Package 'psketti'

March 10, 2021

Type Package

Title Generating Investigatory Plots and Tables for Rasch Analysis

Version 0.1.2

Author Shane B. Galvin

Maintainer Shane B. Galvin <sbmgalvin@gmail.com>

Description psketti generates investigatory plots and tables to assist in Rasch Analysis by using a number of accessor, table, and plotting functions. Data are extracted from dichotmous (RM) and partial credit (PCM) Rasch models fitted by Conditional Maximum Likelihood (CML) estimation in the eRm package. Empirical Item Characteristic Curves (ICC) are computed by dividing the latent dimension into class intervals in which the frequency of response to a category is counted and presented as a proportion of that class interval. Confidence Intervals for the Empirical ICC are also calculated. Infit and Outfit measures are also extracted for presentation as a simple diagnostic plot. Plots are compiled using ggplot2.

Depends R (>= 4.0.0)

Imports eRm (>= 1.0-1),
dplyr (>= 1.0.1),
ggplot2 (>= 3.3.2),
viridis (>= 0.5.1),
RColorBrewer (>= 1.1.2),
stats (>= 4.0.2)

Suggests tidyverse (>= 1.3.0),
knitr,
rmarkdown

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

R topics documented:

`al_dente` *Plot Dichotomous Rasch Model Residuals*

Description

Plot Dichotomous Rasch Model Residuals

Usage

```
al_dente(eRm.obj, style = "present")
```

Arguments

`eRm.obj`	an eRm object of class 'RM'.
`style`	Options for plot colour; a character string for plotting style options are present for coloured plots, or print for black and white plots. Defaults to "present".

Value

A list object containing multiple residual plots and residuals Pearson correlations matrix.

`FakeData` *Fake Data*

Description

Simulated scored data for 23 items and 1200 participants, where items where presented in random order to each participant. Contains responses for dichotomous (X), and polychotomous (K) data. Item delivery order is stored in the Index column Data were simulated according to suggestions in Linacre (2007). Data are stored in long format.

Usage

```
data(FakeData)
```

Format

An object of class "cross"; see read.cross.

Source

Rasch Measurement Transactions

References

Linacre. (2013) Rasch Measurement Transactions 21:3 p. 1125 (Rasch Measurement Transactions)

Examples

```
# Show fake data
data(FakeData)

library(tidyverse)
# Count of item responses by item and polychotomous category
count_by_reponse_K <- FakeData %>% group_by(Item, K) %>%  summarise(Count = n())
```

FakeItems *Fake Items*

Description

Item parameters for simulated scored data for 23 items and 1200 participants, where items where presented in random order to each participant. Contains responses for dichotomous (X), and polychotomous (K) data. Item delivery order is stored in the Index column. Data were simulated according to suggestions in Linacre (2007). Data are stored in long format.

Usage

```
data("FakeItems")
```

Format

An object of class "cross"; see read.cross.

Source

Rasch Measurement Transactions

References

Linacre. (2013) Rasch Measurement Transactions 21:3 p. 1125 (Rasch Measurement Transactions)

Examples

```
# Show fake item parameters
data("FakeItems")
```

FakePCMData *Fake Data*

Description

Simulated scored data for 10 items and 1200 participants, where items where presented in random order to each participant. Contains responses for dichotomous (X), and polychotomous (K) data. Item delivery order is stored in the Index column Data were simulated according to suggestions in Linacre (2007). Data are stored in long format.

Usage

```
data("FakePCMData")
```

Format

An object of class `"cross"`; see `read.cross`.

Source

Rasch Measurement Transactions

References

Linacre. (2013) Rasch Measurement Transactions 21:3 p. 1125 (Rasch Measurement Transactions)

Examples

```
# Show fake data
data("FakePCMData")

library(tidyverse)
# Count of item responses by item and polychotomous category
count_item_by_reponse_K <- FakePCMData %>%
                           pivot_longer(cols = -ID,
                                        names_to = "Item",
                                        values_to = "Response") %>%
                           group_by(Item, Response) %>%
                           summarise(Count = n())
```

FakePCMItems *Fake Items*

Description

Item parameters for simulated scored data for 11 items and 1200 participants, where items where presented in random order to each participant. Contains responses for dichotomous (X), and polychotomous (K) data. Item delivery order is stored in the Index column. Data were simulated according to suggestions in Linacre (2007). These are the resulting item parameters.

