

Package ‘bang’

July 22, 2025

Title Bayesian Analysis, No Gibbs

Version 1.0.4

Date 2024-07-17

Description Provides functions for the Bayesian analysis of some simple commonly-used models, without using Markov Chain Monte Carlo (MCMC) methods such as Gibbs sampling. The ‘rust’ package <https://cran.r-project.org/package=rust> is used to simulate a random sample from the required posterior distribution, using the generalized ratio-of-uniforms method. See Wakefield, Gelfand and Smith (1991) [DOI:10.1007/BF01889987](https://doi.org/10.1007/BF01889987) for details. At the moment three conjugate hierarchical models are available: beta-binomial, gamma-Poisson and a 1-way analysis of variance (ANOVA).

Depends R (>= 3.3.0)

Imports bayesplot (>= 1.1.0), graphics, rust (>= 1.2.2), stats

RoxygenNote 7.2.3

Suggests ggplot2 (>= 2.2.1), knitr, MASS, rmarkdown, testthat

VignetteBuilder knitr

URL <https://paulnorthrop.github.io/bang/>,
<https://github.com/paulnorthrop/bang>

BugReports <https://github.com/paulnorthrop/bang/issues>

License GPL (>= 2)

LazyData true

Encoding UTF-8

Config/testthat/edition 3

NeedsCompilation no

Author Paul J. Northrop [aut, cre, cph],
Benjamin D. Hall [aut, cph]

Maintainer Paul J. Northrop <p.northrop@uc1.ac.uk>

Repository CRAN

Date/Publication 2024-07-17 21:00:02 UTC

Contents

bang-package	2
coagulation	3
hanova1	4
hef	7
plot.hef	11
pp_check.hef	13
print.hef	15
print.summary.hef	16
pump	16
rat	17
set_user_prior	18
sim_pred_beta_binom	19
sim_pred_gamma_pois	20
sim_pred_hanova1	20
summary.hef	21
temp1	22
temp2	23
weight_gain	24
Index	26

bang-package

bang: Bayesian Analysis, No Gibbs

Description

Performs Bayesian analyses using some simple commonly-used models. The multivariate generalized ratio-of-uniforms method is used to simulate random samples from the required posterior distribution. The user can either choose hyperparameter values of a default prior distribution or specify their own prior distribution.

Details

Currently three conjugate hierarchical models are available: beta-binomial, gamma-Poisson and 1-way Analysis of Variance (ANOVA). The function `hef` produces random posterior samples from for the beta-binomial and gamma-Poisson models. The function `hanova1` does this for the 1-way Analysis of Variance (ANOVA). The `rust` package is used to produce these samples.

See `vignette("bang-a-vignette", package = "bang")` for a brief introduction to the package and `vignette("bang-b-hef-vignette", package = "bang")` and `vignette("bang-c-anova-vignette", package = "bang")` for illustrations of the use of the `hef` and `hanova1` functions.

Author(s)

Maintainer: Paul J. Northrop <p.northrop@ucl.ac.uk> [copyright holder]

Authors:

- Benjamin D. Hall <benjamin.hall.16@ucl.ac.uk> [copyright holder]

References

Northrop, P. J. (2017). rust: Ratio-of-Uniforms Simulation with Transformation. R package version 1.2.3. <https://cran.r-project.org/package=rust>.

See Also

[hef](#) for hierarchical exponential family models.

[hanova1](#) for hierarchical one-way analysis of variance (ANOVA).

[set_user_prior](#) to set a user-defined prior.

coagulation

Coagulation time data

Description

Coagulation time in seconds for blood drawn from 24 animals randomly allocated to four different diets from Box, Hunter, and Hunter (1978). The data frame `coagulation` has 24 rows and 2 columns. Each row relates to a different animal. Column 1 contains the coagulation times. Column 2 contains a label for the type of diet: one of A, B, C or D.

Usage

```
coagulation
```

Format

A data frame with 24 rows and 2 columns.

Source

Table 11.2 of Gelman, A., Carlin, J. B., Stern, H. S. Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014) *Bayesian Data Analysis*. Chapman & Hall / CRC. <http://www.stat.columbia.edu/~gelman/book/>

References

Box, G. E. P., Hunter, W. G., and Hunter, J. S. (1978). *Statistics for Experimenters*. New York: Wiley.

hanova1

*Posterior sampling for a 1-way hierarchical ANOVA***Description**

Produces random samples from the posterior distribution of the parameters of a 1-way hierarchical ANOVA model.

