Package: simplexgof (via r-universe)

June 20, 2026

Type Package

Title Bootstrap-Calibrated Goodness-of-Fit Test for Simplex Regression

Version 0.1.0

Date 2026-06-12

Description Implements the bootstrap-calibrated local-influence goodness-of-fit test for simplex regression models with constant or varying dispersion, following the local influence approach of Zhu and Zhang (2004) <doi:10.1093/biomet/91.3.579> and the simplex regression model of Barndorff-Nielsen and Jorgensen (1991) <doi:10.1016/0047-259X(91)90008-P>. The test statistic aggregates individual local-influence measures under case-weight perturbation. Because the first-order asymptotic normal calibration is severely liberal in finite samples, a parametric bootstrap calibration is provided that restores accurate size control and delivers high power against omitted covariates, neglected dispersion, and distributional misspecification. Plotting functions reproduce the figures and tables of the companion methodological paper. Computational kernels are implemented in 'C++' via 'Rcpp' and 'RcppArmadillo' for speed, and two real datasets are bundled.

License GPL-3

URL `https://github.com/Raydonal/simplexgof`, `https://raydonal.github.io/simplexgof/`

BugReports `https://github.com/Raydonal/simplexgof/issues`

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.0), parallel, stats, graphics, grDevices

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

NeedsCompilation yes

Author Raydonal Ospina [aut, cre]
(<`https://orcid.org/0000-0002-9884-9090`>)

Maintainer Raydonal Ospina `<raydonal@de.ufpe.br>`

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-06-13 00:00:28 UTC

RemoteUrl https://github.com/raydonal/simplexgof

RemoteRef HEAD

RemoteSha 8b96e5237e334d8bfa8fb98c69cc7167621fd0b3

Contents

`ammonia` *Ammonia Oxidation Data (Brownlee, 1965)*

Description

Data from an industrial ammonia oxidation process recorded over 21 days (Brownlee, 1965, p. 45). The response is the proportion of ammonia not converted to nitric acid; three process variables are included.

Usage

```
ammonia
```

Format

A data frame with 21 rows and 4 columns:

`perda` Proportion of ammonia loss (response), in (0, 1).

`corr_ar` Air flow rate (arbitrary units).

`temp_agua` Cooling water inlet temperature (degrees Celsius).

`conc_acido` Nitric acid concentration (percent).

Details

The model selected in Espinheira et al. (2026) is:

$$\text{logit}(\mu_t) = \beta_1 + \beta_2 x_{t2} + \beta_3 x_{t3} + \beta_4(x_{t2} \times x_{t3})$$

$$\log(\sigma_t^2) = \gamma_1 + \gamma_2 x_{t3} + \gamma_3(x_{t2} \times x_{t3})$$

where x_{t2} = `corr_ar`, x_{t3} = `temp_agua`. The variable `conc_acido` is included in the dataset for completeness but was not used in the selected model.

Source

Brownlee, K.A. (1965). *Statistical Theory and Methodology in Science and Engineering*, 2nd ed. Wiley, New York, p. 45.

References

Espinheira P.L., Silva F.C., Barros M., Ospina R. (2026). A Bootstrap-Calibrated Local Influence Goodness-of-Fit Procedure for Simplex Regression Models.

Examples

```
data(ammonia)
str(ammonia)
with(ammonia, plot(corr_ar, perda, xlab = "Air flow", ylab = "Ammonia loss"))
```

`paper_ammonia` *Reproduce Ammonia Application (Paper Section 7.1)*

Description

Fits the simplex regression model to the Brownlee (1965) ammonia-oxidation data, runs the bootstrap U_n test, and optionally produces the influence index plot and the half-normal envelope plot, reproducing Section 7.1 and Tables 5–6 of Ospina et al. (2026).

Usage

```
paper_ammonia(B = 1000, seed = 123, plot = TRUE, verbose = TRUE)
```

Arguments

`B`	Integer; bootstrap replicates. Default 1000.
`seed`	Integer; random seed for reproducibility. Default 123.
`plot`	Logical; whether to produce the two diagnostic plots. Default `TRUE`.
`verbose`	Logical; whether to print progress. Default `TRUE`.

Value

A list (invisibly) with components:

`fit` The "`simplexfit`" object.

