Package 'AATtools'

November 22, 2019

Type Package

Title Tools for Analyzing the Approach-Avoidance Task

Version 0.0.1

Author Sercan Kahveci

Description Compute approach bias scores using different scoring algorithms,
compute split-half reliability of the AAT using bootstrapping,
and compute confidence intervals for individual AAT scores using bootstrapping.

Depends R (¿= 3.6.1), magrittr, dplyr, doParallel, lmerTest

Imports tidyr

License GPL-3

Encoding UTF-8

BugReports https://github.com/Spiritspeak/AATtools/issues

LazyData true

ByteCompile true

RoxygenNote 6.1.1

R topics documented:

`aat_bootstrap` *Compute bootstrapped approach-bias scores*

Description

Compute bootstrapped approach-bias scores with confidence intervals.

Compute bootstrapped approach-bias scores with confidence intervals.

Usage

```
aat_bootstrap(ds, subjvar, pullvar, targetvar, rtvar, iters, plot = T,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff",
  "aat_dscore", "aat_dscore_multiblock", "aat_multilevelscore"),
  trialdropfunc = c("prune_nothing", "trial_prune_3SD"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus"),
  ...)

aat_bootstrap(ds, subjvar, pullvar, targetvar, rtvar, iters, plot = T,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff",
  "aat_dscore", "aat_dscore_multiblock", "aat_multilevelscore"),
  trialdropfunc = c("prune_nothing", "trial_prune_3SD"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus"),
  ...)
```

Arguments

`ds`	a longformat data.frame
`subjvar`	Quoted name of the participant identifier column
`pullvar`	Quoted name of the column indicating pull trials. Pull trials should either be represented by 1, or by the second level of a factor.
`targetvar`	Name of the column indicating trials featuring the target stimulus. Target stimuli should either be represented by 1, or by the second level of a factor.
`rtvar`	Name of the reaction time column.
`iters`	Total number of desired iterations. At least 200 are required to get confidence intervals that make sense.
`plot`	Plot the bias scores and their confidence intervals after computation is complete. This gives a good overview of the data.
`algorithm`	Function (without brackets or quotes) to be used to compute AAT scores. See aat_doublemeandiff for a list of usable algorithms.
`trialdropfunc`	Function (without brackets or quotes) to be used to exclude outlying trials in each half. `prune_nothing` excludes no trials, while `trial_prune_3SD` excludes trials deviating more than 3SD from the mean per participant.
`errortrialfunc`	Function (without brackets or quotes) to apply to an error trial. `error_replace_blockmeanplu` replaces error trial reaction times with the block mean plus an arbitrary extra amount of time. If used, the following additional arguments are required: • `blockvar` - Quoted name of the block variable • `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE • `errorbonus` - Amount to add to the reaction time of error trials. Default is 0.6 (recommended by `Greenwald,Nosek,& Banaji,2003`)
`...`	Other arguments, to be passed on to the algorithm functions (see `algorithm` above)
`ds`	a longformat data.frame
`subjvar`	Quoted name of the participant identifier column
`pullvar`	Quoted name of the column indicating pull trials. Pull trials should either be represented by 1, or by the second level of a factor.

targetvar	Name of the column indicating trials featuring the target stimulus. Target stimuli should either be represented by 1, or by the second level of a factor.
rtvar	Name of the reaction time column.
iters	Total number of desired iterations. At least 200 are required to get confidence intervals that make sense.
plot	Plot the bias scores and their confidence intervals after computation is complete. This gives a good overview of the data.
algorithm	Function (without brackets or quotes) to be used to compute AAT scores. See aat_doublemeandiff for a list of usable algorithms.
trialdropfunc	Function (without brackets or quotes) to be used to exclude outlying trials in each half. prune_nothing excludes no trials, while trial_prune_3SD excludes trials deviating more than 3SD from the mean per participant.
errortrialfunc	Function (without brackets or quotes) to apply to an error trial. error_replace_blockmeanplu replaces error trial reaction times with the block mean plus an arbitrary extra amount of time. If used, the following additional arguments are required: • blockvar - Quoted name of the block variable • errorvar - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE • errorbonus - Amount to add to the reaction time of error trials. Default is 0.6 (recommended by Greenwald,Nosek,& Banaji,2003)
...	Other arguments, to be passed on to the algorithm functions (see algorithm above)

Value

A list, containing bootstrapped bias scores, a data frame with bootstrapped 95 the number of iterations, and a matrix of bias scores for each iteration.

