

# Package: betaregscale (via r-universe)

May 23, 2026

**Type** Package

**Title** Beta Regression for Interval-Censored Scale-Derived Outcomes

**Version** 2.7.1

**Maintainer** José Evandeilton Lopes <evandeilton@gmail.com>

**Description** Maximum-likelihood estimation of beta regression models for responses derived from bounded rating scales. Observations are treated as interval-censored on (0, 1) after a scale-to-unit transformation, and the likelihood is built from the difference of the beta CDF at the interval endpoints. The complete likelihood supports mixed censoring types: uncensored, left-censored, right-censored, and interval-censored observations. Both fixed- and variable-dispersion submodels are supported, with flexible link functions for the mean and precision components. A compiled C++ backend (via 'Rcpp' and 'RcppArmadillo') provides numerically stable, high-performance log-likelihood evaluation. Standard S3 methods (print(), summary(), coef(), fitted(), residuals(), predict(), plot(), confint(), vcov(), logLik(), AIC(), BIC()) are available for fitted objects.

**License** MIT + file LICENSE

**URL** <https://evandeilton.github.io/betaregscale/>

**BugReports** <https://github.com/evandeilton/betaregscale/issues>

**Depends** R (>= 4.1.0)

**Imports** Formula, ggplot2, numDeriv, Rcpp (>= 1.0.12), rlang, stats

**Suggests** betareg, gridExtra, knitr, MASS, rmarkdown, testthat (>= 3.0.0)

**LinkingTo** Rcpp, RcppArmadillo, RcppEigen

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** yes

**RoxygenNote** 7.3.3  
**Config/testthat/edition** 3  
**Repository** <https://evandeilton.r-universe.dev>  
**Date/Publication** 2026-05-23 19:07:39 UTC  
**RemoteUrl** <https://github.com/evandeilton/betaregscale>  
**RemoteRef** HEAD  
**RemoteSha** b823795be0f207ab52c86a0437ef008ec6b874eb

## Contents

AIC.brs	3
AIC.brsmm	4
anova.brs	5
anova.brsmm	6
autoplot.brs	7
autoplot.brs_bootstrap	9
autoplot.brs_marginaleffects	11
autoplot.brsmm	12
BIC.brs	14
BIC.brsmm	15
brs	16
brs_bootstrap	17
brs_cens	19
brs_check	21
brs_coef	23
brs_cv	25
brs_est	26
brs_gof	27
brs_hessian	28
brs_marginaleffects	29
brs_predict_scoreprob	31
brs_prep	33
brs_repar	36
brs_sim	37
brs_table	39
brsmm	41
brsmm_re_study	43
coef.brs	44
coef.brsmm	45
confint.brs	46
confint.brsmm	47
fitted.brs	48
fitted.brsmm	49
formula.brs	50
formula.brsmm	51
logLik.brs	52

logLik.brsmm . . . . .	53
model.matrix.brs . . . . .	54
model.matrix.brsmm . . . . .	55
nobs.brs . . . . .	56
nobs.brsmm . . . . .	57
plot.brs . . . . .	58
plot.brsmm . . . . .	59
predict.brs . . . . .	61
predict.brsmm . . . . .	62
print.brs . . . . .	63
print.brsmm . . . . .	64
print.brsmm_re_study . . . . .	65
print.summary.brs . . . . .	66
print.summary.brsmm . . . . .	67
ranef . . . . .	68
ranef.brsmm . . . . .	69
residuals.brs . . . . .	70
residuals.brsmm . . . . .	71
summary.brs . . . . .	72
summary.brsmm . . . . .	73
vcov.brs . . . . .	74
vcov.brsmm . . . . .	75

<b>Index</b>	<b>76</b>
--------------	-----------

---

AIC.brs	<i>Akaike information criterion</i>
---------	-------------------------------------

---

## Description

Akaike information criterion

## Usage

```
## S3 method for class 'brs'
AIC(object, ..., k = 2)
```

## Arguments

object	A fitted "betaregscale" object.
...	Ignored.
k	Penalty per parameter (default 2).

## Value

Scalar AIC value.

**See Also**

[brs](#), [logLik.brs](#), [BIC.brs](#), [brs\\_gof](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
AIC(fit)
```

---

AIC.brsmm

*AIC for brsmm models*

---

**Description**

AIC for brsmm models

**Usage**

```
## S3 method for class 'brsmm'
AIC(object, ..., k = 2)
```

**Arguments**

object	A fitted "brsmm" object.
...	Currently ignored.
k	Numeric penalty per parameter.

**Value**

Numeric scalar.

**See Also**

[brsmm](#), [logLik.brsmm](#), [BIC.brsmm](#), [brs\\_gof](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
AIC(fit)

```

anova.brs

*Model comparison by analysis of deviance (LR test) for 'brs'***Description**

Model comparison by analysis of deviance (LR test) for 'brs'

**Usage**

```

## S3 method for class 'brs'
anova(object, ..., test = c("Chisq", "none"))

```

**Arguments**

object	A fitted "brs" model.
...	Additional fitted "brs" and/or "brsmm" models to compare.
test	Character; "Chisq" (default) or "none".

**Value**

An object of class "anova" and "data.frame" with model-wise log-likelihood, information criteria, and (optionally) LR test columns.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Ferrari, S. L. P., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501

**See Also**

[brs](#), [logLik.brs](#), [AIC.brs](#), [BIC.brs](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
m1 <- brs(y ~ 1, data = prep)
m2 <- brs(y ~ x1, data = prep)
m3 <- brs(y ~ x1 + x2, data = prep)
anova(m1, m2, m3)

```

---

anova.brsmm

---

*Model comparison by analysis of deviance (LR test) for 'brsmm'*


---

**Description**

Model comparison by analysis of deviance (LR test) for 'brsmm'

**Usage**

```

## S3 method for class 'brsmm'
anova(object, ..., test = c("Chisq", "none"))

```

**Arguments**

object	A fitted "brsmm" model.
...	Additional fitted "brsmm" and/or "brs" models to compare.
test	Character; "Chisq" (default) or "none".

**Value**

An object of class "anova" and "data.frame" with model-wise log-likelihood, information criteria, and (optionally) LR test columns.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Ferrari, S. L. P., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501

**See Also**

[brsmm](#), [logLik.brsmm](#), [AIC.brsmm](#), [BIC.brsmm](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
m1 <- brs(y ~ 1, data = prep)
m2 <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
anova(m1, m2)
```

---

autoplot.brs

*ggplot2 autoplot for brs models*

---

**Description**

Produces ggplot2 diagnostics tailored to interval-censored scale models.

**Usage**

```
## S3 method for class 'brs'
autoplot(
  object,
  type = c("calibration", "score_dist", "cdf", "residuals_by_delta", "all"),
  bins = 10L,
  scores = NULL,
  newdata = NULL,
  n_grid = 200L,
  max_curves = 6L,
  residual_type = "rqr",
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  ncol = 2L,
  theme = ggplot2::theme_minimal(),
  ...
)
```

## Arguments

object	A fitted "brs" object.
type	Plot type: "calibration", "score_dist", "cdf", "residuals_by_delta", or "all" (produces all panels in a single grid).
bins	Number of bins for "calibration".
scores	Optional integer vector of scores for "score_dist". Defaults to all scores from 0 to ncuts.
newdata	Optional data frame of covariate scenarios for "cdf".
n_grid	Number of points on (0, 1) used to draw CDF curves.
max_curves	Maximum number of CDF curves shown when newdata is not provided.
residual_type	Residual type for "residuals_by_delta"; passed to <code>residuals.brs</code> .
title	Optional character: override the plot title via <code>ggplot2::labs(title = ...)</code> . Ignored when type = "all".
xlab	Optional character: override the x-axis label. Ignored when type = "all".
ylab	Optional character: override the y-axis label. Ignored when type = "all".
ncol	Number of columns for the grid when type = "all". Defaults to 2.
theme	A ggplot2 theme object (e.g., <code>ggplot2::theme_bw()</code> ) or a theme function. Applied to every panel. Defaults to <code>ggplot2::theme_minimal()</code> .
...	Additional arguments forwarded to <code>ggplot2::theme()</code> and applied on top of theme. Use named theme element arguments, e.g. <code>legend.position = "none"</code> .

## Details

type = "calibration" bins predictions and compares mean observed vs mean predicted response in each bin.

type = "score\_dist" compares observed score frequencies against expected frequencies implied by the fitted beta interval model.

## Value

A ggplot2 object.

