Package: EmStanS (via r-universe)

November 29, 2026

Type Package

Title Embedded Standard Setting

Version 1.1.0

Maintainer Sooyong Lee <sooyongl09@gmail.com>

Description Provides functions for embedded standard setting, including data preparation, item-level summaries, cut score estimation, visualization, reporting, and simulation. The package implements embedded standard setting approaches such as ESS-Count and ESS-Weight to support the estimation of performance standards from item-achievement level descriptor alignments and empirical item information.

Depends R (>= 4.1.0)

Imports bslib, data.table, dplyr, DT, ggplot2, htmltools, mgcv, plotly, psych, readxl, stats, stringi, stringr, shiny, tibble, utils

Suggests testthat (>= 3.0.0), withr

License GPL (>= 3)

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Config/Needs/website quarto

URL https://sooyongl.github.io/EmStanS/

Collate 'EmStanS-package.R' 'ess_data.R' 'ess_estimate.R' 'ess_diagnose.R' 'ess_main.R' 'ess_bootstrap.R' 'ess_locations.R' 'ess_methods.R' 'ess_plots.R' 'ess_profile.R' 'ess_report.R' 'validate.R' 'z_simulation_function.R'

RoxygenNote 8.0.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev zlib1g-dev

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

Date/Publication 2026-11-22 01:38:41 UTC

RemoteUrl https://github.com/sooyongl/emstans

RemoteRef HEAD

RemoteSha 0a982327ec1ade87d72141c52520171c6390c189

Contents

bootstrapping . . . 2
emstans . . . 3
ess_data . . . 5
ess_locations . . . 7
ess_read_excel . . . 10
extract . . . 11
generateMockDataSet . . . 12
plot.ESS . . . 14
report . . . 14
summary.ESS . . . 15
update.ESS . . . 16
validateData . . . 17

Index **18**

`bootstrapping` *Bootstrap the modal cut scores of one unit*

Description

Resamples the unit's items with replacement, keeping each item's location and modal aligned level, and re-estimates the cut scores in every sample.

Usage

```
bootstrapping(x, ...)

## S3 method for class 'ESS'
bootstrapping(
  x,
  unit,
  n = 100,
  seed = NULL,
  grid = c("items", "jittered"),
  ...
)
```

Arguments

x	An object of class ‘"ESS"‘.
...	Former argument names, still accepted with a warning: ‘gca_name‘ (now ‘unit‘) and ‘boot_num‘ (now ‘n‘).
unit	The unit to bootstrap, as named in ‘extract(x, "cutscore")‘.
n	Number of bootstrap samples. Default ‘100‘.
seed	Optional random seed, for reproducible draws.
grid	Candidate cuts in each sample: * ‘"items"‘ (default): the sampled items’ locations, with the exact ESS minimum, the same estimator as [emstans()] (with ‘smooth = TRUE‘ in the original fit, the GAM-smoothed minimum). * ‘"jittered"‘: an evenly spaced grid over the location range (step 0.1 when the range is under 10, else 1), each point randomly moved by up to one step, shared by all cut levels; the cut is the minimum of a GAM-smoothed inconsistency curve over that grid. This is the bootstrap used by EmStanS before the 0.1.0 rewrite.

Value

An object of class ‘"bootESS"‘: use ‘print()‘, ‘summary()‘, ‘extract()‘, ‘report()‘.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")
boot <- bootstrapping(res, unit = "M1-All", n = 50, seed = 1)
boot
head(extract(boot, "draws"))
extract(boot, "plot")
if (interactive()) report(boot)

# The jittered-grid bootstrap
bootstrapping(res, unit = "M1-All", n = 10, seed = 1, grid = "jittered")
```

---

emstans *Run Embedded Standard Setting Analysis*

Description

‘emstans()‘ estimates cut scores with Embedded Standard Setting (ESS) from panelists’ item-level alignments and item locations. Each grade and round is analysed separately: cut scores are estimated for every panelist and for the modal alignment across panelists (per table and for all tables), and the median and average of the panelist cuts are reported. For the modal result of each grade, the function also reports agreement between aligned and empirical levels, an item review, and impact data.

Usage

