Package ‘FlexReg’

October 12, 2022

Title Regression Models for Bounded and Binomial Responses

Version 1.2

Author Agnese M. Di Brisco [aut, cre], Roberto Ascari [aut, cre], Sonia Migliorati [aut], Andrea Ongaro [aut]

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

Description Functions to fit regression models for bounded (e.g., proportions and rates) and binomial responses. In case of bounded responses, 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>). In case of binomial responses, available models are the flexible beta-binomial (Ascari, R., Migliorati, S. (2021) <doi:10.1002/sim.9005>), the beta-binomial, and the binomial. 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

LazyData true

Biarch true

Depends R (>= 3.5.0)

Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.0.0), loo, bayesplot, ggplot2, Formula, utils, grDevices, faraway

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

SystemRequirements GNU make

RoxygenNote 7.1.2

Suggests testthat

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-09-16 09:46:07 UTC

R topics documented:

`FlexReg-package` *The 'FlexReg' package.*

Description

This package provides functions and methods to implement several types of regression models for continuous bounded responses (e.g., proportions and rates) and binomial data. Inferential statistical analysis is dealt with by Bayesian estimation based on the Hamiltonian Monte Carlo (HMC) algorithm through the **'rstan'** package.

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., 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

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

---

`Atomic` *Atomic bombs data*

Description

Counts of chromosome aberrations in atomic bombs survivors.

Format

A data frame containing 1039 observations on the following 4 variables.

`y` the number of cells with chromosomal abnormalities.

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

`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

Counts of eggs parasitized by female parasitoids.

Format

A data frame containing 70 observations on the following 4 variables.

`y` the total 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 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 household.

`Age` the age of the head of 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` has been created by dividing the variable '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 diagnostics 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()
)
```

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`, `gaftery`, `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)

Details

- `R-hat` 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% and last 50% of the chain. The fraction to use from the first and last part of the chain can be edited through the additional arguments `frac1` and `frac2`. The sum of `frac1` and `frac2` has to be strictly less than 1. See `coda::geweke.diag` for further details.
- `raftery` returns the length of "burn-in" (M), the required sample size (N), the minimum sample size for a chain with zero autocorrelation ($Nmin$), and the estimate of the "dependence factor" ($I = (M + N)/Nmin$). Values of I greater than 5 may be indicative of 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 list in the `additional.args` argument. See `coda::raftery.diag` for further details.
- `heidel` returns a p-value 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 diagnostics. 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 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

---

`convergence.plot` *Convergence plots*

Description

The function produces a .pdf file containing some convergence plots for 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 (with extension .pdf).
`plotfun`	an optional character vector of diagnostics plots. The default is to compute `all` plots, otherwise one can specify a selection 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 (`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

The plots can be further customized using the `ggplot2` package.

- `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.

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

---

`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 plot the probability of augmentation in zero/one.

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 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 if `type` is `"BetaBin"` or `"FBB"`). It must lie in (0, 1).
`phi`	the precision parameter (an alternative way to specify the `theta` parameter if `type` is `"BetaBin"` or `"FBB"`). It must be a positive real value.

`p`	the mixing weight (to be specified if `type` is `"FB"` or `"VIB"`). It must lie in (0, 1).
`w`	the normalized distance among clusters of the FB distribution (to be specified if `type = "FB"`). It must lie in (0, 1).
`k`	the extent of the variance inflation (to be specified if `type = "VIB"`). It must lie in (0, 1).
`q0`	the probability of augmentation in zero (it can be specified only if `type` is `"Beta"`, `"FB"`, or `"VIB"`). It must lie in (0, 1). In case of no augmentation is `NULL` (default).
`q1`	the probability of augmentation in one (it can be specified only if `type` is `"Beta"`, `"FB"`, or `"VIB"`). It must lie in (0, 1). In case of no augmentation 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

```
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)
```

`dBetaBin` *Beta-binomial probability mass function*

Description

The function computes the probability mass function of the beta-binomial distribution.

Usage

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

Arguments

`x`	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. It is an alternative way to specify the `theta` parameter. It must be a positive real value.

Details

The beta-binomial distribution has probability mass function

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

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

A vector with the same length as `x`.

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)
```

`dBeta_mu` *Beta probability density function*

Description

The function computes the probability density function of the beta distribution with a mean-precision parameterization. It can also compute the probability density function of the augmented beta distribution by assigning positive probabilities to zero and one and a (continuous) beta density to the interval (0,1).