Usage

```
hanova1(
  n = 1000,
  resp,
  fac,
  ...,
  prior = "default",
  hpars = NULL,
  param = c("trans", "original"),
  init = NULL,
  mu0 = 0,
  sigma0 = Inf,
  nrep = NULL
)
```

Arguments

n	A numeric scalar. The size of posterior sample required.
resp	A numeric vector. Response values.
fac	A vector of class <code>factor</code> indicating the group from which the corresponding element of <code>resp</code> originates. Must have the same length as <code>resp</code> .
...	Optional further arguments to be passed to <code>ru</code> .
prior	The log-prior for the parameters of the hyperprior distribution. If the user wishes to specify their own prior then <code>prior</code> must be an object returned from a call to <code>set_user_prior</code> . Otherwise, <code>prior</code> is a character scalar giving the name of the required in-built prior. If <code>prior</code> is not supplied then a default prior is used. See Details .
hpars	A numeric vector. Used to set parameters (if any) in an in-built prior. If <code>prior = cauchy</code> then <code>hpars</code> is a numeric vector of length 2 giving the respective scale parameters of the half-Cauchy priors for σ_α and σ .
param	A character scalar. If <code>param = "trans"</code> (the default) then the marginal posterior of hyperparameter vector ϕ is reparameterized in terms of $\log\sigma_\alpha, \log\sigma$. If <code>param = "original"</code> the original parameterization, i.e. σ_α, σ is used. The former tends to make the optimizations involved in the ratio-of-uniforms algorithm more stable and to increase the probability of acceptance, but at the expense of slower function evaluations.

<code>init</code>	A numeric vector. Optional initial estimates sent to <code>ru</code> in the search for the mode of the posterior density of (perhaps a subset of) the hyperparameter vector ϕ . If an in-built prior is used then <code>ru</code> is used to sample from the marginal posterior density of (σ_α, σ) , so <code>init</code> must have length 2. Otherwise, <code>init</code> has length equal to the argument <code>anova_d</code> supplied to <code>set_user_prior</code> .
<code>mu0, sigma0</code>	A numeric scalar. Mean and standard deviation of a normal prior for μ . Only used if an in-built prior is used or if <code>anova_d = 2</code> is supplied in a call to <code>set_user_prior</code> to set a user-defined prior. The default, <code>sigma0 = Inf</code> , sets an improper uniform prior for μ .
<code>nrep</code>	A numeric scalar. If <code>nrep</code> is not NULL then <code>nrep</code> gives the number of replications of the original dataset simulated from the posterior predictive distribution. Each replication is based on one of the samples from the posterior distribution. Therefore, <code>nrep</code> must not be greater than <code>n</code> . In that event <code>nrep</code> is set equal to <code>n</code> .

Details

Consider I independent experiments in which the n_i responses y_i from experiment/group i are normally distributed with mean θ_i and standard deviation σ . The population parameters $\theta_1, \dots, \theta_I$ are modelled as random samples from a normal distribution with mean μ and standard deviation σ_α . Let $\phi = (\mu, \sigma_\alpha, \sigma)$. Conditionally on $\theta_1, \dots, \theta_I, y_1, \dots, y_I$ are independent of each other and are independent of ϕ . A *hyperprior* is placed on ϕ . The user can either choose parameter values of a default hyperprior or specify their own hyperprior using `set_user_prior`.

The `ru` function in the `rust` package is used to draw a random sample from the marginal posterior of the hyperparameter vector ϕ . Then, conditional on these values, population parameters are sampled directly from the conditional posterior density of $\theta_1, \dots, \theta_I$ given ϕ and the data. See the vignette("bang-c-anova-vignette", package = "bang") for details.

The following priors are specified up to proportionality.

Priors:

`prior = "bda"` (the default): $\pi(\mu, \sigma_\alpha, \sigma) = 1/\sigma$, that is, a uniform prior for $(\mu, \sigma_\alpha, \log\sigma)$, for $\sigma_\alpha > 0$ and $\sigma > 0$. The data must contain at least 3 groups, that is, `fac` must have at least 3 levels, for a proper posterior density to be obtained. [See Sections 5.7 and 11.6 of Gelman et al. (2014).]

`prior = "unif"`: $\pi(\mu, \sigma_\alpha, \sigma) = 1$, that is, a uniform prior for $(\mu, \sigma_\alpha, \sigma)$, for $\sigma_\alpha > 0$ and $\sigma > 0$. [See Section 11.6 of Gelman et al. (2014).]

`prior = "cauchy"`: independent half-Cauchy priors for σ_α and σ with respective scale parameters A_α and A , that is, $\pi(\sigma_\alpha, \sigma) = 1/[(1 + \sigma_\alpha^2/A_\alpha^2)(1 + \sigma^2/A^2)]$. [See Gelman (2006).] The scale parameters (A_α, A) are specified using `hpars = (A_\alpha, A)`. The default setting is `hpars = c(10, 10)`.

