Package: normref (via r-universe)

November 20, 2026

Title Continuous Norming

Version 1.0.0

Description A toolbox for continuous norming of psychological and educational tests, supporting regression-based norming where norms can vary as a continuous function of age or another norm predictor. Norms are estimated using Generalized Additive Models for Location, Scale, and Shape (GAMLSS), enabling flexible modelling of the full score distribution in a normative sample. The package supports applications in psychometrics and psychological testing, and includes functions for model selection, reliability estimation, and norm calculation, including confidence intervals, and normed score precision evaluation and visualization. For more details, see Timmerman et al. (2021) <doi:10.1037/met0000348>.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5.0)

Imports dplyr, foreach, gamlss, gamlss.dist, ggplot2, openxlsx2, rlang, Rdpack, stats, utils, withr, scales, lpSolve, doParallel

RdMacros Rdpack

Suggests gamlss.tr, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Klazien de Vries [aut] (ORCID: <https://orcid.org/0009-0007-9302-1562>), Hannah Heister [aut] (ORCID: <https://orcid.org/0009-0001-1512-5549>), Julian Urban [aut] (ORCID: <https://orcid.org/0000-0001-8886-4724>), Lieke Voncken [ctb] (ORCID: <https://orcid.org/0000-0002-6710-271X>), Casper Albers [ctb] (ORCID: <https://orcid.org/0000-0002-9213-6743>), Marieke Timmerman

[aut, cre] (ORCID: <https://orcid.org/0000-0003-3480-5918>),
Dutch Research Council, NWO [fnd] (GrantID:
10.61686/YHUOB23106)

Maintainer Marieke Timmerman <m.e.timmerman@rug.nl>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-20 14:30:29 UTC

RemoteUrl https://github.com/cran/normref

RemoteRef HEAD

RemoteSha 892da85acb1beb01d61872bbfa33884c3a7724ae

Contents

boot_centiles 2
boot_composite 4
boot_heatmap 7
boot_sample 9
boot_subtest 11
centiles_bin 14
composite_shape 17
cotapp_data 18
different_rel 18
fb_select 20
ids_data 22
ids_kn_data 23
ids_rel_data 23
normtable_create 24
plot_drel 27
plot_normtable 28
reliability_window 30
sample_size_poly 31
shape_data 33

Index **36**

`boot_centiles` *Plot bootstrap standard errors across continuous predictor with reference thresholds*

Description

Visualizes bootstrap-based standard errors (SE) across a continuous predictor (e.g., age, grade) for selected centiles, and compares them to reference standard errors derived under a traditional norming framework.

Usage

```
boot_centiles(
  boot,
  centiles = c(5, 25, 50, 75, 95),
  age_range = NULL,
  benchmark_n = c(200, 300, 400),
  interpolate = TRUE
)
```

Arguments

boot	Object of class "normref_bootstrap".
centiles	Numeric vector of centiles (0–100 scale) to evaluate (e.g., c(5, 10, 50)).
age_range	Optional numeric vector of length 2 specifying the range of the independent variable to include.
benchmark_n	Numeric vector of sample sizes used to compute benchmark standard errors.
interpolate	Logical. If TRUE (default), a two-step interpolation is performed.

Value

A ggplot2 object.

See Also

boot_subtest(), boot_heatmap()

Examples

```
# 1. Prepare example data from normref
invisible(data("ids_data", package = "normref"))
mydata.BB <- shape_data(
  ids_data,
  age_name = "age",
  score_name = "y7",
  family = "BB")

# 2. Fit a GAMLSS model
mod <- fb_select(
  data = mydata.BB,
  age_name = "age",
  score_name = "shaped_score",
  family = "BB",
  selcrit = "BIC")

# 3. Run nonparametric bootstrap on subtest norms
boot_res <- boot_subtest(
  model = mod,
  age_name = "age",
  normtype = "IQ",
```

```
  boot_type = "nonparametric",
  B = 10, # 1000
  min_age = 5,
  max_age = 20,
  seed = 42,
  parallel = FALSE # TRUE
)

# 4. Standard error along centile trajectories
boot_centiles(boot_res, centiles = c(5, 50, 95), benchmark_n = c(200, 400))
```

`boot_composite` *Bootstrap uncertainty for composite norm tables*

Description

Computes bootstrap estimates, standard errors (SE), Monte Carlo standard errors of the SE, and percentile confidence intervals for composite norm tables and sample observations derived from multiple subtests.

Usage

```
boot_composite(
  subtest_models,
  composite_model,
  age_name = "age",
  composite_score_name = NULL,
  subtest_score_names = NULL,
  normtype = "Z",
  weights = NULL,
  min_age = NULL,
  max_age = NULL,
  step_size_age = NULL,
  min_score = NULL,
  max_score = NULL,
  step_size_score = 1,
  cont_cor = TRUE,
  ci_level = 0.95,
  B = 500,
  reselect = FALSE,
  seed = NULL,
  parallel = FALSE,
  ncores = NULL,
  trim = 3,
  keep_boot = FALSE,
  fb_control = NULL
)
```

Arguments

subtest_models	Named list of fitted gamlss objects for each subtest. The data used for model fitting should be available in the global environment.
composite_model	Fitted gamlss object for the composite score.
age_name	Character. Name of the continuous independent variable (e.g. "age", "grade"). Default is "age".
composite_score_name	Optional character. Label for the composite score variable in output tables. If NULL (default), "Composite Score" is used.
subtest_score_names	Optional named list or character vector of labels for each subtest score. Defaults to "<subtestname>_testscore".
normtype	Character. Norm score type: "Z" (N(0,1); default), "T" (N(50,10)), "IQ" (N(100,15)), or "SS" (N(10,3)).
weights	Optional numeric weights for the subtests.
min_age, max_age	Numeric. Bounds for the independent variable grid.
step_size_age	Numeric. Increment for the continuous variable grid.
min_score, max_score	Numeric. Score grid bounds for composite score.
step_size_score	Numeric. Increment for composite score grid.
cont_cor	Logical. Apply continuity correction to subtests.
ci_level	Numeric. Nominal confidence level for bias-corrected percentile CIs (default 0.95).
B	Integer. Number of bootstrap replicates (default 500).
reselect	Logical or named list controlling model reselection per subtest / composite.
seed	Optional integer seed for reproducibility.
parallel	Logical. If TRUE, runs replicates in parallel.
ncores	Integer. Number of worker cores when parallel = TRUE.
trim	Numeric or named list. Truncation threshold for extreme scores (in Z units). If numeric (default 3), subtests remain untrimmed (Inf) during composite score calculation to avoid artificial tail truncation artifacts, and only the final composite norms are trimmed at trim. Alternatively, supply a named list specifying thresholds per subtest and for "composite".
keep_boot	Logical. Whether to retain raw bootstrap replicate matrices.
fb_control	Named list of control arguments for fb_select(), to be used when reselect = TRUE.

