Package ‘eefAnalytics’

January 9, 2025

Type Package

Title Robust Analytical Methods for Evaluating Educational Interventions using Randomised Controlled Trials Designs

Version 1.1.5

Description Analysing data from evaluations of educational interventions using a randomised controlled trial design. Various analytical tools to perform sensitivity analysis using different methods are supported (e.g. frequentist models with bootstrapping and permutations options, Bayesian models). The included commands can be used for simple randomised trials, cluster randomised trials and multisite trials. The methods can also be used more widely beyond education trials. This package can be used to evaluate other intervention designs using Frequentist and Bayesian multilevel models.

License AGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

LazyData true

Depends R (>= 3.6.0)

Imports R2jags (>= 0.7), ggplot2 (>= 3.4.0), lme4 (>= 1.1.0), methods, graphics, stats, mvtnorm (>= 1.1.2), coda (>= 0.19), MCMCvis (>= 0.15)

NeedsCompilation no

Author Germaine Uwimpuhwe [aut, cre],
Qing Zhang [aut],
Akansha Singh [aut],
Dimitris Vallis [aut],
Jochen Einbeck [aut],
Steve Higgins [aut],
ZhiMin Xiao [aut],
Ewoud De Troyer [aut],
Adetayo Kasim [aut]

Maintainer Germaine Uwimpuhwe <`germaine.uwimpuhwe@durham.ac.uk`>

Repository CRAN

Date/Publication 2025-01-09 10:20:02 UTC

Contents

`ComparePlot` *A plot function to compare different eefAnalytics S3 objects from the eefAnalytics package.*

Description

It generates bar plot that compares the effect size from eefAnalytics' methods.

Usage

```
ComparePlot(
  eefAnalyticsList,
  group,
  Conditional = TRUE,
  ES_Total = TRUE,
  modelNames
)
```

Arguments

`eefAnalyticsList`	A list of eefAnalytics S3 objects from eefAnalytics package.
`group`	a string/scalar value indicating which intervention to plot. This must be one of the values of intervention variable excluding the control group. For a two arm trial, the maximum number of values to consider is 1 and 2 for three arm trial.

`Conditional`	a logical value to indicate whether to plot conditional effect size. The default is Conditional=TRUE, otherwise Conditional=FALSE should be specified for plot based on unconditional effect size. Conditional variance is total or residual variance a multilevel model with fixed effects, whilst unconditional variance is total variance or residual variance from a multilevel model with only intercept as fixed effect.
`ES_Total`	A logical value indicating whether to plot the effect size based on total variance or within school variance. The default is ES_Total=TRUE, to plot effect size using total variance. ES_Total=FALSE should be specified for effect size based on within school or residuals variance.
`modelNames`	a string factor containing the names of model to compare. See examples below.

Details

`ComparePlot` produces a bar plot which compares the effect sizes and the associated confidence intervals from the different models. For a multilevel model, it shows the effect size based on residual variance and total variance.

Value

Returns a bar plot to compare the different methods. The returned figure can be further modified as any `ggplot`

Examples

```
if(interactive()){

data(mstData)
###############
##### SRT #####
###############

outputSRT <- srtFREQ(Posttest~ Intervention + Prettest,
                     intervention = "Intervention", data = mstData)

outputSRTBoot <- srtFREQ(Posttest~ Intervention + Prettest,
                         intervention = "Intervention",nBoot=1000, data = mstData)

###############
##### MST #####
###############

outputMST <- mstFREQ(Posttest~ Intervention + Prettest,
                     random = "School", intervention = "Intervention", data = mstData)

outputMSTBoot <- mstFREQ(Posttest~ Intervention + Prettest,
                         random = "School", intervention = "Intervention",
                         nBoot = 1000, data = mstData)

##################
##### Bayesian #####
```

```
##################

outputSRTbayes <- srtBayes(Posttest~ Intervention + Prettest,
                           intervention = "Intervention",
                           nsim = 2000, data = mstData)

## comparing different results

ComparePlot(list(outputSRT,outputSRTBoot,outputMST,outputMSTBoot,outputSRTbayes),
            modelNames =c("ols", "olsBoot","MLM","MLMBoot","OLSBayes"),group=1)


}
```

`crtBayes` *Bayesian analysis of Cluster Randomised Education Trials (CRT) using Vague Priors.*

Description

`crtBayes` performs Bayesian multilevel analysis of cluster randomised education trials, utilising vague priors and JAGS language to fit the model. It assumes hierarchical clustering, such as students within schools, and estimates treatment effects while accounting for this structure.