Parameterizations for sampling:

`param = "original"` is $(\mu, \sigma_\alpha, \sigma)$, `param = "trans"` (the default) is $\phi_1 = \mu, \phi_2 = \log\sigma_\alpha, \phi_3 = \log\sigma$.

Value

An object (list) of class "hef", which has the same structure as an object of class "ru" returned from `ru`. In particular, the columns of the `n`-row matrix `sim_vals` contain the simulated values of ϕ . In addition this list contains the arguments `model`, `resp`, `fac` and `prior` detailed above and an `n` by

I matrix `theta_sim_vals`: column *i* contains the simulated values of θ_i . Also included are `data = cbind(resp, fac)` and `summary_stats` a list containing: the number of groups *I*; the numbers of responses each group *n_i*; the total number of observations; the sample mean response in each group; the sum of squared deviations from the group means *s*; the arguments to `hanova1` `mu0` and `sigma0`; `call`: the matched call to `hanova1`.

References

Gelman, A., Carlin, J. B., Stern, H. S. Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014) *Bayesian Data Analysis*. Chapman & Hall / CRC.

Gelman, A. (2006) Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3), 515-533. doi:10.1214/06BA117A.

See Also

The `ru` function in the `rust` package for details of the arguments that can be passed to `ru` via `hanova1`.

[hef](#) for hierarchical exponential family models.

[set_user_prior](#) to set a user-defined prior.

Examples

```
# ===== Late 21st Century Global Temperature Data =====

# Extract data for RCP2.6
RCP26_2 <- temp2[temp2$RCP == "rcp26", ]

# Sample from the posterior under the default 'noninformative' flat prior
# for (mu, sigma_alpha, log(sigma)). Ratio-of-uniforms is used to sample
# from the marginal posterior for (log(sigma_alpha), log(sigma)).
temp_res <- hanova1(resp = RCP26_2[, 1], fac = RCP26_2[, 2])

# Plot of sampled values of (sigma_alpha, sigma)
plot(temp_res, params = "ru")

# Plot of sampled values of (log(sigma_alpha), log(sigma))
# (centred at (0,0))
plot(temp_res, ru_scale = TRUE)

# Plot of sampled values of (mu, sigma_alpha, sigma)
plot(temp_res)

# Estimated marginal posterior densities of the mean for each GCM
plot(temp_res, params = "pop", which_pop = "all", one_plot = TRUE)

# Posterior sample quantiles
probs <- c(2.5, 25, 50, 75, 97.5) / 100
round(t(apply(temp_res$sim_vals, 2, quantile, probs = probs)), 2)

# Ratio-of-uniforms information and posterior sample summaries
summary(temp_res)
```

```

# ===== Coagulation time data, from Table 11.2 Gelman et al (2014) =====

# With only 4 groups the posterior for sigma_alpha has a heavy right tail if
# the default `noninformative' flat prior for (mu, sigma_alpha, log(sigma))
# is used. If we try to sample from the marginal posterior for
# (sigma_alpha, sigma) using the default generalized ratio-of-uniforms
# runing parameter value r = 1/2 then the acceptance region is not bounded.

# Two remedies: reparameterize the posterior and/or increase the value of r.

# (log(sigma_alpha), log(sigma)) parameterization, ru parameter r = 1/2
coag1 <- hanova1(resp = coagulation[, 1], fac = coagulation[, 2])

# (sigma_alpha, sigma) parameterization, ru parameter r = 1
coag2 <- hanova1(resp = coagulation[, 1], fac = coagulation[, 2],
                 param = "original", r = 1)

# Values to compare to those in Table 11.3 of Gelman et al (2014)
all1 <- cbind(coag1$theta_sim_vals, coag1$sim_vals)
all2 <- cbind(coag2$theta_sim_vals, coag2$sim_vals)
round(t(apply(all1, 2, quantile, probs = probs)), 1)
round(t(apply(all2, 2, quantile, probs = probs)), 1)

# Pairwise plots of posterior samples from the group means
plot(coag1, which_pop = "all", plot_type = "pairs")

# Independent half-Cauchy priors for sigma_alpha and sigma
coag3 <- hanova1(resp = coagulation[, 1], fac = coagulation[, 2],
                 param = "original", prior = "cauchy", hpars = c(10, 1e6))

```

hef

Hierarchical Exponential Family Model

Description

Produces random samples from the posterior distribution of the parameters of certain hierarchical exponential family models.

