeness, J4Completeness

**MedianClarity** The median clarity of J1Clarity, J2Clarity, J3Clarity, J4Clarity

**MeanClarity** The mean clarity of J1Clarity, J2Clarity, J3Clarity, J4Clarity

**VarCompleteness** The variance of J1Completeness, J2Completeness, J3Completeness, J4Completeness

**VarClarity** The variance clarity of J1Clarity, J2Clarity, J3Clarity, J4Clarity

### Details

Data set was derived from the data reported in the SubjectData data set (subjectdata.txt). It contains the summary completeness and clarity data from 4 judges who assessed the same abstract. Only the initial 5 sites are included.

dinddata.txt

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

KitchenhamMadeyskiBudgen16.DiffInDiffData

---

KitchenhamMadeyskiBudgen16.FINNISH

*KitchenhamMadeyskiBudgen16.FINNISH data*

---

### Description

If you use this data set please cite this R package and the following paper when accepted: Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong, 'Robust Statistical Methods for Empirical Software Engineering', Empirical Software Engineering, vol. 22, no.2, p. 579-630, 2017. DOI: 10.1007/s10664-016-9437-5 (<https://dx.doi.org/10.1007/s10664-016-9437-5>), URL: <https://madeyski.e-informatyka.pl/download/KitchenhamMadeyski>

### Usage

KitchenhamMadeyskiBudgen16.FINNISH

### Format

A data frame with variables:

**Project** Project ID

**DevEffort** Development Effort measured in hours

**UserEffort** Effort provided by the customer/user organisation measured in hours

**Duration** Project duration measured in months

- HWType** A categorical variable defining the hardware type
- AppType** A categorical variable defining the application type
- FP** Function Points measured using the TIEKE organisation method
- Co** A categorical variable defining the company

### Details

Data set collected from 9 Finish companies by Mr Hanna M'aki from the TIEKE organisation see Barbara Kitchenham and Kari Kansala, Inter-item correlations among function points, Proceedings ICSE 15, 1983, pp 477-480

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

KitchenhamMadeyskiBudgen16.FINNISH

---

KitchenhamMadeyskiBudgen16.PolishData

*KitchenhamMadeyskiBudgen16.PolishData data*

---

### Description

If you use this data set please cite this R package and the following paper when accepted: Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong, 'Robust Statistical Methods for Empirical Software Engineering', Empirical Software Engineering, vol. 22, no.2, p. 579-630, 2017. DOI: 10.1007/s10664-016-9437-5 (<https://dx.doi.org/10.1007/s10664-016-9437-5>), URL: <https://madeyski.e-informatyka.pl/download/KitchenhamMadeyski>

### Usage

KitchenhamMadeyskiBudgen16.PolishData

### Format

A data frame with variables:

**Abstract** The abstract identifier

**Site** Numeric identifier for the site

**Treatment** The first three characters of the Abstract field which identifies the journal and time period of the abstract

**Journal** An acronym for the journal from which the abstract was obtained: IST or JSS

**Timeperiod** The Time period in which the abstract was found: 1 or 2

**J1** The identifier for the judge who made the next 2 assessments

**J1Completeness** The average completeness made by judge J1 based on the 8 completeness questions

**J1Clarity** The clarity assessment made by judge J1

**J2** The identifier for the judge who made the next 2 assessments

**J2Completeness** The average completeness made by judge J2 based on the 8 completeness questions

**J2Clarity** The clarity assessment made by judge J2

**J3** The identifier for the judge who made the next 2 assessments

**J3Completeness** The average completeness made by judge J3 based on the 8 completeness questions

**J3Clarity** The clarity assessment made by judge J3

**J4** The identifier for the judge who made the next 2 assessments

**J4Completeness** The average completeness made by judge J4 based on the 8 completeness questions

**J4Clarity** The clarity assessment made by judge J4

**MedianCompleteness** The median of J1Completeness, J2Completeness, J3Completeness, J4Completeness

**MedianClarity** The median of J1Clarity, J2Clarity, J3Clarity, J4Clarity

## Details

Data set derived from PolishSubjects data set collected at Wroclaw University. It summarizes the completeness and clarity data collected from 4 judges about the same abstract.

PolishData.txt

## Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

## Examples

KitchenhamMadeyskiBudgen16.PolishData

---

KitchenhamMadeyskiBudgen16.PolishSubjects

*KitchenhamMadeyskiBudgen16.PolishSubjects data*

---

## Description

If you use this data set please cite this R package and the following paper when accepted: Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong, 'Robust Statistical Methods for Empirical Software Engineering', Empirical Software Engineering, vol. 22, no.2, p. 579-630, 2017. DOI: 10.1007/s10664-016-9437-5 (<https://dx.doi.org/10.1007/s10664-016-9437-5>), URL: <https://madeyski.e-informatyka.pl/download/KitchenhamMadeyski>

**Usage**

KitchenhamMadeyskiBudgen16.PolishSubjects

**Format**

A data frame with variables:

**Judge** The identifier for each subject

**Abstract** The identifier for each abstract - the code starts with a three alphanumeric string that defines the source of the abstract

**OrderViewed** Each judge assessed 4 abstracts in sequence, this data item identifies the order in which the subject viewed the specified abstract

**Completeness1** Assessment by judge of question 1: Is the reason for the project clear? Can take values: Yes/No/Partly

**Completeness2** Assessment by judge of question 2: Is the specific aim/purpose of the study clear? Can take values: Yes/No/Partly

**Completeness3** Assessment by judge of question 3: If the aim is to describe a new or enhanced software technology (e.g. method, tool, procedure or process) is the method used to develop this technology defined? Can take values: Yes/No/Partly/NA

**Completeness4** Assessment by judge of question 4: Is the form (e.g. experiment, general empirical study, data mining, case study, survey, simulation etc.) that was used to evaluate the technology made clear? Can take values: Yes/No/Partly

**Completeness5** Assessment by judge of question 5: Is there a description of how the evaluation process was organised? Can take values: Yes/No/Partly

**Completeness6** Assessment by judge of question 6: Are the results of the evaluation clearly described? Can take values: Yes/No/Partly

**Completeness7** Assessment by judge of question 7: Are any limitations of the study reported?: Yes/No/Partly

**Completeness8** Assessment by judge of question 8: Are any ideas for future research presented?: Yes/No/Partly

**Clarity** Assessment by judge of question regarding the overall understandability of the abstract: Please give an assessment of the clarity of this abstract by circling a number on the scale of 1-10 below, where a value of 1 represents Very Obscure and 10 represents Extremely Clearly Written.

**Completeness1NumValue** A numerical value for completeness question 1 where 0=No, Partly=0.5, yes =1

**Completeness2NumValue** A numerical value for completeness question 2 where 0=No, Partly=0.5, yes =1, NA means not applicable

**Completeness3NumValue** A numerical value for completeness question 3 where 0=No, Partly=0.5, yes =1, NA means not applicable or not answered

**Completeness4NumValue** A numerical value for completeness question 4 where 0=No, Partly=0.5, yes =1, NA means not applicable

**Completeness5NumValue** A numerical value for completeness question 5 where 0=No, Partly=0.5, yes =1, NA means not applicable

**Completeness6NumValue** A numerical value for completeness question 6 where 0=No, Partly=0.5, yes =1, NA means not applicable

**Completeness7NumValue** A numerical value for completeness question 7 where 0=No, Partly=0.5, yes =1, NA means not applicable

**Completeness8NumValue** A numerical value for completeness question 8 where 0=No, Partly=0.5, yes =1, NA means not applicable

**Sum** The sum of the numerical completeness questions excluding those labelled NA

**TotalQuestions** The count of the number of question related to completeness excluding questions considered not applicable

**Completeness** Sum/TotalQuestions

### Details

Data set collected at Wroclaw University of Technology (POLAND) by Lech Madeyski includes separate entries for each abstract assessed by a judge, that is 4 entries for each judge. Data collected from 16 subjects recruited from Wroclaw University of Technology who were each asked to assess 4 abstracts.

Note Only completeness question 2 was expected to be context dependent and have a NA (not applicable) answer, if other completeness answers were left blank, BAK coded the answer as NA  
polishsubjects.txt

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

KitchenhamMadeyskiBudgen16.PolishSubjects

---

KitchenhamMadeyskiBudgen16.SubjectData

*KitchenhamMadeyskiBudgen16.SubjectData*

---

### Description

If you use this data set please cite this R package and the following paper when accepted: Barbara Kitchenham, Lech Madeyski, David Budgen, Jacky Keung, Pearl Brereton, Stuart Charters, Shirley Gibbs, and Amnart Pohthong, 'Robust Statistical Methods for Empirical Software Engineering', Empirical Software Engineering, vol. 22, no. 2, pp. 579–630, 2017. DOI: 10.1007/s10664-016-9437-5 (<https://dx.doi.org/10.1007/s10664-016-9437-5>), URL: <https://madeyski.e-informatyka.pl/download/KitchenhamMa>

### Usage

KitchenhamMadeyskiBudgen16.SubjectData

**Format**

A data frame with variables:

**Judge** Alphanumeric identifier for each judge

**Institution** Numerical value identifying each site from which data was collected

**JudgeID** Numerical value identifying each judge

**Age** Age of the judge in years

**Eng1st** Whether the judge's first language was English: Yes/No

**YearsStudy** The number of years have student been studying computing at University: 1, 2, 3, 4

**AbstractsRead** Number of abstracts the judge had read prior to the study' 0, 1 to 10, 10+

**AbstractsWritten** Whether the judge had ever written an abstract for a scientific report/article

**AbstractID** Alphanumeric identifier for an abstract. The first character identifies the journal, I=IST, J=JSS, the third digit identifies the time period as 1 or 2, the remaining digits identify the abstract number within the set of abstracts found for the specified journal and time period

**Treat** The initial 3 characters of AbstractID

**TreatID** A numeric identifier for the journal and time period, 1=IB1, 2=IB2, 3=JB1, 4=JB2

**Order** The order in which the judge should have viewed the specified abstract

**Completeness1NumValue** The numeric answer to completeness question 1

**Completeness2NumValue** The numeric answer to completeness question 2

**Completeness3NumValue** The numeric answer to completeness question 3

**Completeness4NumValue** The numeric answer to completeness question 4

**Completeness5NumValue** The numeric answer to completeness question 5

**Completeness6NumValue** The numeric answer to completeness question 6

**Completeness7NumValue** The numeric answer to completeness question 7

**Completeness8NumValue** The numeric answer to completeness question 8

**Clarity** The response to the clarity question or NA if not answered

**NumberOfAnsweredCompletenessQuestions** The number of completeness questions excluding those with NA

**TotalScore** Sum of the numeric values of the 8 completeness questions

**MeanScore** Sum of the completeness questions 1 to 8 divided by TotalScore

**Site** The name of the site which provided the data. HongKong refers to the Polytechnic University, HongKong.2 refers to the City University

**Details**

Data set collected from 16 judges assessing 4 abstracts at 6 sites: Lincoln University NZ=1, Hong Kong Polytechnic University=2, PSu Thailand=3, Durham=4, Keele=5, Hong Kong City University=6

subjectdata.txt: Judge Institution JudgeID age eng1st years.study abs.read Absid Treat TreatID Order Com.1 Com.2 Com.3 Com.4 Com.5 Com.6 Com.7 Com.8 Clarity num.questions total.score av.score Site

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

```
KitchenhamMadeyskiBudgen16.SubjectData
```

---

LaplaceDist

*LaplaceDist*

---

**Description**

Returns a sample of N observations from a Laplace distribution with specified mean and spread.

**Usage**

```
LaplaceDist(N, mean, spread, max = 0.5, min = -0.5)
```

**Arguments**

N	is the required sample size
mean	is the required mean
spread	is the spread of the function
max	the upper limit of the distribution. Must be finite.
min	the lower limit of the distribution. Must be finite.