A list, containing bootstrapped bias scores, a data frame with bootstrapped 95 the number of iterations, and a matrix of bias scores for each iteration.

Author(s)

Sercan Kahveci

Sercan Kahveci

---

aat_splithalf *Compute the bootstrapped split-half reliability for approach-avoidance task data*

Description

Compute bootstrapped split-half reliability for approach-avoidance task data. aat_splithalf() uses multicore computation, which is fast, but provides no clear output when there are errors. aat_splithalf_singlecore() is much slower, but more easily debugged.

Compute bootstrapped split-half reliability for approach-avoidance task data. aat_splithalf() uses multicore computation, which is fast, but provides no clear output when there are errors. aat_splithalf_singlecore() is much slower, but more easily debugged.

Usage

```
aat_splithalf(ds, subjvar, pullvar, targetvar, rtvar, iters, plot = T,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff",
  "aat_dscore", "aat_dscore_multiblock", "aat_multilevelscore"),
  trialdropfunc = c("prune_nothing", "trial_prune_3SD",
  "trial_prune_dropcases"), errortrialfunc = c("prune_nothing",
  "error_replace_blockmeanplus", "error_prune_dropcases"),
  casedropfunc = c("prune_nothing", "case_prune_3SD"), ...)

aat_splithalf_singlecore(ds, subjvar, pullvar, targetvar, rtvar, iters,
  plot = T, algorithm = c("aat_doublemeandiff", "aat_doublemediandiff",
  "aat_dscore", "aat_dscore_multiblock", "aat_multilevelscore"),
  trialdropfunc = c("prune_nothing", "trial_prune_3SD"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus"),
  casedropfunc = c("prune_nothing", "case_prune_3SD"), ...)

aat_splithalf(ds, subjvar, pullvar, targetvar, rtvar, iters, plot = T,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff",
  "aat_dscore", "aat_dscore_multiblock", "aat_multilevelscore"),
  trialdropfunc = c("prune_nothing", "trial_prune_3SD",
  "trial_prune_dropcases"), errortrialfunc = c("prune_nothing",
  "error_replace_blockmeanplus", "error_prune_dropcases"),
  casedropfunc = c("prune_nothing", "case_prune_3SD"), ...)

aat_splithalf_singlecore(ds, subjvar, pullvar, targetvar, rtvar, iters,
  plot = T, algorithm = c("aat_doublemeandiff", "aat_doublemediandiff",
  "aat_dscore", "aat_dscore_multiblock", "aat_multilevelscore"),
  trialdropfunc = c("prune_nothing", "trial_prune_3SD"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus"),
  casedropfunc = c("prune_nothing", "case_prune_3SD"), ...)
```

Arguments

ds	a longformat data.frame
subjvar	Quoted name of the participant identifier column
pullvar	Quoted name of the column indicating pull trials. Pull trials should either be represented by 1, or by the second level of a factor.
targetvar	Name of the column indicating trials featuring the target stimulus. Target stimuli should either be represented by 1, or by the second level of a factor.
rtvar	Name of the reaction time column.
iters	Total number of desired iterations. At least 200 are recommended for reasonable confidence intervals; If you want to see plots of your data, 1 iteration is enough.
plot	Create a scatterplot of the AAT scores computed from each half of the data from the last iteration. This is highly recommended, as it helps to identify outliers that can inflate or diminish the reliability.
algorithm	Function (without brackets or quotes) to be used to compute AAT scores. See Algorithms for a list of usable algorithms.
trialdropfunc	Function (without brackets or quotes) to be used to exclude outlying trials in each half. The way you handle outliers for the reliability computation

should mimic the way you do it in your regular analyses. It is recommended to exclude outlying trials when computing AAT scores using the mean double-dfference scores and multilevel scoring approaches, but not when using d-scores or median double-difference scores. `prune_nothing` excludes no trials, `trial_prune_3SD` excludes trials deviating more than 3SD from the mean per participant. `trial_prune_dropcases` allows you to set the maximum standard deviation to include using argument `trialsd` (default is 3) and prune participants altogether if they have more than a certain proportion of outliers using argument `maxoutliers` (default is .15)

`errortrialfunc` Function (without brackets or quotes) to apply to an error trial.