## References

- Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.
- Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543
- Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**See Also**

[brs](#), [plot.brs](#), [autoplot.brs\\_bootstrap](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1 + x2, data = prep)
ggplot2::autoplot(fit, type = "calibration")
ggplot2::autoplot(fit,
  type = "calibration",
  title = "My calibration", ylab = "Observed"
)
ggplot2::autoplot(fit, type = "all")
```

---

autoplot.brs\_bootstrap

*ggplot2 autoplot for bootstrap results*

---

**Description**

Produces visual summaries for objects returned by [brs\\_bootstrap](#).

**Usage**

```
## S3 method for class 'brs_bootstrap'
autoplot(
  object,
  type = c("ci_forest", "dist", "qq", "stability"),
  parameter = NULL,
  title = NULL,
  caption = NULL,
  max_parameters = 12L,
  ci_level = NULL,
  theme = NULL,
  ...
)
```

**Arguments**

object	An object of class "brs_bootstrap".
type	Plot type: "ci_forest", "dist", "qq", or "stability".
parameter	Optional parameter name used by type = "dist", "qq", and "stability". If NULL, the first parameter is used.
title	Optional plot title override.
caption	Optional subtitles/titles for plot types. Accepts: <ul style="list-style-type: none"> <li>• a single string (used for the selected type);</li> <li>• a character vector/list with up to four entries in the order ci_forest, dist, qq, stability.</li> </ul>
max_parameters	Maximum number of parameters shown in type = "ci_forest".
ci_level	Confidence level used in type = "stability". Defaults to the level stored in object.
theme	Optional ggplot2 theme object (e.g., ggplot2::theme_bw()). If NULL, ggplot2::theme_minimal() is used.
...	Currently ignored.

**Details**

For type = "dist", "qq", and "stability", bootstrap draws must be present in attr(object, "boot\_draws"), obtained by fitting with brs\_bootstrap(..., keep\_draws = TRUE).

**Value**

A ggplot2 object.

**See Also**

[brs\\_bootstrap](#), [brs](#), [autoplot.brs](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
boot <- brs_bootstrap(fit, R = 50)
ggplot2::autoplot(boot, type = "ci_forest")
```

---

```
autoplot.brs_marginaleffects  
  ggplot2 autoplot for marginal effects
```

---

## Description

Produces visual summaries for objects returned by [brs\\_marginaleffects](#).

## Usage

```
## S3 method for class 'brs_marginaleffects'  
autoplot(  
  object,  
  type = c("forest", "magnitude", "dist"),  
  variable = NULL,  
  top_n = 12L,  
  title = NULL,  
  caption = NULL,  
  theme = NULL,  
  ...  
)
```

## Arguments

object	An object of class "brs_marginaleffects".
type	Plot type: "forest", "magnitude", or "dist".
variable	Optional variable name for type = "dist".
top_n	Maximum number of variables shown in "magnitude" (ordered by  AME ).
title	Optional plot title override.
caption	Optional subtitle override.
theme	Optional ggplot2 theme object. If NULL, ggplot2::theme_minimal() is used.
...	Currently ignored.

## Details

type = "dist" requires AME simulation draws stored in attr(object, "ame\_draws"), which are available when marginal effects are computed with keep\_draws = TRUE and interval = TRUE.

## Value

A ggplot2 object.

## See Also

[brs\\_marginaleffects](#), [brs](#), [autoplot.brs](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
ame <- brs_marginaleffects(fit)
ggplot2::autoplot(ame, type = "forest")

```

---

autoplot.brsmm

*ggplot2 autoplot for brsmm models*


---

**Description**

Produces ggplot2 diagnostics tailored to mixed beta interval models.

**Usage**

```

## S3 method for class 'brsmm'
autoplot(
  object,
  type = c("calibration", "score_dist", "ranef_qq", "residuals_by_group",
    "ranef_caterpillar", "ranef_density", "ranef_pairs", "shrinkage", "all"),
  bins = 10L,
  scores = NULL,
  residual_type = c("response", "pearson"),
  max_groups = 25L,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  ncol = 2L,
  theme = ggplot2::theme_minimal(),
  ...
)

```

**Arguments**

object	A fitted "brsmm" object.
type	Plot type. One of "calibration", "score_dist", "ranef_qq", "residuals_by_group", "ranef_caterpillar", "ranef_density", "ranef_pairs", "shrinkage", or "all" (produces all panels in a single grid).

bins	Number of bins for "calibration".
scores	Optional integer vector of scores for "score_dist". Defaults to all scores from 0 to ncuts.
residual_type	Residual type for "residuals_by_group"; passed to <code>residuals.brsmm</code> .
max_groups	Maximum groups in "residuals_by_group".
title	Optional character: override the plot title via <code>ggplot2::labs(title = ...)</code> . Ignored when type = "all".
xlab	Optional character: override the x-axis label. Ignored when type = "all".
ylab	Optional character: override the y-axis label. Ignored when type = "all".
ncol	Number of columns for the grid when type = "all". Defaults to 2.
theme	A <code>ggplot2</code> theme object (e.g., <code>ggplot2::theme_bw()</code> ) or a theme function. Applied to every panel. Defaults to <code>ggplot2::theme_minimal()</code> .
...	Additional arguments forwarded to <code>ggplot2::theme()</code> and applied on top of theme. Use named theme element arguments, e.g. <code>legend.position = "none"</code> .

### Value

A `ggplot2` object, or (when type = "all") invisibly returns the list of all panels after rendering the grid.

### Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit_mm <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
ggplot2::autoplot(fit_mm, type = "calibration", bins = 4)
ggplot2::autoplot(fit_mm,
  type = "ranef_caterpillar",
  title = "My title", ylab = "Mode"
)
ggplot2::autoplot(fit_mm, type = "all")
```

BIC.ars

*Bayesian information criterion*

---

**Description**

Bayesian information criterion

**Usage**

```
## S3 method for class 'ars'  
BIC(object, ...)
```

**Arguments**

object	A fitted "betaregscale" object.
...	Ignored.

**Value**

Scalar BIC value.

**See Also**

[ars](#), [logLik.ars](#), [AIC.ars](#), [ars\\_gof](#)

**Examples**

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- ars_prep(dat, ncuts = 100)  
fit <- ars(y ~ x1, data = prep)  
BIC(fit)
```

---

BIC.brsmm	<i>BIC for brsmm models</i>
-----------	-----------------------------

---

## Description

BIC for brsmm models

## Usage

```
## S3 method for class 'brsmm'  
BIC(object, ...)
```

## Arguments

object	A fitted "brsmm" object.
...	Currently ignored.

## Value

Numeric scalar.

## See Also

[brsmm](#), [logLik.brsmm](#), [AIC.brsmm](#), [brs\\_gof](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
BIC(fit)
```

brs

*Fit a beta interval regression model***Description**

Unified interface that dispatches to `brs_fit_fixed` (fixed dispersion) or `brs_fit_var` (variable dispersion) based on the formula structure.

**Usage**

```
brs(
  formula,
  data,
  link = "logit",
  link_phi = "logit",
  ncuts = 100L,
  lim = 0.5,
  reparam = 2L,
  method = c("BFGS", "L-BFGS-B"),
  hessian_method = c("numDeriv", "optim")
)
```

**Arguments**

formula	A <a href="#">Formula</a> -style formula with two parts: $y \sim x_1 + x_2 \mid z_1 + z_2$ .
data	Data frame.
link	Mean link function (default "logit").
link_phi	Dispersion link function (default "logit").
ncuts	Number of scale categories (default 100).
lim	Uncertainty half-width (default 0.5).
reparam	Reparameterization scheme (default 2).
method	Optimization method (default "BFGS").
hessian_method	Character: "numDeriv" or "optim".

**Details**

If the formula contains a `|` separator (e.g.,  $y \sim x_1 + x_2 \mid z_1$ ), the variable-dispersion model is fitted; otherwise, a fixed-dispersion model is used.

**Value**

An object of class "brs".

## References

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
# Fixed dispersion
fit1 <- brs(y ~ x1, data = prep)
print(fit1)
# Variable dispersion
fit2 <- brs(y ~ x1 | x2, data = prep)
print(fit2)
```

---

brs\_bootstrap

*Parametric bootstrap confidence intervals for brs models*


---

## Description

Computes bootstrap-based confidence intervals for the parameters of a fitted "brs" model by repeatedly simulating data from the fitted model and re-estimating parameters. Only "brs" (fixed or variable-dispersion) objects are supported; "brsmm" is not supported.

## Usage

```
brs_bootstrap(
  object,
```

```

R = 199L,
level = 0.95,
ci_type = c("percentile", "basic", "normal", "bca"),
max_tries = NULL,
keep_draws = FALSE
)

```

```

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

```

### Arguments

object	A fitted "brs" object (fixed or variable dispersion).
R	Integer: number of bootstrap replicates (default 199).
level	Numeric: confidence level (default 0.95).
ci_type	Character: type of confidence interval. One of "percentile" (default), "basic", "normal", or "bca".
max_tries	Optional integer: maximum number of bootstrap attempts to obtain converged replicates. If NULL, uses $\max(3 * R, 50)$ .
keep_draws	Logical: if TRUE, stores successful bootstrap parameter draws in attribute "boot_draws".
x	Object returned by brs_bootstrap.
...	Ignored.

### Details

For each replicate, data are simulated via [brs\\_sim](#) using the estimated coefficients (on the link scale) and the original design. The model is then re-fitted with [brs](#). Replicates that fail to converge are discarded; if the number of successful replicates is too low, a warning is issued. Intervals are the empirical quantiles of the bootstrap distribution of each parameter.

### Value

A data frame with columns parameter, estimate (original point estimate), se\_boot (bootstrap standard error), ci\_lower, ci\_upper, mcse\_lower, mcse\_upper, wald\_lower, wald\_upper, and level. The attribute "n\_success" gives the number of replicates that converged. Additional attributes include "R", "n\_attempted", "ci\_type", and optionally "boot\_draws".

### Methods (by generic)

- `print(brs_bootstrap)`: Print method for bootstrap results

### See Also

[confint.brs](#) for Wald intervals; [brs\\_sim](#) for simulation; [brs](#) for fitting.

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
boot <- brs_bootstrap(fit, R = 50, level = 0.95)
print(boot)

```

brs\_cens

*Graphical and tabular censoring summary***Description**

Produces a visual summary of the censoring structure in a fitted "brs" model or a response matrix produced by [brs\\_check](#). The summary includes:

1. Bar chart of censoring type counts
2. Histogram of midpoint responses colored by censoring type
3. Interval plot showing  $[l_i, u_i]$  segments
4. Proportion table of censoring types

**Usage**

```

brs_cens(
  object,
  n_sample = 100L,
  which = 1:4,
  caption = NULL,
  gg = FALSE,
  title = "Censoring diagnostic overview",
  sub.caption = NULL,
  theme = NULL,
  palette = NULL,
  inform = FALSE,
  ...
)

```

**Arguments**

object	A fitted "brs" object, a matrix returned by <code>brs_check</code> , or a data frame returned by <code>brs_prep</code> (must contain columns <code>left</code> , <code>right</code> , <code>yt</code> , and <code>delta</code> ).
n_sample	Integer: maximum number of observations to show in the interval plot (default 100). If the data has more observations, a random sample is drawn.
which	Integer vector selecting which panels to draw (default 1:4).
caption	Optional panel captions. Accepts a character vector (or list coercible to character) with up to 4 labels, in the order: burden, midpoint-by-type, width-by-type, ordered interval map.
gg	Logical: use ggplot2? (default FALSE).
title	Optional global title for the plotting page.
sub.caption	Optional subtitle/caption for the plotting page.
theme	Optional ggplot2 theme object (e.g., <code>ggplot2::theme_bw()</code> ). If NULL, a minimal theme is used when <code>gg = TRUE</code> .
palette	Optional named character vector with colors for censoring types Exact, Left, Right, and Interval.
inform	Logical; if TRUE, prints brief interpretation messages about boundary and interval censoring intensity.
...	Further arguments (currently ignored).

**Value**

Invisibly returns a data frame with censoring counts and proportions, percentages, and interpretation flags.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**Examples**

```
y <- c(0, 3, 5, 7, 10)
Y <- brs_check(y, ncuts = 10)
brs_cens(Y)
```

```
prep <- brs_prep(data.frame(y = y), ncuts = 10)
brs_cens(prepare)
```

brs\_check

*Transform and validate a scale-derived response variable***Description**

Takes a discrete (or continuous) response on the scale  $0, 1, \dots, K$  (where  $K = \text{ncuts}$ ) and converts it to a pair of interval endpoints on the open unit interval  $(0, 1)$ . Each observation is classified into one of four censoring types following the complete likelihood used in this package:

$\delta = 0$  **Uncensored (exact)**: the observation is a continuous value already in  $(0, 1)$ . The likelihood contribution is the density  $f(y_i|\theta)$ . Endpoints:  $l_i = u_i = y_i$  (or  $y_i/K$  when on the scale).

$\delta = 1$  **Left-censored**: the latent value is below some upper bound  $u_i$ . The contribution is  $F(u_i|\theta)$ . When the observation is at the scale minimum ( $y = 0$ ), the upper bound is  $u_i = \text{lim}/K$ . When the user forces  $\delta = 1$  on a non-boundary observation ( $y \neq 0$ ), the upper bound is  $u_i = (y + \text{lim})/K$ , preserving observation-specific variation. In both cases  $l_i = \epsilon$ .

$\delta = 2$  **Right-censored**: the latent value is above some lower bound  $l_i$ . The contribution is  $1 - F(l_i|\theta)$ . When the observation is at the scale maximum ( $y = K$ ), the lower bound is  $l_i = (K - \text{lim})/K$ . When the user forces  $\delta = 2$  on a non-boundary observation ( $y \neq K$ ), the lower bound is  $l_i = (y - \text{lim})/K$ , preserving observation-specific variation. In both cases  $u_i = 1 - \epsilon$ .

$\delta = 3$  **Interval-censored**: the standard case for scale data. The contribution is  $F(u_i|\theta) - F(l_i|\theta)$  with midpoint interval endpoints  $[(y - \text{lim})/K, (y + \text{lim})/K]$ .

**Usage**

```
brs_check(y, ncuts = 100L, lim = 0.5, delta = NULL)
```

**Arguments**

<code>y</code>	Numeric vector: the raw response. Can be either integer scores on the scale $\{0, 1, \dots, K\}$ or continuous values already in $(0, 1)$ .
<code>ncuts</code>	Integer: number of scale categories $K$ (default 100). Must be $\geq \max(y)$ .
<code>lim</code>	Numeric: half-width $h$ of the uncertainty region (default 0.5). Controls the width of the interval around each scale point.
<code>delta</code>	Integer vector or NULL. If NULL (default), censoring types are inferred automatically from the boundary rules described above. If provided, must have the same length as <code>y</code> , with every element in $\{\emptyset, 1, 2, 3\}$ . The supplied values override the automatic classification on a per-observation basis, and the endpoint formulas adapt to non-boundary observations as described in the table above. This parameter is used internally by the simulation functions when the analyst forces a specific censoring type (e.g., <code>brs_sim(..., delta = 2)</code> ).

## Details

### Automatic classification (delta = NULL):

If the entire input vector is already in  $(0, 1)$  (i.e., all values satisfy  $0 < y < 1$ ), all observations are treated as uncensored ( $\delta = 0$ ).

Otherwise, for scale (integer) data:

- $y = 0$ : left-censored ( $\delta = 1$ ).
- $y = K$ : right-censored ( $\delta = 2$ ).
- $0 < y < K$ : interval-censored ( $\delta = 3$ ).

### User-supplied delta (delta vector):

When the delta argument is provided, the user-supplied censoring indicators override the automatic boundary-based rules on a per-observation basis. This is the mechanism used by `brs_sim` when the analyst forces a specific censoring type in Monte Carlo studies.

The endpoint formulas for each delta value are:

$\delta$	Condition	$l_i$ (left)	$u_i$ (right)
0	$y \in (0, 1)$	$y$	$y$
0	$y$ on scale	$y/K$	$y/K$
1	$y = 0$ (boundary)	$\epsilon$	$\text{lim}/K$
1	$y \neq 0$ (forced)	$\epsilon$	$(y + \text{lim})/K$
2	$y = K$ (boundary)	$(K - \text{lim})/K$	$1 - \epsilon$
2	$y \neq K$ (forced)	$(y - \text{lim})/K$	$1 - \epsilon$
3	midpoint interval	$(y - \text{lim})/K$	$(y + \text{lim})/K$

All endpoints are clamped to  $[\epsilon, 1 - \epsilon]$  with  $\epsilon = 10^{-5}$  to avoid boundary issues in the beta likelihood.

The midpoint approximation  $y_t$  is computed as:

- $y_t = y$  when  $y \in (0, 1)$  (continuous data).
- $y_t = y/K$  when  $y$  is on the integer scale.

This value is used exclusively as an initialization aid for starting-value computation and does not enter the likelihood.

### Interaction with the fitting pipeline:

This function is called internally by `.extract_response()` when the data does *not* carry the "is\_prepared" attribute. If data has already been processed by `brs_prep` or by simulation with forced delta (`brs_sim` with `delta != NULL`), the pre-computed columns are used directly and `brs_check()` is skipped.

## Value

A numeric matrix with  $n$  rows and 5 columns:

`left` Lower endpoint  $l_i$  on  $(0, 1)$ , clamped to  $[\epsilon, 1 - \epsilon]$ .

`right` Upper endpoint  $u_i$  on  $(0, 1)$ , clamped to  $[\epsilon, 1 - \epsilon]$ .

yt Midpoint approximation  $y_t$  for starting-value computation (does not enter the likelihood).  
y Original response value (preserved unchanged).  
delta Censoring indicator: 0 = exact (density), 1 = left-censored  $F(u)$ , 2 = right-censored  $1 - F(l)$ ,  
3 = interval-censored  $F(u) - F(l)$ .

## References

- Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.
- Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543
- Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

## See Also

[brs\\_prep](#) for the analyst-facing pre-processing function; [brs\\_sim](#) for simulation with forced delta.

## Examples

```
# Scale data with boundary observations
y <- c(0, 3, 5, 7, 9, 10)
brs_check(y, ncuts = 10)

# Force all observations to be exact (delta = 0)
brs_check(y, ncuts = 10, delta = rep(0L, length(y)))

# Force delta = 1 on non-boundary observations:
# endpoints use actual y values, preserving variation
y2 <- c(30, 60)
brs_check(y2, ncuts = 100, delta = c(1L, 1L))
# left = (eps, eps), right = (30.5/100, 60.5/100)
```

---

brs\_coef

*Internal coefficient table (deprecated, use brs\_est() or summary())*


---

## Description

Deprecated convenience wrapper. Use [brs\\_est](#) for coefficient estimates or [summary.brs](#) for a full model summary.

**Usage**

```
brs_coef(fit, alpha = 0.05)
```

**Arguments**

fit	A fitted "brs" object.
alpha	Significance level.

**Value**

A list with components est (from [brs\\_est](#)) and gof (from [brs\\_gof](#)).