Details

`boot_composite()` operates directly on complete observations across all subtests. Fitting data and response variables are extracted directly from the model data and `model$mu.formula` of each subtest model in `subtest_models` and `composite_model`.

Important: All subtest models in `subtest_models` must have been fitted on the exact same sample of complete observations with identical row counts and row ordering in their model data.

Value

An object of class `"normref_bootstrap"` containing:

normtable Tidy data.frame of normtable-level bootstrap statistics.

sample Data.frame of observation-level bootstrap statistics with all subtest and composite scores.

mcse_normtable Monte Carlo standard error metrics for the norm table.

mcse_sample Monte Carlo standard error metrics for the sample.

B, trueB Requested and successfully converged replicate counts.

model_freq_table Model selection frequencies across replicates.

model_selection_history Model selection history per subtest and composite.

meta Metadata including weights, correlation matrix, models, data, and normtype.

grid Evaluation grid for the continuous variable and composite scores.

boot Optional raw bootstrap results (if `keep_boot = TRUE`).

See Also

`boot_subtest()`, `boot_heatmap()`, `boot_centiles()`, `boot_sample()`

Examples

```
# 1. Prepare example data from normref
invisible(data("ids_data", package = "normref"))

# 2. Fit subtest models
mydata1 <- shape_data(ids_data, age_name = "age", score_name = "y7",  family = "BB")
mod1 <- fb_select(mydata1, age_name = "age", score_name = "shaped_score", family = "BB")
mydata2 <- shape_data(ids_data, age_name = "age", score_name = "y14", family = "NO")
mod2 <- fb_select(mydata2, age_name = "age", score_name = "shaped_score", family = "NO")

normtable1 <- normtable_create(mod1, mydata1, age_name = "age", score_name = "shaped_score",
                trim = Inf)
normtable2 <- normtable_create(mod2, mydata2, age_name = "age", score_name = "shaped_score",
                trim = Inf)
# 3. Construct composite score
comp_data <- composite_shape(list(y7 = normtable1, y14 = normtable2))
comp_model <- fb_select(comp_data, age_name = "age", score_name = "z_sum", family = "NO")

# 4. Bootstrap composite score precision
boot_comp <- boot_composite(
```

```
  subtest_models = list(y7 = mod1, y14 = mod2),
  composite_model = comp_model,
  age_name = "age",
  composite_score_name = "z_sum",
  step_size_score = 0.1,
  B = 10, # 1000
  seed = 42,
  parallel = FALSE #TRUE
)

head(boot_comp$sample)

# Observed sample-level standard error ratios by score bin
boot_sample(
  boot_comp,
  type = "ratio",
  benchmark_n = 300,
  z_bins = seq(-3, 3, by = 1)
  )
```

---

boot_heatmap *Plot heatmap of bootstrap norm table uncertainty*

Description

Visualizes bootstrap-based uncertainty from normtables as a heatmap over the independent variable and score. Regions trimmed or set to NA appear blank.

Usage

```
boot_heatmap(
  boot,
  type = "se",
  benchmark_n = NULL,
  centiles = NULL,
  add_points = FALSE,
  point_style = "hollow",
  add_contour = FALSE,
  age_range = NULL,
  score_range = NULL,
  limits = NULL,
  centile_label_nudge = 0.5,
  print_pct = FALSE
)
```

Arguments

boot	Object of class "normref_bootstrap".
type	Character. Either "se", "difference", or "ratio". • "se": Displays raw bootstrap standard errors. • "difference": Displays the difference between the bootstrap SE and the reference SE ($SE_{\text{boot}} - SE_{\text{ref}}$). • "ratio": Displays the relative ratio of the bootstrap SE to the reference SE ($SE_{\text{boot}}/SE_{\text{ref}}$).
benchmark_n	Numeric. Reference sample size used to compute benchmark standard errors (required when type = "difference" or type = "ratio").
centiles	Optional numeric vector of centiles (0–100 scale) to overlay.
add_points	Logical. If TRUE, overlay raw sample data points.
point_style	Character. Style for points: "hollow" (default) or "solid".
add_contour	Logical. If TRUE and type == "difference" or type == "ratio", adds a reference contour line (break at 0 for difference, 1 for ratio).
age_range	Numeric vector of length 2 specifying bounds for the continuous variable.
score_range	Numeric vector of length 2 specifying score limits for plotting.
limits	Optional numeric vector of length 2 for color scale limits.
centile_label_nudge	Numeric. Horizontal offset for centile labels.
print_pct	Logical. If TRUE and type == "difference" or type == "ratio", prints the percentage of table cells where the bootstrap SE exceeds the reference SE. Default is FALSE.

Value

A ggplot2 object.

See Also

boot_subtest(), boot_centiles()

Examples

```
# 1. Prepare example data from normref
invisible(data("ids_data", package = "normref"))
mydata.BB <- shape_data(
  ids_data,
  age_name = "age",
  score_name = "y7",
  family = "BB")

# 2. Fit a GAMLSS model
mod <- fb_select(
  data = mydata.BB,
  age_name = "age",
```

```
  score_name = "shaped_score",
  family = "BB",
  selcrit = "BIC")

# 3. Run nonparametric bootstrap on subtest norms
boot_res <- boot_subtest(
  model = mod,
  age_name = "age",
  normtype = "IQ",
  boot_type = "nonparametric",
  B = 10, #1000
  min_age = 5,
  max_age = 20,
  seed = 42,
  parallel = FALSE # TRUE
)

# 4. Standard error heatmap across age and score
boot_heatmap(
  boot_res,
  type = "se",
  add_points = TRUE,
  centiles = c(1, 5, 25, 50, 75, 95, 99),
  centile_label_nudge = 2.0
)

# 5. Benchmark difference heatmap against discrete norming (N = 300 per group)
boot_heatmap(
  boot_res,
  type = "difference",
  benchmark_n = 300,
  add_points = TRUE,
  add_contour = TRUE,
  centiles = c(5,25,50,75,95)
)
```

`boot_sample` *Plot sample-level bootstrap precision across age, faceted by score bins*

Description

Visualizes person-level bootstrap standard errors, differences, or relative ratios against a continuous predictor (such as age), faceted into standard-score bins (e.g., z-score bins from -3 to +3). Individual points represent observed sample cases.