Usage

```
crtBayes(
  formula,
  random,
  intervention,
  baseln,
  nsim = 10000,
  data,
  alpha = 0.05,
  digits = 3,
  threshold = c(0, 0.05, seq(0.1, 1, 0.1)),
  condopt,
  uncopt,
  ...
)
```

Arguments

`formula`	The model to be analysed. It should be of the form y ~ x1 + x2 + ..., where y is the outcome variable and Xs are the predictors.
`random`	A string specifying the "clustering variable" (e.g., schools or sites) as found in the dataset.
`intervention`	A string specifying the "intervention variable" as it appears in the formula.

`baseln`	A string specifying the reference category for the intervention variable. If not provided, the first level will be used as the reference (e.g., baseln = "0" for an intervention with levels 0 and 1).
`nsim`	Number of MCMC iterations to be performed. A minimum of 10,000 is recommended to ensure convergence.
`data`	A data frame containing the variables referenced in the formula, including predictors, the clustering variable, and the intervention.
`alpha`	significant level, default alpha = 0.05.
`digits`	number of decimal places, by default digits=3
`threshold`	a scalar or vector of pre-specified threshold(s) for estimating Bayesian posterior probability such that the observed effect size is greater than or equal to the threshold(s).
`condopt`	additional arguments of `jags` to be passed exclusively to the conditional model (e.g., defining n.chains only for the conditional model, etc.).
`uncopt`	additional arguments of `jags` to be passed exclusively to the unconditional model (e.g., defining n.chains only for the unconditional model, etc.).
`...`	Common additional arguments of `jags` to be passed to both the conditional and unconditional model specifications

Details

The function provides posterior estimates for fixed effects (predictors) and random effects (clustering) under a Bayesian framework. Effect sizes are computed using Hedges' g, and variance components are decomposed into between-cluster and within-cluster variances.

Value

S3 object; a list consisting of:

- `Beta`: Estimates and credible intervals for the predictors specified in the model (posterior distributions).
- `ES`: Hedges' g effect size for the intervention(s). If bootstrapping is not used, 95% credible intervals are computed based on MCMC sampling.
- `covParm`: Variance components broken down into between-cluster variance (e.g., between schools), within-cluster variance (e.g., within pupils), and intra-cluster correlation (ICC)..
- `SchEffects`: A vector of the estimated deviation of each school from the intercept.
- `ProbES`: A matrix showing the probability of observing an effect size larger than various thresholds (0, 0.05, 0.10, ...).
- `Model`: A model object from `jags` and an `MCMCsummary` object containing only the mean and credible intervals (CIs) as columns.
- `Unconditional`: A list containing the unconditional effect size and variance decomposition.

Examples

```
if(interactive()){

  data(crtData)

  ########################################################
  ## Bayesian analysis of cluster randomised trials     ##
  ########################################################

  output <- crtBayes(formula = Posttest ~ Prettest + Intervention,
                   random = "School",
                   intervention = "Intervention",
                   alpha = 0.05,
                   digits = 3,
                   nsim = 10000,
                   data = crtData)
  output

  ### Fixed effects
  beta <- output$Beta
  beta

  ### Effect size
  ES1 <- output$ES
  ES1

  ## Covariance matrix
  covParm <- output$covParm
  covParm

  ## Prob ES
  ProbES <- output$ProbES
  ProbES

  ## Unconditional
  Unconditional <- output$Unconditional
  Unconditional


  ### plot random effects for schools

  plot(output)

 ### plot posterior probability of an effect size to be bigger than a pre-specified threshold

  plot(output,group=1)


}
```

`crtData` *Cluster Randomised Trial Data.*

Description

A cluster randomised trial dataset containing 22 schools. The data contains a random sample of test data of pupils and not actual trial data.

Format

A data frame with 265 rows and 5 variables

Details

- Posttest: posttest scores
- Prettest: prettest scores
- Intervention: the indicator for intervention groups in a two arm trial, coded as 1 for intervention group and 0 for control group.
- Intervention2: a simulated indicator for intervention groups in a three arm trial.
- School: numeric school identifier

`crtFREQ` *Analysis of Cluster Randomised Education Trials using Multilevel Model under a Frequentist Setting.*

Description

`crtFREQ` performs analysis of cluster randomised education trials using a multilevel model under a frequentist setting.