**Value**

N values from a Laplace distribution

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
set.seed(123)
LaplaceDist(10, 0, 1)
# [1] -0.55311564 0.85946218 -0.20094937 1.45258293 2.12808209 -2.39565480 0.05785263
# [8] 1.53636446 0.10855453 -0.09076809
```

---

Madeyski15EISEJ.OpenProjects

*Madeyski15EISEJ.OpenProjects data*

---

## Description

If you use this data set please cite: Marian Jureczko and Lech Madeyski, 'Cross-project defect prediction with respect to code ownership model: An empirical study', e-Informatica Software Engineering Journal, vol. 9, no. 1, pp. 21-35, 2015. DOI: 10.5277/e-Inf150102 (<https://dx.doi.org/10.5277/e-Inf150102>) URL: <https://madeyski.e-informatyka.pl/download/JureczkoMadeyski15.pdf>)

## Usage

Madeyski15EISEJ.OpenProjects

## Format

A data frame with variables:

**PROP** The percentage of classes of proprietary (i.e., industrial) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on open source projects.

**NOTOPEN** The percentage of classes of projects which are not open source projects that must be tested in order to find 80% of defects in case of software defect prediction models built on open source projects.

**STUD** The percentage of classes of student (i.e., academic) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on open source projects.

**OPEN** The percentage of classes of open source projects that must be tested in order to find 80% of defects in case of software defect prediction models built on open source projects.

## Details

This paper presents an analysis of 84 versions of industrial, open-source and academic projects. We have empirically evaluated whether those project types constitute separate classes of projects with regard to defect prediction. The predictions obtained from the models trained on the data from the open source projects were compared with the predictions from the other models (built on proprietary, i.e. industrial, student, open source, and not open source projects).

## Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

## Examples

Madeyski15EISEJ.OpenProjects

---

Madeyski15EISEJ.PropProjects

*Madeyski15EISEJ.PropProjects data*

---

## Description

If you use this data set please cite: Marian Jureczko and Lech Madeyski, 'Cross-project defect prediction with respect to code ownership model: An empirical study', e-Informatica Software Engineering Journal, vol. 9, no. 1, pp. 21-35, 2015. DOI: 10.5277/e-Inf150102 (<https://dx.doi.org/10.5277/e-Inf150102>) URL: <https://madeyski.e-informatyka.pl/download/JureczkoMadeyski15.pdf>

## Usage

Madeyski15EISEJ.PropProjects

## Format

A data frame with variables:

**NOTPROP** The percentage of classes of non-proprietary (i.e., non-industrial) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on proprietary (i.e., industrial) projects.

**OPEN** The percentage of classes of open source projects that must be tested in order to find 80% of defects in case of software defect prediction models built on proprietary (i.e., industrial) projects.

**STUD** The percentage of classes of student (i.e., academic) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on proprietary (i.e., industrial) projects.

**PROP** The percentage of classes of proprietary (i.e., industrial) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on proprietary (i.e., industrial) projects.

## Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

## Examples

Madeyski15EISEJ.PropProjects

---

Madeyski15EISEJ.StudProjects

*Madeyski15EISEJ.StudProjects data*

---

## Description

If you use this data set please cite: Marian Jureczko and Lech Madeyski, 'Cross-project defect prediction with respect to code ownership model: An empirical study', e-Informatica Software Engineering Journal, vol. 9, no. 1, pp. 21-35, 2015. DOI: 10.5277/e-Inf150102 (<https://dx.doi.org/10.5277/e-Inf150102>) URL: <https://madeyski.e-informatyka.pl/download/JureczkoMadeyski15.pdf>

## Usage

Madeyski15EISEJ.StudProjects

## Format

A data frame with variables:

**PROP** The percentage of classes of proprietary (i.e., industrial) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on student (i.e., academic) projects.

**NOTSTUD** The percentage of classes of projects which are not student projects that must be tested in order to find 80% of defects in case of software defect prediction models built on student (i.e., academic) projects.

**STUD** The percentage of classes of student (i.e., academic) projects that must be tested in order to find 80% of defects in case of software defect prediction models built on student (i.e., academic) projects.

**OPEN** The percentage of classes of open source projects that must be tested in order to find 80% of defects in case of software defect prediction models built on student (i.e., academic) projects.

## Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

## Examples

Madeyski15EISEJ.StudProjects

---

Madeyski15SQJ.NDC

*Madeyski15SQJ.NDC data*

---

### Description

If you use this data set please cite: Lech Madeyski and Marian Jureczko, 'Which Process Metrics Can Significantly Improve Defect Prediction Models? An Empirical Study,' *Software Quality Journal*, vol. 23, no. 3, pp.393-422, 2015. DOI: 10.1007/s11219-014-9241-7

### Usage

Madeyski15SQJ.NDC

### Format

A data frame with variables:

**Project** In case of open source projects this field includes the name of the project as well as its version. In case of industrial projects this field includes the string 'proprietary' (we were not allowed to disclose the names of the analyzed industrial software projects developed by Capgemini Polska).

**simple** The percentage of classes that must be tested in order to find 80% of defects in case of simple defect prediction models, i.e., using only software product metrics as predictors.

**advanced** The percentage of classes that must be tested in order to find 80% of defects in case of advanced defect prediction models, using not only software product metrics but also the NDC (Number of distinct committers) process metric.

### Details

'This paper presents an empirical evaluation in which several process metrics were investigated in order to identify the ones which significantly improve the defect prediction models based on product metrics. Data from a wide range of software projects (both, industrial and open source) were collected. The predictions of the models that use only product metrics (simple models) were compared with the predictions of the models which used product metrics, as well as one of the process metrics under scrutiny (advanced models). To decide whether the improvements were significant or not, statistical tests were performed and effect sizes were calculated. The advanced defect prediction models trained on a data set containing product metrics and additionally Number of Distinct Committers (NDC) were significantly better than the simple models without NDC, while the effect size was medium and the probability of superiority (PS) of the advanced models over simple ones was high ( $p=.016$ ,  $r=-.29$ ,  $PS=.76$ ), which is a substantial finding useful in defect prediction. A similar result with slightly smaller PS was achieved by the advanced models trained on a data set containing product metrics and additionally all of the investigated process metrics ( $p=.038$ ,  $r=-.29$ ,  $PS=.68$ ). The advanced models trained on a data set containing product metrics and additionally Number of Modified Lines (NML) were significantly better than the simple models without NML, but the effect size was small ( $p=.038$ ,  $r=.06$ ). Hence, it is reasonable to recommend the NDC process metric in building the defect prediction models.' [<https://dx.doi.org/10.1007/s11219-014-9241-7>]

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

Madeyski15SQJ.NDC

---

MadeyskiKitchenham.EUBASdata

*MadeyskiKitchenham.EUBASdata data*

---

**Description**

If you use this data set please cite this R package and the paper where we analyze the data set: Lech Madeyski and Barbara Kitchenham, 'Effect Sizes and their Variance for AB/BA Crossover Design Studies', Empirical Software Engineering, vol. 24, no.4, p. 1982-2017, 2018. DOI: 10.1007/s10664-017-9574-5

**Usage**

MadeyskiKitchenham.EUBASdata

**Format**

A data frame with variables:

**ID** Project ID

**TimePeriod** Period of time (run): R1, R2

**SequenceGroup** Sequence group: G1, G2, G3, G4

**System** Software system identifier indicates the system (i.e., S1 or S2) used as the experimental object: S1. A software system to sell and manage CDs/DVDs in a music shop, S2. A software system to book and buy theater tickets

**Technique** The independent variable. It is a nominal variable that can assume the following two values: AM (analysis models plus source code) and SC (source code alone)

**Comp\_Level** This denotes the comprehension level of the source code achieved by a software engineer

**Modi\_Level** This denotes the capability of a maintainer to modify source code

**Details**

Data set comes from an experiment conducted in Italy at the University of Basilicata (with 24 first-year students from the Master's Program in Computer Science) to answer the question 'Do the software models produced in the requirements analysis process aid in the comprehensibility and modifiability of source code?', see G. Scanniello, C. Gravino, M. Genero, J. A. Cruz-Lemus, and G. Tortora, 'On the Impact of UML Analysis Models on Source-code Comprehensibility and Modifiability,' ACM Transactions on Software Engineering and Methodology, vol. 23, pp. 13:1-13:26, Apr. 2014. However, the inconsistent subject data for subject 2 was removed, see the aforementioned paper by Madeyski and Kitchenham.

**Source**

<https://madeyski.e-informatyka.pl/reproducible-research/>

**Examples**

MadeyskiKitchenham.EUBASdata

---

MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR

*MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR data*

---

**Description**

Data form a set of primary studies on reading methods for software inspections. They were analysed by Lech Madeyski and Barbara Kitchenham, 'How variations in experimental designs impact the construction of comparable effect sizes for meta-analysis', 2015.

**Usage**

MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR

**Format**

A data frame with 17 rows and 26 variables:

**Study** Name of empirical study

**Ref.** Reference to the paper reporting primary study or experimental run where data were originally reported

**Teams** The number of teams including both, PBR and Control teams

**DesignDesc** Experimental design description: Before-after, Between-groups, Cross-over

**ExpDesign** Experimental design: between-groups (BG), within-subjects cross-over (WSCO), within-subjects before-after (WSBA)

**M\_PBR** The average proportion of defects found by teams using PBR

- M\_C** The average proportion of defects found by teams using Control treatment: Check-Based Reading (CBR) or Ad-Hoc Reading (AR)
- Diff** The difference between M\_PBR and M\_C, i.e.  $Diff = M\_PBR - M\_C$
- Inc** The percentage increase in defect rate detection, i.e.  $Inc = 100 * [(M\_PBR - M\_C) / M\_C]$
- SD\_C\_ByAuthors** The standard deviation of the control group values reported by the original Authors, i.e., obtained from the papers/raw data
- SD\_C** The standard deviation of the control group values equals SD\_C\_ByAuthors for studies for which the data was available OR the weighted average of SD\_C\_ByAuthors (i.e., 0.169) for studies where SD\_C\_ByAuthors is missing.
- V\_C** The variance of the Control group observations, i.e., the variance obtained from the teams using the Control method  $V\_C = SD\_C^2$
- V\_D** The variance of the unstandardized mean difference D (between the mean value for the treatment group and the mean value for the Control group)
- SD\_C\_Alt** This is the equivalent of SD\_C (the standard deviation of the control group) based on a different variance for the student studies or the practitioner studies depending on the subject type of the study with the missing value.
- V\_Alt** The variance of the mean difference in the meta-analysis based on SD\_C\_Alt
- SS\_C** The sum of squares of the Control group values. For within subjects studies  $SS = V\_C * (n - 1)$ . For between subjects studies  $SS = V\_C * (n_C - 1)$
- n\_PBR** The number of PBR teams
- n\_C** The number of Control (CBR or AR) teams
- ControlType** Type of Control treatment: CRB or AR
- ParticipantsType** Type of participants: Engineers or Students
- TeamType** Type of team: Nominal or Real
- TwoPersonTeamVsLargerTeam** Reflects size of the teams: 2-PersonTeam or LargerTeam
- ArtefactType** The type of artefact: Requirements or Other
- AssociatedWithBasili** Whether study is associated with Basili (the forerunner): Yes or No
- ControlType\_Basili** Combined ControlType and AssociatedWithBasili: AH\_AssociatedWithBasili, CBR\_AssociatedWithBasili, CBR\_NotAssociatedWithBasili

### Details

If you use this data set please cite: Lech Madeyski and Barbara Kitchenham, 'How variations in experimental designs impact the construction of comparable effect sizes for meta-analysis', 2015.

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR

---

MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20190324

*MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20190324*  
*data*

---

## Description

This data is used in the paper: Tomasz Lewowski and Lech Madeyski, 'Creating Evolving Project Data Sets in Software Engineering', vol. 851 of Studies in Computational Intelligence, pp. 1–14. Cham: Springer, 2020. DOI: 10.1007/978-3-030-26574-8\_1

## Usage

MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20190324

## Format

A text file with variables:

**rowID** unique id assigned to projects before filtering (source: API)

**id** GitHub repository ID (source: API)

**repository owner** the organization or user owning the repository (source: API)

**project name** name of the project (source: API)

**manual** link to best found project documentation - wiki, webpage, documentation directory or readme. Projects with limited documentation were marked with (limited) and ones that had documentation in Chinese - (Chinese) (source: manual)

**installation** the recommended installation medium(s) for the project. Some mediums may be missing for projects with multiple recommendations. (source: manual)

**support** channel(s) that can be used to get support and/or report bugs. Some channels may be missing for projects with multiple ones. Abbreviations used (source: manual): GH GitHub Issues SO Stack Overflow GG Google Groups ML Mailing list FB Facebook MM Mattermost LI LinkedIn ? not found

**is not sample/playground/docs/...** 1 if the project is an actual application or library, 0 if it is a set of samples, only documentation or some experimental area (source: manual)

**is industrial** whether the project can be treated as industrial quality one. Values and their meanings: 1 the repository can be classified as industrial grade; 0,5 the repository can sometimes be classified as industrial grade, but it is either a minor project or its documentation or support may be lacking the depth; 0 the repository cannot be classified as industrial-grade; -1 the repository is no longer actively maintained as of the date of data acquisition; -2 the repository is no longer in Java as of the date of data acquisition. (source: manual)

**createdAt** the date at which the repository was created (source: API)

**updatedAt** the date of last repository update - including changes in projects, watchers, issues etc. (source: API)

- pushedAt** the date of last push to the repository - NOT the date of last pushed commit (source: API)
- diskUsage** total number of bytes on disk that are needed to store the repository (source: API)
- forkCount** number of existing repository forks (independent copies managed by other entities) (source: API)
- isArchived** true if the repository is archived (no longer maintained), false otherwise (source: API)
- isFork** true if the repository is a fork (not the main repository), false otherwise (source: API)
- isMirror** true if the repository is a mirror, false otherwise (source: API)
- sshUrlOfRepository** URL that can be used to immediately clone the repository (source: API)
- licenseInfo.name** name of license under which the project is distributed. Names are the same as in <https://choosealicense.com/appendix/> (source: API)
- commitSHA** unique Git identifier of commit that was top of the main branch at the time of data acquisition (source: API)
- defaultBranchRef.target.history.totalCount** number of commits on the default branch in the repository (usually master) at the time of data acquisition (source: API)
- stargazers.totalCount** number of stargazers for the repository at the time of data acquisition (source: API)
- watchers.totalCount** number of watchers for the repository at the time of data acquisition (source: API)
- languages.totalSize** total size of all source code files (source: API)
- Java.byte.count** total size of Java files (source: API)
- Language** main programming language used in the repository, i.e. one that the most code is written in (source: API)
- searchQuery** query used during search that obtained this project (source: API)

## Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

## Examples

MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20190324

---

MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022

*MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022*  
*data*

---

## Description

This data is used in the paper: Tomasz Lewowski and Lech Madeyski, 'How do software engineering data sets evolve? A reproduction study', 2020 (submitted). Generated by: `token <- '...'`  
`MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022<-searchForIndustryRelevantGitHubProjects(token, '2019-03-01', '2018-08-01')`  
`usethis::use_data(MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022)`

**Usage**

MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022

**Format**

A text file with variables:

**rowID** unique id assigned to projects before filtering (source: API)

**id** GitHub repository ID (source: API)

**repository owner** the organization or user owning the repository (source: API)

**project name** name of the project (source: API)

**manual** link to best found project documentation - wiki, webpage, documentation directory or readme. Projects with limited documentation were marked with (limited) and ones that had documentation in Chinese - (Chinese) (source: manual)

**installation** the recommended installation medium(s) for the project. Some mediums may be missing for projects with multiple recommendations. (source: manual)

**support** channel(s) that can be used to get support and/or report bugs. Some channels may be missing for projects with multiple ones. Abbreviations used (source: manual): GH GitHub Issues SO Stack Overflow GG Google Groups ML Mailing list FB Facebook MM Mattermost LI LinkedIn ? not found

**is not sample/playground/docs/...** 1 if the project is an actual application or library, 0 if it is a set of samples, only documentation or some experimental area (source: manual)

**is industrial** whether the project can be treated as industrial quality one. Values and their meanings: 1 the repository can be classified as industrial grade; 0,5 the repository can sometimes be classified as industrial grade, but it is either a minor project or its documentation or support may be lacking the depth; 0 the repository cannot be classified as industrial-grade; -1 the repository is no longer actively maintained as of the date of data acquisition; -2 the repository is no longer in Java as of the date of data acquisition. (source: manual)

**createdAt** the date at which the repository was created (source: API)

**updatedAt** the date of last repository update - including changes in projects, watchers, issues etc. (source: API)

**pushedAt** the date of last push to the repository - NOT the date of last pushed commit (source: API)

**diskUsage** total number of bytes on disk that are needed to store the repository (source: API)

**forkCount** number of existing repository forks (independent copies managed by other entities) (source: API)

**isArchived** true if the repository is archived (no longer maintained), false otherwise (source: API)

**isFork** true if the repository is a fork (not the main repository), false otherwise (source: API)

**isMirror** true if the repository is a mirror, false otherwise (source: API)

**sshUriOfRepository** URL that can be used to immediately clone the repository (source: API)

**licenseInfo.name** name of license under which the project is distributed. Names are the same as in <https://choosealicense.com/appendix/> (source: API)

- commitSHA** unique Git identifier of commit that was top of the main branch at the time of data acquisition (source: API)
- defaultBranchRef.target.history.totalCount** number of commits on the default branch in the repository (usually master) at the time of data acquisition (source: API)
- stargazers.totalCount** number of stargazers for the repository at the time of data acquisition (source: API)
- watchers.totalCount** number of watchers for the repository at the time of data acquisition (source: API)
- languages.totalSize** total size of all source code files (source: API)
- Java.byte.count** total size of Java files (source: API)
- Language** main programming language used in the repository, i.e. one that the most code is written in (source: API)
- searchQuery** query used during search that obtained this project (source: API)

### Source

<https://madeyski.e-informatyka.pl/reproducible-research/>

### Examples

MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20191022

---

metaanalyse.Cliffd     *metaanalyse.Cliffd*

---

### Description

This function provides a simple meta-analysis of experiments using Cliff's  $d$  as an effect size. It returns the  $100 \cdot (1 - \alpha/2)$

### Usage

```
metaanalyse.Cliffd(  
  Cliffd,  
  Cliffdvar,  
  df = 0,  
  alternative = "two.sided",  
  alpha = 0.05  
)
```

**Arguments**

Cliffd	A vector of one or more numerical values, identifying the effect sizes to be meta-analysed
Cliffdvar	A vector of the estimates variance of each of the effect sizes
df	The total degrees of freedom for the set of effect sizes. If $df > 0$ , the pvalues and significance test use the t-distribution probability values. If $df = 0$ (default) the pvalues and significance test use the normal distribution probability values. The confidence intervals are always based on the normal probability values, as recommended by Cliff.
alternative	Specifies the type of significance test and can take the values "two.sided" (default), "less" or "greater".
alpha	The significance level used to control the significance tests and calculation of confidence limits (default 0.05).

**Value**

Estimate	The overall estimate of Cliff's d obtained from the set of experiments
UpperCI	The upper $100 \cdot (1 - \alpha / 2)$
LowerCI	The lower $100 \cdot (1 - \alpha / 2)$
	The variance of the Estimate
tvalue	The value of the t-statistic
df	The supplied degrees of freedom or NA if the input parameter df was set to zero
AltHyp	Defines the alternative hypothesis used for significance testing and depends on the value of the input parameter alternative. It takes the values "Not=0", ">0", or "<0"
NullHyp	Defines the null hypothesis and depends on the value of the input parameter alternative. It takes the values "~0", "<0", or ">0"
pvalue	The p-value of the t-test if the parameter $df > 0$ , or the normal probability value if $d = 0$
RejectNullHyp	"Yes" or "No" depending on whether or not the null hypothesis should be rejected at the $\alpha / 2$ level for two-sided tests and alpha level for one-sided tests
	The Q homogeneity statistic
	The I-squared estimate of the extent of heterogeneity
ProbQHomogeneous	The probability that the set of Cliff's d values come from a set of homogeneous experiments.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
Cliffd=c(0.84,0.2,-0.04,0.44,0.76)
CliffdvarInvalid=c(0.04,0.18,0.21,0.15)
Cliffdvar=c(0.04,0.18,0.21,0.15,0.06)
CliffdvarInvalid=c(0.04,0.18,0.21,0.15)
```

```

df=45
as.data.frame(metaanalyse.Cliffd(Cliffd=Cliffd,Cliffdvar=Cliffdvar,df=df,alternative="greater",
  alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue
#1 0.44 0.6601381 0.1502568 0.0256 2.75 45 >0 <=0 0.004275955
# RejectNullHyp Q I.square ProbQHomogeneous
#1 Yes 21.5 81.39535 0.0002519835
as.data.frame(metaanalyse.Cliffd(Cliffd=Cliffd,Cliffdvar=Cliffdvar,df=df,alternative="less",
  alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue RejectNullHyp
#1 0.44 0.6601381 0.1502568 0.0256 2.75 45 <0 >=0 0.995724 No
# Q I.square ProbQHomogeneous
#1 21.5 81.39535 0.0002519835
as.data.frame(metaanalyse.Cliffd(Cliffd=Cliffd,Cliffdvar=Cliffdvar,df=df,alternative="two.sided",
  alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue
#1 0.44 0.692073 0.09227496 0.0256 2.75 45 Not=0 ~0 0.008551911
# RejectNullHyp Q I.square ProbQHomogeneous
#1 Yes 21.5 81.39535 0.0002519835
as.data.frame(metaanalyse.Cliffd(Cliffd=Cliffd,Cliffdvar=Cliffdvar,df=df,alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue
#1 0.44 0.692073 0.09227496 0.0256 2.75 45 Not=0 ~0 0.008551911
# RejectNullHyp Q I.square ProbQHomogeneous
#1 Yes 21.5 81.39535 0.0002519835
metaanalyse.Cliffd(Cliffd=Cliffd,Cliffdvar=Cliffdvar,df=0,alternative="two.sided",alpha=0.05)
#Error in testfunctionParameterChecks(alternative = alternative, alpha = alpha, :
# Invalid alternative parameter, choose one of two.sided, greater or less
# metaanalyse.Cliffd(Cliffd=Cliffd,Cliffdvar=CliffdvarInvalid,df=df,alternative="greater",
# alpha=0.05)
#Error in metaanalyse.Cliffd(Cliffd = Cliffd, Cliffdvar = CliffdvarInvalid, :
# Length of Cliffdvar parameter must equal the length of the Cliffd parameter

```

---

metaanalyse.PHat

*metaanalyse.PHat*

---

## Description

This function performs a meta-analysis of experiments using PHat as an effect size. It returns the  $100 \cdot (1 - \alpha/2)$

## Usage

```

metaanalyse.PHat(
  PHat,
  PHatvar,
  DFUnknown,
  df,
  alternative = "two.sided",
  alpha = 0.05
)

```

**Arguments**

PHat	The estimates of PHat obtained from a group of experiments to be meta-analysed
PHatvar	The estimate of the variance of each PHat estimate
DFUnknown	If DFUnknown=FALSE the degrees of freedom for each experiment is known, and the df parameter must be a vector specifying the effect size of each experiment, otherwise the df parameter is ignored.
df	If DFUnknown is TRUE, this parameter is a vector of numerical values specifying the degrees of freedom for each experiment, and the confidence intervals, pvalues and significance test use the t-distribution probability values. If the parameter DFUnknown is FALSE, the confidence intervals, pvalues and significance test use the normal distribution probability values.
alternative	Specifies the type of significance test and can take the values "two.sided" (default), "less" or "greater".
alpha	The significance level (default 0.05) used to control the significance tests and calculation of confidence limits.

