Package 'FlexReg'

February 4, 2026

Title Regression Models for Bounded Continuous and Discrete Responses

Version 1.4.2

Description Functions to fit regression models for bounded continuous and discrete responses. In case of bounded continuous responses (e.g., proportions and rates), available models are the flexible beta (Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018) <doi:10.1214/17-BA1079>), the variance-inflated beta (Di Brisco, A. M., Migliorati, S., Ongaro, A. (2020) <doi:10.1177/1471082X18821213>), the beta (Ferrari, S.L.P., Cribari-Neto, F. (2004) <doi:10.1080/0266476042000214501>), and their augmented versions to handle the presence of zero/one values (Di Brisco, A. M., Migliorati, S. (2020) <doi:10.1002/sim.8406>) are implemented. In case of bounded discrete responses (e.g., bounded counts, such as the number of successes in n trials), available models are the flexible beta-binomial (Ascari, R., Migliorati, S. (2021) <doi:10.1002/sim.9005>), the beta-binomial, and the binomial are implemented. Inference is dealt with a Bayesian approach based on the Hamiltonian Monte Carlo (HMC) algorithm (Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (2014) <doi:10.1201/b16018>). Besides, functions to compute residuals, posterior predictives, goodness of fit measures, convergence diagnostics, and graphical representations are provided.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

Depends R (>= 3.5.0)

Imports Formula, bayesplot, ggplot2, loo, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4.0)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

NeedsCompilation yes

Author Roberto Ascari [aut, cre], Agnese M. Di Brisco [aut],

Sonia Migliorati [aut],
Andrea Ongaro [aut]

Maintainer Roberto Ascari <roberto.ascari@unimib.it>

Repository CRAN

Date/Publication 2026-02-04 15:50:02 UTC

Contents

Atomic *Atomic bombs data*

Description

Count/Percentage of chromosome aberrations in atomic bombs survivors.

Format

A `data.frame` containing 1039 observations on the following 5 variables:

`y.perc` the percentage of cells with chromosomal abnormalities.

`y` the number of cells with chromosomal abnormalities.

`n` the number of analyzed cells. It is fixed to 100 for all the survivors.

`dose` a quantitative measure of the radiation exposure level, expressed in rads.

`bomb` a factor, indicating which bomb the subject survived (H = Hiroshima, N = Nagasaki).

Details

The data have been originally analyzed by Otake and Prentice (1984) and successively by Ascari and Migliorati (2021).

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Otake, M., Prentice, R.L. (1984). The analysis of chromosomally aberrant cells based on beta-binomial distribution. Radiat Res. **98**, 456–470.

Bacteria *Bacteria data*

Description

Count/Percentage of eggs parasitized by female parasitoids.

Format

A `data.frame` containing 70 observations on the following 5 variables:

`y.perc` the percentage of parasitized eggs.

`y` the number of parasitized eggs.

`n` the maximum number of eggs that female parasitoids could parasitized. It is fixed to 128 for all the observations.

`females` the number of female parasitoids.

`females_std` the standardized version of `females`.

Details

The data have been originally analyzed by Demétrio et al (2014) and successively by Ascari and Migliorati (2021). Data come from a completely randomized experiment with 10 replicates for each specification of number of females.

Source

Demétrio et al., (2014). Models for overdispersed data in entomology.

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Demétrio, C.G.B., Hinde, J., Moral, R.A. (2014). Models for overdispersed data in entomology. Ecological Modelling Applied to Entomology. Entomology in Focus Switzerland: Springer International Publishing, 219–259.

Consumption *Italian Households Consumption data*

Description

This dataset is a subset from the 2016 Survey on Household Income and Wealth data, a statistical survey conducted by the Bank of Italy. The statistical units are the households and the head of the household is conventionally selected as the major income earner.

Format

A `data.frame` containing 568 observations on the following 8 variables:

`NComp` the number of household members.

`Sex` the sex of the head of the household.

`Age` the age of the head of the household.

`NEarners` the number of household income earners.

`Area` a factor indicating the geographical area where the household is located.

`Citizenship` a factor indicating the citizenship of the head of household.

`Income` the net disposable income.

`Consumption` the propensity to consume, defined as the percentage of `Income` that is spent rather than saved.

Details

Full data are available on the website of the Bank of Italy. `Consumption` is computed as the ratio between the amount of `consumption' over the` net disposable income'.

Source

Bank of Italy, Survey on Household Income and Wealth, 2016.

Survey description.

`convergence.diag` *Convergence diagnostics*

Description

The function returns some diagnostic measures to check for convergence to the equilibrium distribution of the Markov Chain(s). Moreover, it prints the number (and percentage) of iterations that ended with a divergence and that saturated the max treedepth, and the E-BFMI values for each chain for which E-BFMI is less than 0.2.

Usage

```
convergence.diag(
  model,
  diagnostics = "all",
  pars = NULL,
  additional.args = list()
)

## S3 method for class 'convergence.diag.flexreg'
print(x, ...)
```

Arguments

`model`	an object of class `` `flexreg` ``.
`diagnostics`	an optional character vector of diagnostics names. The default is to compute `"all"` diagnostics, otherwise one can specify a selection of diagnostics among `"Rhat"`, `"geweke"`, `"raftery"`, `"heidel"`, and `"gelman"`.
`pars`	an optional character vector of parameter names. If `pars` is not specified, all parameters in the regression models are evaluated.
`additional.args`	a list containing additional arguments (see details)
`x`	an object of class `` `convergence.diag.flexreg` ``.
`...`	additional arguments. Currently not used.

Details

- `"Rhat"` returns the potential scale reduction factor on split chains. An R-hat greater than 1 is indicative of a bad mix of the chains. At convergence R-hat has to be less than 1.05. See `rstan::Rhat` for further details.

- "geweke" returns the z-scores, one for each parameter, for a test of equality between the means of the first 10\
- "raftery" returns the estimate of the "dependence factor" I. Values of I greater than 5 may indicate a strong autocorrelation. Additional parameters such as the quantile to be estimated (q), the desired margin of error of the estimate (r), and the probability (s) of obtaining an estimate between $q - r$ and $q + r$ can be passed as a list in the additional.args argument. See coda::raftery.diag for further details.
- "heidel" returns the p-values, one for each parameter, referred to a convergence test where the null hypothesis is that the sampled values come from a stationary distribution. It is possible to set the target value for ratio of halfwidth to sample mean (eps) and the significance level of the test (pvalue) into the additional.args argument. See coda::heidel.diag for further details.
- "gelman" returns the estimate of the potential scale reduction factor and the upper confidence limit. At least two chains are needed to compute the Gelman and Rubin's convergence diagnostic. Additional parameters such as the confidence level (confidence), a logical flag indicating whether variables should be transformed (transform), a logical flag indicating whether only the second half of the series should be used in the computation (autoburnin), and a logical flag indicating whether the multivariate potential scale reduction factor should be calculated for multivariate chains (multivariate) can be passed as a list in the additional.args argument. See coda::gelman.diag for further details.

Value

A print from check_hmc_diagnostics function and a list of convergence diagnostics.

References

Brooks, SP., Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. Journal of Computational and Graphical Statistics, **7**, 434-455.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bayesian Statistics 4 (ed JM Bernado, JO Berger, AP Dawid and AFM Smith). Clarendon Press, Oxford, UK.

Heidelberger P., Welch P.D. (1981). A spectral method for confidence interval generation and run length control in simulations. Comm. ACM. **24**, 233-245.

Raftery, A.E. and Lewis, S.M. (1992). One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. Statistical Science, **7**, 493-497.