`gof` The "`simplexgof`" object.

`diag` The `simplex_diag()` output.

`table_params` Data frame of parameter estimates (Table 5).

`table_gof` Data frame of GoF test results (Table 6).

See Also

`paper_pbsc`, `simplex_gof`

Examples

```
res <- paper_ammonia(B = 200, seed = 123)  # B = 1000 in the paper
print(res$table_params)
print(res$table_gof)
```

`paper_fig1` *Reproduce Figure 1: Null Distribution of U_n*

Description

Reproduces Figure 1 of Ospina et al. (2026): QQ-plots and histograms of the asymptotic U_n statistic against the standard normal, for three ranges of μ and two dispersion levels ($\sigma^2 \in \{0.5, 16\}$). Also returns the table of characteristic measures (mean, variance, kurtosis, skewness).

Usage

```
paper_fig1(n = 40, R = 1000, sigma2 = c(0.5, 16), seed = 185, plot = TRUE)
```

Arguments

`n`	Sample size. Default 40.
`R`	Number of Monte Carlo replications. Default 1000.
`sigma2`	Dispersion values to study. Default `c(0.5, 16)`.
`seed`	Random seed for the (fixed) covariate design and the MC loop. Default 185 (chosen to match the μ ranges in the paper).
`plot`	Logical; produce the QQ and histogram panels. Default `TRUE`.

Details

The true β vectors are those of Table 1 of the paper, chosen so that the fitted means fall in $(0.019, 0.147)$, $(0.205, 0.886)$ and $(0.903, 0.995)$. Covariates are $x_{ti} \sim U(0, 1)$, generated once and held fixed.

Value

Invisibly, a list with `Un` (named list of U_n vectors) and `measures` (data frame of characteristic measures).

See Also

`simplex_Un_asymptotic`, `paper_ammonia`

Examples

```
res <- paper_fig1(n = 40, R = 200)   # R = 1000 in the paper
print(res$measures)
```

`paper_pbsc` *Reproduce PBSC Application (Paper Section 7.2)*

Description

Fits the simplex regression model to the PBSC transplant dataset (Edmonton Hematopoietic Institute), runs the bootstrap U_n test, and optionally produces diagnostic plots, reproducing Section 7.2 and Tables 7–8 of Ospina et al. (2026).

Usage

```
paper_pbsc(B = 1000, seed = 456, plot = TRUE, verbose = TRUE)
```

Arguments

`B`	Integer; bootstrap replicates. Default 1000.
`seed`	Integer; random seed. Default 456.
`plot`	Logical; whether to produce diagnostic plots. Default `TRUE`.
`verbose`	Logical; print progress. Default `TRUE`.

Value

A list (invisibly) with components `fit`, `gof`, `diag`, `table_params`, `table_gof`.

See Also

`paper_ammonia`, `simplex_gof`

Examples

```
res <- paper_pbsc(B = 200, seed = 456)
print(res$table_params)
```

`pbsc` *Peripheral Blood Stem Cell (PBSC) Transplant Data*

Description

Data from a study of 242 patients at the Edmonton Hematopoietic Institute - Alberta Health Services (Espinheira et al., 2026). The response is the CD34+ cell recovery rate (ratio of viable CD34+ cells post-thaw to pre-freeze), a continuous proportion in (0, 1).

Usage

```
pbsc
```

Format

A data frame with 242 rows and 3 columns:

`recovery` CD34+ recovery rate (response), in (0, 1).

`adj_age` Adjusted patient-age variable.

`chemo` Chemotherapy protocol indicator: 0 = one-day protocol, 1 = three-day protocol.

Details

The model selected in Espinheira et al. (2026) is:

$$\mathrm{logit}(\mu_t) = \beta_1 + \beta_2 x_{t2} + \beta_3 x_{t3}$$

$$\log(\sigma_t^2) = \gamma_1 + \gamma_2 x_{t2} + \gamma_3 x_{t1}$$

where x_{t1} = `chemo` and x_{t2} = x_{t3} = `adj_age`.

Source

Edmonton Hematopoietic Institute - Alberta Health Services. Used under the data-sharing policy of that institution.