Usage

```
data("FakePCMItems")
```

Format

An object of class "cross"; see read.cross.

Source

Rasch Measurement Transactions

References

Linacre. (2013) Rasch Measurement Transactions 21:3 p. 1125 (Rasch Measurement Transactions)

Examples

```
# Show fake item parameters
data("FakePCMItems")
```

---

ingrediente *Score Report Tables*

Description

Generates score report for dataframe.

Usage

```
ingrediente(x, ID, Item, Score, K, K_options, Index = NULL)
```

Arguments

x	a dataframe.
ID	column name for ID column
Item	column name for Item column
Score	a column name for response scores
K	column name for column containing multiple choice responses
K_options	An ordered factor object to arrange column order in the score table.
Index	Column name for order of administration per participant. This can also be an ordered factor for the item names. Orders the Response string. Defaults to 'NULL', using the items to order the response string.

Value

Score reports for participants, with counts of category selection and a score string ordered by score string index

Examples

```
# Example 1
# For dichotomous Rasch model
library(psketti)
data("FakeData")

K_opt <- factor(LETTERS[1:5], levels = LETTERS[1:5], ordered = TRUE)
score_report <- ingrediente(x = FakeData,
                            Item = "Item",
                            ID = "ID",
                            Score = "X",
                            K = "K",
                            K_options = K_opt,
                            Index = "Index")

# show score report for values with a total score <= 5
score_report[score_report$total_score <= 1, ]
# Score report ordering response string by item difficulty
data("FakeItems")
FI2 <- FakeItems[order(FakeItems$Beta),]
row.names(FI2)<- NULL
FI_factor <- factor(FI2$Item, levels = FI2$Item, ordered = TRUE)



score_report2 <- ingrediente(x = FakeData,
                             Item = "Item",
                             ID = "ID",
                             Score = "X",
                             K = "K",
                             K_options = K_opt,
                             Index = FI_factor)

# show score report for values with a total score <= 5
score_report2[score_report2$total_score == 21, ]

## Not run:
# Example 2
# For Rasch partial credit model
library(dplyr)
library(tidyr)
data("FakePCMData")
data("FakePCMItems")

# Arrange Data, wide to long
fpcm <- FakePCMData %>%
  pivot_longer(cols = -ID, values_to = "Response", names_to = "Item") %>%
  mutate(X = Response) %>%
  mutate(K = as.character(Response)) %>%
  mutate(K = recode(K, "0" = "A", "1" = "B", "2" = "C", "3" = "D"))

# factor variable: Index for item order
F2            <- FakePCMItems[, c("Item", "Beta")] # extract relevant cols
F2            <- F2[order(F2$Beta),]               # order dataframe
row.names(F2) <-  NULL                             # drop rownames
```

```
# create factor variable
F_factor <- factor(F2$Item,
                   levels = F2$Item,
                   ordered = TRUE)
#apply factor to data frame
fpcm$Index <- fpcm$Item                          # Item -> Index
fpcm$Index <- factor(fpcm$Index,
                     levels = levels(F_factor),
                     ordered = TRUE)

fpcm      <- as.data.frame(fpcm) # ensure this is a dataframe!!

# factor variable for K categories
K_opt     <- factor(LETTERS[1:4],
                    levels = LETTERS[1:4],
                    ordered = TRUE)
# produce score report
score_pcm <- ingrediente(x = fpcm,
                         Item = "Item",
                         ID = "ID",
                         Score = "X",
                         K = "K",
                         Index = "Index",
                         K_options = K_opt)

score_pcm[score_pcm$total_score < 2, ] # print out score report

## End(Not run)
```

`item_fit_table` *Create Item Fit Table*

Description

This function extracts Item fit statistics from eRm 'itemfit()'

Usage

```
item_fit_table(eRm.obj)
```

Arguments

`eRm.obj`	input data, generated using a Rasch model estimation function from eRm.