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Ferrari, S. L. P., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**See Also**

[brs\\_est](#), [brs\\_gof](#), [summary.brs](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
suppressWarnings(brs_coef(fit)) # deprecated; use brs_est()
```

---

brs\_cv *K-fold cross-validation for brs models*

---

### Description

Performs repeated k-fold cross-validation for `brs` models.

### Usage

```
brs_cv(formula, data, k = 5L, repeats = 1L, ...)
```

### Arguments

formula	Model formula passed to <code>brs</code> .
data	Data frame.
k	Number of folds.
repeats	Number of repeated k-fold runs.
...	Additional arguments forwarded to <code>brs</code> (e.g., <code>repar</code> , <code>link</code> , <code>method</code> ).

### Details

The `log_score` is the mean log predictive contribution under the complete likelihood contribution implied by each observation's censoring type (`delta`).

### Value

A data frame with one row per fold and columns: `repeat`, `fold`, `n_train`, `n_test`, `log_score`, `rmse_yt`, `mae_yt`, `converged`, and `error`. The object has class `"brs_cv"`.

### References

- Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.
- Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543
- Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
cv <- brs_cv(y ~ x1, data = prep, k = 3, repeats = 1)
cv

```

brs\_est

*Coefficient estimates with inference***Description**

Coefficient estimates with inference

**Usage**

brs\_est(object, alpha = 0.05)

**Arguments**

object	A fitted "betaregscale" object.
alpha	Significance level (default 0.05).

**Value**

Data frame of estimates, standard errors, z-values, and p-values.

**References**

- Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.
- Ferrari, S. L. P., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501
- Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

### See Also

[brs](#), [brs\\_gof](#), [brs\\_hessian](#), [summary.brs](#)

### Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
brs_est(fit)
```

---

brs\_gof

*Goodness-of-fit measures*

---

### Description

Goodness-of-fit measures

### Usage

```
brs_gof(object)
```

### Arguments

object            A fitted "brs" or "brsmm" object.

### Value

Data frame with logLik, AIC, BIC, and pseudo-R-squared.

## References

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

## See Also

[brs](#), [brs\\_est](#), [brs\\_hessian](#)

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
brs_gof(fit)
```

---

brs\_hessian

*Extract the Hessian matrix*

---

## Description

Extract the Hessian matrix

## Usage

```
brs_hessian(object)
```

## Arguments

object            A fitted "brs" or "brsmm" object.

**Value**

Numeric Hessian matrix.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**See Also**

[brs](#), [vcov.brs](#), [brs\\_est](#)

**Examples**

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brs(y ~ x1, data = prep)  
brs_hessian(fit)
```

---

brs\_marginaleffects    *Marginal effects for brs models*

---

**Description**

Computes average marginal effects (AME) for numeric covariates in the mean or precision sub-model of a fitted "brs" object.

**Usage**

```
brs_marginaleffects(
  object,
  newdata = NULL,
  model = c("mean", "precision"),
  type = c("response", "link"),
  variables = NULL,
  h = 1e-05,
  interval = TRUE,
  level = 0.95,
  n_sim = 400L,
  keep_draws = FALSE
)
```

**Arguments**

object	A fitted "brs" object.
newdata	Optional data frame for evaluation; defaults to the data used in fitting.
model	Character; "mean" (default) or "precision".
type	Character prediction scale: "response" (default) or "link".
variables	Optional character vector of covariate names. Defaults to all numeric covariates in the selected submodel.
h	Finite-difference step for non-binary numeric covariates.
interval	Logical; compute interval estimates via simulation.
level	Confidence level for interval estimates.
n_sim	Number of parameter draws when interval = TRUE.
keep_draws	Logical; if TRUE and interval = TRUE, stores AME simulation draws in attribute "ame_draws".

**Details**

AMEs are computed by finite differences on predictions:

$$\text{AME}_j = \frac{1}{n} \sum_{i=1}^n \frac{\hat{g}_i(x_{ij} + h) - \hat{g}_i(x_{ij})}{h},$$

where  $\hat{g}_i$  is the selected prediction scale.

For binary covariates coded as 0/1, the effect is computed as the average discrete difference  $\hat{g}(x_j = 1) - \hat{g}(x_j = 0)$ .

If interval = TRUE, uncertainty is approximated by asymptotic parameter simulation from  $\mathcal{N}(\hat{\theta}, \hat{V})$ .

**Value**

A data frame with one row per variable and columns: variable, ame, std.error, ci.lower, ci.upper, model, type, and n. The returned object has class "brs\_marginaleffects" and attributes with analysis metadata.

## References

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
brs_marginaleffects(fit, model = "mean", type = "response")
```

---

brs\_predict\_scoreprob *Predict score probabilities from a fitted brs model*

---

## Description

Computes predicted probabilities for integer scores on the original scale  $\{0, 1, \dots, K\}$  implied by the fitted beta interval model.

## Usage

```
brs_predict_scoreprob(
  object,
  newdata = NULL,
  scores = NULL,
  format = c("matrix", "long"),
  id_col = "id"
)
```

**Arguments**

object	A fitted "brs" object.
newdata	Optional data frame for prediction.
scores	Optional integer vector of scores to evaluate. Defaults to all scores in $0:ncuts$ .
format	Output format: "matrix" (default) or "long".
id_col	Column name for observation id when format = "long".

**Details**

For a score  $s$  and  $K = ncuts$ , probabilities are computed as:

- $P(Y = s) = F(\lim/K)$  for  $s = 0$ ,
- $P(Y = s) = 1 - F((K - \lim)/K)$  for  $s = K$ ,
- $P(Y = s) = F((s + \lim)/K) - F((s - \lim)/K)$  for  $s \in \{1, \dots, K - 1\}$ ,

where  $F$  is the beta CDF under the fitted  $(\mu_i, \phi_i)$ .

**Value**

If format = "matrix", a numeric matrix with one row per observation and one column per requested score.

If format = "long", a long data frame with columns id\_col, score, and prob.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
```

```

)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
pmat <- brs_predict_scoreprob(fit)
head(pmat[, 1:5])
plong <- brs_predict_scoreprob(fit, scores = 0:10, format = "long")
head(plong)

```

brs\_prep

*Pre-process analyst data for beta interval regression*

## Description

Validates and transforms raw data into the format required by `brs`. The analyst can supply data in several ways:

1. **Minimal (Mode 1)**: only the score  $y$ . Censoring is inferred automatically:  $y = 0 \rightarrow \delta = 1$ ,  $y = K \rightarrow \delta = 2$ ,  $0 < y < K \rightarrow \delta = 3$ ,  $y \in (0, 1) \rightarrow \delta = 0$ .
2. **Classic (Mode 2)**:  $y$  + explicit delta. The analyst declares the censoring type; interval endpoints are computed using the actual  $y$  value.
3. **Interval (Mode 3)**: left and/or right columns (on the original scale). Censoring is inferred from the NA pattern.
4. **Full (Mode 4)**:  $y$ , left, and right together. The analyst's own endpoints are rescaled directly to  $(0, 1)$ .

All covariate columns are preserved unchanged in the output.

## Usage

```

brs_prep(
  data,
  y = "y",
  delta = "delta",
  left = "left",
  right = "right",
  ncuts = 100L,
  lim = 0.5
)

```

## Arguments

<code>data</code>	Data frame containing the response variable and covariates.
<code>y</code>	Character: name of the score column (default "y").
<code>delta</code>	Character: name of the censoring indicator column (default "delta"). Values must be in $\{0, 1, 2, 3\}$ .

left	Character: name of the left-endpoint column (default "left").
right	Character: name of the right-endpoint column (default "right").
ncuts	Integer: number of scale categories (default 100).
lim	Numeric: half-width of the uncertainty region (default 0.5). Used only when constructing intervals from y alone.

## Details

**Priority rule:** if delta is provided (non-NA), it takes precedence over all automatic classification rules. When delta is NA, the function infers the censoring type from the pattern of left, right, and y:

left	right	y	delta	Interpretation	Inferred $\delta$
NA	5	NA	NA	Left-censored (below 5)	1
20	NA	NA	NA	Right-censored (above 20)	2
30	45	NA	NA	Interval-censored [30, 45]	3
NA	NA	50	NA	Exact observation	0
NA	NA	50	3	Analyst says interval	3
NA	NA	0	1	Analyst says left-censored	1
NA	NA	99	2	Analyst says right-censored	2

When y, left, and right are all present for the same observation, the analyst's left/right values are used directly (rescaled by  $K = \text{ncuts}$ ) and delta is set to 3 (interval-censored) unless the analyst supplied delta explicitly.

### Endpoint formulas for Mode 2 (y + explicit delta):

When the analyst supplies delta explicitly, the endpoint computation uses the actual y value to produce observation-specific bounds. This is the same logic used by [brs\\_check](#) with a user-supplied delta vector:

$\delta$	Condition	$l_i$ (left)	$u_i$ (right)
0	(any)	$y/K$	$y/K$
1	$y = 0$	$\epsilon$	$\text{lim}/K$
1	$y \neq 0$	$\epsilon$	$(y + \text{lim})/K$
2	$y = K$	$(K - \text{lim})/K$	$1 - \epsilon$
2	$y \neq K$	$(y - \text{lim})/K$	$1 - \epsilon$
3	type "m"	$(y - \text{lim})/K$	$(y + \text{lim})/K$

**Consistency warnings:** when the analyst supplies delta values that are unusual for the given y (e.g.,  $\delta = 1$  but  $y \neq 0$ ), the function emits a warning but proceeds. This is by design for Monte Carlo workflows where forced delta on non-boundary observations is intentional.

All endpoints are clamped to  $[\epsilon, 1 - \epsilon]$  with  $\epsilon = 10^{-5}$ .

**Value**

A data.frame with the following columns appended or replaced:

left Lower endpoint on (0, 1).

right Upper endpoint on (0, 1).

yt Midpoint approximation on (0, 1).

y Original scale value (preserved for reference).

delta Censoring indicator: 0 = exact, 1 = left, 2 = right, 3 = interval.

Covariate columns are preserved. The output carries attributes "is\_prepared" (TRUE), "ncuts" and "lim" so that `brs` can detect prepared data and skip the internal `brs_check` call.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**See Also**

`brs_check` for the automatic classification of raw scale scores; `brs` for fitting the model.

**Examples**