**Value**

Estimate. The simple average of the PHat values recommended by Kromrey as the best estimator for meta-analysis.

UpperCI The upper  $100*(1-\alpha/2)$

LowerCI The lower  $100*(1-\alpha/2)$

Variance The variance of the Estimate output

tvalue The value of the t-statistic

df Either NA if the parameter DFUnknown is TRUE, or sum of the degrees of freedom for each experiment.

AltHyp Defines the alternative hypothesis used for significance testing and depends on the value of the input parameter alternative. It takes the values "Not=0.5", ">0.5", or "<0.5".

NullHyp Defines the null hypothesis and depends on the value of the input parameter alternative. It takes the values "~0.5", "<0.5", or ">0.5".

pvalue The p-value of the t-test if the parameter DFUnknown is FALSE, otherwise the normal probability value.

RejectNullHyp "Yes" or "No" depending on whether or not the null hypothesis should be rejected at the  $\alpha/2$  level for two-sided tests and alpha level for one-sided tests

The I-squared estimate of the extent of heterogeneity

The Q homogeneity statistic

ProbQHomogeneous. The probability that the set of Phat values come from a set of homogeneous experiments.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

PHat=c(0.92,0.6,0.48,0.72,0.88)
PHatvar=c(0.01,0.04,0.05,0.04,0.01)
PHatdf=c(6.63,6.63,5.08,5.61,8)
PHatInvalid=c(0.92,0.6,0.48,0.72)
as.data.frame(metaanalyse.PHat(PHat=PHat,PHatvar=PHatvar,DFUnknown=FALSE,df=PHatdf,
alternative="greater",alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue RejectNullHyp..
# 1 0.72 0.8777899 0.5622101 0.006 2.840188 31.95 >0.5 <=0.5 0.003890609 Yes..
as.data.frame(metaanalyse.PHat(PHat=PHat,PHatvar=PHatvar,DFUnknown=TRUE,df=PHatdf,
alternative="greater",alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue RejectNullHyp..
# 1 0.72 0.8718182 0.5681818 0.006 2.840188 NA >0.5 <=0.5 0.002254349 Yes..
as.data.frame(metaanalyse.PHat(PHat=PHat,PHatvar=PHatvar,DFUnknown=FALSE,df=PHatdf,
alternative="two.sided",alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue RejectNullH..
#1 0.72 0.8777899 0.5622101 0.006 2.840188 31.95 Not=0.5 ~0.5 0.007781218 Yes..
as.data.frame(metaanalyse.PHat(PHat=PHat,PHatvar=PHatvar,DFUnknown=TRUE,df=PHatInvalid,
alpha=0.05))
# Estimate UpperCI LowerCI Variance tvalue df AltHyp NullHyp pvalue RejectNullHyp I..
#1 0.72 0.8718182 0.5681818 0.006 2.840188 NA Not=0.5 ~0.5 0.004508698 Yes 82..

```

---

metaanalyseSmallSampleSizeExperiments

*metaanalyseSmallSampleSizeExperiments*

---

**Description**

Implements analysis of small sample size experiments based on Hedges and Olkin p128-131.

**Usage**

```
metaanalyseSmallSampleSizeExperiments(d, f, A)
```

**Arguments**

**d** a vector of standardized mean differences for different experiments (not adjusted for small sample size)

**f** a vector defining the degrees for each experiment

**A** a vector defining the relationship between d and its related t value for each experiment

**Value**

**UnweightedMean** The unweighted mean of the small size adjusted standardized mean differences

**WeightedMean** The weighted mean of the small size adjusted standardized mean differences

**VarWeightedMean** The variance of the weighted mean

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
d=c(.461, .782, .513, .612, -0.131, -0.018, 0.774, 0.138, -0.482, 0.333, .701, -0.222,
    .399, .538, -0.19, 0.833, 0.512, 0.601, -0.366, .510)
A=c(2/5, 2/5, 2/7, 2/10, 2/8, 2/10, 2/6, 2/6, 2/4, 2/9, 2/9, 2/7, 2/7, 2/5, 2/5, 2/4, 2/10, 2/8, 2/4, 2/8)
f=c(8, 8, 12, 18, 14, 18, 10, 10, 6, 16, 16, 12, 12, 8, 8, 6, 18, 14, 6, 14)
metaanalyseSmallSampleSizeExperiments(d, f, A)
```

---

MetaAnalysisSimulations

*MetaAnalysisSimulations*

---

**Description**

This function simulates data from many families of experiments. The number of families simulated is defined by the Replications parameter. The parameter Exp determines the number of experiments in each family. The function simulates data from one of four distributions and uses the data to construct two of groups of equal size (GroupSize). The experimental design of individual experiments in each family is determined by the FourGroup parameter. If FourGroup=FALSE, the basic experimental design is a balanced two group randomized experiment, otherwise the experimental design is a balanced four group experiment corresponding to a randomized blocks experiment. The function calls either NP2GMetaAnalysisSimulation or NP2GMetaAnalysisSimulation to generate and analyse data for each individual family. The function either returns the meta-analysed data from each experiment or provides summary statistics.

**Usage**

```
MetaAnalysisSimulations(
  mean = 0,
  sd = 1,
  diff = 0.5,
  GroupSize = 10,
  type = "n",
  Replications = 50,
  Exp = 5,
  seed = 456,
  alpha = 0.05,
  FourGroup = FALSE,
  StdAdj = 0,
  BlockEffect = 0,
  BlockStdAdj = 0,
  StdExp = 0,
  MAMethod = "PM",
  returnES = FALSE,
```

```

    AlwaysTwoSidedTests = FALSE
  )

```

### Arguments

mean	the value used for the mean of control group in the simulated data. It can be any real number including zero.
sd	the value used for the spread of the control group and the spread of the treatment group in the simulated data. The value must be a real value greater than 0.
diff	mean+diff is the value used for the mean of the treatment group. It can be zero.
GroupSize	is the size of each of the groups comprising one experiment. Groupsize should be an integer of 4 or more
type	specifies the distribution being simulated. The permitted values are 'n' for the normal distribution (default), 'l' for the lognormal distribution, 'g' for the gamma distribution and 'lap' for the Laplace distribution.
Replications	The number of times the set of experiments is simulated.
Exp	is the number of experiments in each family of experiments being simulated. Exp should be an integer of 2 or more (default 5).
seed	specifies the seed to be used to initiate the simulation, so the simulation is repeatable.
alpha	The significance level used for tests and confidence intervals (default 0.05)
FourGroup	is a Boolean variable that determines whether the experiment is a two group experiments or a 4-Group randomised block experiment. It defaults to FALSE which means a two-group experiment is the default condition
StdAdj	If non-zero that can be used to introduced variability into the treatment spread/variance (default 0). Not appropriate for gamma data.
BlockEffect	A factor used to change the mean difference between blocks (default 0.5)
BlockStdAdj	if non-zero this can be used to change the BlockEffect from a fixed to random effect (default 0).
StdExp	if non-zero it simulates a random effect between experiments in the same family (default 0).
MAMethod	specifies the model to be used when experimental effect sizes ar aggregated using the R metafor package.
returnES	if TRUE the function outputs the summary statistics otherwise it outputs the meta-analysis results for each family (default FALSE)
AlwaysTwoSidedTests	This parameter can be used to override the one-sided tests used as default if the diff parameter is non-zero (default FALSE). This should only be set to TRUE to check simulation reported in other papers that seem to have used two-sided tests.

### Value

The parameter either returns the meta-analysis values obtained from each family or the average values of the meta-analysis over all replications.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

as.data.frame(
  MetaAnalysisSimulations(
    mean=0, sd=1, diff=0.5, GroupSize=10, type='n', Replications=5, Exp=5,
    seed=456, alpha=0.05, FourGroup=FALSE, StdAdj=0, BlockEffect=0.5,
    BlockStdAdj=0, StdExp=0, MAMethod='PM', returnES=FALSE))
# AverageCliffd AverageCliffdvar AverageCliffdsig Averagephat Averagephatvar
#1 0.3336 0.0132419 0.8 0.6668 0.003214756
#Averagephatsig AveMStd AveMStdvar AveMStd sig MAMean.phat MAphat.var
#1 0.9 0.6176206 0.04278117 0.9 0.689908 0.003888047
#MAphat.sig MAMean.Cliffd MACliffd.var MACliffd.sig Mean.StdMDUnweighted
#1 0.9 0.37984 0.01575063 0.9 0.6449963
#StdMDUnweighted.var StdMDUnweighted.sig Mean.StdMDAdjUnweighted
#1 0.04299001 0.9 0.6145034
#StdMDAdjUnweighted.var StdMDAdjUnweighted.sig Mean.HedgesMA Hedges.var
#1 0.04192908 0.9 0.6150575 0.04455833
#Hedges.sig Mean.StdMDAdjMA.exact StdMDAdjMA.exact.var StdMDAdjMA.exact.sig
#1 0.9 0.5834754 0.05171067 0.8
#Mean.StdMDAdjMA.approx StdMDAdjMA.approx.var StdMDAdjMA.approx.sig
#1 0.58643 0.04749064 0.9
#Mean.StdMDMA.exact StdMDMA.exact.var StdMDMA.exact.sig Mean.StdMDMA.approx
#1 0.6134374 0.05711235 0.8 0.6165884
#StdMDMA.approx.var StdMDMA.approx.sig
#1 0.05242339 0.9
#as.data.frame(
  # MetaAnalysisSimulations(
  # mean=0, sd=1, diff=0.5, GroupSize=10, type='n', Replications=50, Exp=5,
  # seed=456, alpha=0.05, FourGroup=FALSE, StdAdj=0, BlockEffect=0.5,
  # BlockStdAdj=0, StdExp=0, MAMethod='PM', returnES=FALSE))
# AverageCliffd AverageCliffdvar AverageCliffdsig Averagephat Averagephatvar
#1 0.29808 0.01333744 0.74 0.64904 0.003236444
# Averagephatsig AveMStd AveMStdvar AveMStd sig MAMean.phat MAphat.var
# 0.78 0.5450377 0.04217901 0.78 0.6677884 0.004538661
# MAphat.sig MAMean.Cliffd MACliffd.var MACliffd.sig Mean.StdMDUnweighted
#1 0.72 0.3356298 0.01833956 0.72 0.5686653
# StdMDUnweighted.var StdMDUnweighted.sig Mean.StdMDAdjUnweighted
#1 0.04237386 0.82 0.5419554
#StdMDAdjUnweighted.var StdMDAdjUnweighted.sig Mean.HedgesMA Hedges.var
# 0.04138573 0.78 0.5420552 0.04388383
# Hedges.sig Mean.StdMDAdjMA.exact StdMDAdjMA.exact.var StdMDAdjMA.exact.sig
#1 0.76 0.5163304 0.05874152 0.72
#Mean.StdMDAdjMA.approx StdMDAdjMA.approx.var StdMDAdjMA.approx.sig
#1 0.5203279 0.05591752 0.74
# Mean.StdMDMA.exact StdMDMA.exact.var
# 0.5418705 0.06468786
# StdMDMA.exact.sig Mean.StdMDMA.approx StdMDMA.approx.var StdMDMA.approx.sig
# 0.72 0.5461255 0.06159257 0.74

```

```

#as.data.frame(
#  MetaAnalysisSimulations(
#    mean=0, sd=1, diff=0.5, GroupSize=10, type='n', Replications=50, Exp=5,
#    seed=456, alpha=0.05, FourGroup=TRUE, StdAdj=0, BlockEffect=0.5,
#    BlockStdAdj=0, StdExp=0, MAMethod='PM', returnES=FALSE))
# AverageCliffd AverageCliffdvar AverageCliffdsig Averagephat ...
#1      0.27968      0.00683327      0.92      0.63984 ...
# as.data.frame(
#  MetaAnalysisSimulations(
#    mean=0, sd=1, diff=0.5, GroupSize=10, type='n', Replications=10, Exp=5,
#    seed=456, alpha=0.05, FourGroup=TRUE, StdAdj=0, BlockEffect=0.5,
#    BlockStdAdj=0, StdExp=0, MAMethod='PM', returnES=TRUE))
#Family NumExp GroupSize AveCliffd AveCliffdvar AveCliffdsig Avephat ...
#  1      1      5      10      0.252 0.007423693 TRUE ...
# Family NumExp GroupSize AveCliffd AveCliffdvar AveCliffdsig Avephat ...
#1      1      5      10      0.252 0.007423693 TRUE 0.626 ...

```

---

NP2GMetaAnalysisSimulation

*NP2GMetaAnalysisSimulation*


---

## Description

This function simulates data from a family of experiments. The parameter `Exp` determines the number of experiments in the family. The function simulates data from one of four distributions and uses the data to construct two of groups of equal size (`GroupSize`). The distribution for one of the groups corresponds to the control and is based on the given mean and spread, the distribution for the other group corresponds to the treatment group and is based on the mean+diff and the spread plus any variance adjustment requested (determined by the parameter `StdAdj`). The data from each experiment is analysed separately to estimate three non-parametric effect sizes: the Cliff's *d* and the probability of superiority referred to as *phat* and their variances. Parametric effect sizes Cohen's *d* (also known as the standardized means difference, SMD) and the small sample size adjusted standardized mean difference *g* are also calculated together with their variances. The effect sizes are then meta-analysed using various methods: the simple average of the effect size and the variance weighted averages (using the exact and approximate normal variance and the weighted and unweighted standardized mean difference). The function uses the `metafor` package for formal meta-analysis, and the specific method of formal meta-analysis used is determined by the `MAMethod`. All tests of significance are done at the 0.05 level. If the parameter `returnES` is `TRUE`, the function returns the effect sizes for each experiment in the family, otherwise it returns the meta-analysis results.

## Usage