```
emstans(
  file = NULL,
  data = NULL,
  ...,
  grades = NULL,
  location = "RP67",
  method = c("weight", "count"),
  descriptors = c("threshold", "range"),
  smooth = FALSE,
  by_domain = FALSE,
  domains = NULL,
  font_size = 14,
  digits = 3
)
```

Arguments

file	Path to an EmStanS Excel workbook (sheets Setup, Panelists, Ratings, ItemMetaData, ExamineeMetaData). Either ‘file‘ or ‘data‘ must be given.
data	An object from [ess_data()] or [ess_read_excel()], or a list of data frames in the workbook’s sheet order (setup, panelists, ratings, item data, examinee data).
...	Former argument names, still accepted with a warning. All arguments after ‘data‘ must be named in full. Former names: ‘filePath‘ (now ‘file‘), ‘grade‘ (‘grades‘), ‘loc‘ (‘location‘), ‘WESS‘ (‘method‘), ‘gamest‘ (‘smooth‘), ‘domain‘ and ‘select_domain‘ (‘by_domain‘, ‘domains‘), ‘targets‘ (use the ‘columns‘ argument of [ess_data()]) and ‘median‘ (no longer needed: modal, median and average cuts are always reported).
grades	Grades to analyse: ‘NULL‘ for all, a numeric vector of positions, or a character vector of grade names. Default ‘NULL‘.
location	Name of the item location column, such as ‘"RP67"‘ or ‘"RP50"‘. Default ‘"RP67"‘.
method	‘"weight"‘ for ESS-Weight (minimise the summed distance of inconsistent items) or ‘"count"‘ for ESS-Count (minimise their number). Default ‘"weight"‘.
descriptors	How the level descriptors relate to the cut scores (Lewis & Cook, 2020). ‘"threshold"‘ (default): each descriptor describes students at the threshold of a level, who are expected to answer every item aligned to that level at the response probability, so the cut that opens a level lies at or above the items aligned to it. The cut between the items aligned to levels j and j + 1 opens level j; examinees below the lowest cut are reported as "Below" the lowest level; and the highest level is an item-only category ("beyond" the highest threshold) that bounds the cut below it but opens no examinee level, so add a level above the highest descriptor if its cut is wanted. ‘"range"‘: each descriptor describes the whole range of a level, so items aligned to a level are expected at or above that level’s cut, and each cut separates the items aligned below a level from those aligned at or above it. The two forms give similar cuts (the same ESS-Weight minimum, usually at the

	same or a neighbouring item location) but attach them to different levels. An item exactly at a cut belongs to the level below it (threshold) or above it (range). Default ‘"threshold"‘. Versions before 1.1.0 used range descriptors by default; set ‘descriptors = "range"‘ to reproduce their results.
smooth	If ‘TRUE‘, cut scores are the minima of a GAM-smoothed inconsistency profile instead of the exact minima. Default ‘FALSE‘.
by_domain	If ‘TRUE‘, each content domain (the item data’s Domain column) is also analysed separately, in addition to all items of each grade. Default ‘FALSE‘.
domains	With ‘by_domain = TRUE‘, the domains to analyse; ‘NULL‘ for all of them.
font_size	Font size for plots. Default ‘14‘.
digits	Decimal places in printed tables. Default ‘3‘.

Value

An object of class ‘"ESS"‘. Use [extract()] to get its tables, [report()] for an HTML report, [plot()] for plots, [update()] to set cut scores manually and [bootstrapping()] for bootstrap cut scores.

Examples

```
# Simulated study: 2 grades, 2 tables of 3 panelists, 30 items per grade
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")   # threshold descriptors
res
extract(res, "cutscore")

# ESS-Count, with each content domain also analysed separately
res_count <- emstans(data = sim, location = "Loc_RP50", method = "count",
                     by_domain = TRUE)
names(extract(res_count, "cutscore"))