```
# --- Mode 1: y only (automatic classification, like brs_check) ---
d1 <- data.frame(y = c(0, 3, 5, 7, 10), x1 = rnorm(5))
brs_prep(d1, ncuts = 10)

# --- Mode 2: y + explicit delta ---
d2 <- data.frame(
  y = d1$y,
  delta = c(0, 3, 3, 3, 0), # Force interval-censoring for 3,5,7
  x1 = d1$x1
)
brs_prep(d2, ncuts = 100)

# --- Mode 3: left/right with NA patterns ---
d3 <- data.frame(
  left = c(NA, 20, 30, NA),
  right = c(5, NA, 45, NA),
```

```

    y = c(NA, NA, NA, 50),
    x1 = d1$x1[1:4]
  )
brs_prep(d3, ncuts = 100)

# --- Mode 4: y + left + right (analyst-supplied intervals) ---
d4 <- data.frame(
  y = c(50, 75),
  left = c(48, 73),
  right = c(52, 77),
  x1 = rnorm(2)
)
brs_prep(d4, ncuts = 100)

# --- Fitting after prep ---

dat5 <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep5 <- brs_prep(dat5, ncuts = 100)
fit5 <- brs(y ~ x1, data = prep5)
summary(fit5)

```

---

brs\_repar

*Reparameterize (mu, phi) into beta shape parameters*


---

### Description

Converts a mean–dispersion pair  $(\mu, \phi)$  to the shape parameters  $(a, b)$  of the beta distribution under one of three reparameterization schemes.

### Usage

```
brs_repar(mu, phi, repar = 2L)
```

### Arguments

mu	Numeric vector of mean values in $(0, 1)$ .
phi	Numeric vector (or scalar) of dispersion values.
repar	Integer (0, 1, or 2) selecting the scheme.

## Details

The three schemes are:

repar = 0 Direct:  $a = \mu$ ,  $b = \phi$ .

repar = 1 Ferrari–Cribari-Neto:  $a = \mu\phi$ ,  $b = (1 - \mu)\phi$ , where  $\phi$  acts as a precision parameter.

repar = 2 Mean–variance:  $a = \mu(1 - \phi)/\phi$ ,  $b = (1 - \mu)(1 - \phi)/\phi$ , where  $\phi \in (0, 1)$  is analogous to a coefficient of variation.

## Value

A data.frame with columns shape1 and shape2.

## References

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master’s dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

## Examples

```
brs_repar(mu = 0.5, phi = 0.2, repar = 2)
```

---

brs\_sim

*Simulate data from beta interval models*


---

## Description

Simulates interval-censored responses from fixed- or variable-dispersion beta regression models.

## Usage

```
brs_sim(
  formula,
  data,
  beta,
  phi = 1/5,
```

```

    zeta = NULL,
    link = "logit",
    link_phi = "logit",
    ncuts = 100L,
    lim = 0.5,
    repar = 2L,
    delta = NULL
  )

```

### Arguments

formula	Model formula with one (mean) or two parts (mean   precision). A left-hand-side response is allowed but ignored.
data	Data frame with predictor variables.
beta	Numeric vector of mean-model coefficients.
phi	Scalar dispersion parameter (link scale), used only for one-part formulas.
zeta	Numeric vector of precision-model coefficients (link scale), required for two-part formulas.
link	Mean link function.
link_phi	Precision link function.
ncuts	Number of scale categories.
lim	Half-width used in interval construction.
repar	Reparameterization scheme.
delta	Forced censoring type (0, 1, 2, 3) or NULL.

### Details

The model structure is controlled by `formula` in the same style as `brs`:

- one-part formula ( $\sim x_1 + x_2$  or  $y \sim x_1 + x_2$ ): fixed dispersion using scalar `phi`.
- two-part formula ( $\sim x_1 + x_2 | z_1$  or  $y \sim x_1 + x_2 | z_1$ ): variable dispersion using coefficient vector `zeta`.

The `delta` argument can force a single censoring type (0, 1, 2, 3) for all observations; otherwise, censoring is classified automatically from simulated scale values via `brs_check`.

### Value

A data frame with columns `left`, `right`, `yt`, `y`, `delta`, plus simulated predictor columns from the model matrices. When `delta != NULL`, the output carries `attr(, "is_prepared") = TRUE`.

### References

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

## Examples

```
dat <- data.frame(
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
# Fixed dispersion
sim_fixed <- brs_sim(
  formula = ~ x1 + x2, data = dat,
  beta = c(0.2, -0.5, 0.3), phi = 1 / 5
)
# Variable dispersion
sim_var <- brs_sim(
  formula = ~ x1 | x2, data = dat,
  beta = c(0.2, -0.5), zeta = c(0.5, -0.5)
)
```

---

brs\_table

*Compare fitted brs models in a single table*


---

## Description

Builds a comparison table for one or more fitted "brs" objects, summarizing fit statistics and (optionally) censoring composition.

## Usage

```
brs_table(
  ...,
  models = NULL,
  include_censoring = TRUE,
  sort_by = c("none", "AIC", "BIC", "logLik"),
  decreasing = FALSE,
  digits = 4L
)
```

**Arguments**

...	Fitted "brs" objects passed individually.
models	Optional list of fitted "brs" objects. Use either ... or models, not both.
include_censoring	Logical; include censoring counts/proportions. Default is TRUE.
sort_by	Character; optional sort criterion: "none" (default), "AIC", "BIC", or "logLik".
decreasing	Logical; sort direction when sort_by != "none".
digits	Integer number of digits used for numeric rounding.

**Value**

A data frame with one row per model.

**References**

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Hawker, G. A., Mian, S., Kendzerska, T., and French, M. (2011). Measures of adult pain: Visual Analog Scale for Pain (VAS Pain), Numeric Rating Scale for Pain (NRS Pain), McGill Pain Questionnaire (MPQ), Short-Form McGill Pain Questionnaire (SF-MPQ), Chronic Pain Grade Scale (CPGS), Short Form-36 Bodily Pain Scale (SF-36 BPS), and Measure of Intermittent and Constant Osteoarthritis Pain (ICOAP). *Arthritis Care and Research*, 63(S11), S240-S252. doi:10.1002/acr.20543

Hjermstad, M. J., Fayers, P. M., Haugen, D. F., et al. (2011). Studies comparing Numerical Rating Scales, Verbal Rating Scales, and Visual Analogue Scales for assessment of pain intensity in adults: a systematic literature review. *Journal of Pain and Symptom Management*, 41(6), 1073-1093. doi:10.1016/j.jpainsymman.2010.08.016

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  x2 = rep(c(0, 0, 1, 1), 5)
)
prep <- brs_prep(dat, ncuts = 100)
m1 <- brs(y ~ 1, data = prep)
m2 <- brs(y ~ x1, data = prep)
brs_table(null = m1, x1 = m2, sort_by = "AIC")
```

brsmm

*Fit a mixed-effects beta interval regression model***Description**

Fits a beta interval-censored mixed model with Gaussian random intercepts/slopes using marginal maximum likelihood. The implementation supports random-effects formulas such as  $\sim 1 \mid \text{group}$  and  $\sim 1 + x \mid \text{group}$ , and offers three integration methods for the random effects: Laplace approximation, Adaptive Gauss-Hermite Quadrature (AGHQ), and Quasi-Monte Carlo (QMC).

**Usage**

```
brsmm(
  formula,
  random = ~1 | id,
  data,
  link = "logit",
  link_phi = "logit",
  reparam = 2L,
  ncuts = 100L,
  lim = 0.5,
  int_method = c("laplace", "aghq", "qmc"),
  n_points = 11L,
  qmc_points = 1024L,
  start = NULL,
  method = c("BFGS", "L-BFGS-B"),
  hessian_method = c("numDeriv", "optim"),
  control = list(maxit = 2000L)
)
```

**Arguments**

formula	Model formula. Supports one- or two-part formulas: $y \sim x_1 + x_2$ or $y \sim x_1 + x_2 \mid z_1 + z_2$ .
random	Random-effects specification of the form $\sim \text{terms} \mid \text{group}$ , e.g. $\sim 1 \mid \text{id}$ or $\sim 1 + x \mid \text{id}$ .
data	Data frame.
link	Mean link function.
link_phi	Precision link function.
reparam	Beta reparameterization code (0, 1, 2).
ncuts	Number of categories on the original scale.
lim	Half-width used to construct interval endpoints.
int_method	Integration method: "laplace" (default), "aghq", or "qmc".
n_points	Number of quadrature points for <code>int_method="aghq"</code> . Ignored for other methods. Default is 11.

qmc_points	Number of QMC points for <code>int_method="qmc"</code> . Default is 1024.
start	Optional numeric vector of starting values (beta, gamma, and packed lower-Cholesky random parameters).
method	Optimizer passed to <code>optim</code> .
hessian_method	"numDeriv" (default) or "optim".
control	Control list for <code>optim</code> .

## Details

The conditional contribution for each observation follows the same mixed censoring likelihood used by `brs`:

1.  $\delta = 0$ : exact contribution via beta density,
2.  $\delta = 1$ : left-censored contribution via beta CDF,
3.  $\delta = 2$ : right-censored contribution via survival CDF,
4.  $\delta = 3$ : interval contribution via CDF difference.