```

NP2GMetaAnalysisSimulation(
  mean,
  sd,
  diff,
  GroupSize,
  Exp = 5,

```

```

type = "n",
StdAdj = 0,
alpha = 0.05,
seed = 123,
StdExp = 0,
MAMethod,
returnES = FALSE,
AlwaysTwoSidedTests = FALSE
)

```

### Arguments

mean	the value used for the mean of control group in the simulated data. It can be any real number including zero.
sd	the value used for the spread of the control group and the spread of the treatment group in the simulated data. The value must be a real value greater than 0.
diff	mean+diff is the value used for the mean of the treatment group. It can be zero.
GroupSize	is the size of each of the 2 groups comprising one experiment. Groupsize should be an integer of 4 or more
Exp	is the number of experiments being simulated. Exp should be an integer of 2 or more. It defaults to 5.
type	specifies the distribution being simulated. The permitted values are "n" for the normal distribution, "l" for the lognormal distribution, "g" for the gamma distribution and "lap" for the Laplace distribution. The parameter defaults to "n".
StdAdj	specifies a level used to adjust the treatment variance. It allows heterogeneity to be modelled. It defaults to zero meaning no variance heterogeneity is introduced.
alpha	the Type 1 error rate level use for statistical tests.
seed	specifies the seed to be used to initiate the simulation, so the simulation is repeatable. It defaults to 123.
StdExp	defines whether any additional heterogeneity is introduced between families. The value (set to 0 or 0.5 for our simulations) is used when we generate a deviation to be added to the control mean (control rate for gamma data) for each family. The deviation is generated from a Normal distribution with mean 0 and standard deviation=0.5. If StdExp=0 we do not add any deviations to the mean.
MAMethod	the meta-analysis method needed for the call to the metafor package rma algorithm
returnES	Determines the format of the output. It defaults to FALSE which causes the function to output the meta-analysis results for the family of experiments. If set to TRUE it returns the effect sizes for each experiment.
AlwaysTwoSidedTests	If FALSE the function performs one-sided tests if diff!=0, and two-sided tests if diff=0. If set to TRUE the function always does two-sided tests.

**Value**

Depending on the value of the returnES parameter, the function either returns the effect sizes for each experiment or the aggregated results for the family

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
as.data.frame(NP2GMetaAnalysisSimulation(mean=0,sd=1,diff=0.5,GroupSize=10,
  Exp=5,type="n",StdAdj=0,alpha=0.05,seed=457,StdExp=1,MAMethod="PM",
  returnES=FALSE))
# NumExp GroupSize AveCliffd AveCliffdvar AveCliffdsig Avephat Avephatvar Avephatsig AveMStd..
# 5 10 0.252 0.01499003 TRUE 0.626 0.003645333 TRUE 0.4883188..
# AveMStdSig Maphat Maphatvar Maphatsig MACliffd MACliffdvar MACliffdsig StdMDAdjUnweighted..
#1 TRUE 0.6288 0.003620188 TRUE 0.2575 0.01490134 TRUE 0.4748065..
# StdMDAdjUnweightedvar StdMDAdjUnweightedSig StdMDUnweighted StdMDUnweightedvar StdMDUnweight..
#1 0.04065614 TRUE 0.4980148 0.04157691 TRUE
# HedgesMA.Weighted HedgesMA.Weightedvar HedgesMA.WeightedSig StdMDAdjMAexact StdMDAdjMAexactvar
#1 0.4755316 0.04307274 TRUE 0.4725834 0.04315211
# StdMDAdjMAexactSig StdMDAdjMAapprox StdMDAdjMAapproxvar StdMDAdjMAapproxSig StdMDMAapprox St..
#1 TRUE 0.4716 0.03762363 TRUE 0.4955783 ..
# StdMDMAapproxSig StdMDMAexact StdMDMAexactvar StdMDMAexactSig
#1 TRUE 0.4966121 0.04756193 TRUE
as.data.frame(NP2GMetaAnalysisSimulation(mean=0,sd=1,diff=0.5,GroupSize=10,Exp=5,type="n",
  StdAdj=0,alpha=0.05,seed=457,StdExp=1,MAMethod="PM",returnES=TRUE))
# MeanExp VarExp StdMD df tval t.sig Cliffd Cliffdvar Cliffd.sig PHat PHat..
#1 0.5641594 1.437447 0.4705502 17.77980 1.0521822 FALSE 0.26 0.08149818 FALSE 0.63 0.02..
#2 0.6400936 1.081352 0.6155452 17.23411 1.3764009 FALSE 0.36 0.06527192 FALSE 0.68 0.01..
#3 0.8199650 1.698610 0.6291418 15.42141 1.4068038 FALSE 0.28 0.07362909 FALSE 0.64 0.01..
#4 0.2970819 1.709441 0.2272214 13.87833 0.5080824 FALSE 0.04 0.07936485 FALSE 0.52 0.01..
#5 0.5688567 1.079082 0.5476154 16.79899 1.2245053 FALSE 0.32 0.07498667 FALSE 0.66 0.01..
# Phat.sig StdMDAdj StdMDAdjvar.exact StdMDAdjvar.approx StdMDvar.exact StdMDvar.approx
#1 FALSE 0.4503698 0.2129598 0.1884384 0.2324722 0.2057040
#2 FALSE 0.5882961 0.2182075 0.1918563 0.2388898 0.2100409
#3 FALSE 0.5979539 0.2211428 0.1911344 0.2448130 0.2115926
#4 FALSE 0.2146782 0.2105671 0.1800107 0.2358918 0.2016604
#5 FALSE 0.5227345 0.2162500 0.1896495 0.2373259 0.2081330
as.data.frame(NP2GMetaAnalysisSimulation(mean=0,sd=1,diff=0.724,GroupSize=10,Exp=5,type="1",
  StdAdj=0,alpha=0.05,seed=123,StdExp=1,MAMethod="PM",returnES=FALSE))
# NumExp GroupSize AveCliffd AveCliffdvar AveCliffdsig Avephat Avephatvar Avephatsig AveMStd..
#1 5 10 0.344 0.01288023 TRUE 0.672 0.003118222 TRUE 0.483665..
# AveMStdSig Maphat Maphatvar Maphatsig MACliffd MACliffdvar MACliffdsig StdMDAdjUnweighted
#1 TRUE 0.7014 0.004229764 TRUE 0.403 0.01690867 TRUE 0.5722448
# StdMDAdjUnweightedvar StdMDAdjUnweightedSig StdMDUnweighted StdMDUnweightedvar StdMDUnweight..
#1 0.04146189 TRUE 0.6046947 0.04260837 TRUE
# HedgesMA.Weighted HedgesMA.Weightedvar HedgesMA.WeightedSig StdMDAdjMAexact StdMDAdjMAexactvar
#1 0.5742311 0.04453436 TRUE 0.5405307 0.0450343
# StdMDAdjMAexactSig StdMDAdjMAapprox StdMDAdjMAapproxvar StdMDAdjMAapproxSig StdMDMAapprox S..
#1 TRUE 0.5411 0.03819079 TRUE 0.5737401 0...
# StdMDMAapproxSig StdMDMAexact StdMDMAexactvar StdMDMAexactSig
```

#1	TRUE	0.5727409	0.05042801	TRUE
----	------	-----------	------------	------

---

 NP4GMetaAnalysisSimulation

*NP4GMetaAnalysisSimulation*


---

### Description

This function simulates data from a family of experiments, where the number of experiments in a family is defined by the parameter `Exp`. It simulates data from one of four distributions and uses the data to construct four of groups of equal size (`GroupSize`). Two groups are assigned as control groups and their distribution is based on the parameter, mean, and the parameter, spread. However, the mean and spread for the control group in Block 2 can be adjusted using the parameters `BlockEffect` and `BlockStdAdj` respectively. The other two groups are treatment groups and their distribution is based on the mean+diff and the spread parameter, but the distributions can be adjusted using the `StdAdj`, `BlockEffect` and `BlockStdAdj` parameters. The data from each experiment is analysed separately to estimate the non-parametric statistics P-hat, Cliff's d and their variances. In addition, the estimates of the standardized mean difference and the small sample size adjusted standardized mean difference are calculated. The effect size statistics are then meta-analysed using the method specified by the `MAMethod` parameter. We output both the average non-parametric effect statistics across the `Exp` experiment analysed as if they arose from a single large experiment and also the results of meta-analysing each non-parametric effect size. We use the standard parametric effect sizes and their meta-analysis as baselines. Tests of significance are one-sided if the mean difference is non-zero. If the mean difference is zero, two-sided tests are used. In addition, the user can force the use of two-sided tests using the parameter `AlwaysTwoSidedTests`. This should only be used for comparison with results reported in other simulation studies. The `alpha` parameter determines the significance level used in the tests.

### Usage

```
NP4GMetaAnalysisSimulation(
  mean,
  sd,
  diff,
  GroupSize,
  Exp = 5,
  type = "n",
  alpha = 0.05,
  seed = 123,
  StdAdj = 0,
  BlockEffect = 0,
  BlockStdAdj = 0,
  StdExp = 0,
  MAMethod,
  returnES = FALSE,
  AlwaysTwoSidedTests = FALSE
)
```

**Arguments**

mean	The default value used for the group means in the simulated data. It can be any real number including zero.
sd	The default value used for the spread of the control group and the spread of the treatment group in the simulated data. The value must be a real value greater than 0.
diff	mean+diff is the value used for the mean of the treatment group. It can be zero.
GroupSize	is the size of each of the 4 groups comprising one experiment. Groupsize should be an integer of 4 or more
Exp	is the number of experiments being simulated. Exp should be an integer of 2 or more. It defaults to 5.
type	specifies the distribution being simulated. The permitted values are "n" for the normal distribution, "l" for the lognormal distribution, "g" for the gamma distribution and "lap" for the Laplace dsitribution. The parameter defaults to "n".
alpha	the Type 1 error rate level use for statistical tests.
seed	specifies the seed to be used to initiate the simulation, so the simulation is repeatable. It defaults to 123.
StdAdj	The value used to introduce heterogeneity into the treatment groups variance if required.
BlockEffect	is the effect of having two different blocks
BlockStdAdj	is the variance associated with the Block. If BlockStdAdj is zero it means we are treating the block effect as a fixed effect. If BlockStdAdj>0, we treat the block effect as a random effect and increase the variance of Block 2 data.
StdExp	defines whether any additional heterogeneity is introduced between families. The value (set to 0 or 0.5 for our simulations) is used when we generate a deviation to be added to the control mean (control rate for gamma data) for each family. The deviation is generated from a Normal distribution with mean 0 and standard deviation=0.5. If StdExp=0 we do not add any deviations to the mean.
MAMethod	defines the method used for meta-analysis
returnES	This determines the format of the output. If returnES=FALSE it returns the summary meta-analysis statistics otherwise it returns the effect sizes and their variances for each experiment in the family
AlwaysTwoSidedTests	If this parameter is TRUE, the function always does two-sided tests. IF the parameter is FALSE, the function does two-sided statistical tests if the difference between treatment groups is 0, if the difference is not 0, it does one-sided tests

**Value**

If returnES is FALSE, the function returns the summary meta-analysis summary statistics otherwise the function returns the effect sizes for each experiment

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

as.data.frame(NP4GMetaAnalysisSimulation(mean=0,sd=1,diff=0.8,GroupSize=5,Exp=5,type="n",
alpha=0.05,seed=457,StdAdj=0,BlockEffect=0.5,BlockStdAdj=0,StdExp=0,MAMethod="FE",returnES=TRUE))
# MeanExp VarExp StdMD df tval t.sig Cliffd Cliffdvar PHat PHatvar PH..
#1 1.0761565 1.3874542 0.91362108 14.42773 2.0429188 TRUE 0.52 0.05530667 0.76 0.0132 13..
#2 0.1012680 0.9779431 0.10240368 12.74930 0.2289816 FALSE 0.20 0.09048000 0.60 0.0224 10..
#3 1.2100986 0.9909894 1.21558760 11.16850 2.7181365 TRUE 0.64 0.04720000 0.82 0.0110 13..
#4 -0.1452027 2.3106703 -0.09552252 11.93764 -0.2135949 FALSE 0.04 0.09888000 0.52 0.0244 10..
#5 1.1701075 0.9623530 1.19277505 12.72802 2.6671261 TRUE 0.52 0.05048000 0.76 0.0124 15..
# StdMDAdj StdMDAdjvar.exact StdMDAdjvar.approx StdMDvar.exact StdMDvar.approx
#1 0.86514731 0.2357247 0.2025998 0.2664156 0.2259389
#2 0.09623845 0.2098977 0.1769637 0.2377103 0.2003632
#3 1.13176658 0.2732955 0.2230773 0.3262821 0.2573441
#4 -0.08937076 0.2106627 0.1753619 0.2407210 0.2003345
#5 1.12084087 0.2623764 0.2201822 0.3050637 0.2493511
as.data.frame(NP4GMetaAnalysisSimulation(mean=0,sd=1,diff=0.8,GroupSize=5,Exp=5,type="n",
alpha=0.05,seed=457,StdAdj=0,BlockEffect=0.5,BlockStdAdj=0,StdExp=0,MAMethod="FE",returnES=FALSE))
# NumExp GroupSize AveCliffd AveCliffdvar AveCliffdsig Avephat Avephatvar Avephatsig AveMDStd..
#1 5 5 0.384 0.01369387 TRUE 0.692 0.003336 TRUE 0.5927084..
# AveMDStdsig Maphat Maphatvar MAphatsig MACliffd MACliffdvar MACliffdsig StdMDAdjUnweigh..
#1 TRUE 0.7253858 0.00300356 TRUE 0.4471125 0.01246219 TRUE 0.6249..
# StdMDAdjUnweightedvar StdMDAdjUnweightedvar StdMDUnweighted StdMDUnweightedvar StdMDUnweight..
#1 0.04220968 TRUE 0.665773 0.04366035 TRUE
# HedgesMA.Weighted HedgesMA.Weightedvar HedgesMA.Weightedsig StdMDAdjMAexact StdMDAdjMAexactvar
#1 0.6250243 0.04574766 TRUE 0.5709401 0.04711703
# StdMDAdjMAexactsig StdMDAdjMAapprox StdMDAdjMAapproxvar StdMDAdjMAapproxsig StdMDMAapprox St..
#1 TRUE 0.5715637 0.03950437 TRUE 0.6090632 0..
# StdMDMAapproxsig StdMDMAexact StdMDMAexactvar StdMDMAexactsig
#1 TRUE 0.6013198 0.05417894 TRUE

```

---

```
percentageInaccuracyOfLargeSampleVarianceApproximation
```

```
percentageInaccuracyOfLargeSampleVarianceApproximation
```

---

**Description**

Plot the extent of inaccuracy using the large sample approximate effect size variance on 4 related graphs corresponding to the four different correlation values. Plot visualizes the relationship between sample size and effect size and the percentage inaccuracy of the large sample variance approximation. Function is used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

**Usage**

```
percentageInaccuracyOfLargeSampleVarianceApproximation(data)
```

**Arguments**

data - data behind the plot returned by `getSimulatedCrossoverDataSets()` or stored in `reproducer::KitchenhamMadeyski.SimulatedCrossoverDataSets`

**Value**

plot described in description

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
data <- KitchenhamMadeyski.SimulatedCrossoverDataSets
myPlot <- percentageInaccuracyOfLargeSampleVarianceApproximation(data)
```

---

PHat.test

*PHat.test*

---

**Description**

This function provides single-sided and two-sided tests of the probability of superiority (phat).

**Usage**

```
PHat.test(x, y, alpha = 0.05, alternative = "two.sided", sigfig = -1)
```

**Arguments**

x	The data from one group
y	The data from the alternative group
alpha	The significance level of tests which also controls the values of the confidence interval (default 0.05)
alternative	This defines whether a one-sided test or a two-sided (default) test is required. For a one-sided test use parameter values 'greater' or 'less' to define whether the d-value should be greater or less than zero.
sigfig	is the number of significant digits in the data.

**Value**

The values of phat and its standard error, the t-value, its pvalue and the upper and lower confidence interval.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

set.seed(456)
x <- rnorm(10, 0, 1)
y <- rnorm(10, 0.8, 1)
PHat.test(x, y, alpha = .05, alternative = "greater", sigfig = -1)
# A tibble: 1 x 8
#   phat sqse.phat phat.df phat.tvalue phat.pvalue phat.ci.lower phat.ci.upper phat.sig
#   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <lg1>
# 1  0.79    0.0118    13.6     2.67    0.00924    0.599     1 TRUE
PHat.test(x, y, alpha = .05, alternative = "two.sided", sigfig = -1)
# A tibble: 1 x 8
#   phat sqse.phat phat.df phat.tvalue phat.pvalue phat.ci.lower phat.ci.upper phat.sig
#   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <lg1>
# 1  0.79    0.0118    13.6     2.67    0.0185     0.557     1 TRUE

```

---

PHatonesidedTestStatistics

*PHatonesidedTestStatistics*


---

**Description**

This function is a helper function for meta-analysis of experiments using PHat as an effect size. It returns the  $100*(1-\alpha)$

**Usage**

```

PHatonesidedTestStatistics(
  effectsize,
  effectsize.variance,
  effectsize.df = 0,
  alpha = 0.05,
  alternative = "greater"
)

```

**Arguments**

effectsize	The overall estimate of the centralized PHat (i.e. Phat-0.5) from a group of effect sizes to be meta-analysed
effectsize.variance	The estimate of the variance of the overall estimate of PHat
effectsize.df	The total degrees of freedom for the set of effect sizes. If effectsize.df>0, the confidence intervals, pvalues and significance test use the t-distribution probability values. If effectsize.df=0 (default), the confidence intervals, the pvalues and significance test use the normal distribution probability values.
alpha	The significance level (default 0.05) used to control the significance tests and calculation of confidence limits.
alternative	Specifies the type of significance test and can take the values "less" or "greater" (default).

**Value**

ES.test The value of the t-statistic

ES.pvalue The p-value of the two-sided t-test if the parameter d.df>0, or the normal probability value if d.df=0

ES.sig The significance of the statistical test of the d.tvalue return value at the alpha level for one sided tests and aplha/2 for two sided tests as specified by the input parameter alternative.

ES.ci.lower The lower  $100*(1-\alpha/2)$

ES.ci.upper The upper  $100*(1-\alpha/2)$

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
PHatES=mean(c(0.92,0.6,0.48,0.72,0.88))-0.5
PHatESvar=sum(c(0.01,0.04,0.05,0.04,0.01))/25
PHatdf=sum(c(6.63,6.63,5.08,5.61,8))
#PHatonesidedTestStatistics(effectsize=PHatES, effectsize.variance=PHatESvar, effectsize.df=PHatdf)
# A tibble: 1 x 5
#   ES.test ES.pvalue ES.sig ES.ci.lower ES.ci.upper
#   <dbl>    <dbl> <lgl>    <dbl>    <dbl>
#1    2.84    0.00389 TRUE      0.0888    0.351
#PHatonesidedTestStatistics(effectsize=PHatES, effectsize.variance=PHatESvar, effectsize.df=0,
# alternative="less")
# A tibble: 1 x 5
#   ES.test ES.pvalue ES.sig ES.ci.lower ES.ci.upper
#   <dbl>    <dbl> <lgl>    <dbl>    <dbl>
#1    2.84    0.998 FALSE    0.0926    0.347
```

---

PHattwosidedTestStatistics

*PHattwosidedTestStatistics*

---

**Description**

This function is a helper function for meta-analysis of experiments using PHat as an effect size. It returns the  $100*(1-\alpha/2)$

**Usage**

```
PHattwosidedTestStatistics(
  effectsize,
  effectsize.variance,
  effectsize.df = 0,
  alpha = 0.05
)
```

**Arguments**

effectsize	The overall estimate of the centralized PHat (ie.Phat-0.5) from a group of effect sizes to be meta-analysed
effectsize.variance	The estimate of the variance of the overall estimate ofPHat
effectsize.df	The total degrees of freedom for the set of effect sizes. If effectsize.df>0, the confidence intervals, pvalues and significance test use the t-distribution probability values. If effectsize.df=0 (default), the confidence intervals, the pvalues and significance test use the normal distribution probability values.
alpha	The significance level (default 0.05) used to control the significance tests and calculation of confidence limits.

