Package 'AATtools'

May 11, 2020

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 bootstrapped and exact split-half reliability of the AAT, and compute confidence intervals for individual AAT scores.

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

License GPL-3

Encoding UTF-8

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

LazyData true

ByteCompile true

RoxygenNote 7.0.2

R topics documented:

`aat_bootstrap` *Compute bootstrapped approach-bias scores*

Description

Compute bootstrapped approach-bias scores with confidence intervals.

Usage

```
aat_bootstrap(
  ds,
  subjvar,
  pullvar,
  targetvar = NULL,
  rtvar,
  iters,
  plot = T,
  include.raw = F,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff", "aat_dscore",
    "aat_dscore_multiblock", "aat_regression", "aat_standardregression",
    "aat_doublemeanquotient", "aat_doublemedianquotient", "aat_singlemeandiff",
    "aat_singlemediandiff"),
 trialdropfunc = c("prune_nothing", "trial_prune_3SD", "trial_prune_SD_dropcases",
  "trial_recode_SD", "trial_prune_percent_subject", "trial_prune_percent_sample"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus",
    "error_prune_dropcases"),
  ...
)

## S3 method for class 'aat_bootstrap'
print(x)

## S3 method for class 'aat_bootstrap'
plot(x)
```

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. 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 regression scoring approaches, but not when using d-scores or median double-difference scores. • prune_nothing excludes no trials (default)

- `trial_prune_3SD` excludes trials deviating more than 3SD from the mean per participant.
- `trial_prune_SD_dropcases` removes trials deviating more than a specific number of standard deviations from the participant's mean, and removes participants with an excessive percentage of outliers. Required arguments:
  - `trialsd` - trials deviating more than `trialsd` standard deviations from the participant's mean are excluded (optional; default is 3)
  - `maxoutliers` - participants with a higher percentage of outliers are removed from the data. (optional; default is .15)
- `trial_recode_SD` recodes outlying reaction times to the nearest non-outlying value, with outliers defined as reaction times deviating more than a certain number of standard deviations from the participant's mean. Required argument:
  - `trialsd` - trials deviating more than this many standard deviations from the mean are classified as outliers.
- `trial_prune_percent_subject` and `trial_prune_percent_sample` remove trials below and/or above certain percentiles, on a subject-by-subject basis or sample-wide, respectively. The following arguments are available:
  - `lowerpercent` and `uppperpercent` (optional; defaults are .01 and .99).

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

- `prune_nothing` removes no errors (default).
- `error_replace_blockmeanplus` replaces error trial reaction times with the block mean, plus an arbitrary extra quantity. If used, the following additional arguments are required:
  - `blockvar` - Quoted name of the block variable (mandatory)
  - `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE (mandatory)
  - `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 more errors than a given percentage. The following arguments are available:
  - `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE (mandatory)
  - `maxerrors` - participants with a higher percentage of errors are excluded from the dataset. Default is .15.

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

Value

A list, containing bootstrapped bias scores, their variance, bootstrapped 95 percent confidence intervals, the number of iterations, and a matrix of bias scores for each iteration.

Author(s)

Sercan Kahveci

aat_compute *Compute simple AAT scores*

Description

Compute simple AAT scores, with optional outlier exclusion and error trial recoding.

Usage

```
aat_compute(
  ds,
  subjvar,
  pullvar,
  targetvar = NULL,
  rtvar,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff", "aat_dscore",
    "aat_dscore_multiblock", "aat_regression", "aat_standardregression",
    "aat_doublemeanquotient", "aat_doublemedianquotient", "aat_singlemeandiff",
    "aat_singlemediandiff"),
 trialdropfunc = c("prune_nothing", "trial_prune_3SD", "trial_prune_SD_dropcases",
  "trial_recode_SD", "trial_prune_percent_subject", "trial_prune_percent_sample"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus",
    "error_prune_dropcases"),
  ...
)
```

Arguments

ds	a long-format data.frame
subjvar	column name of subject variable
pullvar	column name of pull/push indicator variable, must be numeric or logical (where pull is 1 or TRUE)
targetvar	column name of target stimulus indicator, must be numeric or logical (where target is 1 or TRUE)
rtvar	column name of reaction time variable
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 regression scoring approaches, but not when using d-scores or median double-difference scores. • prune_nothing excludes no trials (default) • trial_prune_3SD excludes trials deviating more than 3SD from the mean per participant. • trial_prune_SD_dropcases removes trials deviating more than a specific number of standard deviations from the participant's mean, and removes participants with an excessive percentage of outliers. Required arguments:

  - `trialsd` - trials deviating more than `trialsd` standard deviations from the participant's mean are excluded (optional; default is 3)
  - `maxoutliers` - participants with a higher percentage of outliers are removed from the data. (optional; default is .15)
- `trial_recode_SD` recodes outlying reaction times to the nearest non-outlying value, with outliers defined as reaction times deviating more than a certain number of standard deviations from the participant's mean. Required argument:
  - `trialsd` - trials deviating more than this many standard deviations from the mean are classified as outliers.
- `trial_prune_percent_subject` and `trial_prune_percent_sample` remove trials below and/or above certain percentiles, on a subject-by-subject basis or sample-wide, respectively. The following arguments are available:
  - `lowerpercent` and `uppperpercent` (optional; defaults are .01 and .99).

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

- `prune_nothing` removes no errors (default).
- `error_replace_blockmeanplus` replaces error trial reaction times with the block mean, plus an arbitrary extra quantity. If used, the following additional arguments are required:
  - `blockvar` - Quoted name of the block variable (mandatory)
  - `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE (mandatory)
  - `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 more errors than a given percentage. The following arguments are available:
  - `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE (mandatory)
  - `maxerrors` - participants with a higher percentage of errors are excluded from the dataset. Default is .15.

