

# Package: ivd (via r-universe)

October 18, 2026

**Type** Package

**Title** ``Individual Variance Detection"

**Version** 1.0.0.9000

**Maintainer** Philippe Rast <rast.ph@gmail.com>

**Description** Fit mixed-effects location scale models with spike-and-slab priors on the scale random effects to identify units with unusual residual variances. The method is described in detail in Carmo, Williams and Rast (2026) <doi:10.3102/10769986261426004>.

**License** GPL (>=3)

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**Imports** nimble (>= 1.1.0), coda (>= 0.19.4), ggplot2, patchwork, future, utils, rstan, ggrepel

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**BugReports** <https://github.com/consistentlyBetter/ivd/issues>

**Depends** R (>= 4.0.0)

**Config/roxygen2/version** 8.0.0

**Config/pak/sysreqs** cmake libgmp-dev make libxml2-dev

**Repository** <https://consistentlybetter.r-universe.dev>

**Date/Publication** 2026-07-20 11:53:23 UTC

**RemoteUrl** <https://github.com/consistentlybetter/ivd>

**RemoteRef** HEAD

**RemoteSha** 48c10cb525a8b386fadb493e6e882ba1a21172c9

Contents

|                                   |           |
|-----------------------------------|-----------|
| .require_suggest . . . . .        | 2         |
| build_ivd_model . . . . .         | 3         |
| codaplot . . . . .                | 4         |
| coef.ivd . . . . .                | 5         |
| fixef . . . . .                   | 5         |
| ivd . . . . .                     | 6         |
| pip . . . . .                     | 8         |
| pip_sensitivity . . . . .         | 9         |
| plot.ivd . . . . .                | 10        |
| plot.pip_sensitivity . . . . .    | 12        |
| print.ivd . . . . .               | 13        |
| print.summary.ivd . . . . .       | 13        |
| ranef . . . . .                   | 14        |
| run_MCMC_compiled_model . . . . . | 15        |
| saeb . . . . .                    | 16        |
| summary.ivd . . . . .             | 17        |
| <b>Index</b>                      | <b>19</b> |

---

|                  |                                              |
|------------------|----------------------------------------------|
| .require_suggest | <i>Helper to check for suggested package</i> |
|------------------|----------------------------------------------|

---

Description

Helper to check for suggested package

Usage

.require\_suggest(pkg, feature)

Arguments

|         |                   |
|---------|-------------------|
| pkg     | requested package |
| feature | requested feature |

Author(s)

philippe

---

|                 |                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| build_ivd_model | <i>Build and compile NIMBLE model and MCMC once This function is exported for use in 'future' and is not meant to be called by user.</i> |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|

---

## Description

Build and compile NIMBLE model and MCMC once This function is exported for use in 'future' and is not meant to be called by user.

## Usage

```
build_ivd_model(
  code,
  constants,
  dummy_data,
  dummy_inits,
  useWAIC = TRUE,
  monitor_pointwise = FALSE
)
```

## Arguments

|                   |                                                                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| code              | Nimble code                                                                                                                                                                                                                            |
| constants         | Constants                                                                                                                                                                                                                              |
| dummy_data        | Data                                                                                                                                                                                                                                   |
| dummy_inits       | inits                                                                                                                                                                                                                                  |
| useWAIC           | Defaults to TRUE. Nimble argument                                                                                                                                                                                                      |
| monitor_pointwise | Also monitor the per-observation $\mu$ and $\tau$ . Defaults to FALSE; only needed to reconstruct the pointwise log-likelihood. Monitoring them costs $O(N \times \text{iterations})$ RAM per chain, so they are off unless requested. |

## Value

A named list with two elements:

- `cmodel`: The compiled NIMBLE model object produced by `compileNimble()`.
- `cmcmc`: The compiled NIMBLE MCMC object, created using `buildMCMC()` and `compileNimble()`, configured to monitor the model parameters (including WAIC monitors if `useWAIC = TRUE`).

The function is intended for internal use (e.g., within parallel workers) and is not meant to be called directly by end users.

