Package 'mtscr'

July 9, 2025

Type Package

Title Multidimensional Top Scoring for Creativity Research

Version 2.0.0

Description Implementation of Multidimensional Top Scoring method for creativity assessment proposed in Boris Forthmann, Maciej Karwowski, Roger E. Beaty (2023) <doi:10.1037/aca0000571>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL https://github.com/jakub-jedrusiak/mtscr

BugReports https://github.com/jakub-jedrusiak/mtscr/issues

RoxygenNote 7.3.2

Depends R (>= 4.2.0)

Imports broom.mixed, cli, dplyr (>= 1.1.0), glmmTMB, glue, lifecycle, methods, purrr, readr, rlang, stats, stringr, tibble, tidyr

Suggests shiny, covr, datamods, DT, roxygen2, shinyWidgets, testthat (>= 3.0.0), withr, writexl

Config/testthat/edition 3

NeedsCompilation no

Author Jakub Jędrusiak [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-6481-8210>, affiliation: University of Wrocław),
Boris Forthmann [aut, rev] (ORCID: <https://orcid.org/0000-0001-9755-7304>, affiliation: University of Münster),
Roger E. Beaty [aut] (ORCID: <https://orcid.org/0000-0001-6114-5973>, affiliation: Pennsylvania State University),
Maciej Karwowski [aut] (ORCID: <https://orcid.org/0000-0001-6974-1673>, affiliation: University of Wrocław)

Maintainer Jakub Jędrusiak <jakub.jedrusiak2@uwr.edu.pl>

Repository CRAN

Date/Publication 2025-07-09 13:30:02 UTC

Contents

`mtscr` *Create MTS model*

Description

Create MTS model for creativity analysis. Use with `summary.mtscr()` and `predict.mtscr()`.

Usage

```
mtscr(
  df,
  id_column,
  score_column,
  item_column = NULL,
  top = 1,
  ties_method = c("random", "average"),
  normalise = TRUE,
  self_ranking = NULL
)
```

Arguments

`df`	Data frame in long format.
`id_column`	Name of the column containing participants' id.
`score_column`	Name of the column containing divergent thinking scores (e.g. semantic distance).
`item_column`	Optional, name of the column containing distinct trials (e.g. names of items in AUT).
`top`	Integer or vector of integers (see examples), number of top answers to prepare indicators for. Default is 1, i.e. only the top answer.
`ties_method`	Character string specifying how ties are treated when ordering. Can be `"average"` (better for continuous scores like semantic distance) or `"random"` (default, better for ratings). See `rank()` for details.
`normalise`	Logical, should the creativity score be normalised? Default is `TRUE` and it's recommended to leave it as such.

`self_ranking` Name of the column containing answers' self-ranking. Provide if model should be based on top answers self-chosen by the participant. Every item should have its own ranks. The top answers should have a value of 1, and the other answers should have a value of 0. In that case, the `top` argument doesn't change anything and should be left as `top = 1`. `ties_method` is not used if `self_ranking` was provided. See mtscr_self_rank for example.

Value

The return value depends on length of the `top` argument. If `top` is a single integer, a `mtscr` model is returned. If `top` is a vector of integers, a `mtscr_list` object is returned, with names corresponding to the `top` values, e.g. `top1`, `top2`, etc.

See Also

- `summary.mtscr()` for the fit measures of the model.
- `predict.mtscr()` for getting the scores.

Examples

```
data("mtscr_creativity", package = "mtscr")

mtscr_creativity <- mtscr_creativity |>
  dplyr::slice_sample(n = 500) # for performance, ignore

# single model for top 1 answer
mtscr(mtscr_creativity, id, SemDis_MEAN, item) |>
  summary()

# three models for top 1, 2, and 3 answers
fit3 <- mtscr(
  mtscr_creativity,
  id,
  SemDis_MEAN,
  item,
  top = 1:3,
  ties_method = "average"
)

# add the scores to the database
predict(fit3)

# get the socres only
predict(fit3, minimal = TRUE)
```

`mtscr_app` *Shiny GUI for mtscr*

Description

Shiny app used as graphical interface for mtscr. Simply invoke `mtscr_app()` to run.

Usage

```
mtscr_app()
```

Details

To use the GUI you need to have the following packages installed: `DT`, `broom.mixed`, `datamods`, `writexl`.

First thing you see after running the app is `datamods` window for importing your data. You can use the data already loaded in your environment or any other option. Then you'll see four dropdown lists used to choose arguments for the functions. Consult the documentation for more details (execute `?mtscr` in the console). When the parameters are chosen, click "Generate model" button. After a while (up to a dozen or so seconds) models' parameters and are shown along with a scored dataframe.

You can download your data as a .csv or an .xlsx file using buttons in the sidebar. You can either download the scores only (i.e. the dataframe you see displayed) or your whole data with scores columns added.