Usage

```
crtFREQ(
  formula,
  random,
  intervention,
  baseln,
  nPerm,
  nBoot,
  type,
  ci,
  seed,
  data
)
```

Arguments

`formula`	the model to be analysed is of the form y ~ x1+x2+.... Where y is the outcome variable and Xs are the independent variables.
`random`	a string variable specifying the "clustering variable" as contained in the data. See example below.
`intervention`	a string variable specifying the "intervention variable" as appearing in the formula and the data. See example below.
`baseln`	A string variable allowing the user to specify the reference category for intervention variable. When not specified, the first level will be used as a reference.
`nPerm`	number of permutations required to generate a permutated p-value.
`nBoot`	number of bootstraps required to generate bootstrap confidence intervals.
`type`	method of bootstrapping including case re-sampling at student level "case(1)", case re-sampling at school level "case(2)", case re-sampling at both levels "case(1,2)" and residual bootstrapping using "residual". If not provided, default will be case re-sampling at student level.
`ci`	method for bootstrap confidence interval calculations; options are the Basic (Hall's) confidence interval "basic" or the simple percentile confidence interval "percentile". If not provided default will be percentile.
`seed`	seed required for bootstrapping and permutation procedure, if not provided default seed will be used.
`data`	data frame containing the data to be analysed.

Value

S3 object; a list consisting of

- `Beta`: Estimates and confidence intervals for variables specified in the model.
- `ES`: Conditional Hedges' g effect size and its 95% confidence intervals. If nBoot is not specified, 95% confidence intervals are based on standard errors. If nBoot is specified, they are non-parametric bootstrapped confidence intervals.
- `covParm`: A vector of variance decomposition into between cluster variance (Schools) and within cluster variance (Pupils). It also contains intra-cluster correlation (ICC).
- `SchEffects`: A vector of the estimated deviation of each school from the intercept.
- `Perm`: A "nPerm x 2w" matrix containing permutated effect sizes using residual variance and total variance. "w" denotes number of intervention. "w=1" for two arm trial and "w=2" for three arm trial excluding the control group. It is produced only when `nPerm` is specified.
- `Bootstrap`: A "nBoot x 2w" matrix containing the bootstrapped effect sizes using residual variance (Within) and total variance (Total). "w" denotes number of intervention. "w=1" for two arm trial and "w=2" for three arm trial excluding the control group. It is only produced when `nBoot` is specified.
- `Unconditional`: A list of unconditional effect sizes, covParm, Perm and Bootstrap obtained based on variances from the unconditional model (model with only the intercept as a fixed effect).

Examples

```
if(interactive()){

data(crtData)

########################################################
## MLM analysis of cluster randomised trials + 1.96SE ##
########################################################

output1 <- crtFREQ(Posttest~ Intervention+Prettest,random="School",
intervention="Intervention",data=crtData)


### Fixed effects
beta <- output1$Beta
beta

### Effect size
ES1 <- output1$ES
ES1

## Covariance matrix
covParm <- output1$covParm
covParm

### plot random effects for schools

plot(output1)

##################################################
## MLM analysis of cluster randomised trials    ##
## with residual bootstrap confidence intervals ##
##################################################

output2 <- crtFREQ(Posttest~ Intervention+Prettest,random="School",
intervention="Intervention",nBoot=1000,type="residual",data=crtData)


### Effect size

ES2 <- output2$ES
ES2

### plot bootstrapped values

plot(output2, group=1)

#######################################################################
## MLM analysis of cluster randomised trials with permutation p-value##
#######################################################################

output3 <- crtFREQ(Posttest~ Intervention+Prettest,random="School",
```

```
intervention="Intervention",nPerm=1000,data=crtData)

### Effect size

ES3 <- output3$ES
ES3


### plot permutated values

plot(output3, group=1)
}
```

`GainIndex` *Calculate the Gain Index (GI) using JAGS*

Description

This function computes the Gain Index and other related statistics for educational trials. Gain index provides a proportion of pupils who would not have make good progress without intervention. This function supports flexible configurations for JAGS modeling.