**Examples**

```
## Not run:
library(nimble)
# Generic nimble example
code <- nimbleCode({
  mu ~ dnorm(0, 1)
  x ~ dnorm(mu, 1)
})

constants <- list()
dummy_data <- list(x = 0)
dummy_inits <- list(mu = 0)

out <- build_ivd_model(
  code = code,
  constants = constants,
  dummy_data = dummy_data,
  dummy_inits = dummy_inits,
  useWAIC = FALSE
)

str(out)

## End(Not run)
```

---

codaplot

*Traceplot from the coda package*


---

**Description**

For more plots see coda

**Usage**

```
codaplot(obj, parameters = NULL, type = "traceplot", askNewPage = TRUE)
```

**Arguments**

|            |                                                                                                                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| obj        | ivd object                                                                                                                                                                                    |
| parameters | Provide parameters of interest using names from the summary() output (e.g., "Intc", "scl_Intc", "sd_Intc", "R\[scl_Intc, Intc\]", "pip\[Intc, 5\]"). Defaults to NULL (plots all parameters). |
| type       | Coda plot. Defaults to 'traceplot'. See coda for more options such as 'acfplot', 'densplot' etc.                                                                                              |
| askNewPage | Should user be prompted for next plot. Defaults to TRUE                                                                                                                                       |

**Value**

Specified coda plot

**Author(s)**

Philippe Rast

---

coef.ivd*Extract coefficients from a fitted ivd model*

---

**Description**

Extract coefficients

**Usage**

```
## S3 method for class 'ivd'  
coef(object, ...)
```

**Arguments**

|        |                         |
|--------|-------------------------|
| object | An object of class ivd. |
| ...    | Not used.               |

**Details**Convenience alias for [fixef\(\)](#): posterior means of the fixed location and scale effects.**Value**

Named numeric vector of posterior means.

**Author(s)**

Philippe Rast

---

fixef*Extract fixed effects from a fitted ivd model*

---

**Description**

Extract fixed effects

**Usage**

```
fixef(object, ...)  
  
## S3 method for class 'ivd'  
fixef(object, ...)
```

**Arguments**

|        |                                       |
|--------|---------------------------------------|
| object | An object of class <code>ivd</code> . |
| ...    | Not used.                             |

**Details**

Posterior means of the fixed effects, named after the model variables. Location effects keep their variable names; scale effects are prefixed with `sc1_`, matching `summary.ivd()`.

The generic is defined here so that `ivd` does not depend on `nlme/lme4`; if either package is attached, its same-named generic dispatches to this method just the same.

**Value**

Named numeric vector of posterior means.

**Author(s)**

Philippe Rast

---

|                  |                                                                                                                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ivd</code> | <i>Main function to set up and run parallel MCMC using nimble and future. <code>ivd</code> computes a mixed effects location and scale model with Spike and Slab regularization on the scale random effects.</i> |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Description**

Main function to set up and run parallel MCMC using nimble and future. `ivd` computes a mixed effects location and scale model with Spike and Slab regularization on the scale random effects.

**Usage**

```
ivd(
  location_formula,
  scale_formula,
  data,
  niter,
  nburnin = NULL,
  WAIC = TRUE,
  workers = 4,
  n_eff = "local",
  ss_prior_p = 0.5,
  thin = 1,
  return_logLik = FALSE,
  seed = NULL,
  progress = interactive(),
  ...
)
```

**Arguments**

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>location_formula</code> | A formula for the location model                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>scale_formula</code>    | A formula for the scale model                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>data</code>             | Data frame in long format for analysis                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>niter</code>            | Total number of MCMC iterations after burnin                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>nburnin</code>          | Number of burnin iterations, defaults to the same as <code>niter</code>                                                                                                                                                                                                                                                                                                                                                            |
| <code>WAIC</code>             | Compute WAIC, defaults to 'TRUE'                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>workers</code>          | Number of parallel R processes – doubles as 'chains' argument                                                                                                                                                                                                                                                                                                                                                                      |
| <code>n_eff</code>            | Use <code>stan::monitor</code> function or built local: 'stan' vs. 'local'                                                                                                                                                                                                                                                                                                                                                         |
| <code>ss_prior_p</code>       | Prior inclusion probability. Defaults to '.5'.                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>thin</code>             | Thinning interval for stored posterior draws. Defaults to 1 (keep every iteration). Larger values cut stored-sample RAM linearly.                                                                                                                                                                                                                                                                                                  |
| <code>return_logLik</code>    | Store the pointwise log-likelihood array (iterations x chains x N) for use with e.g. <code>loo</code> . Defaults to FALSE. When TRUE, <code>tau</code> is also monitored. The array scales with N and is the single largest element of the returned object, so it is opt-in.                                                                                                                                                       |
| <code>seed</code>             | Optional integer for full reproducibility. When supplied, it seeds both the random initial values and a distinct per-chain MCMC seed, so repeated calls return identical draws without needing an external <code>set.seed()</code> . Defaults to NULL (inits drawn from the ambient RNG; chains seeded 1:workers – the previous behaviour).                                                                                        |
| <code>progress</code>         | Show a live, per-chain progress line while the chains compile and sample, and suppress NIMBLE's (buffered) per-worker console output. Defaults to <code>interactive()</code> . Set to FALSE to restore NIMBLE's verbose model-building output and disable the progress line. Progress granularity is per chain: each tick marks a chain finishing (compile + sample), since multisession workers cannot stream sub-chain progress. |
| <code>...</code>              | Currently not used                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Value**