Value

A data.frame of class ItemFit.

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
```

```
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm     <- RM(Fake_Data_scores)    # Estimate Rasch model

itemFit_psk <- item_fit_table(fake_rm) # item fit stats

itemFit_psk # output
```

---

psketti *Plot Your pskettified Data*

Description

This function extracts data from an eRm object of class 'RM' and converts to a format for plotting data. This also computes the empirical response values and empirical confidence intervals

Usage

```
psketti(
  pskettified_data,
  p.style = "present",
  p.IRFLocation = TRUE,
  p.empCI = TRUE,
  p.empICC = TRUE,
  p.empPoints = TRUE,
  Force_no_facet = FALSE
)
```

Arguments

pskettified_data	a list object generated from eRm object class'RM' using 'pskettify()'.
p.style	a character string for plotting style options are present for coloured plots, or print for black and white plots. Defaults to "present".
p.IRFLocation	logical, plots reference lines for Rasch IRF location. Defaults to 'TRUE'.
p.empCI	logical, plots confidence intervals for empirical points, calculated using 'pskettify()'. Defaults to 'TRUE'.
p.empICC	logical, plots empirical ICC for item. Defaults to 'TRUE'.
p.empPoints	logical, plots empirical points for based on class intervals/ score groups generated with 'pskettify()'. Defaults to 'TRUE'.
Force_no_facet	a logical, forces psketti to not use facets for polytomous models. Defaults to 'FALSE'.

Value

A list object containing multiple psketto plots.

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm   <- RM(Fake_Data_scores) # Estimate Rasch model

psk_data  <- pskettify(fake_rm)   # pskettify data

# multiple plots
multi_plot <- psketti(pskettified_data = psk_data)
multi_plot # plot call instructions

# print first item plot
multi_plot$Plot.List[['i01']][[1]]
```

`pskettify` *pskettify your data*

Description

This function extracts data from an eRm object of class 'RM' and converts to a format for plotting data. This also computes the empirical response values and empirical confidence intervals.

Usage

```
pskettify(eRm.obj, conf.level = 0.95, Theta.lwr = -6, Theta.upr = 6)
```

Arguments

`eRm.obj`	an eRm object of class 'RM'.
`conf.level`	the confidence level for empirical response curve. Defaults to 0.95.
`Theta.lwr`	The lowest limit of the latent dimension. Defaults to -6.
`Theta.upr`	The highest limit of the latent dimension. Defaults to 6.

Value

output_list containing presp and emp_ICC.

presp a data frame of ability (Theta) and conditional response probabilities to each item.

emp_ICC a dataframe containing proportion values and confidence intervals for ability class intervals to each item.

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm   <- RM(Fake_Data_scores) # Estimate Rasch model

psk_data  <- pskettify(fake_rm)   # pskettify data
```

---

psketti_distractor *Distractor Analysis Plots*

Description

Implementation of a graphical (Asril and Marais, 2011) approach to assigning a partial credit scoring system to data previously estimated with a dichotomous Rasch model. The function console output prints object details, a list of items, and generic example of how to call the plot.

Usage

```
psketti_distractor(
  x,
  ID,
  Item,
  K,
  response_options,
  eRm.obj,
  p.style = "present",
  distractor_colours = NULL,
  ncut = 10
)
```

Arguments

x	A long formatted dataframe
ID	column name for ID column
Item	column name for Item column
K	column name for column containing multiple choice responses.
response_options	An ordered factor object to arrange column order in the distractor table.
eRm.obj	An object of class eRm and model RM. Use 'eRm::RM(score_data)' to create this object. To plot empirical values for PCM see 'pskettify', 'psketti' and 'psketto'.
p.style	Plot output style, "print" for black and white, or "present" for color. Defaults to "present".
distractor_colours	An optional vector of colours for distractor plot lines. Must be the same length as response_options. Defaults to 'NULL' for viridis color palette.
ncut	Number of cut points to use for the theta axis. Defaults to ncut = 10. You can also set ncut = "Raw" to use the raw theta scores; which is only advisable if the ability data is uniformly distributed.

Value

Plot.List is a list object containing plots of empirical distractor proportions plotted against the dichotomous Rasch IRF.

Examples