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.19.3. https://mc-stan.org

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, type = "FB")
convergence.diag(FB,  diagnostics = c("Rhat", "geweke"), pars = "beta")
```

```
## End(Not run)
```

`convergence.plot` *Convergence plots*

Description

The function produces some convergence plots from the Monte Carlo draws.

Usage

```
convergence.plot(
  model,
  file = "convergence-output.pdf",
  plotfun = "all",
  pars = NULL,
  point_est = "median",
  prob = 0.5,
  prob_outer = 0.9,
  lags = 10,
  warmup = F,
  width = 7,
  height = 7
)
```

Arguments

`model`	an object of class `` `flexreg` ``.
`file`	a character string giving the name of the file (including the extension .pdf) containing the convergence plots. If `NULL`, the convergence plots are printed in the graphics window.
`plotfun`	an optional character vector of diagnostics plots. The default is to compute `"all"` plots, otherwise one can specify a subset of plots among `"density"`, `"trace"`, `"intervals"`, `"rate"`, `"rhat"`, and `"acf"`.
`pars`	an optional character vector of parameter names. If `pars` is not specified, all parameters in the regression models are evaluated.
`point_est`	an optional character to specify the point estimate to be shown between `"median"` (the default), `"mean"`, or `"none"`.
`prob`	the probability mass to be included in the inner interval (for `"intervals"` plot) or in the shaded region (for `"density"` plot). The default is 0.5.
`prob_outer`	the probability mass to be included in the outer interval of the `"intervals"` plot. The default is 0.9.

lags	the number of lags to be shown in the "acf" plot. The default is 10.
warmup	a logical scalar indicating whether to include the warmup draws or not (default).
width, height	the width and height of the graphics region of each plot in inches. The default values are 7.

Details

- "density" returns a density plot for each parameter in pars computed from the posterior draws. See bayesplot::mcmc_areas for further details.
- "trace" returns a trace plot for each parameter in pars computed from the posterior draws. See bayesplot::mcmc_trace for further details.
- "intervals" returns a plot of uncertainty interval for each parameter in pars computed from the posterior draws. See bayesplot::mcmc_intervals for further details.
- "rate" returns a plot for each parameter in pars with the number of iterations on the x-axis and the Monte Carlo mean until iteration i-th on the y-axis.
- "rhat" returns a plot with the Rhat values for each parameter in pars. See bayesplot::mcmc_rhat for further details.
- "acf" returns the autocorrelation plots (one for each parameter in pars). See bayesplot::mcmc_acf for further details.

Moreover, the convergence plots can be further customized using the **ggplot2** package.

Value

A .pdf file with one plot per page.

References

Brooks, SP., Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. Journal of Computational and Graphical Statistics, **7**, 434-455.

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.19.3. https://mc-stan.org

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, type = "FB")
convergence.plot(FB, file = "Convergence_plot_Output.pdf", pars = "beta")

## End(Not run)
```

curve.density *Draw density plots*

Description

The function draws a curve corresponding to the probability density/mass function of the specified distribution (beta, flexible beta, variance-inflated beta, binomial, beta-binomial, or flexible beta-binomial). For beta, flexible beta, and variance-inflated beta, it also allows to include the representation of the probability of augmentation in zero and/or one values.

Usage

```
curve.density(
  type = NULL,
  size = NULL,
  mu = NULL,
  theta = NULL,
  phi = NULL,
  p = NULL,
  w = NULL,
  k = NULL,
  q0 = NULL,
  q1 = NULL,
  ...
)
```

Arguments

type	a character specifying the distribution type to be plotted ("Beta", "FB", "VIB", "Bin", "BetaBin", or "FBB").
size	the total number of trials (to be specified only if type is "Bin", "BetaBin", or "FBB").
mu	the mean parameter of the distribution. It must lie in (0, 1).
theta	the overdispersion parameter (to be specified only if type is "BetaBin" or "FBB"). It must lie in (0, 1).
phi	the precision parameter (if type is "BetaBin" or "FBB", it represents an alternative way to specify the theta parameter). It must be a real positive value.
p	the mixing weight (to be specified only if type is "FB", "VIB", or "FBB"). It must lie in (0, 1).
w	the normalized distance among component means of the FB and FBB distributions (to be specified only if type = "FB", or type = "FBB"). It must lie in (0, 1).
k	the extent of the variance inflation (to be specified only if type = "VIB"). It must lie in (0, 1).

q0	the probability of augmentation in zero (to be specified only if type is "Beta", "FB", or "VIB"). It must lie in (0, 1). In case of no augmentation, it is NULL (default).
q1	the probability of augmentation in one (to be specified only if type is "Beta", "FB", or "VIB"). It must lie in (0, 1). In case of no augmentation, it is NULL (default).
...	additional arguments of stat_function.

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Di Brisco, A. M., Migliorati, S. (2020). A new mixed-effects mixture model for constrained longitudinal data. Statistics in Medicine, **39**(2), 129–145. doi:10.1002/sim.8406

Di Brisco, A. M., Migliorati, S., Ongaro, A. (2020). Robustness against outliers: A new variance inflated regression model for proportions. Statistical Modelling, **20**(3), 274–309. doi:10.1177/1471082X18821213

Ferrari, S.L.P., and Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. Journal of Applied Statistics, **31**(7), 799–815. doi:10.1080/0266476042000214501

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018). A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Examples

```
## Not run:
curve.density("Beta", mu=.5, phi=20)
curve.density("Beta", mu=.5, phi=20, q1 = .3)
curve.density("FB", mu=.5, phi=20, p=.4, w=.8)
curve.density("FB", mu=.5, phi=20, p=.4, w=.8, q0= .1)
curve.density("VIB", mu=.5, phi=20, p=.9, k=.8, col=3)
curve.density("VIB", mu=.5, phi=20, p=.9, k=.8, col=3, q0=.1, q1=.3)

curve.density("Bin", size=10, mu=.7)
curve.density("BetaBin", size=10, mu=.7, phi=10)
curve.density("FBB", size=10, mu=.7, phi=10, p=.2,w=.7)

## End(Not run)
```

---

dBeta *The Mean-Precision Parameterized Beta Distribution*

Description

Density function, distribution function, quantile function, and random generation for the (augmented) beta distribution with the mean-precision parameterization.

Usage

```
dBeta(x, mu, phi, q0 = NULL, q1 = NULL, log = FALSE)

qBeta(prob, mu, phi, q0 = NULL, q1 = NULL, log.prob = FALSE)

pBeta(q, mu, phi, q0 = NULL, q1 = NULL, log.prob = FALSE)

rBeta(n, mu, phi, q0 = NULL, q1 = NULL)
```

Arguments

x, q	a vector of quantiles.
mu	the mean parameter. It must lie in (0, 1).
phi	the precision parameter. It must be a real positive value.
q0	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation, it is NULL (default).
q1	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation, it is NULL (default).
log	logical; if TRUE, densities are returned on log-scale.
prob	a vector of probabilities.
log.prob	logical; if TRUE, probabilities prob are given as log(prob).
n	the number of values to generate. If length(n) > 1, the length is taken to be the number required.

Details

The beta distribution has density

$$f_B(x;\mu,\phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} x^{\mu\phi-1}(1-x)^{(1-\mu)\phi-1}$$

for $0 < x < 1$, where $0 < \mu < 1$ identifies the mean and $\phi > 0$ is the precision parameter.

The augmented beta distribution has density

- q_0, if $x = 0$
- q_1, if $x = 1$
- $(1 - q_0 - q_1) f_B(x;\mu,\phi)$, if $0 < x < 1$

where $0 < q_0 < 1$ identifies the augmentation in zero, $0 < q_1 < 1$ identifies the augmentation in one, and $q_0 + q_1 < 1$.

Value

The function dBeta returns a vector with the same length as x containing the density values. The function pBeta returns a vector with the same length as q containing the values of the distribution function. The function qBeta returns a vector with the same length as prob containing the quantiles. The function rBeta returns a vector of length n containing the generated random values.