An object of class "ivd" (and "list"), which contains the results from fitting a mixed-effects location-scale model with Spike-and-Slab regularization using NIMBLE and parallel MCMC sampling.

The returned object is a named list with the following components:

- `samples`: An `mcmc.list` object containing posterior samples for all monitored parameters across all chains.
- `logLik_array`: Only present when `return_logLik = TRUE`. A 3D array of pointwise log-likelihood values with dimensions iterations × chains × N.
- `rhat_values`: Vector of split- $\hat{R}$  convergence diagnostics (Vehtari et al., 2021).
- `n_eff`: Vector of effective sample sizes, either computed internally ("local") or via `stan::monitor()` ("stan").

- `nimble_constants`: List of model constants used by the underlying NIMBLE model (e.g., number of groups, number of parameters).
- `X_location_names`, `Z_location_names`: Names of fixed and random effects in the location submodel.
- `X_scale`, `Z_scale`: Matrices used for the scale submodel's fixed and random effects.
- `X`, `Z`: Location-submodel fixed/random design matrices, retained so the outcome plot can re-construct the posterior mean of  $\mu$ .
- `Y`: Data frame with the response vector and group identifiers.
- `group_labels`: Character vector mapping the internal cluster index  $j$  back to the user's original grouping IDs.
- `location_formula`, `scale_formula`: The model formulas.
- `ss_prior_p`: The prior inclusion probability used in the fit.
- `workers`: Number of parallel chains used.
- ...: Additional elements created internally and used for downstream S3 methods (`print()`, `summary()`, etc.).

The object is designed to support S3 methods for printing, summarizing, and extracting results from the `ivd` model.

## Examples

```
out <- ivd(location_formula = math_proficiency ~ 1 + (1 | school_id),
  scale_formula = ~ 1 + (1 | school_id),
  data = saeb,
  niter = 1000,
  nburnin = 2000,
  WAIC = TRUE,
  workers = 1) ## Workers = 1 for CRAN server - not ideal for individual use
## Posterior inclusion probability plot (PIP)
plot(out, type = "pip")
## PIP vs. Within-cluster SD
plot(out, type = "funnel")
## Diagnostic plots based on coda plots:
library(coda)
codaplot(out, parameters = "Intc")
codaplot(out, parameters = "R[scl_Intc, Intc]")
```

---

`pip`

*Extract PIPs from a fitted ivd model*

---

## Description

Extract posterior inclusion probabilities

**Usage**

```
pip(object, ...)

## S3 method for class 'ivd'
pip(object, ...)
```

**Arguments**

`object`            An object of class `ivd`.  
`...`                Not used.

**Details**

Returns the package's headline output – the posterior inclusion probability (PIP) of every cluster's scale random effect(s) – as a tidy data frame, one row per cluster and scale random effect.

**Value**

A `data.frame` with one row per cluster x scale random effect:

- `scale_var`: Name of the scale random effect.
- `cluster_index`: Internal cluster index 1..J (matches the default labels in `summary.ivd()` and `plot.ivd()`).
- `cluster_id`: The user's original grouping ID, parsed back to numeric when the original IDs were numeric (equals `cluster_index` for objects fitted before labels were stored).
- `pip`: Posterior inclusion probability of the cluster's scale random effect.
- `u_mean`, `u_sd`: Posterior mean and SD of the scale random effect `u`.

**Author(s)**

Philippe Rast

---

|                              |                                            |
|------------------------------|--------------------------------------------|
| <code>pip_sensitivity</code> | <i>Analytic prior-sensitivity for PIPs</i> |
|------------------------------|--------------------------------------------|

---

**Description**

Prior-sensitivity of the posterior inclusion probabilities

**Usage**

```
pip_sensitivity(fit, prior_p = seq(0.05, 0.95, by = 0.05))
```

**Arguments**

|         |                                                                                                                     |
|---------|---------------------------------------------------------------------------------------------------------------------|
| fit     | A fitted ivd object.                                                                                                |
| prior_p | Numeric vector of prior inclusion probabilities to evaluate, all in (0, 1). Defaults to seq(0.05, 0.95, by = 0.05). |

**Details**

Shows how each cluster's PIP would change under different prior inclusion probabilities `ss_prior_p`, computed analytically from the fitted model – no refitting. The posterior odds are converted to a prior-independent Bayes factor and re-weighted to each value of `prior_p` (see [.pip\\_reweight\(\)](#)).