# Range descriptors: each cut opens the level of the items aligned at or
# above it (the default before EmStanS 1.1.0; Lewis & Cook, 2020)
res_rng <- emstans(data = sim, location = "Loc_RP50", descriptors = "range")
extract(res_rng, "cutscore")

# From an EmStanS Excel workbook (a simulated one ships with the package)
path <- system.file("extdata", "example_study.xlsx", package = "EmStanS")
res_xlsx <- emstans(path, location = "Loc_RP50")
summary(res_xlsx)
```

---

ess_data *Prepare Embedded Standard Setting input data*

Description

Builds the input for [emstans()] from data frames or files, in any column layout. Ratings and items are required; setup information (level labels, SD, EC) can come from a setup table, from arguments, or both, with arguments taking precedence.

Usage

```
ess_data(
  ratings,
  items,
  setup = NULL,
  examinees = NULL,
  levels = NULL,
  sd = NULL,
  ec = NULL,
  columns = NULL,
  quiet = TRUE
)
```

Arguments

ratings	Panelists' item alignments: a data frame or a path to a '.csv' file. One row per panelist, item and round.
items	Item information, including the location column(s) such as 'RP67': a data frame or a path to a '.csv' file.
setup	Optional setup table (one row per grade), a data frame or a path to a '.csv' file. Its 'Level Options' column lists the level labels from lowest to highest, separated by commas.
examinees	Optional score frequencies for impact data: a data frame or a path to a '.csv' file in long format, one row per grade and score, with three columns: grade ('GCA' or 'Grade'), 'Score' (on the same scale as the item locations) and 'Freq' (number of examinees with that score). Wide tables with repeated Score/Freq/Grade blocks side by side are not accepted.
levels	Level labels from lowest to highest. A character vector applies to every grade; a named list gives labels per grade, for example 'list(R1 = c("E", "M", "L"), R2 = c("B", "P", "A"))'. Overrides the setup table.
sd, ec	Scaling SD and essential-consistency threshold. A single number applies to every grade; a named vector gives values per grade. Override the setup table. Defaults are SD = 1 and EC = 0.
columns	Optional column mapping for tables whose column names differ from the defaults. A named list per table, each a named list or vector of input column names or positions, for example 'list(ratings = list(item_id = "ItemID", level = 8))'. The standard names are: ratings 'grade', 'round', 'table', 'panelist', 'item_id', 'level'; items 'grade', 'item_id', 'domain', 'order'; setup 'grade', 'levels', 'sd', 'ec'; examinees 'grade', 'score', 'freq'.
quiet	If 'FALSE', report the column mapping that was used.

Value

An object of class ‘"ess_data"‘.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)

# Data frames with the default column names
d <- ess_data(ratings = sim$rating, items = sim$item_data,
              setup = sim$setup, examinees = sim$examineedata_data)
d

# CSV files with other column names and level labels, and no setup table
ratings <- sim$rating
names(ratings)[names(ratings) == "Panelist"] <- "Rater"
names(ratings)[names(ratings) == "ALD"] <- "Judgment"
ratings$Judgment <- c(Level1 = "Basic", Level2 = "Proficient",
                      Level3 = "Advanced")[ratings$Judgment]
ratings_csv <- tempfile(fileext = ".csv")
items_csv <- tempfile(fileext = ".csv")
write.csv(ratings, ratings_csv, row.names = FALSE)
write.csv(sim$item_data, items_csv, row.names = FALSE)

d <- ess_data(
  ratings = ratings_csv,
  items   = items_csv,
  levels  = c("Basic", "Proficient", "Advanced"),
  columns = list(ratings = list(panelist = "Rater", level = "Judgment"))
)
res <- emstans(data = d, location = "Loc_RP50")
extract(res, "cutscore")
unlink(c(ratings_csv, items_csv))
```

`ess_locations` *Item locations at a response probability*

Description

Converts IRT item parameters into ESS item locations. The location of a dichotomous item is the ability at which an examinee answers it correctly with probability ‘rp‘. A polytomous item with maximum score K gives K locations ("pages"), one for each score point k = 1, ..., K: the ability at which an examinee earns score k or higher with probability ‘rp‘.