Usage

```
GainIndex(
  data,
  formula,
  random,
  intervention,
  NA.omit = TRUE,
  n.iter = 20000,
  n.chains = 3,
  n.burnin = 10000,
  inits = NULL,
  model.file = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A list containing the data for the JAGS model which must include columns: School, Posttest, Pretest, Intervention. Data should not have any missing values in these columns.
`formula`	the model to be analysed is of the form y ~ x1+x2+.... Where y is the outcome variable and Xs are the independent variables. Formula does not need to include `Intervention` variable.
`random`	a string variable specifying the "clustering variable" as contained in the data. See example below.

`intervention`	a string variable specifying the "intervention variable" as appearing in the formula and the data. See example below.
`NA.omit`	Optional; a logic to check if omitting missing value. If NA.omit = TRUE, results will output the percentage of missing value in the four required columns and then JAGS results. If NA.omit = FALSE, will give a warning "Please handle missing values before using GainIndex()." If not provided, the function uses default TRUE.
`n.iter`	Total number of iterations for the MCMC simulation.
`n.chains`	Number of chains to run in the MCMC simulation.
`n.burnin`	Number of burn-in iterations to be discarded before analysis.
`inits`	Optional; a list of initial values for the JAGS model. If NULL, the function generates default initial values.
`model.file`	Optional; a custom path to the JAGS model file. If not provided, the function uses default path.
`alpha`	significant level, default alpha = 0.05.

Value

An S3 object containing the following components:

GI A data frame containing the Gain Index and its 95% confidence intervals, as well as the Progress Index and its 95% confidence intervals.

Proportions A data frame showing the proportion of participants achieving each level of gain (low and high) for both control and intervention groups.

Timing A vector with execution time details, including user and elapsed time in seconds.

Examples

```
######### EXAMPLE ONE: crtData #########
## Not run:
  data(crtData)
 output1 <- GainIndex(data = crtData, formula = Posttest~Prettest, random = "School",n.iter = 200,
                      intervention = "Intervention", NA.omit = T, alpha = 0.05)
  output1


  ########## EXAMPLE TWO: mstData ######
  data(mstData)
 output1 <- GainIndex(data = mstData, formula = Posttest~Prettest, random = "School",n.iter = 200,
                      intervention = "Intervention", NA.omit = T, alpha = 0.05)
  output1

## End(Not run)
```

`mstBayes` *Bayesian analysis of Multisite Randomised Education Trials (MST) using Vague Priors*

Description

`mstBayes` performs Bayesian multilevel analysis of multisite randomised education trials, utilising vague priors and JAGS language to fit the model. It assumes hierarchical clustering, such as students within schools, and estimates treatment effects while accounting for this structure and assuming that all random effects are independent.

Usage

```
mstBayes(
  formula,
  random,
  intervention,
  baseln,
  nsim = 10000,
  data,
  alpha = 0.05,
  digits = 3,
  threshold = c(0, 0.05, seq(0.1, 1, 0.1)),
  condopt,
  uncopt,
  ...
)
```

Arguments

`formula`	The model to be analysed. It should be of the form y ~ x1 + x2 + ..., where y is the outcome variable and Xs are the predictors.
`random`	A string specifying the "clustering variable" (e.g., schools or sites) as found in the dataset.
`intervention`	A string specifying the "intervention variable" as it appears in the formula.
`baseln`	A string specifying the reference category for the intervention variable. If not provided, the first level will be used as the reference (e.g., baseln = "0" for an intervention with levels 0 and 1).
`nsim`	Number of MCMC iterations to be performed. A minimum of 10,000 is recommended to ensure convergence.
`data`	A data frame containing the variables referenced in the formula, including predictors, the clustering variable, and the intervention.
`alpha`	significant level, default alpha = 0.05.
`digits`	number of decimal places, by default digits=3

`threshold`	a scalar or vector of pre-specified threshold(s) for estimating Bayesian posterior probability such that the observed effect size is greater than or equal to the threshold(s).
`condopt`	additional arguments of `jags` to be passed exclusively to the conditional model (e.g., defining n.chains only for the conditional model, etc.).
`uncopt`	additional arguments of `jags` to be passed exclusively to the unconditional model (e.g., defining n.chains only for the unconditional model, etc.).
`...`	Common additional arguments of `jags` to be passed to both the conditional and unconditional model specifications

Details

The function provides posterior estimates for fixed effects (predictors) and random effects (clustering) under a Bayesian framework. Effect sizes are computed using Hedges' g, and variance components are decomposed into between-cluster and within-cluster variances.

Value

S3 object; a list consisting of:

- `Beta`: Estimates and credible intervals for the predictors specified in the model (posterior distributions).
- `ES`: Hedges' g effect size for the intervention(s). If bootstrapping is not used, 95% credible intervals are computed based on MCMC sampling.
- `covParm`: Variance components broken down into between-cluster variance (e.g., between schools), within-cluster variance (e.g., within pupils), and intra-cluster correlation (ICC).
- `randomEffects`: Posterior estimates of random intercepts for each cluster (e.g., schools).
- `ProbES`: A matrix showing the probability of observing an effect size larger than various thresholds (0, 0.05, 0.10, ...).
- `Model`: A model object from `jags` and an `MCMCsummary` object containing only the mean and credible intervals (CIs) as columns.
- `Unconditional`: A list containing the unconditional effect size and variance decomposition.

Examples