PIPs estimated as exactly 0 or 1 are a finite-sample artifact (with  $S$  post-warmup draws a PIP of 1 is only known to be at least  $(S-0.5)/S$ ), so they are clamped to  $0.5/S$  and  $(S-0.5)/S$  before re-weighting; the `at_boundary` column flags those rows. All other PIPs are used as estimated.

**Value**

A data frame of class `pip_sensitivity` with one row per cluster x scale random effect x `prior_p` value: `scale_var`, `cluster_index`, `cluster_id`, `prior_p`, `pip`, and `at_boundary`. The prior used in the fit is stored in `attr(, "p0")`. Plot with [plot.pip\\_sensitivity\(\)](#).

**Author(s)**

Philippe Rast

**Examples**

```
## Not run:
out <- ivd(y ~ 1 + (1 | school), ~ 1 + (1 | school), data, niter = 2000)
sens <- pip_sensitivity(out)
plot(sens)

## End(Not run)
```

---

plot.ivd

---

*Plot method for ivd objects*


---

**Description**

Plot method for ivd objects

**Usage**

```
## S3 method for class 'ivd'
plot(
  x,
  type = "pip",
  pip_level = 0.75,
```

```

    variable = NULL,
    label_points = TRUE,
    labels = c("index", "original"),
    ...
  )

```

## Arguments

|                           |                                                                                                                                                                                                                               |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>            | An object of type <code>ivd</code> .                                                                                                                                                                                          |
| <code>type</code>         | Defaults to 'pip', other options are 'funnel' and 'outcome'.                                                                                                                                                                  |
| <code>pip_level</code>    | Defines a value for the posterior inclusion probability. Defaults to 0.75.                                                                                                                                                    |
| <code>variable</code>     | Name of a specific variable. Defaults to <code>NULL</code>                                                                                                                                                                    |
| <code>label_points</code> | Should points above the pip threshold be labelled? Defaults to <code>TRUE</code> .                                                                                                                                            |
| <code>labels</code>       | Point labels: "index" (default) uses the compact internal 1..J cluster index; "original" uses the user's own grouping IDs (see <code>fit\$group_labels</code> ). Matches the same argument in <a href="#">summary.ivd()</a> . |
| <code>...</code>          | Controls <code>ggrepel</code> arguments.                                                                                                                                                                                      |

## Value

Invisibly returns a `ggplot` object corresponding to the selected plot type. The primary purpose of this method is the side effect of displaying the plot.

The exact plot depends on the value of `type`:

- "pip" — Posterior inclusion probability plot for random scale effects.
- "funnel" — Funnel plot showing the relation between within-cluster standard deviation ( $\tau$ ) and posterior inclusion probabilities.
- "outcome" — Outcome plot relating cluster means ( $\mu$ ), posterior inclusion probability, and within-cluster SD.

When `label_points = TRUE`, labels for clusters exceeding the `pip_level` threshold are added using **ggrepel** (if available).

## Author(s)

Philippe Rast

---

plot.pip\_sensitivity *Plot method for pip\_sensitivity objects*


---

### Description

Plot PIP trajectories across prior inclusion probabilities

### Usage

```
## S3 method for class 'pip_sensitivity'
plot(x, pip_level = 0.75, clusters = NULL, odds_factor = 2, ...)
```

### Arguments

|             |                                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x           | A pip_sensitivity object from <a href="#">pip_sensitivity()</a> .                                                                                           |
| pip_level   | PIP threshold used to judge whether a cluster's classification is prior-sensitive. Defaults to 0.75.                                                        |
| clusters    | Optional vector selecting which clusters to draw, matched against cluster_id (the original grouping IDs) or cluster_index. Defaults to NULL (all clusters). |
| odds_factor | Width of the prior window used to judge sensitivity, as a multiplicative factor on the fitted prior's odds. Defaults to 2.                                  |
| ...         | Not used.                                                                                                                                                   |

### Details

One line per cluster (faceted by scale random effect when there is more than one). By default all clusters are shown: those whose classification at pip\_level is prior-sensitive are drawn in color and labelled, robust clusters are grey. With clusters, only the requested clusters are drawn, each colored and labelled – useful for inspecting individual clusters or small groups. The dashed vertical line marks the prior used in the fit.