For group  $i$ , the random-effects vector  $\mathbf{b}_i \sim N(\mathbf{0}, D)$  is integrated out numerically.

- "laplace": Uses a second-order Laplace approximation at the conditional mode. Fast and generally accurate for  $n_i$  large.
- "aghq": Adaptive Gauss-Hermite Quadrature. Uses `n_points` quadrature nodes centered and scaled by the conditional mode and curvature. More accurate than Laplace, especially for small  $n_i$ .
- "qmc": Quasi-Monte Carlo integration using a Halton sequence. Uses `qmc_points` evaluation points. Suitable for high-dimensional integration (future proofing) or checking robustness.

## Value

An object of class "brsmm".

## References

- Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.
- Ferrari, S. L. P., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
```

```
prep <- brs_prep(dat, ncuts = 100)
fit_mm <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
fit_mm
```

---

brsmm\_re\_study

*Random-effects study for brsmm models*

---

## Description

Provides a compact numeric study of random effects, including: estimated covariance matrix, correlation matrix, per-term standard deviations, empirical mean/SD of posterior modes, shrinkage ratio, and a normality check by Shapiro-Wilk (when applicable).

## Usage

```
brsmm_re_study(object, ...)
```

## Arguments

object	A fitted "brsmm" object.
...	Currently ignored.

## Value

A list with class "brsmm\_re\_study".

## References

Lopes, J. E. (2023). *Modelos de regressao beta para dados de escala*. Master's dissertation, Universidade Federal do Parana, Curitiba. URI: <https://hdl.handle.net/1884/86624>.

Ferrari, S. L. P., and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501

## See Also

[brsmm](#), [ranef.brsmm](#)

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
```

```

)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
rs <- brsmm_re_study(fit)
print(rs)
rs$summary

```

---

coef.brs

*Extract model coefficients*


---

## Description

Extract model coefficients

## Usage

```

## S3 method for class 'brs'
coef(object, model = c("full", "mean", "precision"), ...)

```

## Arguments

object	A fitted "betareg" object.
model	Character: which component to return. "full" (default) returns all parameters, "mean" returns only the mean-model coefficients, "precision" returns only the precision coefficients.
...	Ignored.

## Value

Named numeric vector of estimated parameters.

## See Also

[brs](#), [brs\\_est](#), [vcov.brs](#)

## Examples

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
coef(fit)

```

```
coef(fit, model = "mean")
coef(fit, model = "precision")
```

---

coef.brsmm

*Extract coefficients from a brsmm fit*


---

### Description

Extract coefficients from a brsmm fit

### Usage

```
## S3 method for class 'brsmm'
coef(object, model = c("full", "mean", "precision", "random"), ...)
```

### Arguments

object	A fitted "brsmm" object.
model	Character: "full" (default), "mean", "precision", or "random".
...	Currently ignored.

### Value

Named numeric vector.

### See Also

[brsmm](#), [vcov.brsmm](#), [confint.brsmm](#)

### Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
coef(fit)
coef(fit, model = "mean")
coef(fit, model = "random")
```

---

 confint.brs

*Wald confidence intervals*


---

### Description

Computes Wald confidence intervals for model parameters using the normal approximation.

### Usage

```
## S3 method for class 'brs'
confint(
  object,
  parm,
  level = 0.95,
  model = c("full", "mean", "precision"),
  ...
)
```

### Arguments

object	A fitted "betaregscale" object.
parm	Character or integer: which parameters. If missing, all parameters are returned.
level	Confidence level (default 0.95).
model	Character: "full", "mean", or "precision".
...	Currently ignored.

### Value

Matrix with columns for lower and upper confidence bounds.

### See Also

[brs](#), [coef.brs](#), [vcov.brs](#), [brs\\_est](#)

### Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
confint(fit)
confint(fit, model = "mean")
```

---

`confint.brsmm`*Wald confidence intervals for brsmm models*

---

**Description**

Wald confidence intervals for brsmm models

**Usage**

```
## S3 method for class 'brsmm'
confint(
  object,
  parm,
  level = 0.95,
  model = c("full", "mean", "precision", "random"),
  ...
)
```

**Arguments**

<code>object</code>	A fitted "brsmm" object.
<code>parm</code>	Character or integer: which parameters.
<code>level</code>	Confidence level (default 0.95).
<code>model</code>	Character: "full", "mean", "precision", or "random".
<code>...</code>	Currently ignored.

**Value**

Matrix with columns for lower and upper confidence bounds.

**See Also**

[brsmm](#), [coef.brsmm](#), [vcov.brsmm](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
```

```

prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
confint(fit, model = "mean")

```

---

fitted.brs

*Extract fitted values*


---

### Description

Extract fitted values

### Usage

```

## S3 method for class 'brs'
fitted(object, type = c("mu", "phi"), ...)

```

### Arguments

object	A fitted "betaregscale" object.
type	Character: "mu" (default) or "phi".
...	Currently ignored.

### Value

Numeric vector of fitted values.

### See Also

[brs](#), [residuals.brs](#), [predict.brs](#)

### Examples

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
head(fitted(fit))
head(fitted(fit, type = "phi"))

```

---

fitted.brsmm	<i>Fitted values from a brsmm model</i>
--------------	---

---

### Description

Fitted values from a brsmm model

### Usage

```
## S3 method for class 'brsmm'  
fitted(object, type = c("mu", "phi"), ...)
```

### Arguments

object	A fitted "brsmm" object.
type	Character: "mu" (default) or "phi".
...	Currently ignored.

### Value

Numeric vector.

### See Also

[brsmm](#), [residuals.brsmm](#), [predict.brsmm](#)

### Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
head(fitted(fit))  
head(fitted(fit, type = "phi"))
```

---

formula.brs	<i>Extract model formula</i>
-------------	------------------------------

---

### Description

Extract model formula

### Usage

```
## S3 method for class 'brs'  
formula(x, ...)
```

### Arguments

x	A fitted "betaregscale" object.
...	Ignored.

### Value

The formula used to fit the model.

### See Also

[brs](#), [model.matrix.brs](#), [coef.brs](#)

### Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brs(y ~ x1, data = prep)  
formula(fit)
```

---

formula.brsmm	<i>Extract model formula</i>
---------------	------------------------------

---

## Description

Extract model formula

## Usage

```
## S3 method for class 'brsmm'  
formula(x, ...)
```

## Arguments

x	A fitted "brsmm" object.
...	Ignored.

## Value

The formula used to fit the model.

## See Also

[brsmm](#), [model.matrix.brsmm](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
formula(fit)
```

---

`logLik.brs`*Extract log-likelihood*

---

## Description

Extract log-likelihood

## Usage

```
## S3 method for class 'brs'  
logLik(object, ...)
```

## Arguments

<code>object</code>	A fitted "betaregscale" object.
<code>...</code>	Ignored.

## Value

An object of class "logLik" with attributes `df` (number of estimated parameters) and `nobs` (number of observations).

## See Also

[brs](#), [AIC.brs](#), [BIC.brs](#), [brs\\_gof](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brs(y ~ x1, data = prep)  
logLik(fit)
```

---

logLik.brsmm	<i>Log-likelihood for brsmm models</i>
--------------	--

---

## Description

Log-likelihood for brsmm models

## Usage

```
## S3 method for class 'brsmm'  
logLik(object, ...)
```

## Arguments

object	A fitted "brsmm" object.
...	Currently ignored.

## Value

Object of class "logLik".

## See Also

[brsmm](#), [AIC.brsmm](#), [BIC.brsmm](#), [brs\\_gof](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
logLik(fit)
```

---

model.matrix.brs      *Extract design matrix*

---

## Description

Extract design matrix

## Usage

```
## S3 method for class 'brs'  
model.matrix(object, model = c("mean", "precision"), ...)
```

## Arguments

object	A fitted "betaregscale" object.
model	Character: "mean" (default) or "precision".
...	Ignored.

## Value

The design matrix for the specified submodel.

## See Also

[brs](#), [formula.brs](#), [coef.brs](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brs(y ~ x1, data = prep)  
head(model.matrix(fit))  
head(model.matrix(fit, model = "precision"))
```

---

model.matrix.brsmm     *Extract design matrix*

---

### Description

Extract design matrix

### Usage

```
## S3 method for class 'brsmm'  
model.matrix(object, model = c("mean", "precision", "random"), ...)
```

### Arguments

object	A fitted "brsmm" object.
model	Character: "mean" (default), "precision", or "random".
...	Ignored.

### Value

The design matrix for the specified submodel.

### See Also

[brsmm](#), [formula.brsmm](#)

### Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
head(model.matrix(fit))  
head(model.matrix(fit, model = "random"))
```

---

nobs.brs	<i>Number of observations</i>
----------	-------------------------------

---

**Description**

Number of observations

**Usage**

```
## S3 method for class 'brs'  
nobs(object, ...)
```

**Arguments**

object	A fitted "betaregscale" object.
...	Ignored.

**Value**

Integer: number of observations.