```
if(interactive()){

  data(mstData)

  ########################################################
  ## Bayesian analysis of multisite randomised trials   ##
  ########################################################

  output <- mstBayes(formula = Posttest ~ Prettest + Intervention,
                     random = "School",
                     intervention = "Intervention",
                     alpha = 0.05,
                     digits = 3,
```

```
                     nsim = 10000,
                     data = mstData)
  output

  ### Fixed effects
  beta <- output$Beta
  beta

  ### Effect size
  ES1 <- output$ES
  ES1

  ## Covariance matrix
  covParm <- output$covParm
  covParm

  ## Prob ES
  ProbES <- output$ProbES
  ProbES

  ## Unconditional
  Unconditional <- output$Unconditional
  Unconditional

  ## Random Effect
  randomEffects <- output$SchEffects
  randomEffects


  ### plot random effects for schools

  plot(output)

 ### plot posterior probability of an effect size to be bigger than a pre-specified threshold

  plot(output,group=1)

}
```

`mstData` *Multisite Trial Data.*

Description

A multisite trial dataset containing 54 schools. This data contains a random sample of test data of pupils and not actual trial data.

Format

A data frame with 210 rows and 5 variables

Details

- Posttest: posttest scores
- Prettest: prettest scores
- Intervention: the indicator for the intervention groups in a two arm trial, coded as 1 for intervention group and 0 for control group.
- Intervention2: a simulated indicator for intervention groups in a three arm trial.
- School: numeric school identifier

`mstFREQ` *Analysis of Multisite Randomised Education Trials using Multilevel Model under a Frequentist Setting.*

Description

`mstFREQ` performs analysis of multisite randomised education trials using a multilevel model under a frequentist setting.

Usage

```
mstFREQ(
  formula,
  random,
  intervention,
  baseln,
  nPerm,
  data,
  type,
  ci,
  seed,
  nBoot
)
```

Arguments

`formula`	the model to be analysed is of the form y ~ x1+x2+.... Where y is the outcome variable and Xs are the independent variables.
`random`	a string variable specifying the "clustering variable" as contained in the data. See example below.
`intervention`	a string variable specifying the "intervention variable" as appearing in the formula and the data. See example below.
`baseln`	A string variable allowing the user to specify the reference category for intervention variable. When not specified, the first level will be used as a reference.
`nPerm`	number of permutations required to generate permutated p-value.
`data`	data frame containing the data to be analysed.

`type`	method of bootstrapping including case re-sampling at student level "case(1)", case re-sampling at school level "case(2)", case re-sampling at both levels "case(1,2)" and residual bootstrapping using "residual". If not provided, default will be case re-sampling at student level.
`ci`	method for bootstrap confidence interval calculations; options are the Basic (Hall's) confidence interval "basic" or the simple percentile confidence interval "percentile". If not provided default will be percentile.
`seed`	seed required for bootstrapping and permutation procedure, if not provided default seed will be used.
`nBoot`	number of bootstraps required to generate bootstrap confidence intervals.

Value

S3 object; a list consisting of

- `Beta`: Estimates and confidence intervals for variables specified in the model.
- `ES`: Conditional Hedge's g effect size (ES) and its 95% confidence intervals. If nBoot is not specified, 95% confidence intervals are based on standard errors. If nBoot is specified, they are non-parametric bootstrapped confidence intervals.
- `covParm`: A list of variance decomposition into between cluster variance-covariance matrix (schools and school by intervention) and within cluster variance (Pupils). It also contains intra-cluster correlation (ICC).
- `SchEffects`: A vector of the estimated deviation of each school from the intercept and intervention slope.
- `Perm`: A "nPerm x 2w" matrix containing permutated effect sizes using residual variance and total variance. "w" denotes number of intervention. "w=1" for two arm trial and "w=2" for three arm trial excluding the control group. It is produced only when `nPerm` is specified.
- `Bootstrap`: A "nBoot x 2w" matrix containing the bootstrapped effect sizes using residual variance (Within) and total variance (Total). "w" denotes number of intervention. "w=1" for two arm trial and "w=2" for three arm trial excluding the control group. It is only produced when `nBoot` is specified.
- `Unconditional`: A list of unconditional effect sizes, covParm, Perm and Bootstrap obtained based on variances from the unconditional model (model with only the intercept as a fixed effect).

Examples

```
if(interactive()){

data(mstData)