Usage

```
dBeta_mu(x, mu, phi, q0 = NULL, q1 = NULL)
```

Arguments

`x`	a vector of quantiles.
`mu`	the mean parameter of the beta distribution. It must lie in (0, 1).
`phi`	the precision parameter of the Beta distribution. It must be a positive real value.
`q0`	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation is `NULL` (default).
`q1`	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation is `NULL` (default).

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

A vector with the same length as `x`.

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

`dFB` *Flexible beta probability density function*

Description

The function computes the probability density function of the flexible beta distribution. It can also compute the probability density function of the augmented flexible beta distribution by assigning positive probabilities to zero and one and a (continuous) flexible beta density to the interval (0,1).

Usage

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

Arguments

`x`	a vector of quantiles.
`mu`	the mean parameter of the flexible beta distribution. It must lie in (0, 1).
`phi`	the precision parameter of the flexible beta distribution. It must be a positive real value.
`p`	the mixing weight. It must lie in (0, 1).
`w`	the normalized distance among clusters. It must lie in (0, 1).
`q0`	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation is `NULL` (default).
`q1`	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation is `NULL` (default).

Details

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

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

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, and $0 < w < 1$ is the normalized distance between component means. The latter are $\lambda_1 = \mu + (1-p)\tilde{w}$ and $\lambda_2 = \mu - p\tilde{w}$, where $\tilde{w} = w \min \mu/p, (1-\mu)/(1-p)$.

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

A vector with the same length as `x`.

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)
```

`dFBB` *Flexible beta-binomial probability mass function*

Description

The function computes the probability mass function of the flexible beta-binomial distribution.

Usage

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

Arguments

`x`	a vector of quantiles.
`size`	the total number of trials.
`mu`	the mean parameter. It must lie in (0, 1).
`theta`	the overdispersion paramete. It must lie in (0, 1).
`phi`	the precision parameter. It is an alternative way to specify the `theta` parameter. It must be a positive real value.
`p`	the mixing weight. It must lie in (0, 1).
`w`	the normalized distance among clusters. It must lie in (0, 1).

Details

The FBB distribution is a special mixture of two beta-binomial distributions

$$pBB(x;\lambda_1,\phi) + (1-p)BB(x;\lambda_2,\phi)$$

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

Value

A vector with the same length as `x`.

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)
```

`dVIB` *Variance-inflated beta probability density function*

Description

The function computes the probability density function of the variance-inflated beta distribution. It can also compute the probability density function of the augmented variance-inflated beta distribution by assigning positive probabilities to zero and one and a (continuous) variance-inflated beta density to the interval (0,1).

Usage

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

Arguments

`x`	a vector of quantiles.
`mu`	the mean parameter of the variance-inflated beta distribution. It must lie in (0, 1).
`phi`	the precision parameter of the variance-Inflated distribution. It must be a positive real 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 is `NULL` (default).
`q1`	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation is `NULL` (default).

Details

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

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

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

A vector with the same length as `x`.

