Package ‘GLMBasedRaschEstimation’

May 4, 2026

Title GLM-Based Estimation for Rasch Model Parameters

Version 0.2.0

Description Provides functions for estimating Rasch model parameters using the Generalized Linear Model (GLM) framework. The methods implemented are based on Brown (2018, ISBN:978-3-319-93547-8) <doi:10.1007/978-3-319-93549-2> and Debelak et al. (2022, ISBN:978-1-138-71046-7) <doi:10.1201/9781315200620>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports bslib, DT, graphics, haven, readxl, shiny, stats

URL `https://github.com/DrAhmedSamir/GLMBasedRaschEstimation`

BugReports `https://github.com/DrAhmedSamir/GLMBasedRaschEstimation/issues`

NeedsCompilation no

Author Ahmed Samir Megahed [aut, cre],
Mustafa Ali Khalaf [aut],
Ibraheem Mohamed Mougy [aut]

Maintainer Ahmed Samir Megahed <Asamir@foe.zu.edu.eg>

Repository CRAN

Date/Publication 2026-05-04 14:30:02 UTC

Contents

---

compute_Modified_probabilities
Compute Modified Item Response Probabilities

Description

Compute Modified Item Response Probabilities

Usage

```
compute_Modified_probabilities(results, theta_all)
```

Arguments

`results`	The data frame returned by fit_binary_irt.
`theta_all`	A numeric vector representing ability levels (total scores).

Value

A numeric matrix of predicted probabilities with dimensions `length(theta_all)` rows by `nrow(results)` columns. Each cell contains the probability of a correct response for a given ability level and item, calculated using the logistic function: $exp(\theta - \beta)/(1 + exp(\theta - \beta))$, where β is the item threshold. Columns are named after the items.

Examples

```
set.seed(123)
sample_data <- matrix(sample(c(0, 1), 100, replace = TRUE), ncol = 10)
prepared <- prepare_data(sample_data)
irt_results <- fit_binary_irt(prepared$matrix, prepared$total_score)

abilities <- seq(0, 10, length.out = 5)

probs <- compute_Modified_probabilities(irt_results, abilities)

print(head(probs))
```

`extract_rasch_difficulties_ordered`
Extract Rasch Item Difficulties in Original Order

Description

Extract Rasch Item Difficulties in Original Order

Usage

```
extract_rasch_difficulties_ordered(final_logit_matrix)
```

Arguments

`final_logit_matrix`
The matrix returned by rasch_logit.

Value

A data frame containing the calculated difficulty parameters for each item, with the following columns:

- `Item_Number`: The sequential index of the item.
- `Item_Name`: The name or label of the item.
- `Difficulty_Logit`: The item difficulty parameter (beta) expressed on the logit scale, rounded to four decimal places. Higher values indicate more difficult items.

Examples

```
set.seed(123)
sample_data <- matrix(sample(c(0, 1), 100, replace = TRUE), ncol = 10)
colnames(sample_data) <- paste0("Q", 1:10)
prepared <- prepare_data(sample_data)
irt_results <- fit_binary_irt(prepared$matrix, prepared$total_score)
probs <- compute_Modified_probabilities(irt_results, prepared$total_score)
logits <- rasch_logit(probs)

difficulty_table <- extract_rasch_difficulties_ordered(logits)

print(difficulty_table)
```

fit_binary_irt *Fit Binary IRT Model using GLM*

Description

Fit Binary IRT Model using GLM

Usage

```
fit_binary_irt(data_mat, total_score)
```

Arguments

`data_mat`	A numeric matrix of responses (persons in rows, items in columns).
`total_score`	A numeric vector of total scores for each person.

Value

A data frame with one row per item and the following columns:

- `Item`: The name of the item.
- `Intercept`: The estimated intercept parameter from the GLM logit model.
- `Slope`: The estimated slope parameter (discrimination) from the GLM logit model.
- `threshold`: The calculated item difficulty (also known as the beta parameter), computed as -Intercept / Slope. This represents the point on the ability scale where the probability of a correct response is 0.5.