Usage

```
boot_sample(
  boot,
  benchmark_n = 300,
  type = c("ratio", "difference", "se"),
  z_bins = seq(-3, 3, by = 1),
  age_range = NULL,
  smooth = FALSE,
  point_alpha = 0.5
)
```

Arguments

boot	Object of class "normref_bootstrap".
benchmark_n	Numeric. Reference sample size used to compute benchmark standard errors (default = 300).
type	Character. One of "ratio" (default), "difference", or "se". • "ratio": Relative ratio $\widehat{SE}_{\text{boot}}/SE_{\text{ref}}$. • "difference": Difference $\widehat{SE}_{\text{boot}} - SE_{\text{ref}}$. • "se": Raw bootstrap standard error $\widehat{SE}_{\text{boot}}$.
z_bins	Numeric vector specifying bin edges on the z-score scale (default: seq(-3, 3, by = 1)).
age_range	Optional numeric vector of length 2 specifying bounds for the continuous age variable.
smooth	Logical. If TRUE, adds a LOESS smoother per facet (default = FALSE).
point_alpha	Numeric. Alpha transparency for individual points (default = 0.5).

Value

A ggplot2 object.

See Also

boot_centiles(), boot_heatmap(), boot_composite()

Examples

```
# 1. Prepare example data from normref
invisible(data("ids_data", package = "normref"))

# 2. Fit subtest models
mydata1 <- shape_data(ids_data, age_name = "age", score_name = "y7",  family = "BB")
mod1 <- fb_select(mydata1, age_name = "age", score_name = "shaped_score", family = "BB")
mydata2 <- shape_data(ids_data, age_name = "age", score_name = "y14", family = "NO")
mod2 <- fb_select(mydata2, age_name = "age", score_name = "shaped_score", family = "NO")

# 3. Construct composite score
```

```
normtable1 <- normtable_create(mod1, mydata1, age_name = "age", score_name = "shaped_score",
                trim = Inf)
normtable2 <- normtable_create(mod2, mydata2, age_name = "age", score_name = "shaped_score",
                trim = Inf)
comp_data <- composite_shape(list(y7 = normtable1, y14 = normtable2))
comp_model <- fb_select(comp_data, age_name = "age", score_name = "z_sum", family = "NO")

# 4. Bootstrap composite score precision
boot_comp <- boot_composite(
  subtest_models = list(y7 = mod1, y14 = mod2),
  composite_model = comp_model,
  age_name = "age",
  composite_score_name = "z_sum",
  step_size_score = 0.1,
  B = 10, # 1000
  seed = 42,
  parallel = FALSE #TRUE
)

head(boot_comp$sample)

# 5. Observed sample-level standard error ratios by score bin
boot_sample(
  boot_comp,
  type = "ratio",
  benchmark_n = 400,
  z_bins = seq(-3, 3, by = 1)
  )
```

`boot_subtest` *Bootstrap uncertainty for single GAMLSS norm tables*

Description

Computes bootstrap estimates, standard errors (SE), Monte Carlo standard errors of the SE, and percentile confidence intervals for continuous norm tables and sample observations fitted with `gamlss`. Supports both nonparametric resampling and parametric simulation for sample size planning.

Usage

```
boot_subtest(
  model,
  age_name = "age",
  score_name = NULL,
  boot_type = c("nonparametric", "parametric"),
  n_sim = NULL,
  age_sampling_frame = NULL,
```

```
  normtype = "Z",
  min_age = NULL,
  max_age = NULL,
  min_score = NULL,
  max_score = NULL,
  step_size_score = 1,
  step_size_age = NULL,
  cont_cor = TRUE,
  ci_level = 0.95,
  B = 500,
  seed = NULL,
  parallel = FALSE,
  ncores = NULL,
  trim = 3,
  keep_boot = FALSE,
  reselect = FALSE,
  fb_control = list()
)
```

Arguments

model	A fitted gamlss object (e.g. from fb_select()). The data used for model fitting should be available in the global environment.
age_name	Character. Column name of the continuous independent variable (e.g., "age", "grade"). Default is "age".
score_name	Optional character. Label for the score variable in output tables and plots. If NULL (default), "Subtest Score" is used.
boot_type	Character. Either "nonparametric" (default; resamples empirical observations) or "parametric" (simulates new raw scores from the fitted GAMLSS model).
n_sim	Optional integer. Total sample size to simulate when boot_type = "parametric". Defaults to the original sample size.
age_sampling_frame	Optional function or numeric vector defining the continuous age distribution when boot_type = "parametric".
normtype	Character. One of "Z", "T", or "IQ". Default is "Z".
min_age, max_age	Numeric. Grid bounds for the continuous independent variable. Defaults to the observed range in model data.
min_score, max_score	Numeric. Grid bounds for the score variable. Defaults to the observed range.
step_size_score	Numeric. Increment for the score grid (default 1).
step_size_age	Numeric. Increment for the independent variable grid. Defaults to approximately 100 points.
cont_cor	Logical. Whether to apply continuity correction for discrete families.

ci_level	Numeric. Nominal confidence level for bias-corrected percentile CIs (default 0.95).
B	Integer. Number of bootstrap replicates (default 500).
seed	Optional integer seed for reproducibility.
parallel	Logical. If TRUE, runs replicates in parallel.
ncores	Integer. Number of worker cores when parallel = TRUE.
trim	Numeric. Truncation threshold for extreme scores in Z scale (default 3).
keep_boot	Logical. Whether to retain raw bootstrap replications.
reselect	Logical. Whether to re-select polynomial degrees per replicate.
fb_control	Named list of options passed to fb_select(), to be used when reselect = TRUE.