`error_replace_blockmeanplus` replaces error trial reaction times with the block mean plus an arbitrary extra amount of time. If used, the following additional arguments are required:

- `blockvar` - Quoted name of the block variable
- `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE
- `errorbonus` - Amount to add to the reaction time of error trials. Default is 0.6 (recommended by `Greenwald,Nosek,& Banaji,2003`)

`error_prune_dropcases` removes errors and drops participants if they have a larger proportion of errors than given by argument `maxerrors`, default is .15

`casedropfunc` Function (without brackets or quotes) to be used to exclude outlying participant scores in each half. The way you handle outliers here should mimic the way you do it in your regular analyses. `prune_nothing` excludes no participants, while `case_prune_3SD` excludes participants deviating more than 3SD from the sample mean.

`...` Other arguments, to be passed on to the algorithm functions (see `algorithm` above)

`ds` a longformat data.frame

`subjvar` Quoted name of the participant identifier column

`pullvar` Quoted name of the column indicating pull trials. Pull trials should either be represented by 1, or by the second level of a factor.

`targetvar` Name of the column indicating trials featuring the target stimulus. Target stimuli should either be represented by 1, or by the second level of a factor.

`rtvar` Name of the reaction time column.

`iters` Total number of desired iterations. At least 200 are recommended for reasonable confidence intervals; If you want to see plots of your data, 1 iteration is enough.

`plot` Create a scatterplot of the AAT scores computed from each half of the data from the last iteration. This is highly recommended, as it helps to identify outliers that can inflate or diminish the reliability.

`algorithm` Function (without brackets or quotes) to be used to compute AAT scores. See Algorithms for a list of usable algorithms.

`trialdropfunc` Function (without brackets or quotes) to be used to exclude outlying trials in each half. The way you handle outliers for the reliability computation should mimic the way you do it in your regular analyses. It is recommended to exclude outlying trials when computing AAT scores using the

mean double-dfference scores and multilevel scoring approaches, but not when using d-scores or median double-difference scores. `prune_nothing` excludes no trials, `trial_prune_3SD` excludes trials deviating more than 3SD from the mean per participant. `trial_prune_dropcases` allows you to set the maximum standard deviation to include using argument `trialsd` (default is 3) and prune participants altogether if they have more than a certain proportion of outliers using argument `maxoutliers` (default is .15)

`errortrialfunc` Function (without brackets or quotes) to apply to an error trial.

`error_replace_blockmeanplus` replaces error trial reaction times with the block mean plus an arbitrary extra amount of time. If used, the following additional arguments are required:

- `blockvar` - Quoted name of the block variable
- `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE
- `errorbonus` - Amount to add to the reaction time of error trials. Default is 0.6 (recommended by `Greenwald,Nosek,& Banaji,2003`)

`error_prune_dropcases` removes errors and drops participants if they have a larger proportion of errors than given by argument `maxerrors`, default is .15

`casedropfunc` Function (without brackets or quotes) to be used to exclude outlying participant scores in each half. The way you handle outliers here should mimic the way you do it in your regular analyses. `prune_nothing` excludes no participants, while `case_prune_3SD` excludes participants deviating more than 3SD from the sample mean.

`...` Other arguments, to be passed on to the algorithm functions (see `algorithm` above)

Value

A list, containing the mean bootstrapped split-half reliability, bootstrapped 95 a list of data.frames used over each iteration, and a vector containing the split-half reliability of each iteration.

A list, containing the mean bootstrapped split-half reliability, bootstrapped 95 a list of data.frames used over each iteration, and a vector containing the split-half reliability of each iteration.