References

Espinheira P.L., Silva F.C., Barros M., Ospina R. (2026). A Bootstrap-Calibrated Local Influence Goodness-of-Fit Procedure for Simplex Regression Models.

Examples

```
data(pbsc)
str(pbsc)
hist(pbsc$recovery, breaks = 30, main = "CD34+ Recovery Rate")
```

`plot.simplexfit` *Diagnostic Plots for a Fitted Simplex Regression Model*

Description

`plot` method for objects of class `"simplexfit"`. Produces influence index plots and/or a half-normal plot with simulated envelope.

Usage

```
## S3 method for class 'simplexfit'
plot(
  x,
  which = c("influence", "envelope"),
  ask = length(which) > 1 && dev.interactive(),
  ...
)
```

Arguments

`x`	An object of class `"simplexfit"` returned by `simplex_fit`.
`which`	Character vector indicating which plots to produce: `"influence"` for the influence index plot (see `plot_influence`), and/or `"envelope"` for the half-normal plot with simulated envelope (see `plot_envelope`). Several can be requested at once.
`ask`	Logical; if `TRUE` and more than one plot is requested, the user is asked before each new plot. Defaults to `length(which) > 1 && dev.interactive()`.
`...`	Further arguments passed to `plot_influence` or `plot_envelope`.

Value

The object `x`, invisibly.

See Also

`simplex_fit`, `plot_influence`, `plot_envelope`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
fit <- simplex_fit(ammonia$perda, X, Z)
plot(fit, which = "influence")
```

`plot.simplexgof` *Plots for a Bootstrap Goodness-of-Fit Test Result*

Description

`plot` method for objects of class `"simplexgof"`. Produces the bootstrap distribution of U_n and/or an influence index plot.

Usage

```
## S3 method for class 'simplexgof'
plot(
  x,
  which = c("boot", "influence"),
  ask = length(which) > 1 && dev.interactive(),
  ...
)
```

Arguments

`x`	An object of class `"simplexgof"` returned by `simplex_gof`.
`which`	Character vector indicating which plots to produce: `"boot"` for the bootstrap distribution of U_n (see `plot_gof_boot`), and/or `"influence"` for the influence index plot (see `plot_influence`). Several can be requested at once.
`ask`	Logical; if `TRUE` and more than one plot is requested, the user is asked before each new plot. Defaults to `length(which) > 1 && dev.interactive()`.
`...`	Further arguments passed to `plot_gof_boot` or `plot_influence`.

Value

The object `x`, invisibly.

See Also

`simplex_gof`, `plot_gof_boot`, `plot_influence`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
set.seed(42)
gof <- simplex_gof(ammonia$perda, X, Z, B = 50, verbose = FALSE)
plot(gof, which = "boot")
```

`plot_envelope` *Half-Normal Probability Plot with Simulated Envelope*

Description

Produces a half-normal plot (Atkinson, 1985) with a simulated envelope of absolute deviance residuals with a bootstrap envelope, used to assess the overall fit of a simplex regression model. This replicates the diagnostic plots in Ospina et al. (2026).

Usage

```
plot_envelope(
  fit,
  B = 99,
  conf = 0.95,
  col.obs = "#2c7bb6",
  col.env = "#abd9e9",
  main = "Half-normal plot with bootstrap envelope",
  xlab = "Half-normal quantile",
  ylab = "Absolute deviance residual",
  ...
)
```

Arguments

`fit`	A `"simplexfit"` object.
`B`	Integer; number of simulations for the envelope. Default 99.
`conf`	Numeric in (0,1); confidence level for the envelope. Default 0.95.
`col.obs`	Colour for the observed residuals.
`col.env`	Colour for the envelope band.
`main`, `xlab`, `ylab`	Plot labels.
`...`	Further arguments to `plot`.

Value

Invisibly returns a list with `$residuals` (observed) and `$envelope` (matrix of simulated residuals).

References

Atkinson A.C. (1985). *Plots, Transformations, and Regression*. Oxford University Press.