Value

An object of class "normref_bootstrap" containing:

normtable Tidy data.frame of normtable-level bootstrap statistics.

sample Data.frame of observation-level bootstrap statistics with all subtest scores.

mcse_normtable Monte Carlo standard error metrics for the norm table.

mcse_sample Monte Carlo standard error metrics for the sample.

B, trueB Requested and successfully converged replicate counts.

model_freq_table Model selection frequencies across replicates.

model_selection_history Model selection history (if reselect = TRUE).

meta Metadata including models, data, and normtype.

grid Evaluation grid for the continuous variable and composite scores.

boot Optional raw bootstrap results (if keep_boot = TRUE).

See Also

boot_composite(), boot_heatmap(), boot_centiles()

Examples

```
# # 1. Prepare example data from normref
invisible(data("ids_data", package = "normref"))
mydata.BB <- shape_data(
  ids_data,
  age_name = "age",
  score_name = "y7",
  family = "BB")

# 2. Fit a GAMLSS model
mod <- fb_select(
  data = mydata.BB,
  age_name = "age",
```

```
  score_name = "shaped_score",
  family = "BB",
  selcrit = "BIC")

# 2. Alternatively, fit a spline model:
# library(gamlss)
# mod <- gamlss(shaped_score ~ pb(age), family = "BB", data = mydata.BB, trace = FALSE)

# 3. Run nonparametric bootstrap on subtest norms
boot_res <- boot_subtest(
  model = mod,
  age_name = "age",
  normtype = "IQ",
  boot_type = "nonparametric",
  B = 10, # 1000
  min_age = 5,
  max_age = 20,
  seed = 42,
  parallel = FALSE #TRUE
)
head(boot_res$normtable[1050:1055,]) # inspect results

# 4. Parametric bootstrap for sample size planning (e.g., N = 300)
frame_uniform <- function(n) {runif(n, min = 5, max = 20)} # define sampling frame

boot_plan <- boot_subtest(
  model = mod,
  age_name = "age",
  score_name = "y7",
  boot_type = "parametric",
  n_sim = 300,
  age_sampling_frame = frame_uniform,
  normtype = "Z",
  min_age = 5,
  max_age = 20,
  B = 10, # 1000,
  parallel = FALSE, #TRUE,
  seed = 123
)

boot_centiles(
boot_plan,
centiles = c(5,50,95)
)
```

---

centiles_bin *Plot centiles of a fitted GAMLSS model (binomial-type)*

Description

centiles_bin() plots centile curves and the sample data for binomial-type distributions (see gamlss::.gamlss.bi.list) based on a fitted GAMLSS object.

Usage

```
centiles_bin(
  model,
  xvar,
  cent = c(0.4, 2, 10, 25, 50, 75, 90, 98, 99.6),
  legend = TRUE,
  ylab = "y",
  xlab = "x",
  main = NULL,
  main.gsub = "@",
  xleg = min(xvar),
  yleg = max(model$y),
  xlim = range(xvar),
  ylim = range(model$y),
  save = FALSE,
  plot = TRUE,
  points = TRUE,
  pch = 15,
  cex = 0.5,
  col = "grey",
  col.centiles = seq_along(cent) + 2,
  lty.centiles = 1,
  lwd.centiles = 1,
  colors = "rainbow",
  ...
)
```

Arguments

model	a GAMLSS fitted model, for example the result of fb_select().
xvar	the unique explanatory variable
cent	a vector with elements the % centile values for which the centile curves have to be evaluated
legend	whether a legend is required in the plot or not, the default is legent=TRUE
ylab	the y-variable label
xlab	the x-variable label
main	the main title here as character. If NULL the default title "centile curves using NO" (or the relevant distributions name) is shown
main.gsub	if the main.gsub (with default "@") appears in the main title then it is substituted with the default title.
xleg	position of the legend in the x-axis

yleg	position of the legend in the y-axis
xlim	the limits of the x-axis
ylim	the limits of the y-axis
save	whether to save the sample percentages or not with default equal to FALSE. In this case the sample percentages are printed but are not saved
plot	whether to plot the centiles. This option is useful for centile.split
points	whether the data points should be plotted, default is TRUE for centiles() and FALSE for centiles.fan()
pch	the character to be used as the default in plotting points see par
cex	size of character see par
col	plotting colour see par
col.centiles	Plotting colours for the centile curves
lty.centiles	line type for the centile curves
lwd.centiles	The line width for the centile curves
colors	the different colour schemes to be used for the fan-chart. The following are available c("cm","gray", "rainbow", "heat", "terrain", "topo"),
...	for extra arguments

Value

No return value, only graphical output.

See Also

fb_select()

Examples

```
data("ids_data")

mydata_BB_y14 <- shape_data(
  data       = ids_data,
  age_name   = "age",
  score_name = "y14",
  family     = "BB"
)

mod_BB_y14 <- fb_select(
  data       = mydata_BB_y14,
  age_name   = "age",
  score_name = "shaped_score",
  family     = "BB",
  selcrit    = "BIC"
)

centiles_bin(mod_BB_y14, xvar = age)
```

composite_shape *Shape data for a composite scale based on normalized Z-scores*

Description

composite_shape() creates a data.frame with age values and a weighted sum of normalized z-scores from multiple NormTable objects, suitable for use as input to fb_select().

Usage

```
composite_shape(normtables, weights = NULL)
```

Arguments

normtables	list of NormTable objects created by normtable_create(). Each must contain znorm_sample and norm_sample.
weights	Optional numeric vector of weights for a weighted composite score. If NULL (default), the sum of subtest z-scores is used.

Details

The function combines normalized z-scores across multiple norm tables into a composite score. By default, scores are equally weighted. Optional weights allow differential contribution of subtests to the composite.