**Value**

ES.test	The value of the t-statistic
ES.pvalue	The p-value of the two-sided t-test if the parameter d.df>0, or the normal probability value if d.df=0
ES.sig	The significance of the statistical test of the d.tvalue return value at the alpha level for one sided tests and aplha/2 for two sided tests as specified by the input parameter alternative.
ES.ci.lower	The lower $100*(1-\alpha/2)$
ES.ci.upper	The upper $100*(1-\alpha/2)$

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
PHatES=mean(c(0.92,0.6,0.48,0.72,0.88))-0.5
PHatESvar=sum(c(0.01,0.04,0.05,0.04,0.01))/25
PHatdf=sum(c(6.63,6.63,5.08,5.61,8))
#PHattwosidedTestStatistics(effectsize=PHatES, effectsize.variance=PHatESvar)
# A tibble: 1 x 5
#   ES.test ES.pvalue ES.sig ES.ci.lower ES.ci.upper
#   <dbl>   <dbl> <lg1>   <dbl>   <dbl>
# 1    2.84  0.00451 TRUE     0.0682    0.372
# PHattwosidedTestStatistics(effectsize=PHatES, effectsize.variance=PHatESvar, effectsize.df=PHatdf)
# A tibble: 1 x 5
#   ES.test ES.pvalue ES.sig ES.ci.lower ES.ci.upper
#   <dbl>   <dbl> <lg1>   <dbl>   <dbl>
# 1    2.84  0.00778 TRUE     0.0622    0.378
```

---

```
plotOutcomesForIndividualsInEachSequenceGroup  
  plotOutcomesForIndividualsInEachSequenceGroup
```

---

**Description**

Function to plot a figure on the outcomes for individuals in each sequence group used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham

**Usage**

```
plotOutcomesForIndividualsInEachSequenceGroup(  
  var,  
  covar,  
  meanA1,  
  treatmentDiff,  
  periodEffect,  
  numOfSamples  
)
```

**Arguments**

var	Variance among subjects is a sum of the between subjects variance and the within subjects variance
covar	Covariance equal to the between subjects variance
meanA1	Mean for treatment sequence A1
treatmentDiff	technique effect which is the difference between the effect of technique A and technique B
periodEffect	Period effect which is the difference between period 1 and period 2
numOfSamples	Number of samples ('rows' of data) required for each technique and period

**Value**

plot

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
myPlot <- plotOutcomesForIndividualsInEachSequenceGroup(25, 18.75, 50, 10, 5, 15)
```

---

 PrepareForMetaAnalysisGtoR

*PrepareForMetaAnalysisGtoR*


---

### Description

This function calculates the standardized effect sizes and their confidence intervals, the equivalence point biserial effect size and the  $Z_r$  and  $\text{var}(Z_r)$  needed for input into the metafor rma function (meta analysis). In this function the point bi-serial effect size is based on the adjusted Hedges  $g$  value. The function uses the Hedges  $g$  to  $r$  transformation to prepare for meta-analysing the data where the mean values, the standard deviations, and the number of observations are available.

### Usage

```
PrepareForMetaAnalysisGtoR(Mc, Mt, SDc, SDt, Nc, Nt)
```

### Arguments

Mc	is a vector containing the mean value of the control group for each experiment.
Mt	is a vector containing the mean value of the treatment group for each experiment.
SDc	is a vector of the standard deviations of the control group for each experiment.
SDt	is a vector of the standard deviations of the the treatment group for each experiment.
Nc	is a vector containing the the number of observations (participants) in the control group for each experiment.
Nt	is a vector of the number of observations (participants) in the treatment group for each experiment.

### Value

data frame incl. calculated effect sizes (Hedges'  $g$ , Hedges'  $g$  adjusted), upper and lower confidence bounds on Hedges'  $g$ ,  $z_r$ ,  $v_i$  - variance of  $z_r$ ,  $r$  and  $p$ value

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
PrepareForMetaAnalysisGtoR(c(10, 10), c(12, 14), c(4, 4), c(4, 4), c(20, 20), c(40, 40))
# HGvalues.Hg HGvalues.HgAdjusted Hgupper Hglower zr vi r pvalue
# 0.5 0.4935018 1.082017 -0.06156572 0.2305901 0.01754386 0.2265882 0.0816981743
# 1.0 0.9870036 1.634701 0.40620071 0.4499419 0.01754386 0.4218513 0.0006813222
```

---

printXTable	<i>printXTable</i>
-------------	--------------------

---

**Description**

print data table using xtable R package

**Usage**

```
printXTable(  
  data,  
  selectedColumns,  
  tableType = "latex",  
  alignCells,  
  digits,  
  caption,  
  label,  
  fontSize,  
  captionPlacement = "bottom",  
  alignHeader  
)
```

**Arguments**

data	Data structure including columns to be printed.
selectedColumns	Columns selected to be printed.
tableType	Type of table to produce. Possible values are 'latex' or 'html'. Default value is 'latex'.
alignCells	Defines how to align data cells.
digits	Defines the number of decimal points in each column.
caption	Caption of the table.
label	Label of the table.
fontSize	Size of the font used to produce a table.
captionPlacement	The caption will be have placed at the bottom of the table if captionPlacement is 'bottom' and at the top of the table if it equals 'top'. Default value is 'bottom'.
alignHeader	Defines how to align column headers of a table.

**Value**

A table generated on the fly on a basis of passed data (data, selectedColumns etc.).

**Author(s)**

Lech Madeyski

**Examples**

```
d <- reproducer::MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR
printXTable(d, "Study", "latex", "cc", 0, "C", "L", "tiny", "top", "1")
```

---

`proportionOfSignificantTValuesUsingCorrectAnalysis`  
*proportionOfSignificantTValuesUsingCorrectAnalysis*

---

**Description**

Plots visualize the relationship between sample size, effect size and the proportion of significant t-values using the correct analysis. Function is used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

**Usage**

```
proportionOfSignificantTValuesUsingCorrectAnalysis(data)
```

**Arguments**

`data` - data behind the plot returned by `getSimulatedCrossoverDataSets()` or stored in `reproducer::KitchenhamMadeyski.SimulatedCrossoverDataSets`

**Value**

plot described in description

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
data <- KitchenhamMadeyski.SimulatedCrossoverDataSets
myPlot <- proportionOfSignificantTValuesUsingCorrectAnalysis(data)
```

---

proportionOfSignificantTValuesUsingIncorrectAnalysis  
*proportionOfSignificantTValuesUsingIncorrectAnalysis*

---

**Description**

Plots visualize the relationship between sample size, effect size and the proportion of significant t-values using the incorrect analysis. Function is used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

**Usage**

```
proportionOfSignificantTValuesUsingIncorrectAnalysis(data)
```

**Arguments**

data - data behind the plot returned by `getSimulatedCrossoverDataSets()` or stored in `reproducer::KitchenhamMadeyski.SimulatedCrossoverDataSets`

**Value**

plot described in description

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
data <- KitchenhamMadeyski.SimulatedCrossoverDataSets  
myPlot <- proportionOfSignificantTValuesUsingIncorrectAnalysis(data)
```

---

RandomExperimentSimulations  
*RandomExperimentSimulations*

---

**Description**

This function performs multiple simulations of two-group balanced experiments for one of four distributions and a specific group size. It identifies the average value of  $\phi$ , Cliff's  $d$  and their variances. It either returns the effect sizes for each non-parametric effect size or it reports the number of times the each non-parametric effect size is assessed to be significantly different from zero. We also present the values for the t-test as a comparison. For log-normal data the results of analysing the transformed data are also reported.

**Usage**

```

RandomExperimentSimulations(
  mean,
  sd,
  diff,
  N,
  reps,
  type = "n",
  seed = 123,
  StdAdj = 0,
  alpha = 0.05,
  returnData = FALSE,
  AlwaysTwoSidedTests = FALSE
)

```

**Arguments**

mean	The default mean used for both groups (one treatment group and one control group). It can be changed for the treatment group using the parameter diff
sd	This is the default spread for both groups. It must be a real value greater than 0. It can be adjusted for the treatment group using the parameter StdAdj
diff	This is added to the treatment group mean. It can be a real value and can take the value zero.
N	this is the number of observations in each group. It must be an integer greater than 3.
reps	this identifies the number of times each experiment simulation is replicated.
type	this specifies the underlying distribution used to generate the data. It takes the values 'n' for a normal distribution, 'l' for lognormal distribution, 'g' for a gamma distribution, 'lap' for a Laplace distribution.
seed	This specifies the initial seed for the set of replications (default 123).
StdAdj	this specifies the extent of variance instability introduced by the treatment and it must be non-negative but can be 0.
alpha	This specifies the level of significance used for statistical tests (default 0.05).
returnData	If TRUE, the function returns the individual effect sizes and their variances, otherwise it returns summary statistics (default FALSE).
AlwaysTwoSidedTests	If set to FALSE (default) the algorithm uses one-sided tests if diff!=0 and two-sided tests if diff=0. If set to TRUE the algorithm always uses two-sided tests.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

## Examples

```

as.data.frame(
  RandomExperimentSimulations(
    mean = 0, sd = 1, diff = 0.5, N = 20, reps = 50, type = "n",
    seed = 123, StdAdj = 0, alpha = 0.05))
#   phat   phatvar sigphat emp.phat.var    d    dvar sigd
# 1 0.636675 0.007980072  0.38 0.006413391 0.27335 0.03257962 0.36
#   emp.d.var  tpower    ES Variance   StdES  MedDiff
#1 0.02565356   0.41 0.4849609 0.988889 0.4982554 0.4666802
#as.data.frame(
# RandomExperimentSimulations(
#   mean = 0, sd = 1, diff = 0.5, N = 20, reps = 500, type = "n",
#   seed = 123, StdAdj = 0, alpha = 0.05))
#   phat   phatvar sigphat emp.phat.var    d    dvar sigd emp.d.var
# 1 0.63915 0.007925803  0.444 0.007904962 0.2783 0.03235111 0.414 0.03161985
#   tpower    ES Variance
# 1   0.444 0.4999034 1.002012
#   StdES  MedDiff
# 1 0.5099792 0.4901394

#as.data.frame(
# RandomExperimentSimulations(
#   mean = 0, sd = 1, diff = 0.2, N = 20, reps = 500, type = "n",
#   seed = 123, StdAdj = 0, alpha = 0.05, AlwaysTwoSidedTests = TRUE))
#   phat   phatvar sigphat emp.phat.var    d    dvar sigd emp.d.var
# 1 0.55762 0.008596555  0.092 0.008457325 0.11524 0.03505528 0.076 0.0338293
#   tpower    ES Variance   StdES  MedDiff
# 1   0.1 0.1999034 1.002012 0.2043908 0.1901394

#as.data.frame(
# RandomExperimentSimulations(
#   mean = 0, sd = 1, diff = 0.2, N = 20, reps = 500, type = "n",
#   seed = 123, StdAdj = 0, alpha = 0.05, AlwaysTwoSidedTests = FALSE))
#   phat   phatvar sigphat emp.phat.var    d    dvar sigd emp.d.var
# 1 0.55762 0.008596555  0.154 0.008457325 0.11524 0.03505528 0.146 0.0338293
#   tpower    ES Variance
# 1   0.16 0.1999034 1.002012
#   StdES  MedDiff
# 1 0.2043908 0.1901394

RandomExperimentSimulations(
  mean = 0, sd = 1, diff = 0.5, N = 20, reps = 10, type = "l", seed = 456,
  StdAdj = 0, alpha = 0.05, returnData = TRUE, AlwaysTwoSidedTests = FALSE)
# A tibble: 10 x 6
#   Cliffd CliffdSig PHat PHatSig StdES ESSig
#   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
# 1 -0.185     0 0.407     0 -0.246     0
# 2 -0.08      0 0.46      0 0.185     0
# 3 0.1         0 0.55      0 0.149     0
# 4 0.42        1 0.71      1 0.885     1
# 5 0.51        1 0.755     1 0.827     1
# 6 0.185      0 0.592     0 0.628     1

```

```
# 7 0.465      1 0.732      1 0.818      1
# 8 0.42       1 0.71       1 0.341      0
# 9 0.37       1 0.685      1 0.419      0
# 10 0.115     0 0.557      0 0.273      0
```

---

RandomizedBlockDesignEffectSizes

*RandomizedBlockDesignEffectSizes*

---

### Description

This function finds the theoretical effect sizes for a four-group randomized block experiments assuming one of four different underlying distributions specified by the type parameter. The design assumes two blocks each comprising a control and treatment group. If required a fixed Blocking effect is added to the mean for Block 2.

### Usage

```
RandomizedBlockDesignEffectSizes(
  m1,
  std1,
  m2,
  std2,
  m3,
  std3,
  m4,
  std4,
  BE = 0,
  type = "n"
)
```

### Arguments

m1	The theoretical mean for the control group in Block 1
std1	The theoretical variance for the control group in Block 1
m2	The theoretical mean for the treatment group in Block 1
std2	The theoretical variance for the treatment group in Block 1
m3	The theoretical mean for the control group in Block 2
std3	The theoretical variance for the control group in Block 2
m4	The theoretical mean for the treatment group in Block 2
std4	The theoretical variance for the treatment group in Block 2
BE	A fixed block effect to be added to the Block 2 mean values.
type	String identifying the distribution, 'n' for normal, 'ln' for lognormal, 'lap' for Laplace, 'g' for Gamma

**Value**

dataframe holding the expected unstandardized mean difference effect size, the pooled within group variance, the standardized effect size and the point bi-serial correlation.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
RandomizedBlockDesignEffectSizes(m1=0, std1=1, m2=1, std2=1, m3=0, std3=1, m4=1,
  std4=1, BE = 1, type = 'n')
# ES Var StdES      rPBS
#1 1 1 1 0.4472136
RandomizedBlockDesignEffectSizes(m1=0, std1=1, m2=1, std2=1, m3=0, std3=1, m4=1,
  std4=1, BE = 1, type = 'l')
#      ES      Var      StdES      rPBS
#1 5.266886 82.17791 0.5810004 0.2789675
RandomizedBlockDesignEffectSizes(
  m1=0, std1=1, m2=0.266, std2=1, m3=0, std3=1, m4=0.266, std4=1, BE = 0, type = 'l')
#      ES      Var      StdES      rPBS
#1 0.5024232 6.310995 0.1999957 0.09950162
```

---

RandomizedBlocksAnalysis

*RandomizedBlocksAnalysis*

---

**Description**

The function performs a heteroscedastic test of a two treatment by J blocks randomized blocks effect size. The data are assumed to be stored in  $x$  in list mode. All groups are assumed to be independent. Missing values are not permitted.