```
library(eRm)
library(psketti)
data("FakeData")
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")

# set column names to be equal to original item names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID  # set ID as row names
Fake_Data_scores$ID         <- NULL                 # drop the ID column
fake_rm                     <- RM(Fake_Data_scores) # fit a Rasch Model

# Prepare response options factor
r_o <- factor(sort(unique(FakeData$K)),          # input var
              levels = sort(unique(FakeData$K)), # factor levels
              ordered = TRUE)                    # ordered

# multiple plots
spag_plot <- psketti_distractor(ID = "ID",              # set ID column
                                Item = "Item",          # set Item column
                                K= "K",                 # Set resp categories
```

```
                                x = FakeData,           # select data
                                eRm.obj = fake_rm,      # select eRm object
                                response_options = r_o, # set resp options
                                p.style = "present")    # set plotting style

spag_plot                         # plot call instructions
spag_plot$Plot.List[['i01']][[1]] # plot item 1
```

---

psketto *Plot a Single ICC*

Description

psketto is singlular of psketti; (spaghetti <–> spaghetto). This function plots the data for a single item from the output of 'pskettify()'. 'psketto()' is also used in 'psketti()' to create multiple ICC plots. Based on Asril and Marais (2011).

Usage

```
psketto(
  pskettified_data,
  item,
  item.label,
  style = "present",
  IRFLocation = TRUE,
  empCI = TRUE,
  empICC = TRUE,
  empPoints = TRUE,
  facet_curves = FALSE
)
```

Arguments

pskettified_data	input data, generated using 'pskettify()'.
item	character name of the item to be plotted
item.label	a character string of the item name to use.
style	a character string for plotting style options are present for coloured, or print for black and white. Defaults to "present".
IRFLocation	logical, plots reference lines for Rasch IRF location. Defaults to 'TRUE'
empCI	logical, plots confidence intervals for empirical points, calculated using 'pskettify()'. Defaults to 'TRUE'
empICC	logical, plots empirical ICC for item. Defaults to 'TRUE'
empPoints	logical, plots empirical points for based on class intervals/score groups generated with 'pskettify()'. Defaults to 'TRUE'
facet_curves	logical, should the plot be faceted by category curve? Applies only to polytomous Rasch models. Defaults to 'FALSE',.

Value

psketto plot.

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm   <- RM(Fake_Data_scores) # Estimate Rasch model

psk_data  <- pskettify(fake_rm)   # pskettify data

# plot IRF in default colours
psk_1_present <- psketto(psk_data,
                         style = "present",
                         item = "i01",
                         item.label = "i01")
psk_1_present # plot output

# plot IRF in default greyscale colours
psk_1_print <- psketto(psk_data,
                       style = "print",
                       item = "i01",
                       item.label = "i01")
psk_1_print # plot output
```

`psketto_simple` *Unadorned Rasch IRF*

Description

Unadorned Rasch IRF

Usage

```
psketto_simple(x, item, all.item = FALSE, item.label = FALSE)
```

Arguments

x	an object of class pskettified
item	If you want to plot a single IRF, use this argument to state the name of the item.
all.item	Should all item IRF be plotted? Defaults to FALSE
item.label	Should the item labels be plotted? Defaults to FALSE