Value

A data.frame with:

age	Age values from the first NormTable
z_sum	Weighted sum of normalized z-scores across all objects

See Also

shape_data(), fb_select(), normtable_create()

Examples

```
invisible(data("ids_data"))

# Example with two normtables
mydata1 <- shape_data(ids_data, age_name = "age", score_name = "y7", family = "BCPE")
mod1 <- fb_select(mydata1, age_name = "age", score_name = "shaped_score",
                  family = "BCPE", selcrit = "BIC")
norm1 <- normtable_create(mod1, mydata1, age_name = "age", score_name = "shaped_score")

mydata2 <- shape_data(ids_data, age_name = "age", score_name = "y14", family = "BCPE")
mod2 <- fb_select(mydata2, age_name = "age", score_name = "shaped_score",
```

```
                    family = "BCPE", selcrit = "BIC")
norm2 <- normtable_create(mod2, mydata2, age_name = "age", score_name = "shaped_score")

composite_data <- composite_shape(list(norm1, norm2))
```

`cotapp_data` *cotapp data*

Description

The data are perturbed variants of the scores on the raw speed of block 1 (`rt`) and the raw number of errors of block 7 (`error`) of the normative sample of the cotapp test (Rommelse et al., 2018). More information on the cotapp test: https://www.boom.nl/productgroep/101-45_COTAPP

Usage

```
data(cotapp_data)
```

Format

A dataframe with three columns:

`age` age in years

`rt` reaction time: scores on the raw speed of block 1

`error` number of errors of block 7

References

Rommelse N, Brinkman A, Slaats-Willemse D, Timmerman ME, Voncken L, de Zeeuw P, Luman M, Hartman C (2020). "De Cognitieve Test Applicatie (COTAPP): geavanceerde computertest voor het meten van aandacht, informatieverwerking en executieve functies bij kinderen." *Kind en Adolescent*, **41**, 50–80.

`different_rel` *Estimate reliability across multiple window widths and age steps*

Description

Estimates reliability curves across various combinations of window widths and age step sizes, with optional per-individual estimation.

Usage

```
different_rel(
  data,
  item_variables,
  age_name,
  step_window,
  min_agegroup = NULL,
  max_agegroup = NULL,
  step_agegroup,
  include_window_per_person = FALSE,
  complete.obs = TRUE
)
```

Arguments

`data`	data.frame containing item scores and age variable.
`item_variables`	character vector. Names of the columns with item scores.
`age_name`	string. Name of the age variable. Default is `"age_years"`.
`step_window`	numeric vector. Window widths to evaluate.
`min_agegroup`	numeric. Minimum age to include. Defaults to the floor of the minimum age in the data.
`max_agegroup`	numeric. Maximum age to include. Defaults to the ceiling of the maximum age in the data.
`step_agegroup`	numeric vector. Step sizes between evaluated age points.
`include_window_per_person`	logical. If `TRUE`, also estimates reliability for each individual. Default is `FALSE`.
`complete.obs`	logical. If `TRUE` (default), uses listwise deletion; if `FALSE`, uses pairwise deletion.

Value

An object of class `Drel` (a data.frame) with:

- `rel`: Reliability estimates
- `age`: Corresponding evaluated ages
- `window_width`: Width of the window used
- `age_group_width`: Step size between evaluated age groups
- `version`: Type of estimation (`"step"` or `"window_per_person"`)

See Also

`plot_drel()`

Examples

```
invisible(data("ids_kn_data"))
rel_int <- different_rel(
  data = ids_kn_data,
  item_variables = colnames(ids_kn_data),
  age_name = "age_years",
  step_window = c(0.5, 1, 2, 5, 10, 20),
  min_agegroup = 5,
  max_agegroup = 20,
  step_agegroup = c(0.5, 1, 1.5, 2)
)
```

---

fb_select *Free order model selection procedure*

Description

`fb_select()` applies the free order model selection procedure, using forward–backward selection (Voncken et al. 2019). For a given GAMLSS distribution and model selection criterion, it selects the optimal polynomial degrees for all distribution parameters.

Usage

```
fb_select(
  data,
  age_name,
  score_name,
  family,
  selcrit = "BIC",
  spline = FALSE,
  method = "RS(10000)",
  max_poly = c(5, 5, 2, 2),
  min_poly = c(0, 0, 0, 0),
  start_poly = c(2, 1, 0, 0),
  start.from = NULL,
  trace = TRUE,
  seed = 123
)
```

Arguments

`data`	data.frame. Sample on which to fit the distribution; contains the scores and ages.
`age_name`	string. Name of the age variable.
`score_name`	string. Name of the score variable.

family	string. For example, "BB", "BCPE", "NO", etc. See gamlss.dist::gamlss.family for more information on available distributions. Also accepts user-defined distributions like truncated distributions.
selcrit	string. Model selection criterion: "AIC", "BIC" (default), "GAIC(3)", or "CV" (cross-validated global deviance with 10 folds).
spline	logical. If FALSE (default), estimate polynomial(s) for μ; if TRUE, estimate a p-spline for μ.
method	string. Estimation method for gamlss::gamlss(). Either "RS()", "CG()", or "mixed()", with iteration count. Default is "RS(10000)".
max_poly	vector. Maximum polynomial degrees for each parameter.
min_poly	vector. Minimum polynomial degrees for each parameter.
start_poly	vector. Starting polynomial degrees for each parameter.
start.from	gamlss object (optional). A pre-fitted GAMLSS model to serve as the initial warm start for fitting the starting model (start_poly) using the start.from argument in gamlss::gamlss(). Only utilized when the distribution has 3 or more parameters. Default is NULL.
trace	logical. If TRUE, prints progress during selection.
seed	integer. Random seed for cross-validation folds.

Details

The search history of the free order model selection procedure can be accessed in search_history. This allows the user to investigate whether a more parsimonious model with similar model fit to the selected model exists.

Value

A selected GAMLSS model with the chosen polynomial degrees. This is a gamlss object with additional components

- selcrit: the value of the selection criterion for the chosen model.
- search_history: search history, containing the polynomial degrees for examined models (poly_deg) and their selection criteria (selcrit).

References

Voncken L, Albers CJ, Timmerman ME (2019). "Model selection in continuous test norming with GAMLSS." *Assessment*, **26**(7), 1329–1346. doi:10.1177/1073191117715113.

See Also

shape_data(), fb_select(), normtable_create()

Examples

```
invisible(data("ids_data"))
mydata <- shape_data(ids_data, age_name = "age", score_name = "y14", family = "BB")
mod <- fb_select(mydata, age_name = "age", score_name = "shaped_score",
                 family = "BB", selcrit = "BIC")
mod$search_history
```

`ids_data` *ids data*

Description

The data are perturbed data, based on scores on Test 14 ("naming antonyms") and Test 7 ("naming categories") of the intelligence test IDS-2 (Grob & Hagmann-von Arx, 2018a; Grob et al., 2018b). The data are provided as supplementary material to Timmerman et al. (2021).