**Usage**

```
RandomizedBlocksAnalysis(
  x,
  con = c(-0.5, 0.5, -0.5, 0.5),
  alpha = 0.05,
  alternative = "two.sided"
)
```

**Arguments**

$x$  the structure holding the data. In list format, for a 2 treatment by J block randomized blocks experiments, there are  $2J$  list elements each one specifying the outcome for a specific block and a specific treatment.

con is a 2J list containing the contrast coefficients that are used to calculate the mean effect size.

alpha is the Type 1 error level used for the test of significance (default 0.05)

alternative The type of statistical test. Valid values are one of c('two.sided', 'greater', 'less')

### Value

The t-test and its associated metrics (i.e., critical value standard error and degrees of freedom) and the estimate of the contrast with its upper and lower confidence interval bounds and p-value.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
set.seed(123)
x <- list()
x[[1]] <- rnorm(10, 0, 1)
x[[2]] <- rnorm(10, 0.8, 1)
x[[3]] <- rnorm(10, 0.5, 1)
x[[4]] <- rnorm(10, 1.3, 1)
vec <- c(-1, 1, -1, 1) / 2
RandomizedBlocksAnalysis(x, con = vec, alpha = 0.05)
# $n
# [1] 10 10 10 10
# $test
#      test      crit      se      df
# [1,] 4.432644 2.038622 0.2798104 31.33793
# $psihat
#      psihat ci.lower ci.upper      p.value
# [1,] 1.2403 0.6698721 1.810728 0.0001062952
# $sig
# [1] TRUE
RandomizedBlocksAnalysis(x, con=vec, alpha=0.05, alternative='greater')
# n
# [1] 10 10 10 10
# $test
#      test      crit      se      df
# [1,] 4.432644 1.694956 0.2798104 31.33793
# $psihat
#      psihat ci.lower ci.upper      p.value
# [1,] 1.2403 0.7660336      Inf 5.314762e-05
# $sig
# [1] TRUE
RandomizedBlocksAnalysis(x, con=-vec, alpha=0.05, alternative='greater')
# $n
# [1] 10 10 10 10
# $test
#      test      crit      se      df
# [1,] -4.432644 1.694956 0.2798104 31.33793
# $psihat
```

```

#      psihat ci.lower ci.upper  p.value
#[1,] -1.2403 -1.714566      Inf 0.9999469
# $sig
#[1] FALSE
x[[5]]=rnorm(10,-0.2,1)
x[[6]]=rnorm(10,0.6,1)
vec=c(1,-1,1,-1,1,-1)/3
RandomizedBlocksAnalysis(x,con=vec,alpha=0.05,alternative='less')
# $n
#[1] 10 10 10 10 10 10
# $test
#      test      crit      se      df
#[1,] -4.946987  1.677021  0.236575  48.29776
# $psihat
#      psihat ci.lower  ci.upper  p.value
#[1,] -1.170334      -Inf -0.7735925  4.76961e-06
# $sig
#[1] TRUE

```

---

#### RandomizedBlocksExperimentSimulations

*title RandomizedBlocksExperimentSimulations description This function performs multiple simulations of 4 group balanced randomised Block experiments with two control groups and two treatment groups where one control group and one treatment group are assigned to block 1 and the other control group and treatment group are assigned to block 2. The simulations are based on one of four distributions and a specific group size. The function identifies the average value of the non-parametric effect sizes P-hat, Cliff' d and their variances and whether or not the statistics were significant at the 0.05 level. We also present the values of the t-test as a comparison.*

---

#### Description

title RandomizedBlocksExperimentSimulations description This function performs multiple simulations of 4 group balanced randomised Block experiments with two control groups and two treatment groups where one control group and one treatment group are assigned to block 1 and the other control group and treatment group are assigned to block 2. The simulations are based on one of four distributions and a specific group size. The function identifies the average value of the non-parametric effect sizes P-hat, Cliff' d and their variances and whether or not the statistics were significant at the 0.05 level. We also present the values of the t-test as a comparison.

#### Usage

```

RandomizedBlocksExperimentSimulations(
  mean,
  sd,
  diff,

```

```

N,
reps,
type = "n",
alpha = 0.05,
Blockmean = 0,
BlockStdAdj = 0,
StdAdj = 0,
seed = 123,
returnData = FALSE,
AlwaysTwoSidedTests = FALSE
)

```

### Arguments

mean	The default mean for all 4 groups. The default for the two treatment groups can be altered using the parameter diff and the block mean for block 2 can be altered using the parameter Blockmean.
sd	The default spread for all 4 groups. It must be a real value greater than 0. It can be altered for treatment groups using the parameter StdAdj and for Block 2 groups using BlockStdAdj
diff	The is added to the parameter mean, to define the mean of the other treatment group. It can be a real value ad can take the value zero.
N	this is the number of observations in each group. It must be an integer greater than 3.
reps	this identifies the number of times the simulation is replicated.
type	this specifies the underlying distribution used to generate the data. it takes the values 'n' for a normal distribution, 'l' for lognormal distribution, 'g' for a gamma distribution, 'lap' for a Laplace distribution.
alpha	is the Type 1 error level used for constructing confidence intervals and statistical tests (default 0.05)
Blockmean	is the effect of having two different blocks
BlockStdAdj	is the variance associated with the Block mean. If Blockvar is zero it means we are treat the block effect as a fixed effect. If BlockStdAdj>0, we treat the block effect as a random effect.
StdAdj	The value used to introduce heterogeneity into the treatment groups variance if required.
seed	this specifies the seed value for the simulations and allows the experiment to be repeated.
returnData	if TRUE the function returns the generated data otherwise it returns summary statistics.
AlwaysTwoSidedTests	A boolean variable. If TRUE the simulations always used two-sided tests otherwise the simulations use one-sided tests. return depending on the parameter returnData it returns the generated nonparametric and parametric values and their statistical significance (1 for significant, 0 for not significant) or the summary statistics (averages of effect sizes and their variances and the proportion significant effect sizes)

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```

as.data.frame(
  RandomizedBlocksExperimentSimulations(
    mean = 0, sd = 1, diff = 0.5, N = 10, reps = 50, type = "n",
    alpha = 0.05, Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0, seed = 123,
    AlwaysTwoSidedTests = FALSE))
#   phat   varphat sigphat emp.phat.var   d     vard sigd emp.d.var
#1 0.64415 0.008271389  0.45 0.005888917 0.2883 0.0340919 0.41 0.02355567
#   StdES     ES   Var emp.StdESvar MedDiff tpower
#1 0.5413961 0.5264245 0.9904726 0.08811262 0.5538213 0.46
#as.data.frame(
# RandomizedBlocksExperimentSimulations(
#   mean = 0, sd = 1, diff = 0.5, N = 10, reps = 500, type = "n",
#   alpha = 0.05, Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0, seed = 123,
#   AlwaysTwoSidedTests = FALSE))
#   phat   varphat   sigphat emp.phat.var   d     vard     sigd emp.d.var
# 1 0.63967 0.008322856 0.436 0.007728698 0.27934 0.03430328 0.416 0.03091479
#   StdES     ES   Var emp.StdESvar MedDiff
# 1 0.5130732 0.5029075 1.001602 0.1116687 0.5110203
# tpower
# 1 0.45

#as.data.frame(
# RandomizedBlocksExperimentSimulations(
#   mean = 0, sd = 1, diff = 0.5, N = 10, reps = 500, type = "n",
#   alpha = 0.05, Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0, seed = 123,
#   AlwaysTwoSidedTests = TRUE))
#   phat   varphat sigphat emp.phat.var   d     vard     sigd
# 1 0.63967 0.008322856 0.326 0.007728698 0.27934 0.03430328 0.282
#   emp.d.var   StdES     ES   Var
# 1 0.03091479 0.5130732 0.5029075 1.001602
# emp.StdESvar MedDiff tpower
# 1 0.1116687 0.5110203 0.334

#RandomizedBlocksExperimentSimulations(
#   mean = 0, sd = 1, diff = 0.5, N = 10, reps = 10, type = "n", alpha = 0.05,
#   #Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0, seed = 123, returnData = TRUE)
# A tibble: 10 x 6
#   Cliffd PHat StdES CliffdSig PHatSig ESSig
#   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
# 1 0.58 0.79 1.06 1 1 1
# 2 0.21 0.605 0.383 0 0 0
# 3 0.37 0.685 0.761 1 1 1
# 4 0.44 0.72 0.821 1 1 1
# 5 0.13 0.565 0.240 0 0 0
# 6 0.16 0.58 0.222 0 0 0
# 7 0.38 0.69 0.580 1 1 1
# 8 0.48 0.74 0.882 1 1 1

```

```
# 9  0.11 0.555 0.181      0      0      0
# 10 -0.03 0.485 0.124     0      0      0
```

---

RandomizedDesignEffectSizes

*RandomizedDesignEffectSizes*

---

## Description

This function creates the theoretical effect sizes for data from one of four different distributions for specified parameter values for the distribution specified by the type parameter. It assumes there are two samples, one corresponding to a control group and the other to the treatment group. It returns the theoretical effect sizes for a fully randomized experiment.

## Usage

```
RandomizedDesignEffectSizes(m1, std1, m2, std2, type = "n")
```

## Arguments

m1	The theoretical mean for the control group
std1	The theoretical variance for the control group
m2	The theoretical mean for the treatment group
std2	The theoretical variance for the treatment group
type	String identifying the distribution, 'n' for normal, 'ln' for lognormal, 'lap' for Laplace, 'g' for Gamma

## Value

dataframe containing the expected values of the unstandardized mean difference effect size, the pooled within group variance, the standardized mean difference effect size and the point bi-serial correlation.

## Author(s)

Barbara Kitchenham and Lech Madeyski

## Examples

```
RandomizedDesignEffectSizes(m1=0, std1=1, m2=1, std2=3, type = 'n')
# ES Var StdES rPBS
#1 1 5 0.4472136 0.2182179
RandomizedDesignEffectSizes(m1=0, std1=1, m2=1, std2=3, type = 'l')
# ES Var StdES rPBS
#1 243.0432 242552663 0.0156056 0.007802562
RandomizedDesignEffectSizes(m1=0, std1=1, m2=0.266, std2=1, type = 'l')
# ES Var StdES rPBS
# 1 0.5024232 6.310995 0.1999957 0.09950162
```

---

readExcelSheet	<i>readExcelSheet</i>
----------------	-----------------------

---

**Description**

Function reads data from an Excel file from a specified sheet

**Usage**

```
readExcelSheet(path, sheet, colNames)
```

**Arguments**

path	Path to an Excel file, e.g. /User/lma/datasets/MyDataSet.xls
sheet	Name of a sheet within an Excel file we want to read
colNames	If TRUE, first row of data will be used as column names.

**Author(s)**

Lech Madeyski

**Examples**

```
myPath <- system.file("extdata", "DataSet.xlsx", package = "reproducer")
Madeyski15SQJ.NDC <- readExcelSheet(path = myPath, sheet = "Madeyski15SQJ.NDC", colNames = TRUE)
```

---

reproduceForestPlotRandomEffects	<i>reproduceForestPlotRandomEffects()</i>
----------------------------------	---

---

**Description**

Function reproduces Forest Plot of a Random-Effects Meta-analysis of Mean Differences.

**Usage**

```
reproduceForestPlotRandomEffects()
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceForestPlotRandomEffects()
```

---

```
reproduceMixedEffectsAnalysisWithEstimatedVarianceAndExperimentalDesignModerator  
reproduceMixedEffectsAnalysisWithEstimatedVarianceAndExperimentalDesignModerator()
```

---

**Description**

Function reproduces Mixed-Effects Analysis using Subject Specific Estimated Variance with Experimental Design as a Moderator.

**Usage**

```
reproduceMixedEffectsAnalysisWithEstimatedVarianceAndExperimentalDesignModerator(  
  
)
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceMixedEffectsAnalysisWithEstimatedVarianceAndExperimentalDesignModerator()
```

---

```
reproduceMixedEffectsAnalysisWithExperimentalDesignModerator  
reproduceMixedEffectsAnalysisWithExperimentalDesignModerator()
```

---

**Description**

Function reproduces Mixed-Effects Analysis with Experimental Design as a Moderator.

**Usage**

```
reproduceMixedEffectsAnalysisWithExperimentalDesignModerator()
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceMixedEffectsAnalysisWithExperimentalDesignModerator()
```

---

```
reproduceMixedEffectsForestPlotWithExperimentalDesignModerator  
  reproduceMixedEffectsForestPlotWithExperimentalDesignModerator()
```

---

**Description**

Function reproduces Forest Plot of a Mixed Effects Meta-analysis of Mean Differences with Experimental Design as a Moderator Variable.

**Usage**

```
reproduceMixedEffectsForestPlotWithExperimentalDesignModerator()
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceMixedEffectsForestPlotWithExperimentalDesignModerator()
```

---

```
reproduceSimulationResultsBasedOn500Reps1000Obs  
  reproduceSimulationResultsBasedOn500Reps1000Obs
```

---

**Description**

Function to calculate simulation results based on 500 repetitions of 1000 observation samples. Function is used in a paper 'Effect Sizes and their Variance for AB/BA Crossover Design Studies' by Lech Madeyski and Barbara Kitchenham.

**Usage**

```
reproduceSimulationResultsBasedOn500Reps1000Obs()
```

**Value**

data frame including the following simulation results: # treatmentEffect.Ave - Average Technique Effect # dRM.Ave - Average dRM # dRM.Var - Variance of dRM # dRM.Var.Ave - Average of var(dRM) # dRM.Var.ModerateSampleSizeApprox - # dIG.Ave - Average dIG # dIG.Var - Variance of dIG # dIG.Var.Ave - Average of var(dIG) # dIG.Var.ModerateSampleSizeApprox -

**Author(s)**

Lech Madeyski and Barbara Kitchenham

**Examples**

```
# return simulation results based on 500 repetitions of 1000 observation samples
simulationResultsTable500x1000 <- reproduceSimulationResultsBasedOn500Reps1000Obs()
```

---

```
reproduceTablesOfPaperMetaAnalysisForFamiliesOfExperiments
reproduceTablesOfPaperMetaAnalysisForFamiliesOfExperiments
```

---

**Description**

This function reproduces five of the output tables used in the systematic review paper 'Meta-analysis for Families of Experiments: A Systematic Review and Reproducibility Assessment'. It extracts the reported values for effect sizes, meta-analysis and descriptive statistics in the primary studies. It uses the descriptive statistics to re-calculate effect sizes and then performs a meta-analysis using the constructed effect sizes and compares the calculated values with the reported values.