Usage

```
ess_locations(
  pars,
  model = NULL,
  rp = 0.67,
  D = 1,
  item = 1,
  a = "a",
  b = "b",
  c = "c",
  guessing = c("none", "rescale", "huynh"),
  slope = 1,
  intercept = 0,
  sep = "_"
)
```

Arguments

pars	A data frame of item parameters, one row per item.
model	The IRT model: ‘"rasch"‘, ‘"2pl"‘, ‘"3pl"‘ (dichotomous) or ‘"pcm"‘, ‘"gpcm"‘, ‘"grm"‘ (polytomous). Either one value for every item or one value per row of ‘pars‘, which allows mixed-format tests. If ‘NULL‘, ‘pars‘ must have a column named ‘model‘.
rp	Response probability, between 0 and 1. The default 0.67 is the usual RP67; 0.5 returns the RP50 location (‘b‘ for Rasch and 2PL items).
D	Scaling constant: 1 for the logistic metric (mirt, flexMIRT, Winsteps), 1.702 (or 1.7) for the normal metric (PARSCALE, BILOG-MG and many operational programs).
item	Column of ‘pars‘ holding the item ID, by name or position.
a	Column holding the discrimination (slope). Not used for ‘"rasch"‘ and ‘"pcm"‘ items, where it is fixed at 1.
b	Threshold columns. Either a prefix (the default ‘"b"‘ matches columns ‘b‘, ‘b1‘, ‘b2‘, ...; a column ‘b‘ counts as ‘b1‘) or a character vector naming the threshold columns in order. Dichotomous items use one threshold; polytomous items use thresholds 1 to K and ‘NA‘ after that, so items with different maximum scores can share one table. See Details for what the thresholds mean under each model.
c	Column holding the lower asymptote (pseudo-guessing) of ‘"3pl"‘ items.
guessing	How ‘rp‘ is applied to ‘"3pl"‘ items: * ‘"none"‘ (default): P(correct) = ‘rp‘. Requires ‘rp > c‘. * ‘"rescale"‘: P(correct) = c + (1 - c) ‘rp‘, i.e. ‘rp‘ of the way from the guessing floor to 1. * ‘"huynh"‘: P(correct) = (2 + c) / 3 (Huynh, 1998, 2006). ‘rp‘ is not used for these items; with c = 0 this is 2/3. Other models are not affected.
slope, intercept	Linear transformation to a reporting scale: ‘location = intercept + slope * theta‘. ‘slope‘ must be positive so that item order is kept.
sep	Separator between item ID and score point in the ‘Item_ID‘ of a polytomous score point (‘"Q12_2"‘).

Details

Dichotomous items. With P(theta) = c + (1 - c) / (1 + exp(-D a (theta - b))), the location is ‘theta = b + log((p - c) / (1 - p)) / (D a)‘, where p is the target probability (‘rp‘, or as set by ‘guessing‘). Rasch items have a = 1 and c = 0; 2PL items have c = 0. Under Rasch, changing ‘rp‘ moves every item by the same amount, so item order does not change. Under 2PL and 3PL the shift is ‘1 / (D a)‘ times larger for less discriminating items, so ‘rp‘ can change item order.

Polytomous items. The location of score point k solves P(X >= k | theta) = ‘rp‘. These locations are always increasing in k within an item. * ‘"grm"‘: the thresholds are the category boundaries ‘b_k‘ of P(X >= k) = 1 / (1 + exp(-D a (theta - b_k))), so ‘theta_k = b_k + log(rp / (1 - rp)) / (D a)‘. Boundaries must increase. * ‘"pcm"‘, ‘"gpcm"‘: the thresholds are step difficulties ‘delta_v‘ of P(X = k) proportional to exp(sum over v <= k of D a (theta - delta_v)). They may be disordered. The location is found numerically.

Parameterizations differ between programs; convert before calling. For mirt, ‘"graded"‘ items give ‘b_k = -d_k / a‘, and ‘"gpcm"‘ items (with the default scoring 0, 1, ..., K) give ‘delta_k = -(d_k - d_(k-1)) / a‘. Programs that report an item location b with step deviations d_v use ‘delta_v = b - d_v‘ (PARSCALE) or ‘b + d_v‘, depending on sign convention.