###############################################
## MLM analysis of multisite trials + 1.96SE ##
###############################################

output1 <- mstFREQ(Posttest~ Intervention+Prettest,random="School",
intervention="Intervention",data=mstData)
```

```
### Fixed effects
beta <- output1$Beta
beta

### Effect size
ES1 <- output1$ES
ES1

## Covariance matrix
covParm <- output1$covParm
covParm

### plot random effects for schools

plot(output1)

##################################################
## MLM analysis of multisite trials             ##
## with residual bootstrap confidence intervals ##
##################################################

output2 <- mstFREQ(Posttest~ Intervention+Prettest,random="School",
intervention="Intervention",nBoot=1000,type="residual",data=mstData)

tp <- output2$Bootstrap
### Effect size

ES2 <- output2$ES
ES2

### plot bootstrapped values

plot(output2, group=1)

#######################################################################
## MLM analysis of mutltisite trials with permutation p-value##
#######################################################################

output3 <- mstFREQ(Posttest~ Intervention+Prettest,random="School",
intervention="Intervention",nPerm=1000,data=mstData)

ES3 <- output3$ES
ES3

#### plot permutated values

plot(output3, group=1)
}
```

`plot.eefAnalytics` *A plot method for an eefAnalytics S3 object obtained from the eefAnalytics package.*

Description

Plots different figures based on output from eefAnalytics package.

Usage

```
## S3 method for class 'eefAnalytics'
plot(x, group, Conditional = TRUE, ES_Total = TRUE, slope = FALSE, ...)
```

Arguments

`x`	an output object from the eefAnalytics package.
`group`	a string/scalar value indicating which intervention to plot. This must be one of the values of intervention variable excluding the control group. For a two arm trial, the maximum number of values to consider is 1 and 2 for three arm trial.
`Conditional`	a logical value to indicate whether to plot the conditional effect size. The default is Conditional=TRUE, otherwise Conditional=FALSE should be specified for plot based on the unconditional effect size. Conditional variance is total or residual variance from a multilevel model with fixed effects, whilst unconditional variance is total variance or residual variance from a multilevel model with only intercept as fixed effect.
`ES_Total`	A logical value indicating whether to plot the effect size based on total variance or within school variance. The default is ES_Total=TRUE, to plot the effect size using total variance. ES_Total=FALSE should be specified for the effect size based on within school or residuals variance.
`slope`	A logical value indicating whether to return the plot of random intercept (default is slope=FALSE). return other school-by-intervention interaction random slope (s) is slope=TRUE. This argument is suitable only for mstBayes and mstFREQ functions.
`...`	arguments passed to `plot.default`

Details

Plot produces a graphical visualisation depending on which model is fitted:

- For `srtFREQ()`, plot can only be used when `nBoot` or `nPerm` is specified to visualise the distribution of bootstrapped or permutated values.
- For `crtFREQ()` or `mstFREQ()`, plot shows the distribution of random intercepts when `group=NULL`. It produces histogram of permutated or bootstrapped values when `group` is specified and either `nBoot` or `nPerm` is also specified.

Value

Returns relevant plots for each model.

Examples

```
if(interactive()){

#### read data
data(mstData)
data(crtData)


###############
##### SRT #####
###############

##### Bootstrapped

outputSRTBoot <- srtFREQ(Posttest~ Intervention + Prettest,
                         intervention = "Intervention",nBoot=1000, data = mstData)
plot(outputSRTBoot,group=1)

##### Permutation
outputSRTPerm <- srtFREQ(Posttest~ Intervention + Prettest,
                         intervention = "Intervention",nPerm=1000, data = mstData)

plot(outputSRTPerm,group=1)


###############
##### MST #####
###############


#### Random intercepts
outputMST <- mstFREQ(Posttest~ Intervention + Prettest,
                     random = "School", intervention = "Intervention", data = mstData)
plot(outputMST)


#### Bootstrapped
outputMSTBoot <- mstFREQ(Posttest~ Intervention + Prettest,
                         random = "School", intervention = "Intervention",
                         nBoot = 1000, data = mstData)

plot(outputMSTBoot)
plot(outputMSTBoot,group=1)

#### Permutation
outputMSTPerm <- mstFREQ(Posttest~ Intervention + Prettest,
                         random = "School", intervention = "Intervention",
                         nPerm = 1000, data = mstData)
```

```
plot(outputMSTPerm)
plot(outputMSTPerm,group=1)



###############
##### CRT #####
###############

#### Random intercepts
outputCRT <- crtFREQ(Posttest~ Intervention + Prettest, random = "School",
                     intervention = "Intervention", data = crtData)
plot(outputCRT)