**Usage**

```
reproduceTablesOfPaperMetaAnalysisForFamiliesOfExperiments()
```

**Value**

list incl. the data presented in five of the tables presented in the paper.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
rrData <- reproduceTablesOfPaperMetaAnalysisForFamiliesOfExperiments()
# Reproduce Table 'Overall Mean Values of Effect Sizes Reported and Calculated':
xtable::xtable(rrData$MAStats)
# Reproduce Table 'Calculated and Reported Effect Sizes':
xtable::xtable(rrData$ESdata)
# Report values for 3 papers that reported per document
rrData$MAStatsTP1 <- data.frame(rrData$MAStatsTP1, row.names = NULL)
rrData$ESTP1res <- data.frame(rrData$ESTP1res, row.names = NULL)
xtable::xtable(rrData$MAStatsTP1)
xtable::xtable(rrData$ESTP1res)
# Report extra results for Study 8
# Reproduce Table 'Calculating r_PB Effect Size from Probabilities'
xtable::xtable(rrData$GH2015extra)
```

---

```
reproduceTableWithEffectSizesBasedOnMeanDifferences  
  reproduceTableWithEffectSizesBasedOnMeanDifferences()
```

---

**Description**

Function reproduces Table, which shows the effect sizes based on mean differences.

**Usage**

```
reproduceTableWithEffectSizesBasedOnMeanDifferences()
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceTableWithEffectSizesBasedOnMeanDifferences()
```

---

```
reproduceTableWithPossibleModeratingFactors  
  reproduceTableWithPossibleModeratingFactors()
```

---

**Description**

Function reproduces Table with possible moderating factors.

**Usage**

```
reproduceTableWithPossibleModeratingFactors()
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceTableWithPossibleModeratingFactors()
```

---

```
reproduceTableWithSourceDataByCiolkowski  
  reproduceTableWithSourceDataByCiolkowski
```

---

**Description**

Function reproduces Table, which shows the effect sizes reported by Ciolkowski identifying the type of design used in each study.

**Usage**

```
reproduceTableWithSourceDataByCiolkowski()
```

**Author(s)**

Lech Madeyski

**Examples**

```
reproduceTableWithSourceDataByCiolkowski()
```

---

```
rSimulations      rSimulations
```

---

**Description**

This function simulates many datasets from the same bivariate distribution to investigate the distribution of correlations for specific sample sizes.

**Usage**

```
rSimulations(  
  mean,  
  var,  
  diff,  
  r,  
  N,  
  reps,  
  VarAdj = 0,  
  seed = 123,  
  returntSignificant = F,  
  returndata = F,  
  plothist = F  
)
```

**Arguments**

mean	The mean used for one of bivariate distributions - assumed to be the control condition in an experiment.
var	The variance used for both treatment groups. It must be a real value greater than 0.
diff	This value is added to the parameter mean to specify the mean for the other bivariate distribution - assumed to be the treatment condition in an experiment.
r	This specifies the correlation coefficient to be used for the bivariate normal distribution it must be a value in the range [-1,1].
N	The number of observations in each simulated bivariate normal data set.
reps	The number of bivariate data sets that will be simulated.
VarAdj	This value will be added to the variance of the treatment condition.
seed	This specifies the seed value for the simulations and allows the experiment to be repeated.
returntSignificant	If set to true the percentage of times the t-test delivered a value significant at the 0.05 level is reported (default returntSignificant=F).
returndata	If set to FALSE, the function returns the summary information across all the replications (default returndata=F). If set to TRUE the function outputs the r and variance ratio, and variance accuracy values generated in each replication.
plothist	If set to T, the function outputs a histogram of the r-values, the varprop values and the accuracy values (default plothist=F).

**Value**

output If returndata=F, the output returns summary information about the average of r and the variance properties across the replicated data sets. If returndata=T, the function returns the r-values obtained for each of the simulated data sets to gather with the variance ratio, the variance accuracy measure and a dummy variable indicating whether a test of significance between the mean values was significant (which is indicated by the dummy variable being set to 1) or not (which is indicated by the dummy variable being set to 0)

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
# output=rSimulations(mean=0,var=1,diff=0,r=0.25,N=4,reps=10000)
# reduced reps to pass CRAN time limits
output <- rSimulations(mean = 0, var = 1, diff = 0, r = 0.25, N = 4, reps = 1000)
output <- signif(output, 4)
output
# r.Mean r.Median Var.r PercentNegative Mean.VarProp Variance.VarProp ...
# 1 0.2132 0.3128 0.3126 34.21 0.5036 0.06046 ...
# output=rSimulations(mean=0,var=1,diff=0.8,r=0.25,N=60,reps=10000,returntSignificant=TRUE)
```

```

# reduced reps to pass CRAN time limits
output <- rSimulations(mean = 0, var = 1, diff = 0.8, r = 0.25, N = 60,
  reps = 1000, returntSignificant = TRUE)
output <- signif(output, 4)
output
#   r.Mean r.Median  Var.r PercentNegative Mean.VarProp Variance.VarProp ...
# 1 0.2492  0.2534 0.01529          2.62      0.5009      0.003897 ...
output <- rSimulations(mean = 0, var = 1, diff = 0, r = 0.25, N = 30, reps = 10, returndata = TRUE)
output
#   rvalues  VarProp VarAccuracy VarDiffAccuracy tSig
# 1 0.3981111 0.4276398 0.8630528 0.6974386 0
# 2 0.2104742 0.4994285 0.7812448 0.8224174 0
# 3 0.4252424 0.4933579 1.1568545 0.8866058 0
# 4 0.3502651 0.6004373 0.8710482 0.7628923 0
# 5 0.3845145 0.6029086 0.9618363 0.7998859 0
# 6 0.1397217 0.4201069 1.1817022 1.3582855 0
# 7 0.2311455 0.3894894 0.8322239 0.8594886 0
# 8 0.3725047 0.5985897 1.1742117 0.9938662 0
# 9 0.4881618 0.2712268 0.7585261 0.5723671 0
# 10 0.1568071 0.3936400 0.9869924 1.1143561 0

```

---

```
searchForIndustryRelevantGitHubProjects
```

```
  searchForIndustryRelevantGitHubProjects
```

---

## Description

Function searches for industry relevant software projects available from GitHub. The function was used to deliver data set of software projects in an NCBIr project. More details are described in a report: Lech Madeyski, "Training data preparation method," tech. rep., code quest (research project NCBIr POIR.01.01.01-00-0792/16), 2019, as well as a paper: Tomasz Lewowski and Lech Madeyski, "Creating evolving project data sets in software engineering", 2019. If you use this function or the returned data set than please cite: Tomasz Lewowski and Lech Madeyski, "Creating evolving project data sets in software engineering", 2019

## Usage

```

searchForIndustryRelevantGitHubProjects(
  myToken,
  earliestPushDate,
  latestCreationDate
)

```

## Arguments

myToken            A private token used to access GitHub

earliestPushDate

Only repositories which were pushed after this date will be included in the results (i.e., repositories for which the latest push was before this date will not be included in the results)

latestCreationDate

Only repositories which were created before this date will be included in the results (i.e., repositories created after this date will not be included in the results)

## Value

selected GitHub projects

## Author(s)

Lech Madeyski and Tomasz Lewowski

## Examples

```
# to run this function you need to use your own token as a parameter of the function
# use your own token as the first parameter of the function
# searchForIndustryRelevantGitHubProjects("...", "2019-03-01", "2018-08-01")
```

---

simulate2GExperimentData

*simulate2GExperimentData*

---

## Description

The function returns a two group data set based on one of four different distributions.

## Usage

```
simulate2GExperimentData(  
  mean,  
  sd,  
  diff,  
  GroupSize,  
  type = "n",  
  ExpAdj = 0,  
  StdAdj = 0,  
  BlockEffect = 0,  
  BlockStdAdj = 0  
)
```

**Arguments**

mean	The mean (or rate for gamma data) of the baseline distribution
sd	The standard deviation (or shape for gamma data) of the baseline distribution
diff	The adjustment to the baseline mean for the alternative distribution.
GroupSize	An integer defining the number of data items in each group.
type	A string identifying the distribution used to simulate the data: 'n' for normal, 'ln' for log-normal, 'g' for gamma, 'lap' for Laplace.
ExpAdj	An additional adjustment factor that is added to both the mean value. Defaults to zero.
StdAdj	An additional adjustment factor that is added to both group variance (or rate for gamma data). Defaults to zero.
BlockEffect	An additional factor that is added to the mean of the both groups (shape for the gamma distribution). Defaults to zero.
BlockStdAdj	An additional factor that is added to the variance of both groups (shape for the gamma distribution). Defaults to zero.

**Value**

A table with two columns (BaselineData and AlternativeData) holding the data for each group. For lognormal data an additional two columns are added which return the log transformed data.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
set.seed(236)
simulate2GExperimentData(mean = 0, sd = 1, diff = 0.5, GroupSize = 10,
  type = "n", ExpAdj = 0, StdAdj = 0, BlockEffect = 0, BlockStdAdj = 0)
# A tibble: 10 x 2
#   BaselineData AlternativeData
#   <dbl>          <dbl>
#   <dbl>          <dbl>
# 1    -0.285        -0.255
# 2    -0.972         0.112
# 3    -0.549         1.36
# 4     1.05         1.47
# 5    -0.267         0.107
# 6    -0.137         0.395
# 7     1.30         1.27
# 8    -0.722         1.70
# 9    -0.525         0.264
# 10   -0.0222         0.787
set.seed(345)
simulate2GExperimentData(mean = 0, sd = 1, diff = 0.5, GroupSize = 10,
  type = "l", ExpAdj = 0, StdAdj = 0, BlockEffect = 0, BlockStdAdj = 0)
# A tibble: 10 x 4
```

#	BaselineData	AlternativeData	transBaselineData	transAlternativeData
#	<dbl>	<dbl>	<dbl>	<dbl>
# 1	0.456	10.7	-0.785	2.37
# 2	0.756	0.407	-0.280	-0.900
# 3	0.851	0.705	-0.161	-0.350
# 4	0.748	2.27	-0.291	0.818
# 5	0.935	4.07	-0.0675	1.40
# 6	0.531	0.405	-0.634	-0.903
# 7	0.395	2.91	-0.928	1.07
# 8	5.53	4.69	1.71	1.55
# 9	5.23	0.602	1.65	-0.508
# 10	6.11	2.23	1.81	0.802

---

```
simulate4GExperimentData
```

```
simulate4GExperimentData
```

---

## Description

The function returns a four group data set based on one of four different distributions.

## Usage

```
simulate4GExperimentData(
  mean,
  sd,
  diff,
  GroupSize,
  type = "n",
  ExpAdj = 0,
  StdAdj = 0,
  BlockEffect = 0,
  BlockStdAdj = 0
)
```

## Arguments

mean	The mean (or rate for gamma data) of the baseline distribution
sd	The standard deviation (or shape for gamma data) of the baseline distribution
diff	The adjustment to the baseline mean for the alternative distribution.
GroupSize	An integer defining the number of data items in each group.
type	A string identifying the distribution used to simulate the data: 'n' for normal, 'l' for log-normal, 'g' for gamma, 'lap' for Laplace.
ExpAdj	An additional adjustment factor that is added to both the mean values. Defaults to zero.

StdAdj	An additional adjustment factor that is added to the second group variance (or rate for gamma data). Defaults to zero.
BlockEffect	An additional factor that is added to the mean of the second group groups (shape for the gamma distribution). Defaults to zero.
BlockStdAdj	An additional factor that is added to the variance of the second group (shape for the gamma distribution). Defaults to zero.

### Value

A table with four columns (BaselineData.B1, AlternativeData.B1, BaselineData.B2, AlternativeData.B2), holding the data for each group and block. For lognormal data an additional four columns are added which return the log transformed data for each group.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
set.seed(246)
simulate4GExperimentData(mean = 0, sd = 1, diff = 0.5, GroupSize = 5,
  type = "n", ExpAdj = 0, StdAdj = 0, BlockEffect = 0.5, BlockStdAdj = 0)
# A tibble: 5 x 4
#   BaselineData.B1 AlternativeData.B1 BaselineData.B2 AlternativeData.B2
#   <dbl>           <dbl>           <dbl>           <dbl>
# 1     0.533         1.84             0.749           3.98
# 2     0.251         2.03             1.56            1.09
# 3    -0.290         0.929           0.213           3.94
# 4    -1.48          1.17            1.13            0.106
# 5     0.0340        0.895           0.399           0.879
as.data.frame(
  simulate4GExperimentData(
    mean=0, sd=1, diff=0.5, GroupSize=5, type='l', ExpAdj=0, StdAdj=0,
    BlockEffect = 0.5, BlockStdAdj = 0))
#   BaselineData.B1 AlternativeData.B1 transBaselineData.B1 transAlternativeData.B1
#1  1.4019869      1.049158          0.3378905          0.0479875
#2  3.8514120      0.769227          1.3484398         -0.2623692
#3  6.5162726      1.574126          1.8743025          0.4537002
#4  1.3309218      1.082774          0.2858718          0.0795259
#5  0.2772234      1.630194         -1.2829316          0.4886992
#   BaselineData.B2 AlternativeData.B2 transBaselineData.B2 transAlternativeData.B2
#1  5.4656049      4.6095688          1.6984748          1.5281343
#2  1.6149559      2.0244244          0.4793077          0.7052854
#3  1.7718620      0.5504016          0.5720310         -0.5971070
#4  0.6774067      1.5434812         -0.3894834          0.4340404
#5  0.4507284      5.4987830         -0.7968903          1.7045268
```

---

```
simulateRandomizedBlockDesignEffectSizes
      simulateRandomizedBlockDesignEffectSizes
```

---

### Description

This simulates a two-block and two-treatment design based on one of four distributions, and finds the values of  $k\tau$  and Cliffs  $d$  and their variances. It simulates a randomised blocks experiment with two treatment groups and two control groups each of which being divided into two blocks. It assumes equal group sizes but group spread (standard deviation can be changed, see StAdj). It returns values of both parametric and non-parametric effect sizes and their variance and significance. For the logarithmic distribution it calculates effect sizes based on the log transformed data as well as the raw data.

### Usage

```
simulateRandomizedBlockDesignEffectSizes(
  mean,
  sd,
  diff,
  N,
  type = "n",
  alpha = 0.05,
  Blockmean = 0,
  BlockStdAdj = 0,
  StdAdj = 0,
  AlwaysTwoSidedTests = FALSE,
  ReturnData = FALSE
)
```

### Arguments

mean	The default value for all groups which can be changed for the two treatment groups using the parameter diff and for the two block 2 groups using the parameter Blockmean
sd	The default spread used for all four groups unless adjusted by the StdAdj. It must be a real value greater than 0.
diff	This is added to the parameter mean to obtain the required mean for treatment groups. It can be a real value and can take the value zero.
N	this is the number of observations in each group. It must be an integer greater than 3.
type	this specifies the underlying distribution used to generate the data. it takes the values 'n' for a normal distribution, 'l' for lognormal distribution, 'g' for a gamma distribution, 'lap' for a Laplace distribution.
alpha	The level used for statistical tests (default 0.05).

Blockmean	if >0 an adjustment made to both group means in Block 2
BlockStdAdj	if >0, an adjustment that can be made to the sd of each group in block 2
StdAdj	this specifies the extent of variance instability introduced by the treatment and if >0 will be used to amend the sd parameter for both treatment groups. This value must be positive and less than 0.5
AlwaysTwoSidedTests	Logical variable (default FALSE) if TRUE the function always performs two-sided tests. Otherwise if the parameter diff is not equal to zero, the function performs one-sided tests.
ReturnData	Logical variable, If TRUE, the function simply returns the generated data. If false (which is default value) the function returns various effect sizes and whether the effect sizes are statistically significant.

### Value

data frame incl. either the non-parametric and parametric effect sizes and whether the effect sizes are significant at the 0.05 level or the generated data depending on the value of the ReturnData parameter.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```
set.seed(123)
as.data.frame(
  simulateRandomizedBlockDesignEffectSizes(
    mean = 0, sd = 1, diff = .5, N = 10, type = "n", alpha = 0.05,
    Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0))
# N phat phat.var phat.df phat.test phat.pvalue phat.sig phat.ci.upper phat.ci.lower d
# 1 40 0.79 0.005866667 30.15715 3.786189 0.0003403047 TRUE 1 0.6600213 0.58
#   vard d.sig d.ci.lower d.ci.upper cor sqse ctvar n1 n2 sigCVt sigCVn
# 1 0.02430788 TRUE 0.2775601 1 0.3052632 0.01315789 0.006953352 20 20 TRUE TRUE
#   ttest.sig ES Variance StdES BlockEffect MedianDiff
# 1 TRUE 0.9402999 0.7829385 1.06268 0.307119 1.313642
set.seed(123)
as.data.frame(
  simulateRandomizedBlockDesignEffectSizes(
    mean = 0, sd = 1, diff = 0.5, N = 10, type = "n", alpha = 0.05,
    Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0, AlwaysTwoSidedTests = TRUE)
)
# N phat phat.var phat.df phat.test phat.pvalue phat.sig phat.ci.upper phat.ci.lower
# 1 40 0.79 0.005866667 30.15715 3.786189 0.0006806094 TRUE 0.946392 0.633608
#   d vard d.sig d.ci.lower d.ci.upper cor sqse ctvar n1 n2 sigCVt
# 1 0.58 0.02430788 TRUE 0.2135334 0.8033737 0.3052632 0.01315789 0.006953352 20 20 TRUE
#   ttest.sig ES Variance StdES BlockEffect MedianDiff
# 1 TRUE 0.9402999 0.7829385 1.06268 0.307119 1.313642
set.seed(123)
as.data.frame(
```

```

simulateRandomizedBlockDesignEffectSizes(
  mean = 0, sd = 1, diff = .5, N = 10, type = "1", alpha = 0.05,
  Blockmean = 0.5, BlockStdAdj = 0, StdAdj = 0, ReturnData = TRUE))
# BaselineData.B1 AlternativeData.B1 transBaselineData.B1 transAlternativeData.B1
# 1      0.5709374      5.6073700      -0.56047565      1.72408180
# 2      0.7943926      2.3627208      -0.23017749      0.85981383
# 3      4.7526783      2.4615013      1.55870831      0.90077145
# 4      1.0730536      1.8416883      0.07050839      0.61068272
# 5      1.1380175      0.9456894      0.12928774      -0.05584113
# 6      5.5570366      9.8445021      1.71506499      2.28691314
# 7      1.5855260      2.7124451      0.46091621      0.99785048
# 8      0.2822220      0.2307046      -1.26506123      -1.46661716
# 9      0.5031571      3.3246217      -0.68685285      1.20135590
# 10     0.6404002      1.0275821      -0.44566197      0.02720859
# BaselineData.B2 AlternativeData.B2 transBaselineData.B2 transAlternativeData.B2
# 1      0.5667575      4.163950      -0.5678237      1.4264642
# 2      1.3258120      2.023702      0.2820251      0.7049285
# 3      0.5909615      6.653384      -0.5260044      1.8951257
# 4      0.7954150      6.541284      -0.2288912      1.8781335
# 5      0.8824622      6.181624      -0.1250393      1.8215811
# 6      0.3052289      5.412117      -1.1866933      1.6886403
# 7      3.8106015      4.729964      1.3377870      1.5539177
# 8      1.9220131      2.555092      0.6533731      0.9380883
# 9      0.5282757      2.001781      -0.6381369      0.6940373
# 10     5.7765980      1.858053      1.7538149      0.6195290

```

---