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

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

Description

Compute bootstrapped split-half reliability for approach-avoidance task data.

Usage

```
aat_splithalf(
  ds,
  subjvar,
  pullvar,
  targetvar = NULL,
  rtvar,
  iters,
  plot = T,
  include.raw = F,
  cluster = NULL,
  algorithm = c("aat_doublemeandiff", "aat_doublemediandiff", "aat_dscore",
    "aat_dscore_multiblock", "aat_regression", "aat_standardregression",
    "aat_doublemedianquotient", "aat_doublemeanquotient", "aat_singlemeandiff",
    "aat_singlemediandiff"),
 trialdropfunc = c("prune_nothing", "trial_prune_3SD", "trial_prune_SD_dropcases",
  "trial_recode_SD", "trial_prune_percent_subject", "trial_prune_percent_sample"),
  errortrialfunc = c("prune_nothing", "error_replace_blockmeanplus",
    "error_prune_dropcases"),
  casedropfunc = c("prune_nothing", "case_prune_3SD"),
  ...
)

## S3 method for class 'aat_splithalf'
print(x)

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

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.
include.raw	logical indicating whether raw split-half data should be included in the output object.
cluster	pre-specified registered multi-core DoParallel cluster that can be used to speed up computations if multiple calls to aat_splithalf are made. If no cluster is provided, aat_splithalf will start up a cluster each time it is called, which takes some extra time.

`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 regression scoring approaches, but not when using d-scores or median double-difference scores.

- `prune_nothing` excludes no trials (default)
- `trial_prune_3SD` excludes trials deviating more than 3SD from the mean per participant.
- `trial_prune_SD_dropcases` removes trials deviating more than a specific number of standard deviations from the participant's mean, and removes participants with an excessive percentage of outliers. Required arguments:
  - `trialsd` - trials deviating more than `trialsd` standard deviations from the participant's mean are excluded (optional; default is 3)
  - `maxoutliers` - participants with a higher percentage of outliers are removed from the data. (optional; default is .15)
- `trial_recode_SD` recodes outlying reaction times to the nearest non-outlying value, with outliers defined as reaction times deviating more than a certain number of standard deviations from the participant's mean. Required argument:
  - `trialsd` - trials deviating more than this many standard deviations from the mean are classified as outliers.
- `trial_prune_percent_subject` and `trial_prune_percent_sample` remove trials below and/or above certain percentiles, on a subject-by-subject basis or sample-wide, respectively. The following arguments are available:
  - `lowerpercent` and `uppperpercent` (optional; defaults are .01 and .99).

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

- `prune_nothing` removes no errors (default).
- `error_replace_blockmeanplus` replaces error trial reaction times with the block mean, plus an arbitrary extra quantity. If used, the following additional arguments are required:
  - `blockvar` - Quoted name of the block variable (mandatory)
  - `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE (mandatory)
  - `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 more errors than a given percentage. The following arguments are available:
  - `errorvar` - Quoted name of the error variable, where errors are 1 or TRUE and correct trials are 0 or FALSE (mandatory)
  - `maxerrors` - participants with a higher percentage of errors are excluded from the dataset. 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 (default) • case_prune_3SD excludes participants deviating more than 3SD from the sample mean.
...	Other arguments, to be passed on to the algorithm or outlier rejection functions (see arguments above)
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".

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.

Author(s)

Sercan Kahveci

See Also

q_reliability

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]

#Regression 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_regression,
              formula = "rt ~ is_pull * is_food",
              aatterm = "is_pull:is_food")
#Mean reliability: 0.5313939
#Spearman-Brown-corrected r: 0.6940003
#95%CI: [0.2687186, 0.6749176]
#Coming soon
```

`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_regression` and `aat_standardregression` fit regression models to participants' reaction times and extract a term that serves as AAT score. `aat_regression` extracts the raw coefficient, equivalent to a mean difference score. `aat_standardregression` extracts the t-score of the coefficient, standardized on the basis of the variability of the participant's reaction times. These algorithms can be used to regress nuisance variables out of the data before computing AAT scores. When using these functions, 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_singlemeandiff` and `aat_singlemediandiff` subtract the mean or median approach reaction time from the mean or median avoidance reaction time. These algorithms are only sensible if the supplied data contain a single stimulus category.

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

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

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

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

```
aat_singlemeandiff(ds, subjvar, pullvar, rtvar, ...)

aat_singlemediandiff(ds, subjvar, pullvar, rtvar, ...)
```

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 regression formula to fit to the data to compute an AAT score
aatterm	The formula term representing the approach bias. Usually this is the interaction of the movement-direction and stimulus-category terms.

Value

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

---

q_reliability *Compute psychological experiment reliability*

Description

This function can be used to compute an exact reliability score for a psychological task whose results involve a difference score. The resulting q coefficient is equivalent to the average all possible split-half reliability scores.

Usage

```
q_reliability(ds, subjvar, formula, aatterm = NA)

## S3 method for class 'qreliability'
plot(x)
```

Arguments

ds	a long-format data.frame
subjvar	name of the subject variable
formula	a formula predicting the participant's reaction time using trial-level variables such as movement direction and stimulus category
aatterm	a string denoting the term in the formula that contains the participant's approach bias

Value

a qreliability object, containing the reliability coefficient, and a data.frame with participants' bias scores and score variance.

Author(s)

Sercan Kahveci

Examples

```
q_reliability(ds=dataset,subjvar="subjectid",formula= rt ~ is_pull * is_food, aatterm = "is_pull:is_food")
```

SpearmanBrown *Spearman-Brown corrections for Correlation Coefficients*

Description

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

Usage

```
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.

Details

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

Value

Spearman-Brown-corrected correlation coefficient.

Index