## Bootstrapped
outputCRTBoot <- crtFREQ(Posttest~ Intervention + Prettest, random = "School",
                         intervention = "Intervention", nBoot = 1000, data = crtData)

plot(outputCRTBoot,group=1)


##Permutation
outputCRTPerm <- crtFREQ(Posttest~ Intervention + Prettest, random = "School",
                         intervention = "Intervention", nPerm = 1000, data = crtData)

plot(outputCRTPerm,group=1)
}
```

`print.eefAnalytics` *Print for a fitted model represented by an* `eefAnalytics` *object.*

Description

Print for a fitted model represented by an `eefAnalytics` object.

Usage

```
## S3 method for class 'eefAnalytics'
print(x, ...)
```

Arguments

`x`	Object of class `eefAnalytics`
`...`	Additional arguments of `print`

Value

Print conditional and unconditional effect sizes.

`srtBayes` *Bayesian Analysis of Simple Randomised Education Trials (SRT) using Bayesian Linear Regression Model with Vague Priors.*

Description

`srtBayes` performs Bayesian multilevel analysis of Simple Randomised Education Trials (SRT), utilising vague priors and JAGS language to fit the model. This can also be used with schools as fixed effects.

Usage

```
srtBayes(
  formula,
  intervention,
  baseln,
  nsim = 2000,
  data,
  alpha = 0.05,
  digits = 3,
  threshold = 1:10/10,
  condopt,
  uncopt,
  ...
)
```

Arguments

`formula`	The model to be analysed is of the form y~x1+x2+.... Where y is the outcome variable and Xs are the independent variables.
`intervention`	A string variable specifying the "intervention variable" as appearing in the formula and the data. See example below.
`baseln`	A string variable allowing the user to specify the reference category for intervention variable. When not specified, the first level will be used as a reference.
`nsim`	number of MCMC iterations per chain. Default is 2000.
`data`	Data frame containing the data to be analysed.
`alpha`	significant level, default alpha = 0.05.
`digits`	number of decimal places, by default digits=3
`threshold`	a scalar or vector of pre-specified threshold(s) for estimating Bayesian posterior probability such that the observed effect size is greater than or equal to the threshold(s).
`condopt`	additional arguments of `jags` to be passed only to the conditional model specification (for example, defining priors only for the conditional model, etc.).
`uncopt`	additional arguments of `jags` to be passed only to the unconditional model specification (for example, defining priors only for the unconditional model, etc.).

`...`	Common additional arguments of `jags` to be passed to both the conditional and unconditional model specifications

Value

S3 object; a list consisting of

- `Beta`: Estimates and credible intervals for the variables specified in the model. Use `summary.eefAnalytics` to get Rhat and effective sample size for each estimate.
- `ES`: Conditional Hedges' g effect size and its 95% credible intervals.
- `sigma2`: Residual variance.
- `ProbES`: A matrix of Bayesian posterior probabilities such that the observed effect size is greater than or equal to a pre-specified threshold(s).
- `Model`: A model object from `jags` and an `MCMCsummary` object containing only the mean and credible intervals (CIs) as columns.
- `Unconditional`: A list of unconditional effect sizes, sigma2 and ProbES obtained based on residual variance from the unconditional model (model with only the intercept as a fixed effect).

Examples

```
if(interactive()){

  data(mstData)

  ########################################################
  ## Bayesian analysis of simple randomised trials      ##
  ########################################################

  output <- srtBayes(Posttest~ Intervention+Prettest,
                    alpha = 0.2,
                    digits=4,
                    intervention="Intervention",
                    nsim=10000,
                    data=mstData)

  ### Fixed effects
  beta <- output$Beta
  beta

  ### Effect size
  ES1 <- output$ES
  ES1

  ### Effect size
  ES2 <- output$Unconditional$ES
  ES2

  ## Covariance matrix
  covParm1 <- output$sigma2
```

```
  covParm1


  ## Unconditional Covariance matrix
  covParm2 <- output$Unconditional$sigma2
  covParm2


  ## Prob ES
  ProbES1 <- output$ProbES
  ProbES1


  ## Prob  based on Unconditional ES
  ProbES2 <- output$Unconditional$ProbES
  ProbES2


 ### plot posterior probability of an effect size to be bigger than a pre-specified threshold

  plot(output,group=1)


}
```

`srtFREQ` *Analysis of Simple Randomised Education Trial using Linear Regression Model.*

Description

`srtFREQ` performs analysis of educational trials under the assumption of independent errors among pupils. This can also be used with schools as fixed effects.

Usage