```
simulateRandomizedDesignEffectSizes
```

```
simulateRandomizedDesignEffectSizes
```

---

## Description

This simulates one of four data distributions (normal, log-normal, gamma and Laplace), and finds the values of  $\phi$  and Cliff's  $d$  and their variances. It assumes equal group sizes. It returns values of the effect sizes and their variance for a simulated randomized experiment with two treatments. It returns whether or not each non-parametric effect size was significant. It also returns the parametric (standardized and unstandardized) Effect Size and the whether the t-test was significant.

## Usage

```

simulateRandomizedDesignEffectSizes(
  mean,
  sd,
  diff,
  N,
  type = "n",
  StdAdj = 0,
  alpha = 0.05,
  AlwaysTwoSidedTests = FALSE,

```

```

  Return.Data = FALSE
)

```

### Arguments

mean	The mean used for one of the treatment groups (this is the rate for the gamma data)
sd	The spread used for both treatment groups. It must be a real value greater than 0 (this is the shape for the gamma data).
diff	This is added to the parameter mean, to define the mean of the other treatment group. It can be a real value and can take the value zero.
N	this is the number of observations in each group. It must be an integer greater than 3.
type	this specifies the underlying distribution used to generate the data. it takes the values 'n' for a normal distribution, 'l' for lognormal distribution, 'g' for a gamma distribution, 'lap' for a Laplace distribution.
StdAdj	this specifies the extent of variance instability to be introduced.
alpha	the level for all statistical tests (default 0.05)
AlwaysTwoSidedTests	if set to FALSE (i.e. default) the algorithm uses one-sided tests if diff!=0 and two-sided tests otherwise. If set to TRUE the algorithm always uses two-sided tests.
Return.Data	if set to true the algorithm returns the data not the effect sizes (default FALSE).

### Value

data frame incl. the non-parametric and parametric effect sizes and whether the effect sizes are significant at the specified alpha level. For log-normal data the function returns the effect sizes for the transformed data.

### Author(s)

Barbara Kitchenham and Lech Madeyski

### Examples

```

set.seed(123)
as.data.frame(
  simulateRandomizedDesignEffectSizes(
    mean = 0, sd = 1, diff = 0.8, N = 10, type = "n", StdAdj = 0))
# phat varphat dfphat sigphat d vard sigd cor varcor sigCVt t.value
# 1 0.75 0.01522222 17.46405 TRUE 0.5 0.06237576 TRUE 0.2631579 0.01754995 TRUE 2.095142
# t.se t.df t.lb t.ub t.sig ES Variance StdES MedDiff
# 1 0.4457915 17.87244 0.1606665 Inf TRUE 0.9339963 0.9936502 0.9369759 1.260127
set.seed(123)
as.data.frame(
  simulateRandomizedDesignEffectSizes(
    mean = 0, sd = 1, diff = 0.8, N = 10, type = "n", StdAdj = 0,

```

```

    AlwaysTwoSidedTests = TRUE))
# phat  varphat  dfphat  sigphat  d      vard  sigd      cor
# 1 0.75 0.01522222 17.46405  FALSE 0.5 0.06237576 FALSE 0.2631579
#      varcor  sigCVt  t.value      t.se      t.df      t.lb      t.ub  t.sig
# 1 0.01754995  FALSE 2.095142 0.4457915 17.87244 -0.003056196 1.871049 FALSE
#      ES  Variance      StdES  MedDiff
# 1 0.9339963 0.9936502 0.9369759 1.260127
set.seed(456)
as.data.frame(
  simulateRandomizedDesignEffectSizes(
    mean = 0, sd = 1, diff = 0.8, N = 10, type = "l", StdAdj = 0))
# phat  varphat  dfphat  sigphat  d      vard  sigd      cor      varcor
# 1 0.87 0.008466667 11.1111  TRUE 0.74 0.0350497 TRUE 0.3894737 0.01039674
# sigCVt  t.value      t.se      t.df      t.lb  t.ub  t.sig      ES  Variance
# 1 TRUE 3.599375 2.148297 9.312472 3.809448  Inf TRUE 7.732529 23.07591
#      StdES  MedDiff  transttest  EStrans  StdEstrans  VarTrans
# 1 1.60969 7.77893 0.998772 1.731323 1.598065 1.173728

set.seed(123)
as.data.frame(
  simulateRandomizedDesignEffectSizes(
    mean = 0, sd = 1, diff = 0.8, N = 10, type = "n", StdAdj = 0,
    Return.Data = TRUE))
# BaselineData  AlternativeData
# 1 -0.69470698 1.0533185
# 2 -0.20791728 0.7714532
# 3 -1.26539635 0.7571295
# 4 2.16895597 2.1686023
# 5 1.20796200 0.5742290
# 6 -1.12310858 2.3164706
# 7 -0.40288484 -0.7487528
# 8 -0.46665535 1.3846137
# 9 0.77996512 0.9238542
# 10 -0.08336907 1.0159416

```

---

testfunctionParameterChecks

*testfunctionParameterChecks*


---

## Description

This is a helper function that ensures parameter values used for performing special statistical tests are valid.

## Usage

```
testfunctionParameterChecks(alternative, alpha, stderr)
```

**Arguments**

alternative	The type of statistical test. Valid values are one of c('two.sided', 'greater', 'less')
alpha	The test level. Valid values are between 0.0001 and 0.2
stderr	The standard error of a parameter whose confidence intervals is to be calculated

**Value**

'Success' or an error message.

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
#reproducer:::testfunctionParameterChecks(alternative='larger',alpha=0.1,stderr=0.002)
#Error in testfunctionParameterChecks(alternative = 'larger', alpha = 0.1) :
# Invalid alternative parameter, choose one of two.sided, greater or less
reproducer:::testfunctionParameterChecks(alternative='greater',alpha=0.1,stderr=0.002)
#[1] 'Success'
#reproducer:::testfunctionParameterChecks(alternative='greater',alpha=0.1,stderr=0.000)
#Error in testfunctionParameterChecks(alternative = 'greater', alpha = 0.1, :
# Improbably small variance, data are essentially constant
```

---

transformHgtoR

*transformHgtoR*


---

**Description**

The functions transforms a vector of Hedges g values to their equivalent point bi-serial values.

**Usage**

```
transformHgtoR(g, Nc, Nt)
```

**Arguments**

g	A vector of Hedges g values.
Nc	A vector of numbers identifying the number of control condition participants in each group
Nt	A vector of numbers identifying the number of treatment condition participants in each group

**Value**

value of point biserial r

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
transformHgtoR(0.4, 20, 20)
# [1] 0.1961161
```

---

*transformHgtoZr*      *transformHgtoZr*

---

**Description**

The functions transforms a vector of Hedges g values to their normal approximation of point biserial values.

**Usage**

```
transformHgtoZr(g, Nc, Nt)
```

**Arguments**

g	value of Hedges' g
Nc	the number of observations (participants) in the first (control) group
Nt	the number of observations (participants) in the second (treatment) group

**Value**

value of normal approximation of point biserial r

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
transformHgtoZr(0.5, 20, 20)
# [1] 0.2474665
```

---

transformRtoHg	<i>transformRtoHg</i>
----------------	-----------------------

---

**Description**

This function converts a vector of point bi-serial  $r$  values with associated sample size information back to the mean difference effect size Hedges  $g$ .

**Usage**

```
transformRtoHg(r, Nc, Nt)
```

**Arguments**

$r$	A vector of point bi-serial correlation values.
$Nc$	A vector of the number of observations in the control condition for the related experiments.
$Nt$	A vector of the number of observations in the treatment condition for the related experiments.

**Value**

value of Hedges'  $g$

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
transformRtoHg(c(0.4, 0.2), c(20, 20), c(20, 20))
# [1] 0.8728716 0.4082483
```

---

transformRtoZr	<i>transformRtoZr</i>
----------------	-----------------------

---

**Description**

The function transforms a vector of point biserial  $r$  values to their normal approximation. It also works for the correlation  $r$ .

**Usage**

```
transformRtoZr(r)
```

**Arguments**

r                    A vector of r-values

**Value**

value of normal approximation of point biserial r

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
reproducer::transformRtoZr(0.4)
# [1] 0.4236489
Zr <- reproducer::transformRtoZr(c(0.4, 0.2))
Zr
# [1] 0.4236489 0.2027326
```

---

transformZrtoHg	<i>transformZrtoHg</i>
-----------------	------------------------

---

**Description**

Transforms Zr to Hedge's g.

**Usage**

```
transformZrtoHg(Zr, Nc, Nt)
```

**Arguments**

Zr                    the normal variate  
 Nc                    the number of observations (participants) in the first (control) group  
 Nt                    the number of observations (participants) in the second (treatment) group

**Value**

value of Hedges' g

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
transformZrtoHg(0.5, 20, 20)
# [1] 1.042191
```



**Value**

value of point biserial r

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
transformZrtoR(0.4236489)
# [1] 0.4
transformZrtoR(c(0.4236489, 0.2027326))
# [1] 0.4 0.2
```

---

varStandardizedEffectSize

*varStandardizedEffectSize*

---

**Description**

Function calculates the exact variance of a standardized effect size based on the relationship between t and the standardized effect size, see Morris and DeShon, Combining Effect Size Estimates in Meta-Analysis With Repeated Measures and Independent-Groups Designs, Psychological Methods, 7 (1), pp 105-125.

**Usage**

```
varStandardizedEffectSize(d, A, f, returnVarg = TRUE)
```

**Arguments**

d	An unadjusted standardized effect size
A	The squared constant linking t and d i.e. $t \cdot \sqrt{A} = d$
f	The degrees of freedom of the t value
returnVarg	if set to TRUE return the variance of the small sample size adjusted standardized effect size (g), otherwise returns var(d) where d is the input parameter

**Value**

if returnVarg if set to TRUE, return var(g) otherwise var(d)

**Author(s)**

Barbara Kitchenham and Lech Madeyski

**Examples**

```
d <- 0.5
varStandardizedEffectSize(d, 2 / 20, 38, returnVarg = FALSE)
# [1] 0.1047567
varStandardizedEffectSize(d, 2 / 20, 38, returnVarg = TRUE)
# [1] 0.1090516
```



- calcEffectSizeConfidenceIntervals, [13](#)
- calcPHatConfidenceIntervals, [15](#)
- calcPHatMATestStatistics, [16](#)
- calculate2GBias, [17](#)
- calculate2GType1Error, [19](#)
- calculate4GBias, [21](#)
- calculate4GType1Error, [23](#)
- calculateBasicStatistics, [25](#)
- calculateCliffd, [26](#)
- calculateGroupSummaryStatistics, [27](#)
- calculateHg, [29](#)
- calculateKendalltau, [30](#)
- calculateLargeSampleRandomizedBlockDesignEffectSizes, [75](#)
- [31](#)
- calculateLargeSampleRandomizedDesignEffectSizes, [76](#)
- [33](#)
- CalculateLevel2ExperimentRData, [34](#)
- calculateMABias, [36](#)
- calculateMAType1Error, [39](#)
- calculateNullESAccuracy, [41](#)
- calculatePhat, [43](#)
- calculatePopulationStatistics, [44](#)
- CalculateRLevel1, [45](#)
- calculateSmallSampleSizeAdjustment, [47](#)
- CatchError, [48](#)
- checkIfValidDummyVariable, [49](#)
- Ciolkowski09ESEM.MetaAnalysis.PBRvsCBRorAR, [49](#)
- Cliffd.test, [50](#)
- constructEffectSizes, [51](#)
- ConstructLevel1ExperimentRData, [53](#)
- crossoverResidualAnalysis, [55](#)
- densityCurveOnHistogram, [56](#)
- doLM, [57](#)
- effectSizeCI, [57](#)
- ExtractExperimentData, [59](#)
- ExtractGroupSizeData, [60](#)
- ExtractMAStatistics, [62](#)
- ExtractSummaryStatisticsRandomizedExp, [64](#)
- fmt, [65](#)
- getEffectSizesABBA, [66](#)
- getEffectSizesABBAIgnoringPeriodEffect, [67](#)
- getSimulationData, [68](#)
- getTheoreticalEffectSizeVariancesABBA, [69](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Abrahao13TSE, [70](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Gravino15JVL, [71](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Madeyski10, [72](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Reggio15SSM, [74](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Ricca10TSE, [75](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Ricca14TOSEM, [76](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Romano18ESEM, [77](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14E, [78](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14J, [79](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello14T, [80](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello15E, [81](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Scanniello17T, [83](#)
- KitchenhamEtAl.CorrelationsAmongParticipants.Torchiano17JV, [84](#)
- KitchenhamMadeyski.SimulatedCrossoverDataSets, [85](#)
- KitchenhamMadeyskiBrereton.ABBAMetaAnalysisReportedResults, [86](#)
- KitchenhamMadeyskiBrereton.ABBAReportedEffectSizes, [87](#)
- KitchenhamMadeyskiBrereton.DocData, [88](#)
- KitchenhamMadeyskiBrereton.ExpData, [90](#)
- KitchenhamMadeyskiBrereton.MetaAnalysisReportedResults, [91](#)
- KitchenhamMadeyskiBrereton.ReportedEffectSizes, [92](#)
- KitchenhamMadeyskiBudgen16.COCOMO, [93](#)
- KitchenhamMadeyskiBudgen16.DiffInDiffData, [95](#)
- KitchenhamMadeyskiBudgen16.FINNISH, [96](#)
- KitchenhamMadeyskiBudgen16.PolishData, [97](#)
- KitchenhamMadeyskiBudgen16.PolishSubjects, [98](#)

- KitchenhamMadeyskiBudgen16.SubjectData, [100](#)
- LaplaceDist, [102](#)
- Madeyski15EISEJ.OpenProjects, [103](#)
- Madeyski15EISEJ.PropProjects, [104](#)
- Madeyski15EISEJ.StudProjects, [105](#)
- Madeyski15SQJ.NDC, [106](#)
- MadeyskiKitchenham.EUBASdata, [107](#)
- MadeyskiKitchenham.MetaAnalysis.PBRvsCBRorAR, [108](#)
- MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20192024, [110](#)
- MadeyskiLewowski.IndustryRelevantGitHubJavaProjects20192025, [111](#)
- metaanalyse.Cliffd, [113](#)
- metaanalyse.PHat, [115](#)
- metaanalyseSmallSampleSizeExperiments, [117](#)
- MetaAnalysisSimulations, [118](#)
- NP2GMetaAnalysisSimulation, [121](#)
- NP4GMetaAnalysisSimulation, [124](#)
- percentageInaccuracyOfLargeSampleVarianceApproximation, [126](#)
- PHat.test, [127](#)
- PHatonesidedTestStatistics, [128](#)
- PHattwosidedTestStatistics, [129](#)
- plotOutcomesForIndividualsInEachSequenceGroup, [131](#)
- PrepareForMetaAnalysisGtoR, [132](#)
- printXTable, [133](#)
- proportionOfSignificantTValuesUsingCorrectAnalysis, [134](#)
- proportionOfSignificantTValuesUsingIncorrectAnalysis, [135](#)
- RandomExperimentSimulations, [135](#)
- RandomizedBlockDesignEffectSizes, [138](#)
- RandomizedBlocksAnalysis, [139](#)
- RandomizedBlocksExperimentSimulations, [141](#)
- RandomizedDesignEffectSizes, [144](#)
- readExcelSheet, [145](#)
- reproduceForestPlotRandomEffects, [145](#)
- reproduceMixedEffectsAnalysisWithEstimatedVarianceAndExperimentalDesignModerator, [146](#)
- reproduceMixedEffectsAnalysisWithExperimentalDesignModerator, [146](#)
- reproduceMixedEffectsForestPlotWithExperimentalDesignModerator, [147](#)
- reproduceSimulationResultsBasedOn500Reps1000Obs, [147](#)
- reproduceTablesOfPaperMetaAnalysisForFamiliesOfExperiments, [148](#)
- reproduceTableWithEffectSizesBasedOnMeanDifferences, [149](#)
- reproduceTableWithPossibleModeratingFactors, [149](#)
- searchForIndustryRelevantGitHubProjects, [152](#)
- simulate2GExperimentData, [153](#)
- simulate4GExperimentData, [155](#)
- simulateRandomizedBlockDesignEffectSizes, [157](#)
- simulateRandomizedDesignEffectSizes, [159](#)
- testFunctionParameterChecks, [161](#)
- transformHgtoR, [162](#)
- transformHgtoZr, [163](#)
- transformRtoHg, [164](#)
- transformRtoZr, [164](#)
- transformZrtoHg, [165](#)
- transformZrtoHgapprox, [166](#)
- transformZrtoR, [166](#)
- varStandardizedEffectSize, [167](#)