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm   <- RM(Fake_Data_scores) # Estimate Rasch model

psk_data  <- pskettify(fake_rm)   # pskettify data
psketto_simple(psk_data, item = "i01")                       # single item IRF, no label
psketto_simple(psk_data, item = "i01", item.label = TRUE)    # single item IRF, labeled
psketto_simple(psk_data, item.label = TRUE, all.item = TRUE) # all item IRF labeled
```

---

pskett_msq *Plot a Infit and Outfit MSQ*

Description

This function plots the Infit and Outfit MSQ values from a item of class ItemFit. see 'item_fit_table()'. Based on Yu (2020).

Usage

```
psketti_msq(x, style = "present")
```

Arguments

x	input data, generated using 'item_fit_table()'.
style	a character string for plotting style options are present for coloured, or print for black and white. Defaults to "present".

Value

MSQ plot.

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm     <- RM(Fake_Data_scores)     # Estimate Rasch model

itemFit_psk <- item_fit_table(fake_rm)  # item fit stats
MSQplot     <- psketti_msq(itemFit_psk) # Plot infit and outfit

MSQplot
```

---

report_gen *Generate an Item Report*

Description

report_gen generates a containing folder for an Rmarkdown file, in the root of the project directory. After report_gen has been run, you can go to the containing folder and render the Rmarkdown file. Future versions will allow you to generate the Rmd file and render the html file in one go by selecting html_gen = TRUE. However, you will need to have tidyverse, knitr rmarkdown and kableExtra installed. This function is still undergoing active development.

Usage

```
report_gen(
  item.names,
  eRm.obj,
  data.long,
  Title,
  ptrn,
  ID,
  Item,
  K,
  r_o,
  html_gen = FALSE
)
```

Arguments

item.names	A charcter list of item names
eRm.obj	dichotomous Rasch Model; an eRm object of class 'RM'.
data.long	Long formatted data set with a column for Item names matching item.names, a column for response options, matching r_o, and an ID column name matching ID.
Title	the title of the Rmd report.
ptrn	the prefix label of the item namesL eg i01 is "i".
ID	A character object giving the name of the ID column in data.long
Item	A character object giving the name of the Item column in data.long
K	A character object giving the name of the response column in data.long
r_o	An ordered factor object to arrange columns along by response option order.
html_gen	Generate a html file from the report? Defaults to FALSE. Currently not functional.

Value

A folder containing materials for report generation

If html_gen = TRUE, a html report will be rendered

---

tabliatelle *Distractor Analysis Tables*

Description

Implementation of the tabular (Andrich and Styles, 2009) approach to assigning a partial credit scoring system to data previously modeled with a dichotomous Rasch model.

Usage

```
tabliatelle(x, ID, Item, K, response_options, eRm.obj)
```

Arguments

x	A long formatted dataframe.
ID	column name for ID column.
Item	column name for Item column.
K	column name for column containing multiple choice responses.
response_options	An ordered factor object to arrange column order in the distractor table.
eRm.obj	An object of class eRm and model RM. Use 'eRm::RM(score_data)' to create this object.

Value

tabliatelle returns a list of class tabliatelle.

Examples

```
library(eRm)
library(psketti)

data("FakeData") # load data
# restructure fake data
Fake_Data_scores <- reshape(FakeData[, c("ID", "Item", "X")],
                            timevar = "Item",
                            idvar = "ID",
                            direction = "wide")
# for eRm col names and row names
names(Fake_Data_scores) <- c("ID",
                             paste0("i",
                                    sprintf(fmt  = "%02d", 1:23)))

row.names(Fake_Data_scores) <- Fake_Data_scores$ID
Fake_Data_scores$ID         <- NULL

fake_rm   <- RM(Fake_Data_scores) # Estimate Rasch model

# Prepare response options factor
r_o <- factor(sort(unique(FakeData$K)),          # input var
              levels = sort(unique(FakeData$K)), # factor levels
              ordered = TRUE)                    # ordered
# tabliatellify
tlt_data <- tabliatelle(x = FakeData,
                        eRm.obj = fake_rm,
                        ID = "ID",
                        Item = "Item",
                        K = "K",
                        response_options = r_o)

tlt_data # output
```

Index