Usage

```
data(ids_data)
```

Format

A dataframe with three columns:

`age` age in years

`y7` raw test score on Test 7

`y14` raw test score on Test 14

Source

https://osf.io/p75a6

References

Grob A, Hagmann-von Arx P (2018). *IDS 2: Intelligence and Development Scales-2*. Hogrefe.

Grob A, Hagmann-von Arx P, Ruiter S, Timmerman ME, Visser L (2018). *IDS-2: Intelligentie-en ontwikkelingsschalen voor kinderen en jongeren*. Hogrefe Publishing.

Timmerman ME, Voncken L, Albers CJ (2021). "A tutorial on regression-based norming of psychological tests with GAMLSS." *Psychological methods*, **26**(3), 357. doi:10.1037/met0000348.

ids_kn_data *The ids_kn_data are simulated data for demonstration purposes*

Description

The data are simulated data for demonstration purposes, akin to Test 7 ("naming categories") of the intelligence test IDS-2 (Grob & Hagmann-von Arx, 2018). It consists of the binary scores on 34 items (KN_1,...,KN_34). The raw test score is the sum of the 34 item scores. The data are provided as supplementary material to Heister et al. (2024).

Usage

```
data(ids_kn_data)
```

Format

A dataframe with 36 columns:

`KN_1` binary score on item 1

`KN_2` binary score on item 2 ...

`KN_34` binary score on item 34

`rawscore` raw test score as the unweighted sum of the scores on item 1 to item 34

`age_years` age in year

Source

https://osf.io/dc5k9/files/osfstorage

References

Grob A, Hagmann-von Arx P (2018). *IDS 2: Intelligence and Development Scales-2*. Hogrefe.

Heister HM, Albers CJ, Wiberg M, Timmerman ME (2024). "Item response theory-based continuous test norming." *Psychological methods*. doi:10.1037/met0000686.

ids_rel_data *These fictional reliability data are for demonstration purposes.*

Description

Dataframe with the vectors age and rel, with the ages evaluated, and rel the (fictional) test reliability per age.

Usage

```
data(ids_rel_data)
```

Format

A dataframe with two columns:

`age` age in years

`rel` reliability

Source

constructed by authors

`normtable_create` *Create a norm table based on a GAMLSS fitted model*

Description

`normtable_create()` creates a norm table based on a fitted `GAMLSS` model.

Usage

```
normtable_create(
  model,
  data = NULL,
  age_name,
  score_name,
  datarel = NULL,
  normtype = "Z",
  min_age = NULL,
  max_age = NULL,
  min_score = NULL,
  max_score = NULL,
  step_size_score = 1,
  step_size_age = NULL,
  cont_cor = TRUE,
  ci_level = 0.95,
  trim = 3,
  excel = FALSE,
  excel_name = tempfile("norms", fileext = ".xlsx"),
  new_data = NULL,
  newdata = NULL,
  warn_extrapolation = TRUE
)
```

Arguments

`model` a GAMLSS fitted model, for example the result of `fb_select()`.

data	data.frame. The sample on which the model was fitted. If omitted, it is retrieved directly from model; must contain the score variable (with name given in score_name) and age variable (with name given in age_name).
age_name	string. Name of the age variable.
score_name	string. Name of the score variable.
datarel	data.frame or numeric. If a data.frame, must contain columns age and rel, with estimated test reliability per age. If numeric, a constant reliability is assumed for all ages (optional, only needed for confidence intervals).
normtype	string. Norm score type: "Z" (N(0,1); default), "T" (N(50,10)), "IQ" (N(100,15)), or "SS" (N(10,3)).
min_age	numeric. Lowest age value in the norm table; default is the first integer below the minimum observed age.
max_age	numeric. Highest age value in the norm table; default is the first integer above the maximum observed age.
min_score	numeric. Lowest score value in the norm table; default is the minimum observed score.
max_score	numeric. Highest score value in the norm table; default is the maximum observed score.
step_size_score	numeric. Increment of the scores in the norm table; default is 1.
step_size_age	numeric. Increment of the ages in the norm table; defaults to approximately 100 ages in total.
cont_cor	Logical. If TRUE, apply a continuity correction for discrete families. Default is TRUE.
ci_level	numeric. Confidence interval level (if datarel is provided). Default is 0.95.
trim	numeric. Trim norm scores at ± trim standard deviations. Default is 3.
excel	logical. If TRUE, attempt to write results to an Excel file. Default is FALSE.
excel_name	character. Path to the Excel file. Defaults to a temporary file. Ignored if excel = FALSE.
new_data	logical. Deprecated; kept for backward compatibility. Use newdata instead.
newdata	data.frame. Optional new data on which sample-level norm scores are evaluated. If provided, full norm matrices are omitted.
warn_extrapolation	logical. If TRUE (default), a warning is issued when values in newdata fall outside the range of the normative sample. Set to FALSE in internal resampling or bootstrap routines.

Details

If excel = TRUE, results are written to an Excel file via the **openxlsx2** package. If the package is not installed, a message is printed and the function continues without writing an Excel file. By default, the file is written to a temporary path (see tempfile()); if you want to keep the file permanently, provide your own file name via the excel_name argument (e.g., "norms.xlsx").

Value

A list of class `NormTable` containing:

- `norm_sample`: Estimated norm scores (`normtype`) in the sample, trimmed at `trim`.
- `norm_sample_lower`, `norm_sample_upper`: Lower and upper `ci_level` confidence bounds of `norm_sample`.
- `norm_matrix`: Norm scores (`normtype`) by age.
- `norm_matrix_lower`, `norm_matrix_upper`: Lower and upper `ci_level` bounds of `norm_matrix`.
- `znorm_sample`: Estimated Z scores in the sample.
- `cdf_sample`: Estimated percentiles in the sample.
- `cdf_matrix`: Percentile table by age.
- `data`, `age_name`, `score_name`: Copies of respective function arguments.
- `pop_age`: Evaluated ages in the norm table.

References

Timmerman ME, Voncken L, Albers CJ (2021). "A tutorial on regression-based norming of psychological tests with GAMLSS." *Psychological methods*, **26**(3), 357. doi:10.1037/met0000348.