See Also

`simplex_fit`, `plot_influence`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
fit <- simplex_fit(ammonia$perda, X, Z)
plot_envelope(fit, B = 99)
```

`plot_gof_boot` *Plot Bootstrap Distribution of the U_n Statistic*

Description

Displays the empirical bootstrap distribution of U_n^* together with the observed value U_n and the bootstrap critical values at each significance level, as shown in Ospina et al. (2026).

Usage

```
plot_gof_boot(
  x,
  col.hist = "#abd9e9",
  col.obs = "#d7191c",
  col.cv = "#2c7bb6",
  main = NULL,
  xlab = expression(U[n]^"*"),
  ylab = "Density",
  ...
)
```

Arguments

`x`	A `"simplexgof"` object from `simplex_gof`.
`col.hist`	Fill colour for the histogram.
`col.obs`	Colour for the observed U_n line.
`col.cv`	Colour for the critical-value lines.
`main, xlab, ylab`	Plot labels.
`...`	Further arguments passed to `hist`.

Value

Invisibly returns the `"simplexgof"` object.

See Also

`simplex_gof`, `plot_influence`, `plot.simplexgof`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
set.seed(123)
gof <- simplex_gof(ammonia$perda, X, Z, B = 200, verbose = FALSE)
plot_gof_boot(gof)
```

`plot_influence` *Influence Index Plot*

Description

Plots the total local-influence measure C_{I_t} for each observation under case-weight perturbation, as used in the companion article (Ospina et al., 2026, Figure 2). A horizontal reference line at $2\bar{C}_I$ flags potentially influential observations.

Usage

```
plot_influence(
  x,
  threshold = 2,
  col.bar = "#2c7bb6",
  col.line = "#d7191c",
  label = TRUE,
  main = "Local influence -- case-weight perturbation",
  xlab = "Observation index",
  ylab = expression(C[It]),
  ...
)
```

Arguments

`x`	A `"simplexfit"` object from `simplex_fit`, or a list containing at least `$Cei` (influence vector of length n) as returned by `simplex_diag`.
`threshold`	Numeric scalar; multiplier for the mean of C_{I_t} used to draw the reference line. Default 2.
`col.bar`	Colour for the bars. Default `"#2c7bb6"`.
`col.line`	Colour for the reference line. Default `"#d7191c"`.
`label`	Logical; whether to label bars above the threshold with their index. Default `TRUE`.
`main`, `xlab`, `ylab`	Plot title and axis labels.
`...`	Additional arguments passed to `plot`.

Value

Invisibly returns the numeric vector C_{I_t}.

See Also

`simplex_diag`, `simplex_fit`, `plot.simplexfit`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
fit <- simplex_fit(ammonia$perda, X, Z)
dg  <- simplex_diag(fit)
plot_influence(dg)
```

`rsimplex` *Generate Random Observations from the Simplex Distribution*

Description

Generates independent observations from a simplex distribution $S^{-1}(\mu, \sigma^2)$ using the representation via inverse-Gaussian and chi-squared random variables (Michael, Schucany & Haas, 1976).

Usage

```
rsimplex(n, mu, sigma2)
```

Arguments

`n`	Integer; number of observations.
`mu`	Numeric scalar or vector of means in (0, 1).
`sigma2`	Numeric scalar or vector of dispersion parameters (> 0).

Details

Uses the reparametrisation $\epsilon = \mu/(1-\mu)$ (odds) and $\tau = \sigma^2(1-\mu)^2$ to generate from an inverse-Gaussian mixture. Identical algorithm to the Ox reference implementation.

Value

Numeric vector of length `n` with values in (0, 1).

References

Barndorff-Nielsen O.E., Jorgensen B. (1991). Some parametric models on the simplex. *Journal of Multivariate Analysis*, 39(1), 106–116.

Michael J.R., Schucany W.R., Haas R.W. (1976). Generating random variates using transformations with multiple roots. *The American Statistician*, 30(2), 88–90.

Examples

```
set.seed(1)
y <- rsimplex(200, mu = 0.3, sigma2 = 2)
hist(y, breaks = 20, main = "Simplex(0.3, 2)")
```

`sim_table1` *Monte Carlo Size/Power Simulation for the Bootstrap U_n Test*

Description

Replicates the Monte Carlo study of Espinheira et al. (2026), computing empirical rejection rates of the bootstrap U_n test under correct specification (size) or misspecification (power).

Usage