Usage

```

hef(
  n = 1000,
  model = c("beta_binom", "gamma_pois"),
  data,
  ...,
  prior = "default",
  hpars = NULL,

```

```

  param = c("trans", "original"),
  init = NULL,
  nrep = NULL
)

```

Arguments

<code>n</code>	An integer scalar. The size of the posterior sample required.
<code>model</code>	A character string. Abbreviated name for the response-population distribution combination. For a hierarchical normal model see hanova1 (hierarchical one-way analysis of variance (ANOVA)).
<code>data</code>	A numeric matrix. The format depends on <code>model</code> . See Details .
<code>...</code>	Optional further arguments to be passed to ru .
<code>prior</code>	The log-prior for the parameters of the hyperprior distribution. If the user wishes to specify their own prior then <code>prior</code> must be an object returned from a call to set_user_prior . Otherwise, <code>prior</code> is a character scalar giving the name of the required in-built prior. If <code>prior</code> is not supplied then a default prior is used. See Details .
<code>hpars</code>	A numeric vector. Used to set parameters (if any) in an in-built prior.
<code>param</code>	A character scalar. If <code>param = "trans"</code> (the default) then the marginal posterior of hyperparameter vector ϕ is reparameterized in a way designed to improve the efficiency of sampling from this posterior. If <code>param = "original"</code> the original parameterization is used. The former tends to make the optimizations involved in the ratio-of-uniforms algorithm more stable and to increase the probability of acceptance, but at the expense of slower function evaluations.
<code>init</code>	A numeric vector of length 2. Optional initial estimates for the search for the mode of the posterior density of the hyperparameter vector ϕ .
<code>nrep</code>	A numeric scalar. If <code>nrep</code> is not NULL then <code>nrep</code> gives the number of replications of the original dataset simulated from the posterior predictive distribution. Each replication is based on one of the samples from the posterior distribution. Therefore, <code>nrep</code> must not be greater than <code>n</code> . In that event <code>nrep</code> is set equal to <code>n</code> .

Details

Conditional on population-specific parameter vectors $\theta_1, \dots, \theta_J$ the observed *response* data y_1, \dots, y_J within each population are modelled as random samples from a distribution in an exponential family. The population parameters $\theta_1, \dots, \theta_J$ are modelled as random samples from a common *population distribution*, chosen to be conditionally conjugate to the response distribution, with *hyperparameter* vector ϕ . Conditionally on $\theta_1, \dots, \theta_J, y_1, \dots, y_J$ are independent of each other and are independent of ϕ . A *hyperprior* is placed on ϕ . The user can either choose parameter values of a default hyperprior or specify their own hyperprior using [set_user_prior](#).

The [ru](#) function in the [rust](#) package is used to draw a random sample from the marginal posterior of the hyperparameter vector ϕ . Then, conditional on these values, population parameters are sampled directly from the conditional posterior density of $\theta_1, \dots, \theta_J$ given ϕ and the data.

We outline each `model`, specify the format of the data, give the default (log-)priors (up to an additive constant) and detail the choices of ratio-of-uniforms parameterization `param`.

Beta-binomial: For $j = 1, \dots, J$, $Y_j|p_j$ are i.i.d binomial(n_j, p_j), where p_j is the probability of success in group j and n_j is the number of trials in group j . p_j are i.i.d. beta(α, β), so and $\phi = (\alpha, \beta)$. data is a 2-column matrix: the numbers of successes in column 1 and the corresponding numbers of trials in column 2.

Priors:

prior = "bda" (the default): $\log\pi(\alpha, \beta) = -2.5\log(\alpha + \beta), \alpha > 0, \beta > 0$. [See Section 5.3 of Gelman et al. (2014).]

prior = "gamma": independent gamma priors on α and β , i.e. $\log\pi(\alpha, \beta) = (s_1 - 1)\log\alpha - r_1\alpha + (s_2 - 1)\log\beta - r_2\beta, \alpha > 0, \beta > 0$. where the respective shape (s_1, s_2) and rate (r_1, r_2) parameters are specified using `hpars = (s1, r1, s2, r2)`. The default setting is `hpars = c(1, 0.01, 1, 0.01)`.

Parameterizations for sampling:

param = "original" is (α, β) , param = "trans" (the default) is $\phi_1 = \text{logit}(\alpha/(\alpha+\beta)) = \log(\alpha/\beta), \phi_2 = \log(\alpha + \beta)$. See Section 5.3 of Gelman et al. (2014).