Author(s)

Sercan Kahveci

Sercan Kahveci

See Also

plot.aat_splithalf()

plot.aat_splithalf

Examples

```
#Not Run
aat_splithalf(ds=ds2,subjvar="subjectid",pullvar="is_pull",targetvar="is_food",
              rtvar="rt",iters=1000,trialdropfunc=trial_prune_3SD,
              casedropfunc=case_prune_3SD,plot=T,algorithm=aat_dscore)
#Mean reliability: 0.521959
#Spearman-Brown-corrected r: 0.6859041
#95%CI: [0.4167018, 0.6172474]

#Multilevel Splithalf
aat_splithalf(ds=ds2,subjvar="subjectid",pullvar="is_pull",targetvar="is_food",
              rtvar="rt",iters=100,trialdropfunc=trial_prune_3SD,
              casedropfunc=case_prune_3SD,plot=T,algorithm=aat_multilevelscore,
              formula = "rt ~ is_pull * is_food + (is_pull * is_food | subjectid)",
              aatterm = "is_pull:is_food")
#Mean reliability: 0.5313939
#Spearman-Brown-corrected r: 0.6940003
#95%CI: [0.2687186, 0.6749176]
#Not Run
aat_splithalf(ds=ds2,subjvar="subjectid",pullvar="is_pull",targetvar="is_food",
              rtvar="rt",iters=1000,trialdropfunc=trial_prune_3SD,
              casedropfunc=case_prune_3SD,plot=T,algorithm=aat_dscore)
#Mean reliability: 0.521959
#Spearman-Brown-corrected r: 0.6859041
#95%CI: [0.4167018, 0.6172474]

#Multilevel Splithalf
aat_splithalf(ds=ds2,subjvar="subjectid",pullvar="is_pull",targetvar="is_food",
              rtvar="rt",iters=100,trialdropfunc=trial_prune_3SD,
              casedropfunc=case_prune_3SD,plot=T,algorithm=aat_multilevelscore,
              formula = "rt ~ is_pull * is_food + (is_pull * is_food | subjectid)",
              aatterm = "is_pull:is_food")
#Mean reliability: 0.5313939
#Spearman-Brown-corrected r: 0.6940003
#95%CI: [0.2687186, 0.6749176]
```

`Algorithms` *AAT score computation algorithms*

Description

- `aat_doublemeandiff` computes a mean-based double-difference score:
  `(mean(push_target) -mean(pull_target)) -(mean(push_control) -mean(pull_control))`
- `aat_doublemediandiff` computes a median-based double-difference score:
  `(median(push_target) -median(pull_target)) -(median(push_control) -median(pull_control))`
- `aat_dscore` computes D-scores for a 2-block design (see Greenwald, Nosek, and Banaji, 2003):
  `((mean(push_target) -mean(pull_target)) -(mean(push_control) -mean(pull_control)))`
  `/ sd(participant_reaction_times)`
- `aat_dscore_multiblock` computes D-scores for pairs of sequential blocks and averages the resulting score (see Greenwald, Nosek, and Banaji, 2003). Requires extra `blockvar` argument, indicating the name of the block variable.

- `aat_multilevelscore` fits a multilevel model using lme4 and extracts a random effect serving as AAT score. When using this function, additional arguments must be provided:
  - `formula` - a quoted formula to fit to the data;
  - `aatterm` the quoted random effect within the subject variable that indicates the approach bias; this is usually the interaction of the pull and target terms.

- `aat_doublemeandiff` computes a mean-based double-difference score:
  `(mean(push_target) -mean(pull_target)) -(mean(push_control) -mean(pull_control))`
- `aat_doublemediandiff` computes a median-based double-difference score:
  `(median(push_target) -median(pull_target)) -(median(push_control) -median(pull_control))`
- `aat_dscore` computes D-scores for a 2-block design (see Greenwald, Nosek, and Banaji, 2003):
  `((mean(push_target) -mean(pull_target)) -(mean(push_control) -mean(pull_control)))`
  `/ sd(participant_reaction_times)`
- `aat_dscore_multiblock` computes D-scores for pairs of sequential blocks and averages the resulting score (see Greenwald, Nosek, and Banaji, 2003). Requires extra `blockvar` argument, indicating the name of the block variable.
- `aat_multilevelscore` fits a multilevel model using lme4 and extracts a random effect serving as AAT score. When using this function, additional arguments must be provided:
  - `formula` - a quoted formula to fit to the data;
  - `aatterm` the quoted random effect within the subject variable that indicates the approach bias; this is usually the interaction of the pull and target terms.