```
sim_table1(
  n = 40,
  beta = c(-3, 2, 1, -1, 0.5),
  sigma2 = 0.5,
  R = 5000,
  B = 1000,
  alpha = c(0.01, 0.05, 0.1),
  mu_range = c("low", "mid", "high"),
  ncores = 1,
  seed = NULL
)
```

Arguments

`n`	Sample size.
`beta`	Numeric vector of mean-model coefficients.
`sigma2`	Dispersion parameter (constant model).
`R`	Number of Monte Carlo replications.
`B`	Number of bootstrap replicates per replication.
`alpha`	Significance levels.
`mu_range`	One of `"low"`, `"mid"`, `"high"`; used to select the covariate configuration that places fitted means near 0, near 0.5, or near 1.
`ncores`	Number of parallel workers for the outer MC loop. Default 1.
`seed`	Random seed. Default `NULL`.

Value

A data frame with columns `alpha` and `rej_rate`.

Examples

```
res <- sim_table1(n = 40, beta = c(-3, 2, 1, -1, 0.5),
                  sigma2 = 0.5, R = 100, B = 100,
                  mu_range = "mid", ncores = 1)
print(res)
```

`simplex_diag` *Compute Local-Influence GoF Diagnostic Quantities*

Description

Given a fitted simplex regression model, computes all quantities needed for the U_n goodness-of-fit statistic: the C_{I_t} influence measures, T_n, the J gradient vector, and the individual k_t terms that estimate the asymptotic variance of $T_n/\sqrt{n}$.

Usage

```
simplex_diag(fit, J.method = c("analytic", "finitediff"))
```

Arguments

`fit`	An object of class `"simplexfit"` returned by `simplex_fit`.
`J.method`	Method used to compute the J gradient vector: either `"analytic"` (default) for the closed-form expression, or `"finitediff"` for a numerical finite-difference approximation.

Details

The J vector is computed using the analytic closed-form expressions derived in the article (Ox-compatible implementation). These analytic expressions are faster than numerical differentiation and produce the same test decisions as the original Ox reference implementation.

Value

A list with components:

`Tn`	The numerator of U_n: $\sqrt{n}(\sum C_{I_t} - 2k)$.
`Un`	The test statistic $T_n / s_{k,c}$.
`Cei`	Numeric vector of length n: individual influence values.
`J_vec`	Gradient vector of tr(H_{LD}) w.r.t. θ.
`k_vec`	Numeric vector of length n: the k_t terms.
`A_star`, `B_star`, `A_star_inv`	Estimated matrices.
`Delta`	Perturbation matrix (k x n).
`Hessiana`, `inv_obs`, `inv_fisher`	Matrices from the fit.

See Also

`simplex_fit`, `simplex_gof`

`simplex_fit` *Fit a Simplex Regression Model*

Description

Fits a simplex regression model with logit link for the mean and log link for the dispersion parameter using maximum likelihood via BFGS.

Usage

```
simplex_fit(
  y,
  X,
  Z = NULL,
  start = NULL,
  control = list(maxit = 500, reltol = 1e-10)
)
```

Arguments

`y`	Numeric vector of responses in (0, 1).
`X`	Numeric matrix of covariates for the mean sub-model (including intercept column). Dimension n x p.
`Z`	Numeric matrix of covariates for the dispersion sub-model (including intercept column). Dimension n x q. If `NULL`, defaults to an intercept-only model (constant dispersion).
`start`	Optional numeric vector of starting values of length `p + q`. If `NULL`, OLS-based starting values are computed automatically.
`control`	A list passed to `optim`. Defaults to `list(maxit = 500, reltol = 1e-10)`.

Value

A list of class `"simplexfit"` with components:

`coefficients` Named numeric vector of MLE estimates (beta, gamma).

`loglik` Log-likelihood at the MLE.

`fitted.mu` Fitted means.

`fitted.sigma2` Fitted dispersion values.

`vcov.fisher` Variance-covariance matrix from Fisher information.

`se` Standard errors.

`converged` Logical; whether BFGS converged.

`X` Design matrix for mean sub-model.

`Z` Design matrix for dispersion sub-model.

`y` Response vector.

`n, p, q` Sample size and number of mean/dispersion parameters.

See Also

`simplex_gof`, `simplex_diag`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
fit <- simplex_fit(ammonia$perda, X, Z)
print(fit)
```

`simplex_gof` *Bootstrap-Calibrated Goodness-of-Fit Test for Simplex Regression*

Description

Performs the U_n local-influence goodness-of-fit test for a simplex regression model. The null distribution is calibrated via a parametric bootstrap, which provides accurate size control even in finite samples.