Gamma-Poisson: For $j = 1, \dots, J$, $Y_j|\lambda_j$ are i.i.d Poisson($e_j\lambda_j$), where e_j is the *exposure* in group j , based on the total length of observation time and/or size of the population at risk of the event of interest and λ_j is the mean number of events per unit of exposure. λ_j are i.i.d. gamma(α, β), so $\phi = (\alpha, \beta)$. data is a 2-column matrix: the counts y_j of the numbers of events in column 1 and the corresponding exposures e_j in column 2.

Priors:

prior = "gamma" (the default): independent gamma priors on α and β , i.e. $\log\pi(\alpha, \beta) = (s_1 - 1)\log\alpha - r_1\alpha + (s_2 - 1)\log\beta - r_2\beta, \alpha > 0, \beta > 0$. where the respective shape (s_1, s_2) and rate (r_1, r_2) parameters are specified using `hpars = (s1, r1, s2, r2)`. The default setting is `hpars = c(1, 0.01, 1, 0.01)`.

Parameterizations for sampling:

param = "original" is (α, β) , param = "trans" (the default) is $\phi_1 = \log(\alpha/\beta), \phi_2 = \log(\beta)$.

Value

An object (list) of class "hef", which has the same structure as an object of class "ru" returned from [ru](#). In particular, the columns of the `n`-row matrix `sim_vals` contain the simulated values of ϕ . In addition this list contains the arguments `model`, `data` and `prior` detailed above, an `n` by `J` matrix `theta_sim_vals`: column `j` contains the simulated values of θ_j and `call`: the matched call to `hef`.

If `nrep` is not NULL then this list also contains `data_rep`, a numerical matrix with `nrep` columns. Each column contains a replication of the first column of the original data `data[, 1]`, simulated from the posterior predictive distribution.

References

Gelman, A., Carlin, J. B., Stern, H. S. Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014) *Bayesian Data Analysis*. Chapman & Hall / CRC. <http://www.stat.columbia.edu/~gelman/book/>

See Also

The `ru` function in the [rust](#) package for details of the arguments that can be passed to `ru` via `hef`.

[hanova1](#) for hierarchical one-way analysis of variance (ANOVA).

[set_user_prior](#) to set a user-defined prior.

Examples

```
##### Beta-binomial #####
# ----- Rat tumor data ----- #

# Default prior, sampling on (rotated) (log(mean), log(alpha + beta)) scale
rat_res <- hef(model = "beta_binom", data = rat)

# Hyperparameters alpha and beta
plot(rat_res)
# Parameterization used for sampling
plot(rat_res, ru_scale = TRUE)

summary(rat_res)

# Choose rats with extreme sample probabilities
pops <- c(which.min(rat[, 1] / rat[, 2]), which.max(rat[, 1] / rat[, 2]))
# Population-specific posterior samples: separate plots
plot(rat_res, params = "pop", plot_type = "both", which_pop = pops)
# Population-specific posterior samples: one plot
plot(rat_res, params = "pop", plot_type = "dens", which_pop = pops,
     one_plot = TRUE, add_legend = TRUE)

# Default prior, sampling on (rotated) (alpha, beta) scale
rat_res <- hef(model = "beta_binom", data = rat, param = "original")

plot(rat_res)
plot(rat_res, ru_scale = TRUE)

summary(rat_res)

# To produce a plot akin to Figure 5.3 of Gelman et al. (2014) we
# (a) Use the same prior for (alpha, beta)
# (b) Don't use axis rotation (rotate = FALSE)
# (c) Plot on the scale used for ratio-of-uniforms sampling (ru_scale = TRUE)
# (d) Note that the mode is relocated to (0, 0) in the plot
rat_res <- hef(model = "beta_binom", data = rat, rotate = FALSE)

plot(rat_res, ru_scale = TRUE)

# This is the estimated location of the posterior mode
rat_res$f_mode

# User-defined prior, passing parameters
# (equivalent to prior = "gamma" with hpars = c(1, 0.01, 1, 0.01))
user_prior <- function(x, hpars) {
  return(dexp(x[1], hpars[1], log = TRUE) + dexp(x[2], hpars[2], log = TRUE))
}
user_prior_fn <- set_user_prior(user_prior, hpars = c(0.01, 0.01))
rat_res <- hef(model = "beta_binom", data = rat, prior = user_prior_fn)

plot(rat_res)
```

```

summary(rat_res)

##### Gamma-Poisson #####

# ----- Pump failure data ----- #

pump_res <- hef(model = "gamma_pois", data = pump)
# Hyperparameters alpha and beta

plot(pump_res)

# Parameterization used for sampling
plot(pump_res, ru_scale = TRUE)
summary(pump_res)

# Choose pumps with extreme sample rates
pops <- c(which.min(pump[, 1] / pump[, 2]), which.max(pump[, 1] / pump[, 2]))
plot(pump_res, params = "pop", plot_type = "dens", which_pop = pops)

```

plot.hef

Plot diagnostics for a hef object

Description

plot method for class "hef".