Examples

```
set.seed(42)
sample_data <- matrix(sample(c(0, 1), 50, replace = TRUE), ncol = 5)
colnames(sample_data) <- paste0("Item", 1:5)

prepared <- prepare_data(sample_data)

irt_results <- fit_binary_irt(prepared$matrix, prepared$total_score)

print(irt_results)
```

plot_item_curves *Plot Item Characteristic Curves (ICC)*

Description

Plot Item Characteristic Curves (ICC)

Usage

```
plot_item_curves(theta_all, Modified_prob_matrix, results)
```

Arguments

theta_all	A numeric vector of ability levels.
Modified_prob_matrix	The matrix returned by compute_Modified_probabilities.
results	The data frame returned by fit_binary_irt.

Value

No return value, called for side effects. It generates a series of plots showing the Item Characteristic Curves (ICC) for each item in the model.

Examples

```
set.seed(123)
sample_data <- matrix(sample(c(0, 1), 100, replace = TRUE), ncol = 10)
prepared <- prepare_data(sample_data)
irt_results <- fit_binary_irt(prepared$matrix, prepared$total_score)

theta_range <- seq(0, 10, length.out = 100)
probs <- compute_Modified_probabilities(irt_results, theta_range)

# Plot the curves
plot_item_curves(theta_range, probs, irt_results)
```

plot_rasch_curves *Plot Rasch Item Curves*

Description

Plot Rasch Item Curves

Usage

```
plot_rasch_curves(prob_matrix, final_logit_matrix)
```

Arguments

prob_matrix — The matrix returned by compute_Modified_probabilities.

final_logit_matrix — The matrix returned by rasch_logit.

Value

No return value, called for side effects. It generates a series of plots showing the Rasch Item Characteristic Curves (ICC) with difficulty indicators.

Examples

```
set.seed(123)
sample_data <- matrix(sample(c(0, 1), 100, replace = TRUE), ncol = 10)
prepared <- prepare_data(sample_data)
irt_results <- fit_binary_irt(prepared$matrix, prepared$total_score)
probs <- compute_Modified_probabilities(irt_results, prepared$total_score)
logits <- rasch_logit(probs)

# Plot the Rasch curves
plot_rasch_curves(probs, logits)
```

---

prepare_data — *Prepare Data for IRT Analysis*

Description

Prepare Data for IRT Analysis

Usage

```
prepare_data(data)
```

Arguments

data — A data frame or matrix of binary responses (0 and 1).

Value

A list containing the following two components:

- `matrix`: A numeric matrix of the original data, converted to numeric format and potentially reordered by total score.
- `total_score`: A numeric vector representing the sum of correct responses (row sums) for each person, sorted in descending order if the original data was unsorted.

Examples

```
set.seed(123)
raw_data <- data.frame(
  item1 = c(1, 0, 1, 1),
  item2 = c(0, 0, 1, 1),
  item3 = c(1, 1, 1, 0)
)

prepared <- prepare_data(raw_data)

print(prepared$matrix)
print(prepared$total_score)
```

`rasch_logit` *Compute Logit and Row Means*

Description

Compute Logit and Row Means

Usage

```
rasch_logit(prob_matrix)
```

Arguments

`prob_matrix`	The matrix returned by compute_Modified_probabilities.

Value

A numeric matrix of logit transformations (log-odds) with dimensions `nrow(prob_matrix)` rows by `ncol(prob_matrix)` + 1 columns. The first columns contain the logit values for each item, and the final column, named `Row_Mean_Logit`, contains the mean logit across all items for each person, which serves as an estimate of Student Ability.

Examples

```
set.seed(123)
sample_data <- matrix(sample(c(0, 1), 100, replace = TRUE), ncol = 10)
prepared <- prepare_data(sample_data)
irt_results <- fit_binary_irt(prepared$matrix, prepared$total_score)
probs <- compute_Modified_probabilities(irt_results, prepared$total_score)

logit_results <- rasch_logit(probs)

print(head(logit_results))
```

`runGLMRaschSuite` *Run the GLM-Based Rasch Analysis Suite*

Description

This function launches the interactive Shiny application for Rasch analysis using the GLM framework.

Usage

```
runGLMRaschSuite()
```

Value

This function launches the Shiny app and does not return a value.

Index