References

Ferrari, S.L.P., Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. Journal of Applied Statistics, **31**(7), 799–815. doi:10.1080/0266476042000214501

Examples

```
dBeta(x = c(.5,.7,.8), mu = .3, phi = 20)
dBeta(x = c(.5,.7,.8), mu = .3, phi = 20, q0 = .2)
dBeta(x = c(.5,.7,.8), mu = .3, phi = 20, q0 = .2, q1= .1)

qBeta(prob = c(.5,.7,.8), mu = .3, phi = 20)
qBeta(prob = c(.5,.7,.8), mu = .3, phi = 20, q0 = .2)
qBeta(prob = c(.5,.7,.8), mu = .3, phi = 20, q0 = .2, q1= .1)

pBeta(q = c(.5,.7,.8), mu = .3, phi = 20)
pBeta(q = c(.5,.7,.8), mu = .3, phi = 20, q0 = .2)
pBeta(q = c(.5,.7,.8), mu = .3, phi = 20, q0 = .2, q1= .1)

rBeta(n = 100, mu = .5, phi = 30)
rBeta(n = 100, mu = .5, phi = 30, q0 = .2, q1 = .1)
```

---

dBetaBin *The Beta-Binomial Distribution*

Description

Mass function, distribution function, quantile function, and random generation for the beta-binomial distribution.

Usage

```
dBetaBin(x, size, mu, theta = NULL, phi = NULL, log = FALSE)

qBetaBin(prob, size, mu, theta = NULL, phi = NULL, log.prob = FALSE)

pBetaBin(q, size, mu, theta = NULL, phi = NULL, log.prob = FALSE)

rBetaBin(n, size = NULL, mu = NULL, theta = NULL, phi = NULL)
```

Arguments

x, q	a vector of quantiles.
size	the total number of trials.
mu	the mean parameter. It must lie in (0, 1).
theta	the overdispersion parameter. It must lie in (0, 1).
phi	the precision parameter, an alternative way to specify the overdispersion parameter theta. It must be a real positive value.
log	logical; if TRUE, probabilities are returned on log-scale.
prob	a vector of probabilities.
log.prob	logical; if TRUE, probabilities prob are given as log(prob).
n	the number of values to generate. If length(n) > 1, the length is taken to be the number required.

Details

The beta-binomial distribution has probability mass function

$$f_{BB}(x;\mu,\phi) = \binom{n}{x} \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} \frac{\Gamma(\mu\phi+x)\Gamma((1-\mu)\phi+n-x)}{\Gamma(\phi+n)},$$

for $x \in \{0, 1, \ldots, n\}$, where $0 < \mu < 1$ identifies the mean and $\phi = (1-\theta)/\theta > 0$ is the precision parameter.

Value

The function dBetaBin returns a vector with the same length as x containing the probability mass values. The function pBetaBin returns a vector with the same length as q containing the values of the distribution function. The function qBetaBin returns a vector with the same length as prob containing the quantiles. The function rBetaBin returns a vector of length n containing the generated random values.

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Examples

```
dBetaBin(x = 5, size = 10, mu = .3, phi = 10)
dBetaBin(x = 5, size = 10, mu = .3, theta = 1/(10+1))

qBetaBin(prob = .5, size = 10, mu = .3, phi = 10)
qBetaBin(prob = .5, size = 10, mu = .3, theta = 1/(10+1))

pBetaBin(q = 5, size = 10, mu = .3, phi = 10)
pBetaBin(q = 5, size = 10, mu = .3, theta = 1/(10+1))

rBetaBin(n = 100, size = 40, mu = .5, theta = .4)
```

```
rBetaBin(n = 100, size = 40, mu = .5, phi = 1.5)
```

---

dFB *The Flexible Beta Distribution*

Description

Density function, distribution function, quantile function, and random generation for the (augmented) flexible beta distribution.

Usage

```
dFB(x, mu, phi, p, w, q0 = NULL, q1 = NULL, log = FALSE)

qFB(prob, mu, phi, p, w, q0 = NULL, q1 = NULL, log.prob = FALSE)

pFB(q, mu, phi, p, w, q0 = NULL, q1 = NULL, log.prob = FALSE)

rFB(n, mu, phi, p, w, q0 = NULL, q1 = NULL)
```

Arguments

x, q	a vector of quantiles.
mu	the mean parameter. It must lie in (0, 1).
phi	the precision parameter. It must be a real positive value.
p	the mixing weight. It must lie in (0, 1).
w	the normalized distance among component means. It must lie in (0, 1).
q0	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation, it is NULL (default).
q1	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation, it is NULL (default).
log	logical; if TRUE, densities are returned on log-scale.
prob	a vector of probabilities.
log.prob	logical; if TRUE, probabilities prob are given as log(prob).
n	the number of values to generate. If length(n) > 1, the length is taken to be the number required.

Details

The FB distribution is a special mixture of two beta distributions with probability density function

$$f_{FB}(x; \mu, \phi, p, w) = p f_B(x; \lambda_1, \phi) + (1-p) f_B(x; \lambda_2, \phi),$$

for $0 < x < 1$, where $f_B(x; \cdot, \cdot)$ is the beta density with a mean-precision parameterization. Moreover, $0 < \mu = p\lambda_1 + (1-p)\lambda_2 < 1$ is the overall mean, $\phi > 0$ is a precision parameter, $0 < p < 1$ is the mixing weight, $0 < w < 1$ is the normalized distance between component means, and $\lambda_1 = \mu + (1-p)w$ and $\lambda_2 = \mu - pw$ are the means of the first and second component of the mixture, respectively.

The augmented FB distribution has density

- q_0, if $x = 0$
- q_1, if $x = 1$
- $(1 - q_0 - q_1) f_{FB}(x; \mu, \phi, p, w)$, if $0 < x < 1$

where $0 < q_0 < 1$ identifies the augmentation in zero, $0 < q_1 < 1$ identifies the augmentation in one, and $q_0 + q_1 < 1$.

Value

The function `dFB` returns a vector with the same length as `x` containing the density values. The function `pFB` returns a vector with the same length as `q` containing the values of the distribution function. The function `qFB` returns a vector with the same length as `prob` containing the quantiles. The function `rFB` returns a vector of length `n` containing the generated random values.

References

Di Brisco, A. M., Migliorati, S. (2020). A new mixed-effects mixture model for constrained longitudinal data. Statistics in Medicine, **39**(2), 129–145. doi:10.1002/sim.8406

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018). A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Examples

```
dFB(x = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5)
dFB(x = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5, q0 = .2)
dFB(x = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5, q0 = .2, q1 = .1)

qFB(prob = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5)
qFB(prob = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5, q0 = .2)
qFB(prob = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5, q0 = .2, q1 = .1)

pFB(q = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5)
pFB(q = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5, q0 = .2)
pFB(q = c(.5,.7,.8), mu = .3, phi = 20, p = .5, w = .5, q0 = .2, q1 = .1)

rFB(n = 100, mu = .5, phi = 30,p = .3, w = .6)
rFB(n = 100, mu = .5, phi = 30,p = .3, w = .6, q0 = .2, q1 = .1)
```

dFBB *The Flexible Beta-Binomial Distribution*

Description

Mass function, distribution function, quantile function, and random generation for the flexible beta-binomial distribution.

Usage

```
dFBB(x, size, mu, theta = NULL, phi = NULL, p, w, log = FALSE)

qFBB(prob, size, mu, theta = NULL, phi = NULL, p, w, log.prob = FALSE)

pFBB(q, size, mu, theta = NULL, phi = NULL, p, w, log.prob = FALSE)

rFBB(n, size = NULL, mu, theta = NULL, phi = NULL, p, w)
```

Arguments

x, q	a vector of quantiles.
size	the total number of trials.
mu	the mean parameter. It must lie in (0, 1).
theta	the overdispersion parameter. It must lie in (0, 1).
phi	the precision parameter, an alternative way to specify the overdispersion parameter theta. It must be a real positive value.
p	the mixing weight. It must lie in (0, 1).
w	the normalized distance among component means. It must lie in (0, 1).
log	logical; if TRUE, probabilities are returned on log-scale.
prob	a vector of probabilities.
log.prob	logical; if TRUE, probabilities prob are given as log(prob).
n	the number of values to generate. If length(n) > 1, the length is taken to be the number required.