Usage

```

## S3 method for class 'hef'
plot(
  x,
  y,
  ...,
  params = c("hyper", "ru", "pop"),
  which_pop = NULL,
  plot_type = NULL,
  one_plot = FALSE,
  add_legend = FALSE,
  legend_position = "topright",
  legend_text = NULL,
  num = 100
)

```

Arguments

x an object of class "hef", a result of a call to `ru`.

y Not used.

...	Additional arguments passed to <code>plot.ru</code> , <code>hist</code> or <code>pairs</code> . In particular, <code>ru_scale = TRUE</code> produces a plot using the parameterization used for ratio-of-uniforms sampling.
<code>params</code>	A character scalar that determines to which parameters the plots relate. <ul style="list-style-type: none"> • "hyper": the posterior sample of <i>all</i> hyperparameter values in ϕ is plotted using <code>plot.ru</code>. • "ru": only the posterior sample generated using <code>ru</code> is plotted using <code>plot.ru</code>. This produces a different plot to <code>params = "hyper"</code> if <code>ru</code> is used only on a subset of ϕ. For example, this may be the case if <code>x</code> is the result of a call to <code>hanova1</code>. See <code>vignette("bang-c-anova-vignette", package = "bang")</code> for information. • "pop": posterior samples and/or densities of the population-specific parameter θ are plotted. The population(s) included are determined by <code>which_pop</code> and the type of plot is determined by <code>plot_type</code>. If <code>plot_type</code> is not supplied then it is set to "dens".
<code>which_pop</code>	An integer vector or character scalar. If <code>params = "pop"</code> then <code>which_pop</code> indicates which populations to include in the plot. If <code>which_pop</code> is supplied then <code>params</code> is set to "pop". If <code>which_pop = "all"</code> then all populations are included. If there are many populations then this may fail if <code>plot_type = "pairs"</code> and/or <code>one_plot = FALSE</code> .
<code>plot_type</code>	A character scalar that determines the type of plot produced when <code>params = "pop"</code> . If <code>plot_type</code> is supplied then <code>params</code> is set automatically to "pop". <ul style="list-style-type: none"> • "sim": histograms of the posterior samples of θ for the populations in <code>which_pop</code>. • "dens": estimates of the marginal posterior densities of θ for the populations in <code>which_pop</code>. • "both": both the histograms and estimated posterior densities. • "pairs": pairwise scatter plots of the posterior samples of θ for the populations in <code>which_pop</code>, which must have length greater than one.
<code>one_plot</code> , <code>add_legend</code> , <code>legend_position</code> , <code>legend_text</code>	Only relevant if <code>plot_type = "dens"</code> . If <code>one_plot = TRUE</code> then the estimated marginal posterior densities are plotted in the same graph and if <code>add_legend = TRUE</code> then a legend is added to this plot using <code>legend</code> in the position indicated by the character scalar <code>legend_position</code> . A character vector <code>legend_text</code> may be used to override the default legend text.
<code>num</code>	A numeric scalar. If <code>plot_type == "dens"</code> or <code>plot_type == "both"</code> then <code>num</code> gives the number of points at which the marginal densities are evaluated to produce plots.

Examples

See the examples in `hef` and `hanova1`.

See Also

`plot.ru` for arguments that may be passed via `....`, in particular `ru_scale`.

pp_check.hef

Posterior predictive checks for a hef object

Description

pp_check method for class "hef". This provides an interface to the functions that perform posterior predictive checks in the **bayesplot** package. See [PPC-overview](#) for details of these functions.

Usage

```
## S3 method for class 'hef'
pp_check(object, fun = NULL, raw = FALSE, nrep = NULL, ...)
```

Arguments

object	An object of class "hef", a result of a call to hef or hanova1 .
fun	The plotting function to call. Can be any of the functions detailed at PPC-overview . The "ppc_" prefix can optionally be dropped if fun is specified as a string.
raw	Only relevant if <code>object\$model = "beta_binom"</code> or <code>object\$model = "gamma_pois"</code> . If <code>raw = TRUE</code> then the raw responses are used in the plots. Otherwise, the <i>proportions</i> of successes are used in the <code>beta_binom</code> case and the <i>exposure-adjusted rate</i> in the <code>gamma_pois</code> case. In both cases the values used are <code>object\$data[, 1] / object\$data[, 2]</code> and the equivalent in <code>object\$data_rep</code> .
nrep	The number of predictive replicates to use. If <code>nrep</code> is supplied then the first <code>nrep</code> rows of <code>object\$data_rep</code> are used. Otherwise, or if <code>nrep</code> is greater than <code>nrow(object\$data_rep)</code> , then all rows are used.
...	Additional arguments passed on to <code>bayesplot</code> functions. See Examples below.