**See Also**

[brs](#), [fitted.brs](#), [brs\\_gof](#)

**Examples**

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brs(y ~ x1, data = prep)  
nobs(fit)
```

---

nobs.brsmm	<i>Number of observations in a brsmm fit</i>
------------	--

---

## Description

Number of observations in a brsmm fit

## Usage

```
## S3 method for class 'brsmm'  
nobs(object, ...)
```

## Arguments

object	A fitted "brsmm" object.
...	Currently ignored.

## Value

Integer.

## See Also

[brsmm](#), [fitted.brsmm](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
nobs(fit)
```

plot.brs

*Diagnostic plots for beta interval regression***Description**

Produces up to six diagnostic plots for a fitted "brs" model: residuals vs indices, Cook's distance, residuals vs linear predictor, residuals vs fitted values, a half-normal plot with simulated envelope, and predicted vs observed.

**Usage**

```
## S3 method for class 'brs'
plot(
  x,
  which = 1:4,
  type = "rqr",
  nsim = 100L,
  level = 0.9,
  caption = c("Residuals vs indices", "Cook's distance", "Residuals vs linear predictor",
             "Residuals vs fitted values", "Half-normal plot", "Predicted vs observed"),
  sub.caption = NULL,
  ask = prod(par("mfcol")) < length(which) && dev.interactive(),
  gg = FALSE,
  title = NULL,
  theme = NULL,
  ...
)
```

**Arguments**

x	A fitted "brs" object.
which	Integer vector selecting which plots to draw (default 1:4).
type	Character: residual type passed to <code>residuals.brs</code> (default "rqr").
nsim	Integer: number of simulations for the half-normal envelope (default 100).
level	Numeric: confidence level for the envelope (default 0.9).
caption	Character vector of panel captions.
sub.caption	Subtitle; defaults to the model call.
ask	Logical: prompt before each page of plots?
gg	Logical: use ggplot2? (default FALSE).
title	Optional global title for ggplot output. If NULL, panel captions are used.
theme	Optional ggplot2 theme object (e.g., <code>ggplot2::theme_bw()</code> ). If NULL, a minimal theme is used.
...	Further arguments passed to base <code>plot()</code> .

**Value**

Invisibly returns `x`.

**See Also**

[brs](#), [residuals.brs](#), [autoplot.brs](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
plot(fit, which = 1:4)
```

---

plot.brsmm

*Diagnostic plots for mixed beta interval regression*


---

**Description**

Produces diagnostic plots for fitted "brsmm" models: residuals vs indices, Cook's distance, residuals vs linear predictor, residuals vs fitted values, half-normal envelope, and predicted vs observed.

**Usage**

```
## S3 method for class 'brsmm'
plot(
  x,
  which = 1:4,
  type = c("response", "pearson"),
  nsim = 100L,
  level = 0.9,
  caption = c("Residuals vs indices", "Cook's distance", "Residuals vs linear predictor",
    "Residuals vs fitted values", "Half-normal plot", "Predicted vs observed",
    "Random-effects Q-Q", "Random-effects caterpillar"),
  sub.caption = NULL,
  ask = prod(par("mfcol")) < length(which) && dev.interactive(),
  gg = FALSE,
  title = NULL,
  theme = NULL,
  ...
)
```

**Arguments**

x	A fitted "brsmm" object.
which	Integer vector selecting which panels to draw (default 1:4).
type	Residual type passed to <code>residuals.brsmm</code> ("response" or "pearson").
nsim	Number of simulations for half-normal envelope.
level	Confidence level for the half-normal envelope.
caption	Character vector of plot captions.
sub.caption	Optional subtitle; defaults to model call.
ask	Logical: prompt before each new page?
gg	Logical: use ggplot2 backend?
title	Optional global title for ggplot output. If NULL, panel captions are used.
theme	Optional ggplot2 theme object (e.g., <code>ggplot2::theme_bw()</code> ). If NULL, a minimal theme is used.
...	Further arguments passed to base <code>plot()</code> .

**Value**

Invisibly returns x.

**See Also**

[brsmm](#), [residuals.brsmm](#), [autoplot.brsmm](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
plot(fit, which = 1:4)
```

---

predict.brs	<i>Predict from a fitted model</i>
-------------	------------------------------------

---

## Description

Predict from a fitted model

## Usage

```
## S3 method for class 'brs'
predict(
  object,
  newdata = NULL,
  type = c("response", "link", "precision", "variance", "quantile"),
  at = 0.5,
  ...
)
```

## Arguments

object	A fitted "betaregscale" object.
newdata	Optional data frame for prediction.
type	Prediction type: "response" (default), "link", "precision", "variance", or "quantile".
at	Numeric vector of probabilities for quantile predictions (default 0.5).
...	Currently ignored.

## Value

Numeric vector or matrix.

## See Also

[brs](#), [fitted.brs](#), [brs\\_predict\\_scoreprob](#)

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
head(predict(fit))
```

```
head(predict(fit, type = "precision"))
newdat <- data.frame(x1 = c(1, 2))
predict(fit, newdata = newdat)
```

---

predict.brsmm	<i>Predict from a brsmm model</i>
---------------	-----------------------------------

---

### Description

Predict from a brsmm model

### Usage

```
## S3 method for class 'brsmm'
predict(
  object,
  newdata = NULL,
  type = c("response", "link", "precision", "variance", "quantile"),
  at = 0.5,
  ...
)
```

### Arguments

object	A fitted "brsmm" object.
newdata	Optional data frame.
type	Character: "response" (default), "link", "precision", "variance", or "quantile".
at	Numeric vector of probabilities for quantile predictions (default 0.5).
...	Currently ignored.

### Value

Numeric vector.

### See Also

[brsmm](#), [fitted.brsmm](#), [brs\\_predict\\_scoreprob](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
head(predict(fit))
head(predict(fit, type = "precision"))

```

---

print.brs

*Print a fitted model (brief betareg style)*


---

**Description**

Print a fitted model (brief betareg style)

**Usage**

```

## S3 method for class 'brs'
print(x, digits = max(3, getOption("digits") - 3), ...)

```

**Arguments**

x	A fitted "betaregscale" object.
digits	Number of significant digits.
...	Included for consistency with generic methods. Currently passed to internal methods where applicable.

**Value**

Invisibly returns the input object x. The function is called for its side effect of printing a formatted summary of the fitted model to the console, including the model call, mean coefficients (with link function), and precision coefficients (with link function).

**See Also**

[summary.brs](#), [print.summary.brs](#), [brs](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
print(fit)
```

---

print.brsmm

*Print a fitted brsmm model*

---

**Description**

Print a fitted brsmm model

**Usage**

```
## S3 method for class 'brsmm'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

**Arguments**

x	A fitted "brsmm" object.
digits	Number of digits.
...	Included for consistency with generic methods.

**Value**

Invisibly returns x.

**See Also**

[summary.brsmm](#), [print.summary.brsmm](#), [brsmm](#)

**Examples**

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
```

```

  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
print(fit)

```

---

```
print.brsmm_re_study Print a random-effects study
```

---

### Description

Prints a compact summary of the random-effects study returned by `brsmm_re_study`, including per-term standard deviations, shrinkage ratios, Shapiro-Wilk p-values, and the estimated covariance and correlation matrices.

### Usage

```

## S3 method for class 'brsmm_re_study'
print(x, digits = max(3, getOption("digits") - 3), ...)

```

### Arguments

<code>x</code>	A "brsmm_re_study" object returned by <code>brsmm_re_study</code> .
<code>digits</code>	Integer: number of significant digits for rounding (default <code>max(3, getOption("digits") - 3)</code> ).
<code>...</code>	Currently ignored.

### Value

Invisibly returns `x`. Called for its side-effect of printing the study to the console.

### See Also

[brsmm\\_re\\_study](#), [brsmm](#), [ranef.brsmm](#)

### Examples

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)

```

```
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
rs <- brsmm_re_study(fit)
print(rs)
```

---

```
print.summary.brs      Print a model summary (betareg style)
```

---

## Description

Print a model summary (betareg style)

## Usage

```
## S3 method for class 'summary.brs'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

## Arguments

<code>x</code>	A "summary.betaregscale" object.
<code>digits</code>	Number of digits.
<code>...</code>	Passed to printCoefmat.

## Value

Invisibly returns the input object `x`. The function is called for its side effect of printing a comprehensive summary to the console, including the model call, quantile residuals, coefficient tables for mean and precision submodels with significance stars, goodness-of-fit statistics (log-likelihood, pseudo R-squared), optimization details, and censoring information.

## See Also

[summary.brs](#), [brs](#), [print.brs](#)

## Examples

```
dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
print(summary(fit))
```

---

print.summary.brsmm *Print summary for brsmm models*

---

## Description

Print summary for brsmm models

## Usage

```
## S3 method for class 'summary.brsmm'  
print(x, digits = max(3, getOption("digits") - 3), ...)
```

## Arguments

x	A "summary.brsmm" object.
digits	Number of digits.
...	Passed to printCoefmat.

## Value

Invisibly returns x.

## See Also

[summary.brsmm](#), [brsmm](#), [print.brsmm](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
print(summary(fit))
```

---

ranef	<i>Extract random effects</i>
-------	-------------------------------

---

### Description

Generic function for extracting random effects.

### Usage

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

### Arguments

object	A fitted model object.
...	Additional arguments passed to methods.

### Value

Method-specific; for "brsmm" objects, a matrix or named numeric vector of group-specific random-effect modes.