Details

The FBB distribution is a special mixture of two beta-binomial distributions with probability mass function

$$f_{FBB}(x; \mu, \phi, p, w) = pBB(x; \lambda_1, \phi) + (1-p)BB(x; \lambda_2, \phi),$$

for $x \in \{0, 1, \ldots, n\}$, where $BB(x; \cdot, \cdot)$ is the beta-binomial distribution with a mean-precision parameterization. Moreover, $\phi = (1-\theta)/\theta > 0$ is a precision parameter, $0 < p < 1$ is the mixing weight, $0 < \mu = p\lambda_1 + (1-p)\lambda_2 < 1$ is the overall mean, $0 < w < 1$ is the normalized distance between component means, and $\lambda_1 = \mu + (1-p)w$ and $\lambda_2 = \mu - pw$ are the scaled means of the first and second component of the mixture, respectively.

Value

The function dFBB returns a vector with the same length as x containing the probability mass values. The function pFBB returns a vector with the same length as q containing the values of the distribution function. The function qFBB returns a vector with the same length as prob containing the quantiles. The function rFBB returns a vector of length n containing the generated random values.

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Examples

```
dFBB(x = c(5,7,8), size=10, mu = .3, phi = 20, p = .5, w = .5)
dFBB(x = c(5,7,8), size=10, mu = .3, theta = 1/(20+1), p = .5, w = .5)

qFBB(prob = .5, size=10, mu = .3, phi = 20, p = .5, w = .5)
qFBB(prob = .5, size=10, mu = .3, theta = 1/(20+1), p = .5, w = .5)

pFBB(q = c(5,7,8), size=10, mu = .3, phi = 20, p = .5, w = .5)
pFBB(q = c(5,7,8), size=10, mu = .3, theta = 1/(20+1), p = .5, w = .5)

rFBB(n = 100, size = 40, mu = .5, phi = 5, p = .3, w = .6)
rFBB(n = 100, size = 40, mu = .5, theta = 1/(5+1), p = .3, w = .6)
```

---

dVIB *The Variance-Inflated Beta Distribution*

Description

Density function, distribution function, quantile function, and random generation for the (augmented) variance-inflated beta distribution.

Usage

```
dVIB(x, mu, phi, p, k, q0 = NULL, q1 = NULL, log = FALSE)

qVIB(prob, mu, phi, p, k, q0 = NULL, q1 = NULL, log.prob = FALSE)

pVIB(q, mu, phi, p, k, q0 = NULL, q1 = NULL, log.prob = FALSE)

rVIB(n, mu, phi, p, k, q0 = NULL, q1 = NULL)
```

Arguments

x, q	a vector of quantiles.
mu	the mean parameter. It must lie in (0, 1).
phi	the precision parameter. It must be a real positive value.
p	the mixing weight. It must lie in (0, 1).
k	the extent of the variance inflation. It must lie in (0, 1).
q0	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation, it is NULL (default).
q1	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation, it is NULL (default).
log	logical; if TRUE, densities are returned on log-scale.
prob	a vector of probabilities.
log.prob	logical; if TRUE, probabilities prob are given as log(prob).
n	the number of values to generate. If length(n) > 1, the length is taken to be the number required.

Details

The VIB distribution is a special mixture of two beta distributions with probability density function

$$f_{VIB}(x;\mu,\phi,p,k) = pf_B(x;\mu,\phi k) + (1-p)f_B(x;\mu,\phi),$$

for $0 < x < 1$, where $f_B(x;\cdot,\cdot)$ is the beta density with a mean-precision parameterization. Moreover, $0 < p < 1$ is the mixing weight, $0 < \mu < 1$ represents the overall (as well as mixture component) mean, $\phi > 0$ is a precision parameter, and $0 < k < 1$ determines the extent of the variance inflation. The augmented VIB distribution has density

- q_0, if $x = 0$
- q_1, if $x = 1$
- $(1 - q_0 - q_1)f_{VIB}(x;\mu,\phi,p,k)$, if $0 < x < 1$

where $0 < q_0 < 1$ identifies the augmentation in zero, $0 < q_1 < 1$ identifies the augmentation in one, and $q_0 + q_1 < 1$.

Value

The function dVIB returns a vector with the same length as x containing the density values. The function pVIB returns a vector with the same length as q containing the values of the distribution function. The function qVIB returns a vector with the same length as prob containing the quantiles. The function rVIB returns a vector of length n containing the generated random values.

References

Di Brisco, A. M., Migliorati, S., Ongaro, A. (2020). Robustness against outliers: A new variance inflated regression model for proportions. Statistical Modelling, **20**(3), 274–309. doi:10.1177/1471082X18821213

Examples

```
dVIB(x = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5)
dVIB(x = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5, q1 = .1)
dVIB(x = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5, q0 = .2, q1 = .1)

qVIB(prob = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5)
qVIB(prob = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5, q1 = .1)
qVIB(prob = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5, q0 = .2, q1 = .1)

pVIB(q = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5)
pVIB(q = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5, q1 = .1)
pVIB(q = c(.5,.7,.8), mu = .3, phi = 20, p = .5, k= .5, q0 = .2, q1 = .1)

rVIB(n = 100, mu = .5, phi = 30, p = .3, k = .6)
rVIB(n = 100, mu = .5, phi = 30, p = .3, k = .6, q0 = .2, q1 = .1)
```

Election *Italian Election Results*

Description

Data from the Italian general election held on 4 March 2018.

Format

A `data.frame` containing 232 observations on the following 13 variables:

`NVotes` the number of valid votes.

`FI` the percentage of votes got by `Forza Italia' party.} \item{\code{FDI}}{the percentage of votes got by Fratel` d'Italia' party.

`LEGA` the percentage of votes got by `Lega' party.} \item{\code{LEU}}{the percentage of votes got by` Liberi e Uguali' party.

`M5S` the percentage of votes got by `Movimento 5 Stelle' party.} \item{\code{PD}}{the percentage of votes got by` tito Democratico' party.

`Other` the percentage of votes got by other parties, including blank ballots.

`AgeInd` the age index, defined as the ratio of the number of elderly persons (aged 65 and over) to the number of young persons (from 0 to 14), divided by 10.

`PopDens` the number of inhabitants per square km.

`ER` the employment rate, defined as the ratio of the number of employed persons (aged 15-64) to the number of persons (aged 15-64).

`Illiteracy` the illiteracy rate, defined as the ratio of the number of persons without a qualification (aged 15 and over) to the total number of persons aged 15 and over.

`Foreign` the number of foreigners per 1000 inhabitants.

Details

Data are collected on the 232 electoral districts into which the Italian territory is organized. Distribution of votes for Aosta constituency is not available. Distributions of votes are available on the Italian Ministry of Interior's webpage, whereas constituencies information have been obtained from 2011 Italian Census. The count of votes got by each party can be derived by multiplying the percentage of votes and the number of valid votes.

Source

Italian Ministry of Interior's webpage: https://www.interno.gov.it/it/speciali/2018-elections.

`flexreg` *Flexible Regression Models for Bounded Continuous Responses*

Description

The function fits some flexible regression models for bounded continuous responses (e.g., proportions and rates) via a Bayesian approach to inference based on Hamiltonian Monte Carlo algorithm. Available regression models are the flexible beta regression model (`type = "FB"`, default), the variance inflated beta (`type = "VIB"`), the beta (`type = "Beta"`), as well as their augmented versions.