See Also

`fb_select()`, `plot_normtable()`

Examples

```
# Load example data
invisible(data("ids_data"))

# Prepare data for modeling
mydata_BB_y14 <- shape_data(
  data = ids_data,
  age_name = "age",
  score_name = "y14",
  family = "BB"
)

# Fit model using BIC as selection criterion
mod_BB_y14 <- fb_select(
  data = mydata_BB_y14,
  age_name = "age",
  score_name = "shaped_score",
  family = "BB",
  selcrit = "BIC"
)

# Create norm table from fitted model
norm_mod_BB_y14 <- normtable_create(
  model = mod_BB_y14,
```

```
  data = mydata_BB_y14,
  age_name = "age",
  score_name = "shaped_score"
)

# Calculate norms for a new sample using reliability data
invisible(data("ids_rel_data"))
newdata <- ids_data[1:5, c("age", "y14")]

norm_mod_BB_newdata <- normtable_create(
  model = mod_BB_y14,
  newdata = newdata,
  age_name = "age",
  score_name = "y14",
  datarel = ids_rel_data
)
```

`plot_drel` *Plot reliability estimates over age*

Description

`plot_drel()` plots reliability estimates as a function of age, based on different window widths, using a `Drel` object.

Usage

```
plot_drel(drel, ncol = 3, nrow = 2, ...)
```

Arguments

`drel`	a `Drel` object (created with `different_rel()`).
`ncol`	number of plots per row (default: `3`).
`nrow`	number of plots per column (default: `2`).
`...`	additional arguments passed to plotting functions.

Value

graphical output and the `ggplot` object used to create it.

See Also

`different_rel()`

Examples

```
data("ids_kn_data")

rel_int <- different_rel(
  data           = ids_kn_data,
  item_variables = colnames(ids_kn_data),
  age_name       = "age_years",
  step_window    = c(0.5, 1, 2, 5, 10, 20),
  min_agegroup   = 5,
  max_agegroup   = 20,
  step_agegroup  = c(0.5, 1, 1.5, 2)
)

plot_drel(rel_int, ncol = 2)
```

---

plot_normtable *Plot norm curves from a NormTable object*

Description

`plot_normtable()` plots norm curves as a function of the predictor, along with the sample data, based on a `NormTable` object.

Usage

```
plot_normtable(
  normtable,
  lty = 1,
  lwd = 3,
  pch = 1,
  cex = 0.5,
  col = "aquamarine4",
  xlab = "Age",
  ylab = "Percentile",
  ...
)
```

Arguments

`normtable`	a `NormTable` object (created by `normtable_create()` with `new_data = FALSE`).
`lty`	line type(s) for curves.
`lwd`	line width(s) for curves.
`pch`	symbol for sample points.
`cex`	point size (default: `0.5`).

col	point colour (default: "aquamarine4").
xlab	x-axis label (default: "Age").
ylab	y-axis label (default: "Percentile").
...	additional graphical parameters passed to graphics::plot(), graphics::lines(), or graphics::points().

Value

graphical output and the ggplot object used to create it.

See Also

normtable_create()

Examples

```
data("ids_data")

mydata_BB_y14 <- shape_data(
  data       = ids_data,
  age_name   = "age",
  score_name = "y14",
  family     = "BB"
)

mod_BB_y14 <- fb_select(
  data       = mydata_BB_y14,
  age_name   = "age",
  score_name = "shaped_score",
  family     = "BB",
  selcrit    = "BIC"
)

norm_mod_BB_y14 <- normtable_create(
  model      = mod_BB_y14,
  data       = mydata_BB_y14,
  age_name   = "age",
  score_name = "shaped_score"
)

# default plot
plot_normtable(norm_mod_BB_y14)
```

`reliability_window` *Estimate test reliability by age using a sliding window*

Description

Estimates reliability across age using a sliding window approach, either at fixed age points or per individual.

Usage

```
reliability_window(
  data,
  age_name,
  item_variables,
  window_width,
  window_version = "step",
  min_agegroup = NULL,
  max_agegroup = NULL,
  step_agegroup = 1,
  complete.obs = TRUE
)
```

Arguments

`data`	data.frame containing the item scores and age variable.
`age_name`	string. Name of the age variable.
`item_variables`	numeric or character vector. Column indices or names of the item variables.
`window_width`	numeric. Width of the sliding window used to group individuals by age.
`window_version`	string. Type of windowing: • `"step"` (default): Estimate reliability at fixed age intervals. • `"window_per_person"`: Estimate reliability for each individual.
`min_agegroup`	numeric. Minimum age to include. Defaults to the floor of the minimum age in the data.
`max_agegroup`	numeric. Maximum age to include. Defaults to the ceiling of the maximum age in the data.
`step_agegroup`	numeric. Step size between evaluated ages. Used only when `window_version = "step"`.
`complete.obs`	logical. If `TRUE` (default), uses listwise deletion; if `FALSE`, uses pairwise deletion.

Value

A data.frame with:

- `rel`: Reliability estimates

- age: Corresponding age values
- window_width: The width of the sliding window
- window_per: Description of age step or observation unit

This output can be used as the datarel argument in normtable_create().

References

Heister HM, Albers CJ, Wiberg M, Timmerman ME (2024). "Item response theory-based continuous test norming." *Psychological methods*. doi:10.1037/met0000686.

See Also

normtable_create()

Examples

```
invisible(data("ids_kn_data"))
rel_est <- reliability_window(
  data = ids_kn_data,
  age_name = "age_years",
  item_variables = colnames(ids_kn_data),
  window_width = 2
)
```

---

sample_size_poly *Sample size planning for continuous norming using polynomial regression*

Description

Computes optimal sample sizes per group under a distribution free polynomial regression model for group means, following Hessen (2026).

Usage

```
sample_size_poly(
  n_groups,
  poly_degree,
  n0,
  variances = NULL,
  solution_type = c("balanced", "all", "range")
)
```

Arguments

`n_groups`	Integer. Number of groups (e.g., age groups).
`poly_degree`	Integer scalar or vector. Polynomial degree(s) of polynomial regression model.
`n0`	Integer scalar or vector. Reference sample size per group under traditional norming. Typical values are 200, 300, or 400 (see e.g., Egberink et al. (2026)).
`variances`	Optional numeric vector of length `n_groups`. Estimated variances of the means in each group. If not provided, homoscedastic variances are assumed (`rep(1, n_groups)`).
`solution_type`	Character string indicating which solution(s) to return. Must be one of: **"balanced"** (default) Returns a single optimal solution with the most balanced allocation across groups. **"range"** Returns, for each group, the range of sample sizes across all optimal solutions. **"all"** Returns all optimal solutions.