Usage

```
aat_doublemeandiff(ds, subjvar, pullvar, targetvar, rtvar, ...)

aat_doublemediandiff(ds, subjvar, pullvar, targetvar, rtvar, ...)

aat_dscore(ds, subjvar, pullvar, targetvar, rtvar, ...)

aat_dscore_multiblock(ds, subjvar, pullvar, targetvar, rtvar, blockvar,
  ...)

aat_multilevelscore(ds, subjvar, formula, aatterm, ...)

aat_doublemeandiff(ds, subjvar, pullvar, targetvar, rtvar, ...)

aat_doublemediandiff(ds, subjvar, pullvar, targetvar, rtvar, ...)

aat_dscore(ds, subjvar, pullvar, targetvar, rtvar, ...)

aat_dscore_multiblock(ds, subjvar, pullvar, targetvar, rtvar, blockvar,
  ...)

aat_multilevelscore(ds, subjvar, formula, aatterm, ...)
```

Arguments

`ds`	A long-format data.frame
`subjvar`	Column name of the participant identifier variable
`pullvar`	Column name of the movement variable (0: avoid; 1: approach)
`targetvar`	Column name of the stimulus category variable (0: control stimulus; 1: target stimulus)
`rtvar`	Column name of the reaction time variable
`...`	Other arguments passed on by functions (ignored)
`blockvar`	name of the variable indicating block number
`formula`	A character string containing a formula to fit to the data and derive multilevel scores from
`aatterm`	The random term, grouped under the subject variable, which represents the approach bias. Usually this is the interaction of the pull and target terms.
`ds`	A long-format data.frame
`subjvar`	Column name of the participant identifier variable
`pullvar`	Column name of the movement variable (0: avoid; 1: approach)
`targetvar`	Column name of the stimulus category variable (0: control stimulus; 1: target stimulus)
`rtvar`	Column name of the reaction time variable
`...`	Other arguments passed on by functions (ignored)
`blockvar`	name of the variable indicating block number
`formula`	A character string containing a formula to fit to the data and derive multilevel scores from
`aatterm`	The random term, grouped under the subject variable, which represents the approach bias. Usually this is the interaction of the pull and target terms.

Value

A data.frame containing participant number and computed AAT score.

A data.frame containing participant number and computed AAT score.

`plot.aat_splithalf` *Plot split-half scatterplots*

Description

Plot split-half scatterplots

Plot split-half scatterplots

Usage

```
## S3 method for class 'aat_splithalf'
plot(x, type = c("median", "minimum", "maximum",
  "random"))

## S3 method for class 'aat_splithalf'
plot(x, type = c("median", "minimum", "maximum",
  "random"))
```

Arguments

x	an aat_splithalf object
type	Character argument indicating which iteration should be chosen. Must be an abbreviation of "median" (default), "minimum", "maximum", or "random".
x	an aat_splithalf object
type	Character argument indicating which iteration should be chosen. Must be an abbreviation of "median" (default), "minimum", "maximum", or "random".

Examples

```
#Coming soon
#Coming soon
```

SpearmanBrown *Spearman-Brown corrections for Correlation Coefficients*

Description

Perform a Spearman-Brown correction on the provided correlation score.

Perform a Spearman-Brown correction on the provided correlation score.

Usage

```
SpearmanBrown(corr, ntests = 2, fix.negative = c("nullify",
  "bilateral", "none"))

SpearmanBrown(corr, ntests = 2, fix.negative = c("nullify",
  "bilateral", "none"))
```

Arguments

corr	To-be-corrected correlation coefficient
ntests	An integer indicating how many times larger the full test is, for which the corrected correlation coefficient is being computed. When ntests=2, the formula will compute what the correlation coefficient would be if the test were twice as long.
fix.negative	Determines how to deal with a negative value. "nullify" sets it to zero, "bilateral" applies the correction as if it were a positive number, and then sets it to negative. "none" gives the raw value.

`corr`	To-be-corrected correlation coefficient
`ntests`	An integer indicating how many times larger the full test is, for which the corrected correlation coefficient is being computed. When `ntests=2`, the formula will compute what the correlation coefficient would be if the test were twice as long.
`fix.negative`	Determines how to deal with a negative value. "nullify" sets it to zero, "bilateral" applies the correction as if it were a positive number, and then sets it to negative. "none" gives the raw value.

Details

Correct a correlation coefficient for being based on only a subset of the data.

Correct a correlation coefficient for being based on only a subset of the data.

Value

Spearman-Brown-corrected correlation coefficient.

Spearman-Brown-corrected correlation coefficient.

Index