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)
```

`Election` *Italian Election Results*

Description

Results of the Italian general election held on 4 March 2018 for six parties.

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.

`FDI` the percentage of votes got by 'Fratelli d'Italia' party.

`LEGA` the percentage of votes got by 'Lega' party.

`LEU` the percentage of votes got by 'Liberi e Uguali' party.

`M5S` the percentage of votes got by 'Movimento 5 Stelle' party.

`PD` the percentage of votes got by 'Partito 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), multiplied by 100.

`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.

Source

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

`flexreg` *Flexible Regression Models for Proportions*

Description

The function fits some flexible regression models for continuous bounded 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"`), and 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,
  link.phi = NULL,
  prior.phi = NULL,
  hyperparam.phi = NULL,
  prior.psi = NULL,
  hyperparam.psi = NULL,
  n.iter = 5000,
  burnin.perc = 0.5,
  n.chain = 1,
  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 model (`y ~ x | z`) to be fitted.
`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 `base::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 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 standard deviation parameter for the prior distribution of `beta` regression coefficients. A value of 100 is suggested if the prior is `"normal"`, 2.5 if `"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 standard deviation parameter for the prior distribution of `omega0` regression coefficients. A value of 100 is suggested if the prior is `"normal"`, 2.5 if `"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 standard deviation parameter for the prior distribution of `omega1` regression coefficients. A value of 100 is suggested if the prior is `"normal"`, 2.5 if `"cauchy"`.
`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 (a value of 0.001 is suggested). 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 standaerd deviation parameter for the prior distribution of `psi` regression coefficients. A value of 100 is suggested if the prior is `"normal"`, 2.5 if `"cauchy"`.

`n.iter`	a positive integer specifying the number of iterations for each chain (including warmup). The default is 5000.
`burnin.perc`	the percentage of iterations per chain to discard.
`n.chain`	a positive integer specifying the number of Markov chains. The default is 1.
`thin`	a positive integer specifying the period for saving samples. The default is 1.
`verbose`	`TRUE` (default) or `FALSE`: flag indicating whether to print intermediate output.
`...`	additional arguments for `rstan::sampling`.

Details

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

$$g(\mu_i) = \boldsymbol{x}_i^t \boldsymbol{\beta},$$

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, 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 $q(\cdot)$ specified in the `link.phi` argument:

$$q(\phi_i) = \boldsymbol{z}_i^t \boldsymbol{\psi},$$

where $\boldsymbol{\psi}$ is the vector of regression coefficients for the precision model. The prior distribution and the related hyperparameter of $\boldsymbol{\psi}$ can be specified in `prior.psi` and `hyperparam.psi`. In `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 and 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 responses, one has to link the parameter q_0 to an additional (possibly overlapping) set of covariates through a logit link function:

$$g_0(q_{0i}) = \boldsymbol{x}_{0i}^t \boldsymbol{\omega_0},$$

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 responses, one has to link the parameter q_1 to an additional (possibly overlapping) set of covariates through a logit link function:

$$g_1(q_{1i}) = \boldsymbol{x}_{1i}^t \boldsymbol{\omega_1},$$

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`, the augmented models in zero and/or one can be specified in the `zero.formula` and/or `one.formula` argument 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.
`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`	an object of class `` `stanfit` `` containing 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 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 Binomial data*

Description

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

Usage

```
flexreg_binom(
  formula,
  data,
  type = "FBB",
  n = NULL,
  link.mu = "logit",
  prior.beta = "normal",
  hyperparam.beta = 100,
  hyper.theta.a = NULL,
  hyper.theta.b = NULL,
  link.theta = NULL,
  prior.psi = NULL,
  hyperparam.psi = NULL,
  n.iter = 5000,
  burnin.perc = 0.5,
  n.chain = 1,
  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`).
`data`	an optional data frame, list, or object that is coercible to a data frame through `base::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 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`	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 standard deviation parameter for the prior distribution of `beta` regression coefficients. A value of 100 is suggested if the prior is `"normal"`, 2.5 if `"cauchy"`.
`hyper.theta.a`	a numeric (vector of length 1) specifying the first shape parameter for the beta prior distribution of `theta`.
`hyper.theta.b`	a numeric (vector of length 1) specifying the second shape parameter for the beta prior distribution of `theta`.
`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 standard deviation parameter for the prior distribution of `psi` regression coefficients. A value of 100 is suggested if the prior is `"normal"`, 2.5 if `"cauchy"`.
`n.iter`	a positive integer specifying the number of iterations for each chain (including warmup). The default is 5000.
`burnin.perc`	the percentage of iterations per chain to discard.
`n.chain`	a positive integer specifying the number of Markov chains. The default is 1.
`thin`	a positive integer specifying the period for saving samples. The default is 1.
`verbose`	`TRUE` (default) or `FALSE`: flag indicating whether to print intermediate output.
`...`	additional arguments for `rstan::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 $g(\cdot)$ specified in `link.mu`:

$$g(\mu_i) = x_i^t \boldsymbol{\beta},$$

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 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 $q(\cdot)$ specified in the `link.theta` argument:

$$q(\theta_i) = z_i^t \boldsymbol{\psi},$$

where ψ is the vector of regression coefficients for the overdispersion model. The prior distribution and the related hyperparameter of ψ 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 and 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`	an object of class `` `stanfit` `` containing 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)
```

`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, ...)
```

Arguments

`x`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom`.
`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 to be set as default.
`...`	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 continuous bounded or binomial, respectively. Any other variable specified in the mean model must be set to a default through the `additional.cov.default` argument. If the regression model is of `FB` without augmentation or `FBB` type the function returns a scatterplot with three curves, one corresponding to the overall mean and two corresponding to the component means of the FB distribution, 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` *Posterior Predictives Plot*

Description

Method for plotting the simulated posterior predictive distribution from an object of class `` `flexreg_postpred` ``. 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 or binomial 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 or binomial 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 ~ iq, data = Reading)
pp <- posterior_predict(FB)
plot(pp)

## End(Not run)
```

---

posterior_predict.flexreg
Posterior Predictive

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)
```

Arguments

`model`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom`.
`newdata`	an optional data frame containing variables with which to predict. If omitted, the fitted values are used.

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 responses and for `y/n` in case of binomial 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` *Prediction Method for flexreg Objects*

Description

Method that computes various types of prediction from objects of class `` `flexreg` ``. If the model type is `"FB"` without augmentation or `"FBB"` and `cluster = T`, the function returns also cluster means.

Usage

```
## S3 method for class 'flexreg'
predict(
  object,
  newdata = NULL,
  cluster = F,
  type = "response",
```

```
  estimate = "mean",
  q = NULL,
  ...
)
```

Arguments

`object`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom`.
`newdata`	an optional data frame containing variables with which to predict. If omitted, the fitted values are used.
`cluster`	logical. If the model is `"FB"` without augmentation or `"FBB"`, `cluster = T` returns the cluster means. By default, `cluster = F`.
`type`	a character indicating the type of predictions. Available options are: `"response"`, that returns the marginal fitted means of response/relative response; `"link"`, the linear predictor of the mean model; `"precision"`, the fitted precision parameter phi; `"overdispersion"`, the fitted overdispersion parameter $theta$; `"variance"`, the fitted variance of the response.
`estimate`	the type of estimate: `"mean"` (default), `"median"` or `"quantile"`.
`q`	if `estimate = "quantile"`, numeric value of probability in (0, 1).
`...`	additional arguments. Currently not used.

Details

If `type="response"` the function returns the marginal mean that is μ in case of no augmentation and $q_1 + (1 - q_0 - q_1)\mu$ in case of augmentation. 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. See Di Brisco and Migliorati (2020) for details. The option `type = "overdispersion"` is available only for beta-binomial and flexible beta-binomial models.

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)
```

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

Description

Bayesian version of R-squared for flexible regression models for continuous bounded or binomial responses

Usage

```
R2_bayes(model)
```

Arguments

`model`	an object of class `` `flexreg` ``, usually the result of `flexreg` or `flexreg_binom`.

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.

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

```
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, type = "FB", n.iter=1000)
hist(R2_bayes(FB))
```

`rBetaBin` *Random generator from the beta-binomial distribution*

Description

The function generates random values from the beta-binomial distribution.

Usage

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

Arguments

`n`	the number of values to generate. If `length(n)` > 1, the length is taken to be the number required.
`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. It is an alternative way to specify the `theta` parameter. It must be a positive real value.

Value

A vector of length `n`.

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