As the prior inclusion probability approaches 0 or 1 every PIP follows it, so crossing the threshold at extreme priors is expected and carries no information. A cluster is therefore flagged as prior-sensitive only when its classification changes within a *plausible* perturbation of the fitted prior: priors whose odds lie within a factor odds\_factor of the fitted prior's odds (the default, 2, spans halving to doubling them – for ss\_prior\_p = 0.5 that is priors between 1/3 and 2/3). Because the re-weighted PIP is monotone in the prior, this is evaluated analytically at the two ends of that window, independently of the plotted prior\_p grid.

### Value

A ggplot object.

### Author(s)

Philippe Rast

---

|           |                                     |
|-----------|-------------------------------------|
| print.ivd | <i>Print method for ivd objects</i> |
|-----------|-------------------------------------|

---

**Description**

Print a concise overview of a fitted ivd model

**Usage**

```
## S3 method for class 'ivd'  
print(x, ...)
```

**Arguments**

|     |                         |
|-----|-------------------------|
| x   | An object of class ivd. |
| ... | Not used.               |

**Details**

Shows the model formulas, data dimensions, sampling setup and a one-line convergence note. Use [summary.ivd\(\)](#) for the full posterior table and [pip\(\)](#) for a tidy data frame of the posterior inclusion probabilities.

**Value**

x, invisibly.

**Author(s)**

Philippe Rast

---

|                   |                                             |
|-------------------|---------------------------------------------|
| print.summary.ivd | <i>Print method for summary.ivd objects</i> |
|-------------------|---------------------------------------------|

---

**Description**

Print the posterior summary assembled by [summary.ivd\(\)](#)

**Usage**

```
## S3 method for class 'summary.ivd'  
print(x, ...)
```

**Arguments**

|     |                       |
|-----|-----------------------|
| x   | A summary.ivd object. |
| ... | Not used.             |

**Value**

x, invisibly.

**Author(s)**

Philippe Rast

---

|       |                                                       |
|-------|-------------------------------------------------------|
| ranef | <i>Extract random effects from a fitted ivd model</i> |
|-------|-------------------------------------------------------|

---

**Description**

Extract random effects

**Usage**

```
ranef(object, ...)
```

```
## S3 method for class 'ivd'
ranef(object, labels = c("index", "original"), ...)
```

**Arguments**

|        |                                                                                                                      |
|--------|----------------------------------------------------------------------------------------------------------------------|
| object | An object of class ivd.                                                                                              |
| ...    | Not used.                                                                                                            |
| labels | Row names: "index" (default) uses the internal 1..J cluster index; "original" uses the user's original grouping IDs. |

**Details**

Posterior means of the random effects u, one row per cluster and one column per random effect. Location random effects come first, then the scale random effects (prefixed with scl\_). Note that the scale columns report the unconditional u; multiply by the inclusion indicators (see [pip\(\)](#)) to reproduce the model-implied, spike-and-slab-selected effect.

The generic is defined here so that ivd does not depend on nlme/lme4; if either package is attached, its same-named generic dispatches to this method just the same.

**Value**

Numeric matrix (J x (Kr + Sr)) of posterior means.

**Author(s)**

Philippe Rast

---

run\_MCMC\_compiled\_model

*Run MCMC on an already compiled model Exposed but internal function for future()*

---

## Description

Run MCMC on an already compiled model Exposed but internal function for future()

## Usage

```
run_MCMC_compiled_model(
  compiled,
  seed,
  new_data,
  new_inits,
  niter,
  nburnin,
  useWAIC = TRUE,
  thin = 1,
  ...
)
```

## Arguments

|           |                                                       |
|-----------|-------------------------------------------------------|
| compiled  | Compiled nimble model                                 |
| seed      | Seed, set by future                                   |
| new_data  | Data                                                  |
| new_inits | inits                                                 |
| niter     | Sampling iterations                                   |
| nburnin   | Number of burnin iterations                           |
| useWAIC   | Defaults to TRUE                                      |
| thin      | Thinning interval passed to runMCMC(). Defaults to 1. |
| ...       | Placeholder for nimble arguments                      |

## Value

The output produced by `nimble::runMCMC()` when applied to a compiled NIMBLE MCMC object. The returned value depends on the `useWAIC` argument:

- If `useWAIC = TRUE`, a named list containing:
  - `samples`: A matrix of posterior draws (iterations  $\times$  parameters).
  - `WAIC`: The WAIC value computed by NIMBLE.
  - ...: Additional elements returned by `runMCMC()` when WAIC is enabled.