Details

For details of these functions see [PPC-overview](#). See also the vignettes [Conjugate Hierarchical Models](#), [Hierarchical 1-way Analysis of Variance](#) and the **bayesplot** vignette [Graphical posterior predictive checks](#).

The general idea is to compare the observed data `object$data` with a matrix `object$data_rep` in which each row is a replication of the observed data simulated from the posterior predictive distribution. For greater detail see Chapter 6 of Gelman et al. (2013).

Value

A `ggplot` object that can be further customized using the **ggplot2** package.

References

Jonah Gabry (2016). bayesplot: Plotting for Bayesian Models. R package version 1.1.0. <https://CRAN.R-project.org/package=bayesplot>

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman & Hall/CRC Press, London, third edition. (Chapter 6). <http://www.stat.columbia.edu/~gelman/book/>

See Also

[hef](#) and [hanova1](#) for sampling from posterior distributions of hierarchical models.

bayesplot functions [PPC-overview](#), [PPC-distributions](#), [PPC-test-statistics](#), [PPC-intervals](#), [pp_check](#).

Examples

```
##### Beta-binomial #####
# ----- Rat tumor data ----- #
rat_res <- hef(model = "beta_binom", data = rat, nrep = 50)

# Overlaid density estimates
pp_check(rat_res)

# Overlaid distribution function estimates
pp_check(rat_res, fun = "ecdf_overlay")

# Multiple histograms
pp_check(rat_res, fun = "hist", nrep = 8)

# Multiple boxplots
pp_check(rat_res, fun = "boxplot")
# Predictive medians vs observed median
pp_check(rat_res, fun = "stat", stat = "median")

# Predictive (mean, sd) vs observed (mean, sd)
pp_check(rat_res, fun = "stat_2d", stat = c("mean", "sd"))

##### Gamma-Poisson #####
# ----- Pump failure data ----- #
pump_res <- hef(model = "gamma_pois", data = pump, nrep = 50)

# Overlaid density estimates
pp_check(pump_res)
# Predictive (mean, sd) vs observed (mean, sd)
pp_check(pump_res, fun = "stat_2d", stat = c("mean", "sd"))

##### One-way Hierarchical ANOVA #####
```

```
#----- Late 21st Century Global Temperature Data ----- #
RCP26_2 <- temp2[temp2$RCP == "rcp26", ]
temp_res <- hanova1(resp = RCP26_2[, 1], fac = RCP26_2[, 2], nrep = 50)

# Overlaid density estimates
pp_check(temp_res)
# Predictive (mean, sd) vs observed (mean, sd)
pp_check(temp_res, fun = "stat_2d", stat = c("mean", "sd"))
```

print.hef	<i>Print method for objects of class "hef"</i>
-----------	--

Description

print method for class "hef".

Usage

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

Arguments

x an object of class "hef", a result of a call to [hef](#) or [hanova1](#).
 ... Additional optional arguments. At present no optional arguments are used.

Details

Prints the original call to [hef](#) or [hanova1](#), the name of the model and the number of populations in the hierarchical model.

Value

The argument x, invisibly, as for all [print](#) methods.

See Also

[hef](#) for hierarchical exponential family models.
[hanova1](#) for hierarchical one-way analysis of variance (ANOVA).

```
print.summary.hef      Print method for objects of class "summary.hef"
```

Description

print method for class "summary.hef".

Usage

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

Arguments

x an object of class "summary.hef", a result of a call to [summary.hef](#).
 ... Additional optional arguments to be passed to [print](#).

Details

What is printed depends on the argument params supplied to [summary.hef](#).

Value

The argument x, invisibly, as for all [print](#) methods.

See Also

[summary.hef](#): summary method for class "hef".
[hef](#) for hierarchical exponential family models.
[hanova1](#) for hierarchical one-way analysis of variance (ANOVA).

```
pump                    Pump-failure data
```

Description

Data on pump failures from Gaver, D. P. and O'Muircheartaigh, I. G. (1987). The matrix pump has 10 rows and 2 columns. Each row relates to a different pump system. The first column contains the number of pump failures. The second column contains the length of operating time, in thousands of hours.

Usage

```
pump
```


Format

A matrix 10 rows and 2 columns.

Source

Table 3 of Gaver, D. P. and O’Muircheartaigh, I. G. (1987). See also Gelfand, A. E. and Smith, A. F. M. (1990).