```
rBetaBin(n = 100, size = 40, mu = .5, theta = .4)
rBetaBin(n = 100, size = 40, mu = .5, phi = 1.5)
```

`rBeta_mu` *Random generator from the beta distribution*

Description

The function generates random values from the beta distribution with a mean-precision parameterization, or from the augmented beta distribution.

Usage

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

Arguments

`n`	the number of values to generate. If `length(n) > 1`, the length is taken to be the number required.
`mu`	the mean parameter of the beta distribution. It must lie in (0, 1).
`phi`	the precision parameter of the Beta distribution. It must be a positive real value.
`q0`	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation is `NULL` (default).
`q1`	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation is `NULL` (default).

Value

A vector of length `n`.

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

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

`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 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` without augmentation or `FBB` and `cluster = T`, 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`.
`type`	a character indicating type of residuals (`"raw"` or `"standardized"`).
`cluster`	logical. If the model is `"FB"` without augmentation or `"FBB"`, `cluster = T` returns the cluster means. By default `cluster = F`.
`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

Raw residuals are defined as $r_i = y_i - \hat{\mu}_i$ (or $r_i = y_i/n_i - \hat{\mu}_i$ for binomial data). The values y_i or y_i/n_i are the observed responses which are specified on the left-hand side of `formula` in the `flexreg` or `flexreg_binom` function, respectively. $\hat{\mu}_i$ is the predicted value, the result of the `predict` function with `type = "response"`. Standardized residuals are defined as $\frac{r_i}{\widehat{Var(y_i)}}$ where $\widehat{Var}(y_i)$ is the variance of the dependent variable evaluated at the posterior means (default, otherwise quantile of order q) of the parameters. If the model is `"FB"` without augmentation or `"FBB"` and `cluster = T`, the cluster residuals are computed as the difference between the observed response/relative response and the cluster means $\hat{\lambda}_{1i}$ and $\hat{\lambda}_{2i}$.

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)
```

`rFB` *Random generator from the flexible beta distribution*

Description

The function generates random values from the flexible beta distribution, or from the augmented flexible beta distribution.

Usage

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

Arguments

`n`	the number of values to generate. If `length(n)` > 1, the length is taken to be the number required.
`mu`	the mean parameter of the flexible beta distribution. It must lie in (0, 1).
`phi`	the precision parameter of the flexible beta distribution. It must be a positive real value.
`p`	the mixing weight. It must lie in (0, 1).
`w`	the normalized distance among clusters. It must lie in (0, 1).
`q0`	the probability of augmentation in zero. It must lie in (0, 1). In case of no augmentation is `NULL` (default).
`q1`	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation is `NULL` (default).

Value

A vector of length `n`.

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

```
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)
```

`rFBB` *Random generator from the flexible beta-binomial distribution*

Description

The function generates random values from the flexible beta-binomial distribution.

Usage

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

Arguments

`n`	the number of values to generate. If `length(n) > 1`, the length is taken to be the number required.
`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. It is an alternative way to specify the `theta` parameter. It must be a positive real value.
`p`	the mixing weight. It must lie in (0, 1).
`w`	the normalized distance among clusters. It must lie in (0, 1).

Value

A vector of length `n`.

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

```
rFBB(n = 100, size = 40, mu = .5, theta = .4, p = .3, w = .6)
rFBB(n = 100, size = 40, mu = .5, phi = 1.5, p = .3, w = .6)
```

`rVIB` *Random generation from the variance-inflated beta distribution*

Description

The function generates random values from the variance-inflated beta distribution, or from the augmented variance-inflated beta distribution.

Usage

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

Arguments

`n`	the number of values to generate. If `length(n) > 1`, the length is taken to be the number required.
`mu`	the mean parameter of the variance-inflated distribution. It must lie in (0, 1).
`phi`	the precision parameter of the variance-inflated distribution. It must be a positive real 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 is `NULL` (default).
`q1`	the probability of augmentation in one. It must lie in (0, 1). In case of no augmentation is `NULL` (default).

Value

A vector of length `n`.

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

```
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)
```

`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`.
`...`	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` method.

Examples

```
data("Reading")
FB <- flexreg(accuracy.adj ~ iq, data = Reading, n.iter = 1000)
summary(FB)
```

`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`.
`...`	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::loo_compare` for further details).

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