Value

A data frame with one row per location: ‘Item_ID‘ (the item ID for dichotomous items, item ID, ‘sep‘ and score point for polytomous items), ‘item‘, ‘score‘, ‘max_score‘, ‘model‘, ‘p‘ (the probability the location was solved for), ‘theta‘ and ‘location‘ (‘theta‘ on the reporting scale). Rows keep the order of ‘pars‘; they are not sorted by location.

References

Huynh, H. (1998). On score locations of binary and partial credit items and their applications to item mapping and criterion-referenced interpretation. *Journal of Educational and Behavioral Statistics, 23*(1), 35-56.

Huynh, H. (2006). A clarification on the response probability criterion RP67 for standard settings based on bookmark and item mapping. *Educational Measurement: Issues and Practice, 25*(2), 19-20.

Examples

```
# Mixed-format test: 3PL multiple-choice items and a GPCM constructed-
# response item scored 0-3
pars <- data.frame(
  Item_ID = c("MC1", "MC2", "CR1"),
  model   = c("3pl", "3pl", "gpcm"),
  a  = c(1.2, 0.7, 0.9),
  b1 = c(-0.5, 0.3, -1.0),
  b2 = c(NA, NA, 0.2),
  b3 = c(NA, NA, 1.1),
  c  = c(0.20, 0.25, NA)
)
ess_locations(pars, D = 1.702)

# RP50 and Huynh's guessing-adjusted RP for the 3PL items
```

```
ess_locations(pars, rp = 0.5, D = 1.702)
ess_locations(pars, D = 1.702, guessing = "huynh")

# Locations on a reporting scale (scale score = 500 + 50 * theta)
ess_locations(pars, D = 1.702, slope = 50, intercept = 500)
```

`ess_read_excel` *Read an EmStanS Excel workbook*

Description

Reads the five-sheet EmStanS workbook (Setup, Panelists, Ratings, ItemMetaData, ExamineeMetaData) by sheet name, falling back to sheet position when a name is not found, and returns [ess_data()].

Usage

```
ess_read_excel(
  path,
  sheets = NULL,
  levels = NULL,
  sd = NULL,
  ec = NULL,
  columns = NULL,
  quiet = TRUE
)
```

Arguments

`path`	Path to the '.xlsx' file.
`sheets`	Optional sheet names, a named character vector with any of 'setup', 'ratings', 'items', 'examinees'.
`levels`	Level labels from lowest to highest. A character vector applies to every grade; a named list gives labels per grade, for example 'list(R1 = c("E", "M", "L"), R2 = c("B", "P", "A"))'. Overrides the setup table.
`sd, ec`	Scaling SD and essential-consistency threshold. A single number applies to every grade; a named vector gives values per grade. Override the setup table. Defaults are SD = 1 and EC = 0.
`columns`	Optional column mapping for tables whose column names differ from the defaults. A named list per table, each a named list or vector of input column names or positions, for example 'list(ratings = list(item_id = "ItemID", level = 8))'. The standard names are: ratings 'grade', 'round', 'table', 'panelist', 'item_id', 'level'; items 'grade', 'item_id', 'domain', 'order'; setup 'grade', 'levels', 'sd', 'ec'; examinees 'grade', 'score', 'freq'.
`quiet`	If 'FALSE', report the column mapping that was used.

Value

An object of class ‘"ess_data"‘.

Examples

```
# A simulated workbook that ships with the package
path <- system.file("extdata", "example_study.xlsx", package = "EmStanS")
readxl::excel_sheets(path)
d <- ess_read_excel(path)
d
res <- emstans(data = d, location = "Loc_RP50")
extract(res, "cutscore")
```

`extract` *Extract tables from an ESS result*

Description

Extract tables from an ESS result

Usage

```
extract(x, ...)

## S3 method for class 'ESS'
extract(x, what = "all", ...)
```

Arguments

`x`	An object of class ‘"ESS"‘ returned by [emstans()].
`...`	Not used.
`what`	Which results: ‘"cutscore"‘ (modal cut scores per unit, named by the level each cut opens), ‘"individual"‘ (panelist, modal, median and average cuts), ‘"detailed"‘ (agreement and crosstabs), ‘"summary"‘ (impact data), ‘"review"‘ (item review), ‘"profile"‘ (inconsistency profiles of the modal results) or ‘"all"‘. Default ‘"all"‘.