Details

In the assumed continuous norming model, group means are first estimated and then smoothed using a polynomial model that regresses group means on age. This function determines group sample sizes such that the precision of the estimated means (not higher-order moments) is at least as high as under traditional norming.

This function implements the linear programming (LP) approach described in Hessen (2026). Multiple optimal solutions may exist: different combinations of n_{min} and n_{max} can yield the same minimal total sample size while satisfying the precision constraints.

By default (`solution_type = "balanced"`), a single solution is returned, defined as the solution minimizing the difference between the largest and smallest group sample sizes. This yields a practically balanced design. Alternatively, users can inspect all optimal solutions (`solution_type = "all"`) or examine the range of optimal sample sizes per group (`solution_type = "range"`).

How to use:

- Before data collection: do not provide `variances` (assumes homoscedasticity).
- During/after data collection: provide estimated `variances` per group.

Value

A data.frame. The structure depends on `solution_type`:

solution_type = "balanced" Returns one optimal solution with:

n0 Reference sample size per group
group Group index
poly_degree Polynomial degree
n Sample size per group
variance Expected variance of estimated mean
variance_traditional Variance under traditional norming ($\sigma_j^2/n0$)
total_n Total sample size across all groups

traditional_total_n Total sample size under traditional norming

n_min Lower bound used in LP

n_max Upper bound used in LP

solution_type = "all" Returns all optimal solutions with the same columns as above, plus:

solution_id Identifier for solution number

solution_type = "range" Returns one row per group with:

n0 Reference sample size per group

group Group index

degree Polynomial degree

n_range Range of sample sizes over optimal solutions

References

Egberink I, De Leng W, Evers A, Hemker B, Lucassen W (2026). *COTAN Beoordelingssysteem voor de Kwaliteit van Tests. Geheel herziene versie 2026*. Nederlands Instituut van Psychologen. Hessen DJ (2026). "Richtlijnen voor steekproefgroottes bij continu normeren." Utrecht University, https://github.com/djhessen/COTAN.

Examples

```
# Example 1: Planning before data collection (homoscedastic)

res1 <- sample_size_poly(
  n_groups = 5,
  poly_degree = 2,
  n0 = 10 #400
)

# Example 2: Midway planning with variance estimates (heteroscedastic)
invisible(data("ids_data"))
ids_data$age_group <- cut(ids_data$age, breaks = seq(6, 18, by = 1))
v <- tapply(ids_data$y7, ids_data$age_group, var, na.rm = TRUE)

res2 <- sample_size_poly(
 n_groups = length(v),
 poly_degree = c(1,2,3),
 n0 = c(10,20), #c(300,400),
 variances = v
 )
```

---

shape_data *Shape data as input for* fb_select()

Description

shape_data() reshapes the response variable into the right format for the specified distribution and removes all cases with missing data on the score or age variable. The result is suitable for use as input to fb_select().

Usage

```
shape_data(
  data,
  age_name,
  score_name,
  family,
  max_score = NULL,
  verbose = TRUE
)
```

Arguments

`data`	data.frame. Sample on which to fit the distribution; contains the scores and ages.
`age_name`	string. Name of the age variable.
`score_name`	string. Name of the score variable.
`family`	string. For example, `"BB"`, `"BCPE"`, `"NO"`, etc. See gamlss.dist::gamlss.family for more information on available distributions. Also accepts user-defined distributions like truncated distributions.
`max_score`	numeric. Highest possible score in the norm table. Defaults to the maximum observed score in the sample.
`verbose`	logical. If `TRUE`, messages are printed whenever a transformation is applied.

Details

The function checks whether the response values are valid for the specified GAMLSS distribution family. If not, transformations are applied to ensure compatibility. Messages are printed (if `verbose = TRUE`) to describe each transformation.

Unexpected transformations should prompt inspection of the original data. Note that the function does **not** assess whether the chosen family is appropriate for the data. It only ensures compatibility.

Compatible with all gamlss distributions, with the exception of distributions in the multinomial family (gamlss::.gamlss.multin.list). This includes user-defined distributions, such as truncated distributions.

Value

A data.frame containing the original variables and a new column `shaped_score`, with the response variable in the correct format for GAMLSS modeling.

References

Voncken L, Albers CJ, Timmerman ME (2019). "Model selection in continuous test norming with GAMLSS." *Assessment*, **26**(7), 1329–1346. doi:10.1177/1073191117715113.

See Also

`fb_select()`

Examples

```
invisible(data("ids_data"))
mydata_BB <- shape_data(ids_data, age_name = "age", score_name = "y14", family = "BB")
mydata_BCPE <- shape_data(ids_data, age_name = "age", score_name = "y14", family = "BCPE")
```

Index

∗ **datasets**
    cotapp_data, 18
    ids_data, 22
    ids_kn_data, 23
    ids_rel_data, 23

boot_centiles, 2
boot_centiles(), *6*, *8*, *10*, *13*
boot_composite, 4
boot_composite(), *10*, *13*
boot_heatmap, 7
boot_heatmap(), *3*, *6*, *10*, *13*
boot_sample, 9
boot_sample(), *6*
boot_subtest, 11
boot_subtest(), *3*, *6*, *8*

centiles_bin, 14
composite_shape, 17
cotapp_data, 18

different_rel, 18
different_rel(), *27*

fb_select, 20
fb_select(), *15–17*, *21*, *24*, *26*, *33*, *34*

gamlss.dist::gamlss.family, *21*, *34*
gamlss::.gamlss.bi.list, *15*
gamlss::.gamlss.multin.list, *34*
gamlss::gamlss(), *21*
graphics::lines(), *29*
graphics::plot(), *29*
graphics::points(), *29*

ids_data, 22
ids_kn_data, 23
ids_rel_data, 23

normtable_create, 24
normtable_create(), *17*, *21*, *29*, *31*

plot_drel, 27
plot_drel(), *19*
plot_normtable, 28
plot_normtable(), *26*

reliability_window, 30

sample_size_poly, 31
shape_data, 33
shape_data(), *17*, *21*

tempfile(), *25*