Usage

```
flexreg(
  formula,
  zero.formula = NULL,
  one.formula = NULL,
  data,
  type = "FB",
  link.mu = "logit",
  prior.beta = "normal",
  hyperparam.beta = NULL,
  prior.omega0 = "normal",
  hyperparam.omega0 = NULL,
  prior.omega1 = "normal",
  hyperparam.omega1 = NULL,
  hyperparam.p = NULL,
  hyperparam.w = NULL,
  hyperparam.k = NULL,
  link.phi = NULL,
  prior.phi = NULL,
  hyperparam.phi = NULL,
  prior.psi = NULL,
  hyperparam.psi = NULL,
  n.chain = 1,
  n.iter = 5000,
  warmup.perc = 0.5,
```

```
  thin = 1,
  verbose = TRUE,
  ...
)
```

Arguments

formula	an object of class "formula": a symbolic description of the mean model (y ~ x) or the mean and precision models (y ~ x | z) to be fitted (see Details).
zero.formula	an object of class "formula": a symbolic description of the zero augmented model to be fitted (see Details).
one.formula	an object of class "formula": a symbolic description of the one augmented model to be fitted (see Details).
data	an optional data.frame, list, or object that is coercible to a data.frame through as.data.frame containing the variables in the model. If not found in data, the variables in formula, zero.formula, and one.formula are taken from the environment from which the function flexreg is called.
type	a character specifying the type of regression model. Current options are "FB" (flexible beta, default), "VIB" (variance inflated beta), and "Beta".
link.mu	a character specifying the link function for the mean model (mu). Currently, "logit" (default), "probit", "cloglog", and "loglog" are supported.
prior.beta	a character specifying the prior distribution for the regression coefficients of the mean model, beta. Currently, "normal" (default) and "cauchy" are supported.
hyperparam.beta	a positive numeric (vector of length 1) specifying the hyperprior scale parameter for the prior distribution of beta regression coefficients. The default is 100 if the prior is "normal", 2.5 if it is "cauchy".
prior.omega0	a character specifying the prior distribution for the regression coefficients of the augmented model in zero, omega0. Currently, "normal" (default) and "cauchy" are supported.
hyperparam.omega0	a positive numeric (vector of length 1) specifying the hyperprior scale parameter for the prior distribution of omega0 regression coefficients. The default is 100 if the prior is "normal", 2.5 if it is "cauchy".
prior.omega1	a character specifying the prior distribution for the regression coefficients of the augmented model in one, omega1. Currently, "normal" (default) and "cauchy" are supported.
hyperparam.omega1	a positive numeric (vector of length 1) specifying the hyperprior scale parameter for the prior distribution of omega1 regression coefficients. The default is 100 if the prior is "normal", 2.5 if it is "cauchy".
hyperparam.p	a vector of length 2 with positive elements specifying the hyperparameters for the beta prior distribution of p. The default is c(0.5,2), which corresponds to the uniform prior distribution.

hyperparam.w	a vector of length 2 with positive elements specifying the hyperparameters for the beta prior distribution of w. The default is c(0.5,2), which corresponds to the uniform prior distribution.
hyperparam.k	a vector of length 2 with positive elements specifying the hyperparameters for the beta prior distribution of k. The default is c(0.5,2), which corresponds to the uniform prior distribution.
link.phi	a character specifying the link function for the precision model (phi). Currently, "identity" (default), "log", and "sqrt" are supported.
prior.phi	a character specifying the prior distribution for precision parameter phi if link.phi = "identity". Currently, "gamma" (default) and "unif" are supported.
hyperparam.phi	a positive numeric (vector of length 1) specifying the hyperprior parameter for the prior distribution of phi. If the prior is "gamma", the value identifies the gamma's shape and rate parameters (the default is 0.001). If the prior is "uniform" the hyperparameter must be specified to define the upper limit of the support of phi.
prior.psi	a character specifying the prior distribution for the regression coefficients of the precision model psi (not supported if link.phi = "identity"). Currently, "normal" (default) and "cauchy" are supported.
hyperparam.psi	a positive numeric (vector of length 1) specifying the hyperprior scale parameter for the prior distribution of psi regression coefficients. The default is 100 if the prior is "normal", 2.5 if it is "cauchy".
n.chain	a positive integer specifying the number of Markov chains. The default is 1.
n.iter	a positive integer specifying the number of iterations for each chain (including warm-up). The default is 5000.
warmup.perc	the percentage of iterations per chain to discard.
thin	a positive integer specifying the period for saving samples. The default is 1.
verbose	a logical (with default TRUE) indicating whether to print intermediate output.
...	additional arguments from sampling.

Details

Let Y be a continuous bounded random variable whose distribution can be specified in the type argument and μ be the mean of Y. The flexreg function links the parameter μ to a linear predictor through a function $g_1(\cdot)$ specified in link.mu:

$$g_1(\mu) = \boldsymbol{x}^t\boldsymbol{\beta},$$

where $\boldsymbol{\beta}$ is the vector of regression coefficients for the mean model. The prior distribution and the related hyperparameter of $\boldsymbol{\beta}$ can be specified in prior.beta and hyperparam.beta, respectively. By default, the precision parameter ϕ is assumed to be constant. The prior distribution and the related hyperparameter of ϕ can be specified in prior.phi and hyperparam.phi. It is possible to extend the model by linking ϕ to an additional (possibly overlapping) set of covariates through a proper link function $g_2(\cdot)$ specified in the link.phi argument:

$$g_2(\phi) = \boldsymbol{z}^t\boldsymbol{\psi},$$

where ψ is the vector of regression coefficients for the precision model. The prior distribution and the related hyperparameter of ψ can be specified in `prior.psi` and `hyperparam.psi`. In the function `flexreg`, the regression model for the mean and, where appropriate, for the precision parameter can be specified in the `formula` argument with a formula of type `y ~ x1 + x2 | z1 + z2` where covariates on the left of "|" are included in the regression model for the mean, whereas covariates on the right of "|" are included in the regression model for the precision.

If the second part is omitted, i.e., `y ~ x1 + x2`, the precision is assumed constant for each observation.

In presence of zero values in the response, one has to link the parameter q_0, i.e., the probability of augmentation in zero, to an additional (possibly overlapping) set of covariates through a logit link function:

$$g_3(q_0) = \boldsymbol{x}_0^t \boldsymbol{\omega_0},$$

where $\boldsymbol{\omega_0}$ is the vector of regression coefficients for the augmented model in zero. The prior distribution and the related hyperparameter of $\boldsymbol{\omega_0}$ can be specified in `prior.omega0` and `hyperparam.omega0`. In presence of one values in the response, one has to link the parameter q_1, i.e., the probability of augmentation in one, to an additional (possibly overlapping) set of covariates through a logit link function:

$$g_4(q_1) = \boldsymbol{x}_1^t \boldsymbol{\omega_1},$$

where $\boldsymbol{\omega_1}$ is the vector of regression coefficients for the augmented model in one. The prior distribution and the related hyperparameter of $\boldsymbol{\omega_1}$ can be specified in `prior.omega1` and `hyperparam.omega1`. If both the augmented models in zero and one are specified, the link function is a bivariate logit. In `flexreg` function, the augmented models in zero and/or one can be specified in the `zero.formula` and/or `one.formula` arguments with a formula of type `~ x`. Left hand side in `zero.formula` and `one.formula` can be omitted; if specified, they have to be the same as left hand side in `formula`.

Value

The `flexreg` function returns an object of class `` `flexreg` ``, i.e. a list with the following elements:

`call`	the function call.
`type`	the type of regression model.
`formula`	the overall formula.
`aug`	a character specifing the absence of the augmentation (`"No"`) or the presence of augmentation in zero (`"0"`), one (`"1"`), or both (`"01"`).
`link.mu`	a character specifing the link function in the mean model.
`link.phi`	a character specifing the link function in the precision model.
`model`	a list of objects of class `` `stanfit` `` containing the fitted model(s).
`response`	the response variable, assuming values in (0, 1).
`design.X`	the design matrix for the mean model.
`design.Z`	the design matrix for the precision model (if defined).
`design.X0`	the design matrix for the augmented model in zero (if defined).
`design.X1`	the design matrix for the augmented model in one (if defined).