Value

A data frame or a named list of data frames. Level columns hold the level labels as ordered factors.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")
extract(res, "cutscore")
head(extract(res, "review"))
head(extract(res, "profile")[["M1-All"]])
names(extract(res))
```

`generateMockDataSet` *Generate a Fake Data Set for Embedded Standard Setting*

Description

'generateMockDataSet()' generates a simulated data set that follows the input structure required by 'emstans()'. The generated data can be used to test functions in the EmStanS package, demonstrate the expected data format, or run examples without using operational data.

Usage

```
generateMockDataSet(
  ngca,
  ntable,
  npanelist,
  cor_val,
  nlevel,
  sdinp = 1,
  ecinp = 0,
  n,
  ...,
  domains = c("Alg", "Geom", "Calc", "Prob")
)
```

Arguments

`ngca`	A numeric value indicating the number of grade clusters or GCAs to generate.
`ntable`	A numeric value indicating the number of tables to generate within each GCA.
`npanelist`	A numeric value indicating the number of panelists to generate within each table.
`cor_val`	A numeric value indicating the intended correlation between item locations and aligned level descriptions.
`nlevel`	A numeric value indicating the number of performance levels or aligned level-description categories.
`sdinp`	A numeric value used to adjust the standard deviation of the values used in Weighted Embedded Standard Setting. Default is '1'.

ecinp	A numeric value used to control essentially consistent items in the simulated item ratings. Default is ‘0‘.
n	A numeric value indicating the number of items to generate within each GCA.
...	Additional numeric arguments passed to ‘runif()‘, typically the minimum and maximum score values used to generate item locations. For example, ‘100, 300‘ generates item locations between 100 and 300.
domains	The content subdomains of the items, added to the item data as a ‘Domain‘ column. Either a character vector of subdomain names (for example ‘c("Number", "Algebra", "Geometry")‘), or the number of subdomains, which are then named from ‘"Alg"‘, ‘"Geom"‘, ‘"Calc"‘, ‘"Prob"‘, ‘"Stat"‘, ‘"Num"‘ (or ‘"D1"‘, ‘"D2"‘, ... beyond six). Items are split as evenly as possible across subdomains and assigned at random, so every subdomain spans the whole location range. ‘NULL‘ adds no ‘Domain‘ column. Must be named, since it follows ‘...‘.

Details

This function is mainly intended for package testing, examples, and demonstrations. Use the ‘Domain‘ column with ‘emstans(by_domain = TRUE)‘ to also estimate cut scores within each subdomain. The returned object mimics the data structure expected by ‘emstans()‘, allowing users to inspect the required format before preparing their own data.

Value

A list of simulated data frames formatted for use with ‘emstans()‘. The returned list contains test setup information, panelist information, item ratings, item metadata, and student data.

Examples

```
# 2 grades, 2 tables of 3 panelists, 30 items per grade, locations 300-500
set.seed(1)
fake_data <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                                 cor_val = 0.7, nlevel = 3, sdinp = 1,
                                 ecinp = 0, n = 30, 300, 500)
names(fake_data)
head(fake_data$rating)

# Three named subdomains
fake_data <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 4,
                                 cor_val = 0.6, nlevel = 3, sdinp = 1,
                                 ecinp = 0, n = 45, 300, 500,
                                 domains = c("Number", "Algebra", "Geometry"))
table(fake_data$item_data$GCA, fake_data$item_data$Domain)

# Four levels and no subdomains
fake_data <- generateMockDataSet(ngca = 1, ntable = 2, npanelist = 3,
                                 cor_val = 0.7, nlevel = 4, sdinp = 1,
                                 ecinp = 0, n = 40, 300, 500, domains = NULL)
names(fake_data$item_data)
```

plot.ESS *Plot an ESS result*

Description

Plot an ESS result

Usage