For testing purposes, you may use `mtscr_creativity` dataframe. In the importing window change "Global Environment" to "mtscr" and our dataframe should appear in the upper dropdown list. Use `id` for the ID column, `item` for the item column and `SemDis_MEAN` for the score column.

Value

Runs the app. No explicit return value.

See Also

`mtscr()` for more information on the arguments.

mtscr_creativity for more information about the example dataset.

Forthmann, B., Karwowski, M., & Beaty, R. E. (2023). Don't throw the "bad" ideas away! Multi-dimensional top scoring increases reliability of divergent thinking tasks. Psychology of Aesthetics, Creativity, and the Arts. doi:10.1037/aca0000571

Examples

```
if(interactive()){
mtscr_app()
}
```

mtscr_creativity *Creativity assessment through semantic distance dataset*

Description

A dataset from Forthmann, Karwowski & Beaty (2023) paper. It contains a set of responses in Alternative Uses Task for different items with their semantic distance assessment.

Usage

```
mtscr_creativity
```

Format

`mtscr_creativity`**:**
A `tibble` with 4585 rows and 10 columns:

id patricipant's unique identification number
response response in AUT
item item for which alternative uses were searched for
SemDis_MEAN mean semantic distance

Value

a tibble

Source

https://osf.io/7rgsp/

References

doi:10.1037/aca0000571

mtscr_self_rank *Self-chosen best answers*

Description

An example dataset with best answers self-chosen by the participant. Use with `self_ranking` argument in mtscr().

Usage

```
mtscr_self_rank
```

Format

`mtscr_self_rank`**:**
A tibble with 3225 rows and 4 columns:

subject patricipant's unique identification number

task divergent thinking task number

avr average judges' raiting

top_two indicator of self-chosen two best answers; 1 if chosen, 0 if not

Source

https://osf.io/7rgsp/

References

doi:10.1037/aca0000571

`predict.mtscr` *Extract scores from mtscr model*

Description

Extract the scores from a model fitted with `mtscr()`.

Usage

```
## S3 method for class 'mtscr'
predict(object, ..., minimal = FALSE, id_col = TRUE)

## S3 method for class 'mtscr_list'
predict(object, ..., minimal = FALSE, id_col = TRUE)
```

Arguments

`object`	A model or a model list fitted with `mtscr()`.
`...`	Additional arguments. Currently not used.
`minimal`	If `TRUE`, returns only the person-level scores without the original data.
`id_col`	If `TRUE`, returns the id column in the result. If `FALSE`, the id column is not returned. Only has an impact when `minimal = TRUE`.

Value

The return value is always a tibble but its content depends mainly on the `minimal` argument:

- If `minimal = FALSE` (default), the original data frame is returned with the creativity scores columns added.
- If `minimal = TRUE`, only the creativity scores are returned (i.e., one row per person).

Functions

- `predict(mtscr_list)`: Extract scores from a model list fitted with `mtscr()`.

Examples

```
data("mtscr_creativity", package = "mtscr")

mtscr_creativity <- mtscr_creativity |>
  dplyr::slice_sample(n = 500) # for performance, ignore

fit <- mtscr(mtscr_creativity, id, SemDis_MEAN, item, top = 1:3)

# for a single model from a list
predict(fit$top1)

# for a whole list of models
predict(fit)

# person-level scores only
predict(fit, minimal = TRUE)

# you can also achieve more classic predict() behaviour
mtscr_creativity$score <- predict(fit, id_col = FALSE)

mtscr_creativity |>
  tidyr::unnest_wider(score, names_sep = "_") # Use to expand list-col
```

`summary.mtscr` *Fit measures for mtscr model*

Description

Summarise the overall fit of a single model fitted with `mtscr()`.

Usage

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

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

Arguments

`object`	mtscr model or a mtscr_list object.
`...`	Additional arguments. Currently not used.

Value

A tibble with the following columns:

model The model number (only if a list of models is provided)

nobs Number of observations

sigma The square root of the estimated residual variance

logLik The log-likelihood of the model

AIC The Akaike information criterion

BIC The Bayesian information criterion

df.residual The residual degrees of freedom

emp_rel The empirical reliability

FDI The first difference of the empirical reliability

Functions

- `summary(mtscr_list)`: Get fit measures for a list of models fitted with `mtscr()`.

Examples

```
data("mtscr_creativity", package = "mtscr")

mtscr_creativity <- mtscr_creativity |>
  dplyr::slice_sample(n = 500) # for performance, ignore

fit1 <- mtscr(mtscr_creativity, id, SemDis_MEAN, item, ties_method = "average")
fit3 <- mtscr(mtscr_creativity, id, SemDis_MEAN, item, top = 1:3, ties_method = "average")

summary(fit1)

summary(fit3)
```

Index