References

Di Brisco, A. M., Migliorati, S. (2020). A new mixed-effects mixture model for constrained longitudinal data. Statistics in Medicine, **39**(2), 129–145. doi:10.1002/sim.8406

Di Brisco, A. M., Migliorati, S., Ongaro, A. (2020). Robustness against outliers: A new variance inflated regression model for proportions. Statistical Modelling, **20**(3), 274–309. doi:10.1177/1471082X18821213

Ferrari, S.L.P., Cribari-Neto, F. (2004). Beta Regression for Modeling Rates and Proportions. Journal of Applied Statistics, **31**(7), 799–815. doi:10.1080/0266476042000214501

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018) A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, type="FB")

# Regression model with one augmentation:
AFB1 <- flexreg(accuracy ~ dyslexia | iq + dyslexia + iq:dyslexia,
one.formula = ~ iq + dyslexia, data = Reading, type="FB")

## End(Not run)
```

`flexreg_binom` *Flexible Regression Models for Bounded Discrete Responses*

Description

The function fits some flexible regression models for bounded discrete responses via a Bayesian approach to inference based on Hamiltonian Monte Carlo algorithm. Available regression models are the flexible beta-binomial (`type = "FBB"`, default), the beta-binomial (`type = "BetaBin"`), and the binomial one (`type = "Bin"`).

Usage

```
flexreg_binom(
  formula,
  data,
  type = "FBB",
  n,
  link.mu = "logit",
  prior.beta = "normal",
  hyperparam.beta = 100,
  hyperparam.theta = NULL,
```

```
  hyperparam.p = NULL,
  hyperparam.w = NULL,
  link.theta = NULL,
  prior.psi = NULL,
  hyperparam.psi = NULL,
  n.chain = 1,
  n.iter = 5000,
  warmup.perc = 0.5,
  thin = 1,
  verbose = TRUE,
  ...
)
```

Arguments

formula	an object of class "formula": a symbolic description of the model to be fitted (y ~ x or y ~ x | z, see Details).
data	an optional data.frame, list, or object that is coercible to a data.frame through as.data.frame containing the variables in the model. If not found in data, the variables in formula are taken from the environment from which the function flexreg_binom is called.
type	a character specifying the type of regression model. Current options are "FBB" (flexible beta-binomial, default), "BetaBin" (beta-binomial), and "Bin" (binomial).
n	a character specifying the name of the variable containing the total number of trials.
link.mu	a character specifying the link function for the mean model. Currently, "logit" (default), "probit", "cloglog", and "loglog" are supported.
prior.beta	a character specifying the prior distribution for the regression coefficients of the mean model, beta. Currently, "normal" (default) and "cauchy" are supported.
hyperparam.beta	a positive numeric (vector of length 1) specifying the hyperprior scale parameter for the prior distribution of beta regression coefficients. The default is 100 if the prior is "normal", 2.5 if it is "cauchy".
hyperparam.theta	a vector of length 2 with positive elements specifying the hyperparameters for the beta prior distribution of theta. The default is c(0.5,2), which corresponds to the uniform prior distribution.
hyperparam.p	a vector of length 2 with positive elements specifying the hyperparameters for the beta prior distribution of p. The default is c(0.5,2), which corresponds to the uniform prior distribution.
hyperparam.w	a vector of length 2 with positive elements specifying the hyperparameters for the beta prior distribution of w. The default is c(0.5,2), which corresponds to the uniform prior distribution.
link.theta	a character specifying the link function for the overdispersion model. Currently, "identity" (default), "logit", "probit", "cloglog", and "loglog" are supported. If link.theta = "identity", the prior distribution for theta is a beta.

prior.psi	a character specifying the prior distribution for the regression coefficients of the overdispersion model,psi. Not supported if link.theta="identity". Currently, "normal" (default) and "cauchy" are supported.
hyperparam.psi	a positive numeric (vector of length 1) specifying the hyperprior scale parameter for the prior distribution of psi regression coefficients. The default is 100 if the prior is "normal", 2.5 if it is "cauchy".
n.chain	a positive integer specifying the number of Markov chains. The default is 1.
n.iter	a positive integer specifying the number of iterations for each chain (including warm-up). The default is 5000.
warmup.perc	the percentage of iterations per chain to discard.
thin	a positive integer specifying the period for saving samples. The default is 1.
verbose	a logical (with default TRUE) indicating whether to print intermediate output.
...	additional arguments from sampling.

Details

Let Y be a random variable whose distribution can be specified in the type argument and μ be the mean of Y/n. The flexreg_binom function links the parameter μ to a linear predictor through a function $g_1(\cdot)$ specified in link.mu:

$$g_1(\mu) = x^t\boldsymbol{\beta},$$

where $\boldsymbol{\beta}$ is the vector of regression coefficients for the mean model. The prior distribution and the related hyperparameter of $\boldsymbol{\beta}$ can be specified in prior.beta and hyperparam.beta. By default, link.theta="identity", meaning that the overdispersion parameter θ is assumed to be constant. In that case, the prior distribution for θ is a beta with shape hyperparameters a and b that can be specified in hyper.theta.a and hyper.theta.b. If not specified, $a = b = 1$, otherwise if only one hyperparameter is specified, the other is set equal. It is possible to extend the model by linking θ to an additional (possibly overlapping) set of covariates through a proper link function $g_2(\cdot)$ specified in the link.theta argument:

$$g_2(\theta) = z^t\boldsymbol{\psi},$$

where $\boldsymbol{\psi}$ is the vector of regression coefficients for the overdispersion model. The prior distribution and the related hyperparameter of $\boldsymbol{\psi}$ can be specified in prior.psi and hyperparam.psi. In flexreg_binom, the regression model for the mean and, where appropriate, for the overdispersion parameter can be specified in the formula argument with a formula of type y ~ x1 + x2 | z1 + z2 where covariates on the left of "|" are included in the regression model for the mean, whereas covariates on the right of "|" are included in the regression model for the overdispersion.

If the second part is omitted, i.e., y ~ x1 + x2, the overdispersion is assumed constant for each observation.

Value

The flexreg_binom function returns an object of class `flexreg`, i.e. a list with the following elements:

call	the function call.
type	the type of regression model.

`formula`	the original formula.
`link.mu`	a character specifing the link function in the mean model.
`link.theta`	a character specifing the link function in the overdispersion model.
`model`	a list containing an object of class `` `stanfit` `` with the fitted model.
`response`	the response variable, assuming values in (0, 1).
`design.X`	the design matrix for the mean model.
`design.Z`	the design matrix for the overdispersion model (if defined).

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Examples

```
## Not run:
data(Bacteria)
fbb <- flexreg_binom(y ~ females,  n = "n", data = Bacteria, type = "FBB")

## End(Not run)
```

`model.matrix.flexreg` *Construct Design Matrices for flexreg Objects*

Description

Method for extracting design matrices from fitted regression model objects of class `` `flexreg` ``.

Usage

```
## S3 method for class 'flexreg'
model.matrix(object, ...)
```

Arguments

`object`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom` functions.
`...`	additional arguments. Currently not used.

Details

The method returns a list containing all the design matrices involved in the regression models, namely `` `X` `` (regression on the mean), `` `Z` `` (regression on the precision or overdispersion parameters), `` `X0` `` (regression on the augmentation in zero probability), and/or `` `X1` `` (regression on the augmentation in one probability).

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter = 1000)
model.matrix(FB)

## End(Not run)
```

`model_frame` *Construct Design Matrices for flexreg Objects*

Description

Method for extracting design matrix from fitted regression model objects of class `` `flexreg` ``.

Usage

```
model_frame(object, ...)
```

Arguments

`object`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom` functions.
`...`	additional arguments. Currently not used.

Details

The method returns a list containing all the design matrices involved in the regression models, namely `` `X` `` (regression on the mean), `` `Z` `` (regression on the precision or overdispersion parameters), `` `X0` `` (regression on the augmentation in zero probability), and/or `` `X1` `` (regression on the augmentation in one probability).