```
## S3 method for class 'ESS'
plot(x, type = c("profile", "impact"), unit = NULL, ...)
```

Arguments

x	An object of class ‘"ESS"‘.
type	‘"profile"‘ for the inconsistency profile of a unit’s modal result, or ‘"impact"‘ for cut scores and impact across grades.
unit	For ‘type = "profile"‘, the unit to plot (default: the first). For ‘type = "impact"‘, the item subset (default ‘"All"‘).
...	Not used.

Value

A ggplot object (for ‘"impact"‘, a list of ggplot objects).

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")
plot(res)
plot(res, type = "profile", unit = "M2-All")
plot(res, type = "impact")
```

report *HTML report of an ESS result*

Description

Builds an HTML page (shown in the RStudio viewer or a browser) with the results of [emstans()]. Level labels from the input are used throughout.

Usage

```
report(x, ...)

## S3 method for class 'ESS'
report(x, what = "all", ...)
```

Arguments

x	An object of class ‘"ESS"‘.
...	Not used.
what	Section(s) to show: ‘"all"‘, ‘"individual"‘ (panelist and group cut scores), ‘"detailed"‘ (per-unit profile plot, crosstabs and agreement), ‘"summary"‘ (impact data), ‘"review"‘ (item review) or ‘"setup"‘ (input data overview). Default ‘"all"‘.

Value

A browsable HTML object.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")
rep <- report(res)
rep_summary <- report(res, what = "summary")
if (interactive()) rep
```

---

summary.ESS *Summarise an ESS result*

Description

One row per unit and cut level with the modal, median and average cut scores (all tables), and the agreement between aligned and empirical levels for the modal result.

Usage

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

Arguments

object	An object of class ‘"ESS"‘.
...	Not used.

Value

A data frame, invisibly after printing.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")
summary(res)
```

`update.ESS` *Set cut scores manually*

Description

Replaces the modal (all tables) cut scores of one or more units and recomputes agreement, item review and impact for them, with the descriptor form ‘object‘ was estimated with. Each cut’s page is the last item at or below the cut score (threshold descriptors, the default) or the first item at or above it (range descriptors).

Usage

```
## S3 method for class 'ESS'
update(object, ...)
```

Arguments

`object`	An object of class ‘"ESS"‘.
`...`	Named numeric vectors of cut scores, one value per cut in the order of ‘extract(object, "cutscore")‘, named by unit, for example ‘‘ update(res, ‘R1-All‘ = c(-0.9, 0.4)) ‘‘.

Value

An updated object of class ‘"ESS"‘.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
res <- emstans(data = sim, location = "Loc_RP50")
extract(res, "cutscore")

# Set the cuts of grade M1 by hand; agreement, review and impact are updated
```

```
res_manual <- update(res, `M1-All` = c(350, 440))
extract(res_manual, "cutscore")
summary(res_manual)
```

---

validateData *Check input data*

Description

Kept for compatibility: runs the checks of [ess_data()] on a list of data frames in the workbook's sheet order (setup, panelists, ratings, item data, examinee data) and reports the result as text.

Usage

```
validateData(imprt_data)
```

Arguments

imprt_data	A list of data frames, or an object from [ess_data()].

Value

‘"Data input is validated"‘, or the first problem found.

Examples

```
set.seed(1)
sim <- generateMockDataSet(ngca = 2, ntable = 2, npanelist = 3,
                           cor_val = 0.7, nlevel = 3, sdinp = 1, ecinp = 0,
                           n = 30, 300, 500)
validateData(sim)

# An alignment label that is not one of the grade's levels
sim$rating$ALD[1] <- "Level9"
validateData(sim)
```

Index

bootstrapping, 2

emstans, 3
ess_data, 5
ess_locations, 7
ess_read_excel, 10
extract, 11

generateMockDataSet, 12

plot.ESS, 14

report, 14

summary.ESS, 15

update.ESS, 16

validateData, 17