```
srtFREQ(formula, intervention, baseln, nBoot, nPerm, ci, seed, data)
```

Arguments

`formula`	the model to be analysed is of the form y~x1+x2+.... Where y is the outcome variable and Xs are the independent variables.
`intervention`	a string variable specifying the "intervention variable" as appearing in the formula and the data. See example below.
`baseln`	A string variable allowing the user to specify the reference category for intervention variable. When not specified, the first level will be used as a reference.
`nBoot`	number of bootstraps required to generate bootstrap confidence intervals.
`nPerm`	number of permutations required to generate permutated p-value.

`ci`	method for bootstrap confidence interval calculations; options are the Basic (Hall's) confidence interval "basic" or the simple percentile confidence interval "percentile". If not provided default will be percentile.
`seed`	seed required for bootstrapping and permutation procedure, if not provided default seed will be used.
`data`	data frame containing the data to be analysed.

Value

S3 object; a list consisting of

- `Beta`: Estimates and confidence intervals for the variables specified in the model.
- `ES`: Conditional Hedges'g effect size and its 95% confidence intervals. If nBoot is not specified, 95% confidence intervals are based on standard errors. If nBoot is specified, they are non-parametric bootstrapped confidence intervals.
- `sigma2`: Residual variance.
- `Perm`: A "nPerm x w" matrix containing permutated effect sizes using residual variance. "w" denotes number of intervention. "w=1" for two arm trial and "w=2" for three arm trial excluding the control group. It is produced only if `nPerm` is specified.
- `Bootstrap`: A "nBoot x w" matrix containing the bootstrapped effect sizes using residual variance. "w" denotes number of intervention. "w=1" for two arm trial and "w=2" for three arm trial excluding the control group. It is produced only if `nBoot` is specified.
- `Unconditional`: A list of unconditional effect size, sigma2, Perm and Bootstrap obtained based on variances from the unconditional model (model with only intercept as fixed effect).

Examples

```
if(interactive()){

data(mstData)

###################################################################
## Analysis of simple randomised trials using Hedges Effect Size ##
###################################################################

output1 <- srtFREQ(Posttest~ Intervention+Prettest,
intervention="Intervention",data=mstData )
ES1 <- output1$ES
ES1

###################################################################
## Analysis of simple randomised trials using Hedges Effect Size ##
## with Permutation p-value                                      ##
###################################################################

output2 <- srtFREQ(Posttest~ Intervention+Prettest,
intervention="Intervention",nPerm=1000,data=mstData )

ES2 <- output2$ES
```

```
ES2


#### plot permutated values

plot(output2, group=1)



###################################################################
## Analysis of simple randomised trials using Hedges Effect Size ##
## with non-parametric Basic bootstrap confidence intervals      ##
###################################################################

output3 <- srtFREQ(Posttest~ Intervention+Prettest,
intervention=”Intervention”,nBoot=1000,ci=”basic”,data=mstData)

ES3 <- output3$ES
ES3

### plot bootstrapped values

plot(output3, group=1)

####################################################################
## Analysis of simple randomised trials using Hedges' effect size  ##
##  with schools as fixed effects                                  ##
####################################################################

output4 <- srtFREQ(Posttest~ Intervention+Prettest+as.factor(School),
intervention=”Intervention”,data=mstData )

ES4 <- output4$ES
ES4

####################################################################
## Analysis of simple randomised trials using Hedges' effect size ##
## with schools as fixed effects and with permutation p-value     ##
####################################################################

output5 <- srtFREQ(Posttest~ Intervention+Prettest+as.factor(School),
intervention=”Intervention”,nPerm=1000,data=mstData )

ES5 <- output5$ES
ES5

#### plot permutated values

plot(output5, group=1)

####################################################################
## Analysis of simple randomised trials using Hedges' effect size ##
## with schools as fixed effects and with permutation p-value      ##
```

```
####################################################################

output6 <- srtFREQ(Posttest~ Intervention+Prettest+as.factor(School),
intervention="Intervention",nBoot=1000,data=mstData)

ES6 <- output6$ES
ES6

### plot bootstrapped values

plot(output6, group=1)
}
```

`summary.eefAnalytics` *Summary for a fitted model represented by an* `eefAnalytics` *object.*

Description

Summary for a fitted model represented by an `eefAnalytics` object.

Usage

```
## S3 method for class 'eefAnalytics'
summary(object, ...)
```

Arguments

`object`	Object of class `eefAnalytics`
`...`	Additional arguments of `summary`

Value

Returns relevant summary including Rhat and effective sample sizes.

Index