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter = 1000)
model.frame(FB)

## End(Not run)
```

plot.flexreg *Plot Method for* flexreg *Objects*

Description

Method for plotting regression curves for the mean from fitted regression model objects of class `flexreg`.

Usage

```
## S3 method for class 'flexreg'
plot(
  x,
  name.x,
  additional.cov.default = NA,
  smooth = TRUE,
  cluster = FALSE,
  type = "response",
  ...
)
```

Arguments

x	an object of class `flexreg`, usually the result of flexreg or flexreg_binom functions.
name.x	a character containing the name of the covariate from the mean model to be plotted on the x-axis of the scatterplot.
additional.cov.default	a list of additional covariates from the mean model and their value to be set as default.
smooth	a logical value indicating wheater the curves should be smooth (TRUE) or piece-wise linear (FALSE, default).
cluster	logical. If the model is "FB" or "FBB", cluster = TRUE plots the cluster means. By default, cluster = FALSE.
type	a vector of characters indicating the regression curves to be plotted. Available options are "response" and "response.aug" for augmented models.
...	additional arguments. Currently not used.

Details

The function produces a scatterplot of the covariate from the mean model specified in name.x and y or y/n if the response is bounded continuous or discrete, respectively. Any other variable specified in the mean model must be set to a default through the additional.cov.default argument. The argument type = "response" plots the conditional mean curve (i.e., μ), whereas the argument type = "response.aug", available only for augmented models, plots the augmented mean curve. If the regression model is of "FB" or "FBB" type and cluster = TRUE, then the function returns two additional curves corresponding to the component means, i.e., λ_1 and λ_2.

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq + dyslexia, data = Reading)
plot(FB, name.x="iq", additional.cov.default = list("dyslexia"=1))

## End(Not run)
```

`plot.flexreg_postpred` *Plot Method for* ‘`flexreg_postpred`‘ *objects*

Description

Method for an object of class `` `flexreg_postpred` `` containing the simulated posterior predictive distribution, usually the result of `posterior_predict` function. The plot shows the posterior predictive interval for each statistical unit. Additionally, the mean of the posterior predictives and the values of the observed response (either y or y/n for bounded continuous or discrete responses, respectively) can be added.

Usage

```
## S3 method for class 'flexreg_postpred'
plot(x, prob = 0.9, p_mean = F, response = NULL, ...)
```

Arguments

`x`	an object of class `` `flexreg_postpred` `` containing the simulated posterior predictives, usually the result of `posterior_predict`.
`prob`	the interval probability for the posterior predictives (default is 0.9).
`p_mean`	a logical value indicating whether the posterior predictives' mean should be plotted.
`response`	a numerical vector containing the response (either y or y/n for bounded continuous or discrete responses, respectively) to be added to the plot. If `NULL`, observed values are not plotted.
`...`	additional arguments. Currently not used.

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter=1000)
pp <- posterior_predict(FB)
plot(pp)

## End(Not run)
```

`posterior_predict.flexreg` *Posterior Predictive Method for* `'flexreg'` *objects*

Description

The function takes an object of class `` `flexreg` `` and generates values from the posterior predictive distribution.

Usage

```
## S3 method for class 'flexreg'
posterior_predict(model, newdata = NULL, n.new = NULL)
```

Arguments

`model`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom` functions.
`newdata`	an optional `data.frame` containing variables with which to predict. If omitted, the fitted values are used.
`n.new`	an optional vector containing the total number of trials with which to predict. It must be specified if `newdata` is not `NULL` and the `flexreg` object is the result of the `flexreg_binom` function (i.e., the fitted model is binomial, beta-binomial, or flexible beta-binomial). The vector must have the same length as `nrow(newdata)`.

Details

The function generates values from the posterior predictive distribution, which is the distribution of a future outcome given the observed data. The posterior predictive distribution is computed for y in case of bounded continuous responses and for y/n in case of bounded discrete responses.

Value

An object of class `` `flexreg_postpred` `` containing a matrix with the simulated posterior predictions. Each column refers to a statistical unit to predict.

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Di Brisco, A. M., Migliorati, S., Ongaro, A. (2020). Robustness against outliers: A new variance inflated regression model for proportions. Statistical Modelling, **20**(3), 274–309. doi:10.1177/1471082X18821213

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (2014). Bayesian Data Analysis, 3th edition. Chapman and Hall/CRC. doi:10.1201/b16018

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018). A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter=1000)
pp <- posterior_predict(FB)
plot(pp)

## End(Not run)
```

`predict.flexreg` *Predict Method for* ‘`flexreg`‘ *Objects*

Description

Method that computes various types of predictions from objects of class `` `flexreg` ``.

Usage

```
## S3 method for class 'flexreg'
predict(
  object,
  newdata = NULL,
  n.new = NULL,
  cluster = FALSE,
  type = "response",
  estimate = "mean",
  q = NULL,
  ...
)
```

Arguments

`object`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom` functions.
`newdata`	an optional `data.frame` containing variables with which to predict. If omitted, the fitted values are used.
`n.new`	an optional vector containing the total number of trials with which to predict. It must be specified if `newdata` is not `NULL` and the `flexreg` object is the result of the `flexreg_binom` function (i.e., the fitted model is binomial, beta-binomial, or flexible beta-binomial). The vector must have the same length as `nrow(newdata)`.

cluster	a logical (with default FALSE). The option cluster = TRUE is available only for "FB" and "FBB" models and allows to compute some component-specific predictions (see Details).
type	a character indicating the type of prediction. Available options are: "response", returning the marginal fitted mean of the response/relative response; "link", returning the linear predictor of the mean model; "precision", returning the fitted precision parameter; "overdispersion", returning the fitted overdispersion parameter; "variance", returning the fitted variance of the response.
estimate	a character indicating the type of estimate. Available options are "mean" (default), "median", and "quantile".
q	if estimate = "quantile", a numeric value of probability in (0, 1).
...	additional arguments. Currently not used.

Details

The predict method computes various types of predictions from objects of class `flexreg`. If type = "response", the function returns the marginal mean, i.e., μ. In case of models for continuous bounded responses with augmentation, the function returns also the overall mean $q_1 + (1 - q_0 - q_1)\mu$ and the probabilities of augmentation q_0 and/or q_1. If type = "variance", the function returns $Var(Y|0 < Y < 1)$ in case of no augmentation and $(1 - q_0 - q_1)Var(Y|0 < Y < 1) + q_1^2 + (1 - q_0 - q_1)\mu^2 - (q_1 + (1 - q_0 - q_1)\mu)^2$ in case of augmentation. If cluster = TRUE, for FB and FBB models, the function returns the cluster means (λ_1 and λ_2) when type = "response" and the cluster variances when type = "variance".

The option type = "overdispersion" is available only for beta-binomial and flexible beta-binomial models and returns the fitted overdispersion.

Value

The function returns a data.frame of different dimensions depending on the type of prediction.

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Di Brisco, A. M., Migliorati, S. (2020). A new mixed-effects mixture model for constrained longitudinal data. Statistics in Medicine, **39**(2), 129–145. doi:10.1002/sim.8406

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018). A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data=Reading, type="FB")
predict(FB, type="response", cluster=TRUE)
```

```
## End(Not run)
```

`print.flexreg` *Print Methods for flexreg Objects*

Description

Print Methods for flexreg Objects

Usage

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

Arguments

`x`	an object of class `` `flexreg` ``.
`...`	additional arguments. Currently not used.

`R2_bayes` *Bayesian R-squared for* `flexreg` *Objects*

Description

Bayesian version of R-squared for flexible regression models for bounded continuous and discrete responses.

Usage

```
R2_bayes(model)
```

Arguments

`model`	an object (or a list of objects) of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom` functions.

Details

The function provides a Bayesian version of the R-squared measure, defined as the variance of the predicted values divided by itself plus the expected variance of the errors.

Value

A list with the same length as the number of objects of class `flexreg` passed in the `model` argument. Each element of the list contains a vector of Bayesian R-squared values with the same length as the Markov Chain(s) after warmup.

References

Gelman, A., Goodrich, B., Gabry, J., Vehtari, A. (2019). R-squared for Bayesian Regression Models, The American Statistician, 73:3, 307–309. doi: 10.1080/00031305.2018.1549100

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, type = "FB", n.iter=1000)
hist(R2_bayes(FB)[[1]])

## End(Not run)
```

`Reading` *Reading Skills data*

Description

Data for assessing the contribution of non-verbal IQ to children's reading skills in dyslexic and non-dyslexic children.