Usage

```
simplex_gof(
  y,
  X,
  Z = NULL,
  B = 1000,
  alpha = c(0.01, 0.05, 0.1),
  ncores = 1,
  seed = NULL,
  verbose = TRUE
)
```

Arguments

`y`	Numeric vector of responses in (0, 1).
`X`	Design matrix for the mean sub-model (n x p, including intercept).
`Z`	Design matrix for the dispersion sub-model (n x q, including intercept). If `NULL`, a constant-dispersion model is fitted.
`B`	Integer; number of bootstrap replicates. Default 1000.
`alpha`	Numeric vector of significance levels. Default `c(0.01, 0.05, 0.10)`.
`ncores`	Integer; number of parallel workers. Default 1 (sequential). Set to `NULL` to use all available cores minus 1.
`seed`	Integer random seed for reproducibility. Default `NULL`.
`verbose`	Logical; whether to print progress. Default `TRUE`.

Details

The test statistic is

$$U_n = \frac{\sqrt{n}\left[\sum_{t=1}^{n} C_{I_t} - 2(p+q)\right]}{s_{k,c}}$$

where $C_{I_t} = 2|\Delta_t^\top(-\ddot{\ell})^{-1}\Delta_t|$ is the total local influence of observation t under case-weight perturbation, and $s_{k,c}^2$ is the sample variance of $\{k(y_t;\hat{\theta})\}$.

Because the normal approximation is severely liberal for the simplex class (empirical size 3–7x nominal even at $n = 1000$), critical values from the parametric bootstrap are preferred. Asymptotic $N(0,1)$ critical values are also reported for comparison.

Value

An object of class `"simplexgof"` with components:

`fit`	The `"simplexfit"` object for the original data.
`diag`	The `simplex_diag()` output for the original data.
`Un`	Observed test statistic.
`Tn`	Observed T_n.
`Un_boot`	Numeric vector of B bootstrap U_n^* values.
`results`	Data frame summarising decisions at each level.
`B, alpha`	As input.

References

Espinheira P.L., Silva F.C., Barros M., Ospina R. (2026). A Bootstrap-Calibrated Local Influence Goodness-of-Fit Procedure for Simplex Regression Models.

Zhu H., Zhang H. (2004). A diagnostic procedure based on local influence. *Biometrika*, 91(3), 579–589.

See Also

`simplex_fit`, `simplex_diag`, `rsimplex`

Examples

```
data(ammonia)
X <- cbind(1, ammonia$corr_ar, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
Z <- cbind(1, ammonia$temp_agua,
           ammonia$corr_ar * ammonia$temp_agua)
set.seed(42)
gof <- simplex_gof(ammonia$perda, X, Z, B = 200)
print(gof)
```

`simplex_Un_asymptotic` *Asymptotic* U_n *Statistic (Finite-Difference Calibration)*

Description

Computes the U_n statistic using the finite-difference gradient J, which gives the correct asymptotic variance for the simplex class. This is the version whose null distribution is studied in the simulation section of the companion paper (Figure 1).

Usage

```
simplex_Un_asymptotic(y, X, Z = NULL, eps = 1e-04)
```

Arguments

`y`	Response vector in (0, 1).
`X`	Design matrix for the mean sub-model.
`Z`	Design matrix for the dispersion sub-model (or `NULL`).
`eps`	Finite-difference step. Default `1e-4`.

Details

For the bootstrap test, use `simplex_gof` instead — the analytic gradient is bootstrap-invariant and faster.

Value

Scalar U_n (asymptotic calibration), or `NA` if the model fails to converge.

See Also

`simplex_gof`, `simplex_diag`

Examples

```
set.seed(1)
n  <- 40
X  <- cbind(1, matrix(runif(n * 4), n, 4))
mu <- plogis(drop(X %*% c(2, -0.5, -1.4, 1.25, -2.35)))
y  <- rsimplex(n, mu, 0.5)
Un <- simplex_Un_asymptotic(y, X)
Un
```

Index