- If `useWAIC = FALSE`, a numeric matrix containing the posterior samples (iterations  $\times$  parameters) with no additional elements.

This function is intended for internal use (e.g., within future workers) and is not meant to be called directly by end users.

## Examples

```
## Not run:
library(nimble)
# Generic nimble example
code <- nimbleCode({
  mu ~ dnorm(0, 1)
  x ~ dnorm(mu, 1)
})

constants <- list()
dummy_data <- list(x = 0)
dummy_inits <- list(mu = 0)

out <- build_ivd_model(
  code = code,
  constants = constants,
  dummy_data = dummy_data,
  dummy_inits = dummy_inits,
  useWAIC = FALSE
)

str(out)

## End(Not run)
```

---

saeb

*Basic Education Evaluation System (Saeb) -2021*


---

## Description

The Basic Education Evaluation System (Saeb) is a series of large-scale external assessments conducted by Inep (National Institute for Educational Studies and Research) to diagnose the state of basic education in Brazil and identify factors that may affect student performance. This dataset contains the standardized math scores of 12th graders from 160 randomly selected schools in Rio de Janeiro who took the 2021 exam.

## Usage

saeb

**Format**

A data frame with 11386 student's observations including 5 variables:

**school\_id** A unique identifier for each school in the dataset.

**public** A binary variable indicating the type of school. It takes a value of 1 if the school is public and 0 if the school is private.

**student\_ses** A numerical variable representing the socioeconomic status (SES) of the students.

**math\_proficiency** A numerical variable representing the math proficiency level of the students, standardized with a mean of 0 and a standard deviation of 1.

**location** A numerical variable indicating the geographical location of the school. It takes a value of 1 for urban schools and 2 for rural schools.

**Source**

<https://web.archive.org/web/20250202015037/https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/saeb/resultados>

---

|             |                                     |
|-------------|-------------------------------------|
| summary.ivd | <i>Summary of posterior samples</i> |
|-------------|-------------------------------------|

---

**Description**

Summarize ivd object

**Usage**

```
## S3 method for class 'ivd'
summary(object, digits = 3, pip = "all", labels = c("index", "original"), ...)
```

**Arguments**

|        |                                                                                                                                                                                                                |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object | ivd object                                                                                                                                                                                                     |
| digits | Integer (Default: 2, optional). Number of digits to round to when printing.                                                                                                                                    |
| pip    | Print pip and model parameters ('all'); Only pip ('pip'), or only model parameters ('model'). Defaults to 'all'                                                                                                |
| labels | Cluster labels in the PIP rows: "index" (default) uses the internal 1..J index; "original" uses the user's own grouping IDs (see fit\$group_labels). Matches the same argument in <a href="#">plot.ivd()</a> . |
| ...    | Not used                                                                                                                                                                                                       |

**Value**

A summary.ivd object: a list with the posterior summary matrix (\$table), the pip filter used (\$pip), the number of chains (\$chains), and the chain-averaged WAIC metrics (\$waic, \$lppd, \$pwaic, with \$has\_waic indicating availability). Printed by [print.summary.ivd\(\)](#).

**Author(s)**

Philippe Rast

# Index

- \* **datasets**
  - saeb, [16](#)
  - .pip\_reweight(), [10](#)
  - .require\_suggest, [2](#)
- build\_ivd\_model, [3](#)
- codaplot, [4](#)
- coef.ivd, [5](#)
- fixef, [5](#)
- fixef(), [5](#)
- ivd, [6](#)
- pip, [8](#)
- pip(), [13](#), [14](#)
- pip\_sensitivity, [9](#)
- pip\_sensitivity(), [12](#)
- plot.ivd, [10](#)
- plot.ivd(), [9](#), [17](#)
- plot.pip\_sensitivity, [12](#)
- plot.pip\_sensitivity(), [10](#)
- print.ivd, [13](#)
- print.summary.ivd, [13](#)
- print.summary.ivd(), [17](#)
- ranef, [14](#)
- run\_MCMC\_compiled\_model, [15](#)
- saeb, [16](#)
- summary.ivd, [17](#)
- summary.ivd(), [6](#), [9](#), [11](#), [13](#)