Format

A `data.frame` containing 44 observations on 4 variables.

`accuracy` a reading score.

`accuracy.adj` the adjusted reading score: the observed 1's (perfect reading scores) are substituted with 0.99.

`dyslexia` a factor indicating wheter the child is dyslexic.

`iq` a standardized quantitative measure of the children's non verbal abilities.

Details

The data were originally analyzed by Pammer and Kevan (2004) and successively used by Smithson and Verkuilen (2006) and by Migliorati et al. (2018).

Source

betareg.

References

Cribari-Neto, F., Zeileis, A. (2010). Beta Regression in R. Journal of Statistical Software, 34(2), 1–24.

Di Brisco, A. M., Migliorati, S. (2020). A new mixed-effects mixture model for constrained longitudinal data. Statistics in Medicine, **39**(2), 129–145. doi:10.1002/sim.8406

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018). A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Smithson, M., Verkuilen, J. (2006). A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables. Psychological Methods, **11**(7), 54–71.

`residuals.flexreg` *Residuals Method for flexreg Objects*

Description

Method that computes various types of residuals from objects of class `` `flexreg` ``. If the model type is `FB` or `FBB` and `cluster = TRUE`, the method returns also residuals with respect to cluster means.

Usage

```
## S3 method for class 'flexreg'
residuals(
  object,
  type = "raw",
  cluster = FALSE,
  estimate = "mean",
  q = NULL,
  ...
)
```

Arguments

`object`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom` functions.
`type`	a character indicating type of residuals (`"raw"` or `"standardized"`).
`cluster`	logical. If the model is `"FB"` without augmentation or `"FBB"`, `cluster = TRUE` returns the cluster means. By default `cluster = FALSE`.
`estimate`	a character indicating the type of estimate: `"mean"` (default), `"median"`, or `"quantile"`.
`q`	if `estimate = "quantile"`, a numeric value of probability in (0, 1).
`...`	additional arguments. Currently not used.

Details

The `residuals` method computes raw and standardized residuals from objects of class `` `flexreg` ``. Raw residuals are defined as $r = y - \hat{\mu}$ for bounded continuous responses or as $r = y/n - \hat{\mu}$ for bounded discrete responses. Values y and y/n are the observed responses which are specified on the left-hand side of `formula` in the `flexreg` and `flexreg_binom` functions, respectively. Moreover, $\hat{\mu}$ is the predicted value, the result of the `predict` function with `type = "response"`. Standardized residuals are defined as $\frac{r}{\sqrt{\widehat{Var}(y)}}$ where $\widehat{Var}(y)$ is the variance of the response evaluated at the posterior means –by default, otherwise evaluated at the posterior quantiles of order `q`– of the parameters. If the model is `"FB"` or `"FBB"`, `type = "raw"`, and `cluster = TRUE`, the cluster raw residuals are computed as the difference between the observed response/relative response and the cluster means, i.e., $\hat{\lambda}_1$ and $\hat{\lambda}_2$. If the model is `"FB"` or `"FBB"`, `type = "standardized"` and `cluster = TRUE`, the cluster standardized residuals are computed as the cluster raw residuals divided by the square root of the cluster variances. Cluster residuals, either raw or standardized, can be used for classification purpose. Indeed, with `cluster = TRUE` the `residuals` method returns also a column named `"label"` assigning values 1 or 2 to observations depending on whether they are classified in cluster 1 (if the corresponding cluster residual is smaller) or in cluster 2.

Value

The method returns an array with as many rows as the number of observations in the sample. If `cluster = FALSE`, the array has only one column containing either the raw or standardized residuals. If `cluster = TRUE`, the array has four columns: the first column contains the raw or standardized residuals, the second and third columns contain the cluster residuals, and the fourth column contains the classification labels (see Details).

References

Ascari, R., Migliorati, S. (2021). A new regression model for overdispersed binomial data accounting for outliers and an excess of zeros. Statistics in Medicine, **40**(17), 3895–3914. doi:10.1002/sim.9005

Di Brisco, A. M., Migliorati, S. (2020). A new mixed-effects mixture model for constrained longitudinal data. Statistics in Medicine, **39**(2), 129–145. doi:10.1002/sim.8406

Migliorati, S., Di Brisco, A. M., Ongaro, A. (2018). A New Regression Model for Bounded Responses. Bayesian Analysis, **13**(3), 845–872. doi:10.1214/17-BA1079

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data=Reading, type="FB")
residuals(FB, type="raw", cluster=TRUE)

## End(Not run)
```

Stress *Stress and anxiety data*

Description

Data for assessing the dependency between stress and anxiety in nonclinical women in Townsville, Queensland, Australia.

Format

A `data.frame` containing 166 observations on the following 2 variables:

`stress` defined as rate.

`anxiety` defined as rate.

Details

Both variables are rates obtained as linear transformations from the Depression Anxiety Stress Scales which range from 0 to 42 (Lovibond & Lovibond, 1995). Additional details can be found in Example 2 from Smithson and Verkuilen (2006).

Source

Example 2 from Smithson and Verkuilen (2006).

References

Lovibond, P. F., Lovibond, S. H. (1995). The structure of negative emotional states: Comparison of the Depression Anxiety Stress Scales (DASS) with the Beck Depression and Anxiety Inventories. Behaviour research and therapy, 33(3), 335–343.

Smithson, M., Verkuilen, J. (2006). A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables. Psychological Methods, 11(7), 54–71.

summary.flexreg *Methods for ‘flexreg‘ Objects*

Description

Methods for extracting information from fitted regression model objects of class `` `flexreg` ``.

Usage

```
## S3 method for class 'flexreg'
summary(object, ..., digits = 4)

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

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

Arguments

object	an object of class `flexreg`, usually the result of flexreg or flexreg_binom functions.
...	additional arguments. Currently not used.
digits	an integer indicating the number of decimal places. Default equal to 4.
x	an object of class `summary.flexreg`.

Details

The summary.flexreg method summarizes the results of flexreg and flexreg_binom functions, adding also information from the functions residuals.flexreg and WAIC. The summary.flexreg method returns an object of class `summary.flexreg` containing the relevant summary statistics which can subsequently be printed using the associated print.summary.flexreg method.

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter = 1000)
summary(FB)

## End(Not run)
```

---

summary.flexreg_postpred

Summary Method for ‘flexreg_postpred‘ *objects*

Description

Summary method for an object of class `flexreg_postpred`, containing the simulated posterior predictive distribution.

Usage

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

Arguments

object	an object of class `flexreg_postpred` containing the simulated posterior predictives, usually the result of posterior_predict.
...	additional arguments.

Value

The function summary.flexreg_postpred returns an array with the statistical units by row. The number of rows of the array is equal to the number of columns of the object of class `flexreg_postpred` that is given to the function. By column there are some synthesis values that are the minimum, the first quartile, the media, the mean, the third quartile, and the maximum.

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter=1000)
pp <- posterior_predict(FB)
summary(pp)

## End(Not run)
```

WAIC *WAIC and LOO*

Description

The function computes widely applicable information criterion (WAIC) and efficient approximate leave-one-out cross-validation (LOO) from fitted regression model objects of class `flexreg`.

Usage

```
WAIC(model, ...)

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

Arguments

model	an object (or a list of objects) of class `flexreg`, usually the result of flexreg or flexreg_binom functions.
...	additional arguments.
x	an object of class `WAIC.flexreg`, usually the result of WAIC.

Details

This function takes advantage of the **loo** package to compute the widely applicable information criterion (WAIC) and leave-one-out cross-validation (LOO) for objects of class `flexreg`. If a list of two or more objects of class `flexreg` is provided, the function returns the difference in their expected predictive accuracy (see loo_compare for further details).

Value

A named list with components from loo and waic.

References

Vehtari, A., Gelman, A., Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and Computing. **27**(5), 1413–1432. doi:10.1007/s11222-016-9696-4

Examples

```
## Not run:
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, type="FB", n.iter=1000)
WAIC(FB)

## End(Not run)
```

Index