### See Also

[ranef.brsmm](#), [brsmm\\_re\\_study](#)

### Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
ranef(fit)
```

---

ranef.brsmm	<i>Extract random effects from a brsmm model</i>
-------------	--

---

## Description

Extract random effects from a brsmm model

## Usage

```
## S3 method for class 'brsmm'  
ranef(object, ...)
```

## Arguments

object	A fitted "brsmm" object.
...	Currently ignored.

## Value

A matrix or named numeric vector of group-specific random-effect posterior modes.

## See Also

[brsmm](#), [brsmm\\_re\\_study](#), [ranef](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
ranef(fit)
```

---

residuals.brs	<i>Extract residuals</i>
---------------	--------------------------

---

## Description

Extract residuals

## Usage

```
## S3 method for class 'brs'
residuals(
  object,
  type = c("response", "pearson", "deviance", "rqr", "weighted", "sweighted"),
  ...
)
```

## Arguments

object	A fitted "betaregyscale" object.
type	Residual type. One of "response" (default), "pearson", "deviance", "rqr" (randomized quantile), "weighted", or "sweighted".
...	Currently ignored.

## Details

For Pearson residuals the variance formula depends on the reparameterization stored in `object$repar`:

**repar = 1 (precision)**  $V = \mu(1 - \mu) / (1 + \phi)$

**repar = 2 (mean-variance)**  $V = \mu(1 - \mu) * \phi$

The weighted and sweighted residuals use the digamma/trigamma formulation from the precision parameterization (`repar = 1`), so internal conversion is applied when `repar != 1`.

## Value

Numeric vector of residuals.

## See Also

[brs](#), [fitted.brs](#), [plot.brs](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
head(residuals(fit))
head(residuals(fit, type = "pearson"))

```

---

residuals.brsmm	<i>Residuals from a brsmm model</i>
-----------------	-------------------------------------

---

**Description**

Residuals from a brsmm model

**Usage**

```

## S3 method for class 'brsmm'
residuals(
  object,
  type = c("response", "pearson", "deviance", "rqr", "weighted", "sweighted"),
  ...
)

```

**Arguments**

object	A fitted "brsmm" object.
type	Character: "response" (default), "pearson", "deviance", "rqr", "weighted", or "sweighted".
...	Currently ignored.

**Value**

Numeric vector.

**See Also**

[brsmm](#), [fitted.brsmm](#), [plot.brsmm](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
head(residuals(fit))
head(residuals(fit, type = "pearson"))

```

summary.brs

*Summarize a fitted model (betareg style)***Description**

Summarize a fitted model (betareg style)

**Usage**

```

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

```

**Arguments**

object	A fitted "betaregscale" object.
...	Ignored.

**Value**

A list of class "summary.betaregscale".

**See Also**

[brs](#), [print.summary.brs](#), [brs\\_est](#), [brs\\_gof](#)

**Examples**

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10)
)

```

```

)
prep <- brs_prep(dat, ncuts = 100)
fit <- brs(y ~ x1, data = prep)
s <- summary(fit)
s$coefficients$mean

```

---

summary.brsmm	<i>Summarize a fitted brsmm model</i>
---------------	---------------------------------------

---

### Description

Summarize a fitted brsmm model

### Usage

```

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

```

### Arguments

object	A fitted "brsmm" object.
...	Currently ignored.

### Value

Object of class "summary.brsmm".

### See Also

[brsmm](#), [print.summary.brsmm](#), [brs\\_gof](#), [brsmm\\_re\\_study](#)

### Examples

```

dat <- data.frame(
  y = c(
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15
  ),
  x1 = rep(c(1, 2), 10),
  id = factor(rep(1:4, each = 5))
)
prep <- brs_prep(dat, ncuts = 100)
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)
s <- summary(fit)
s$coefficients$mean

```

---

`vcov.brs`*Variance-covariance matrix of estimated coefficients*

---

## Description

Variance-covariance matrix of estimated coefficients

## Usage

```
## S3 method for class 'brs'  
vcov(object, model = c("full", "mean", "precision"), ...)
```

## Arguments

<code>object</code>	A fitted "betaregscale" object.
<code>model</code>	Character: which component ("full", "mean", or "precision").
<code>...</code>	Ignored.

## Value

A square numeric matrix.

## See Also

[brs](#), [coef.brs](#), [confint.brs](#)

## Examples

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10)  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brs(y ~ x1, data = prep)  
vcov(fit)  
vcov(fit, model = "mean")
```

---

vcov.brsmm	<i>Variance-covariance matrix for brsmm coefficients</i>
------------	--

---

**Description**

Variance-covariance matrix for brsmm coefficients

**Usage**

```
## S3 method for class 'brsmm'  
vcov(object, model = c("full", "mean", "precision", "random"), ...)
```

**Arguments**

object	A fitted "brsmm" object.
model	Character: "full", "mean", "precision", or "random".
...	Currently ignored.

**Value**

Numeric matrix.

**See Also**

[brsmm](#), [coef.brsmm](#), [confint.brsmm](#)

**Examples**

```
dat <- data.frame(  
  y = c(  
    0, 5, 20, 50, 75, 90, 100, 30, 60, 45,  
    10, 40, 55, 70, 85, 25, 35, 65, 80, 15  
  ),  
  x1 = rep(c(1, 2), 10),  
  id = factor(rep(1:4, each = 5))  
)  
prep <- brs_prep(dat, ncuts = 100)  
fit <- brsmm(y ~ x1, random = ~ 1 | id, data = prep)  
vcov(fit, model = "mean")
```

# Index

AIC.brs, [3](#), [5](#), [14](#), [52](#)  
AIC.brsmm, [4](#), [7](#), [15](#), [53](#)  
anova.brs, [5](#)  
anova.brsmm, [6](#)  
autoplot.brs, [7](#), [10](#), [11](#), [59](#)  
autoplot.brs\_bootstrap, [9](#), [9](#)  
autoplot.brs\_marginaleffects, [11](#)  
autoplot.brsmm, [12](#), [60](#)

BIC.brs, [4](#), [5](#), [14](#), [52](#)  
BIC.brsmm, [4](#), [7](#), [15](#), [53](#)  
brs, [4](#), [5](#), [9–11](#), [14](#), [16](#), [18](#), [25](#), [27–29](#), [33](#), [35](#),  
[38](#), [42](#), [44](#), [46](#), [48](#), [50](#), [52](#), [54](#), [56](#), [59](#),  
[61](#), [63](#), [66](#), [70](#), [72](#), [74](#)  
brs\_bootstrap, [9](#), [10](#), [17](#)  
brs\_cens, [19](#)  
brs\_check, [19](#), [20](#), [21](#), [34](#), [35](#), [38](#)  
brs\_coef, [23](#)  
brs\_cv, [25](#)  
brs\_est, [23](#), [24](#), [26](#), [28](#), [29](#), [44](#), [46](#), [72](#)  
brs\_fit\_fixed, [16](#)  
brs\_fit\_var, [16](#)  
brs\_gof, [4](#), [14](#), [15](#), [24](#), [27](#), [27](#), [52](#), [53](#), [56](#), [72](#),  
[73](#)  
brs\_hessian, [27](#), [28](#), [28](#)  
brs\_marginaleffects, [11](#), [29](#)  
brs\_predict\_scoreprob, [31](#), [61](#), [62](#)  
brs\_prep, [20](#), [22](#), [23](#), [33](#)  
brs\_repar, [36](#)  
brs\_sim, [18](#), [22](#), [23](#), [37](#)  
brs\_table, [39](#)  
brsmm, [4](#), [7](#), [15](#), [41](#), [43](#), [45](#), [47](#), [49](#), [51](#), [53](#), [55](#),  
[57](#), [60](#), [62](#), [64](#), [65](#), [67](#), [69](#), [71](#), [73](#), [75](#)  
brsmm\_re\_study, [43](#), [65](#), [68](#), [69](#), [73](#)

coef.brs, [44](#), [46](#), [50](#), [54](#), [74](#)  
coef.brsmm, [45](#), [47](#), [75](#)  
confint.brs, [18](#), [46](#), [74](#)  
confint.brsmm, [45](#), [47](#), [75](#)

fitted.brs, [48](#), [56](#), [61](#), [70](#)  
fitted.brsmm, [49](#), [57](#), [62](#), [71](#)  
Formula, [16](#)  
formula.brs, [50](#), [54](#)  
formula.brsmm, [51](#), [55](#)

logLik.brs, [4](#), [5](#), [14](#), [52](#)  
logLik.brsmm, [4](#), [7](#), [15](#), [53](#)

model.matrix.brs, [50](#), [54](#)  
model.matrix.brsmm, [51](#), [55](#)

nobs.brs, [56](#)  
nobs.brsmm, [57](#)

optim, [42](#)

plot.brs, [9](#), [58](#), [70](#)  
plot.brsmm, [59](#), [71](#)  
predict.brs, [48](#), [61](#)  
predict.brsmm, [49](#), [62](#)  
print.brs, [63](#), [66](#)  
print.brs\_bootstrap (brs\_bootstrap), [17](#)  
print.brsmm, [64](#), [67](#)  
print.brsmm\_re\_study, [65](#)  
print.summary.brs, [63](#), [66](#), [72](#)  
print.summary.brsmm, [64](#), [67](#), [73](#)

ranef, [68](#), [69](#)  
ranef.brsmm, [43](#), [65](#), [68](#), [69](#)  
residuals.brs, [8](#), [48](#), [58](#), [59](#), [70](#)  
residuals.brsmm, [13](#), [49](#), [60](#), [71](#)

summary.brs, [23](#), [24](#), [27](#), [63](#), [66](#), [72](#)  
summary.brsmm, [64](#), [67](#), [73](#)

vcov.brs, [29](#), [44](#), [46](#), [74](#)  
vcov.brsmm, [45](#), [47](#), [75](#)