References

Gaver, D. P. and O’Muircheartaigh, I. G. (1987) Robust Empirical Bayes Analyses of Event Rates. *Technometrics*, **29**, 1-15. doi:10.1080/00401706.1987.10488178

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, **85**(410), 398-409. doi:10.1080/01621459.1990.10476213

rat	<i>Rat tumor data</i>
-----	-----------------------

Description

Tumor incidence in 71 groups of rate from Tarone (1982). The matrix `rat` has 71 rows and 2 columns. Each row relates to a different group of rats. The first column (`y`) contains the number of rats with tumors. The second column (`n`) contains the total number of rats.

Usage

`rat`

Format

A matrix with 71 rows and 2 columns.

Source

Table 5.1 of Gelman, A., Carlin, J. B., Stern, H. S. Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014) *Bayesian Data Analysis*, Chapman & Hall / CRC. <http://www.stat.columbia.edu/~gelman/book/data/rats.asc>

References

Tarone, R. E. (1982) The use of historical information in testing for a trend in proportions. *Biometrics*, **38**, 215-220. doi:10.2307/2530304

set_user_prior	<i>Set a user-defined prior</i>
----------------	---------------------------------

Description

Constructs a user-defined prior distribution for use as the argument prior in [hef](#) or [hanova1](#).

Usage

```
set_user_prior(
  prior,
  ...,
  model = c("beta_binom", "gamma_pois", "anova1"),
  anova_d = 2
)
```

Arguments

prior	An R function returning the log of the prior density for of (perhaps a subset of) the hyperparameter vector ϕ .
...	Further arguments giving the names and values of any parameters involved in the function prior.
model	A character string. Abbreviated name of the model: "beta_binom" for beta-binomial and "gamma_pois" for gamma-Poisson (see hef), "anova1" for 1-way ANOVA (see hanova1).
anova_d	An integer scalar. Only relevant if model = anova1. If anova_d = 2 then prior must return the log-prior density for the standard deviations (σ_α, σ) and a normal prior with mean μ_0 and standard deviation σ_0 is used for μ . The values of $\mu_0 = 0$ and $\sigma_0 = \text{Inf}$ are set in the call to hanova1 , with default values $\mu_0 = 0$ and $\sigma_0 = \text{Inf}$. If anova_d = 3 then prior must return the log-prior density for $(\mu, \sigma_\alpha, \sigma)$.

Details

For details of the hyperparameters in ϕ see the **Details** section of [hef](#) for the models beta_binom and gamma_pois and of [hanova1](#) for the model anova1.

Value

A list of class "bang_prior". Will contain the component prior, the user-supplied function to evaluate the log of the prior, and any arguments supplied in

See Also

[hef](#) for hierarchical exponential family models.

[hanova1](#) for hierarchical one-way analysis of variance (ANOVA).

Examples

```
# User-defined prior, passing parameters
# (equivalent to prior = "gamma" with hpars = c(1, 0.01, 1, 0.01))
user_prior <- function(x, hpars) {
  return(dexp(x[1], hpars[1], log = TRUE) + dexp(x[2], hpars[2], log = TRUE))
}
user_prior_fn <- set_user_prior(user_prior, hpars = c(0.01, 0.01))
```

sim_pred_beta_binom *Simulate from a beta-binomial posterior predictive distribution*

Description

Simulates `nrep` draws from the posterior predictive distribution of the beta-binomial model described in [hef](#). This function is called within [hef](#) when the argument `nrep` is supplied.

Usage

```
sim_pred_beta_binom(theta_sim_vals, data, nrep)
```

Arguments

`theta_sim_vals` A numeric matrix with `nrow(data)` columns. Each row of `theta_sim_vals` contains binomial success probabilities simulated from their posterior distribution.

`data` A 2-column numeric matrix: the numbers of successes in column 1 and the corresponding numbers of trials in column 2.

`nrep` A numeric scalar. The number of replications of the original dataset simulated from the posterior predictive distribution. If `nrep` is greater than `nrow(theta_sim_vals)` then `nrep` is set equal to `nrow(theta_sim_vals)`.

Value

A numeric matrix with `nrep` columns. Each column contains a draw from the posterior predictive distribution of the number of successes.

Examples

```
rat_res <- hef(model = "beta_binom", data = rat)
rat_sim_pred <- sim_pred_beta_binom(rat_res$theta_sim_vals, rat, 50)
```

sim_pred_gamma_pois *Simulate from a gamma-Poisson posterior predictive distribution*

Description

Simulates `nrep` draws from the posterior predictive distribution of the beta-binomial model described in [hef](#). This function is called within [hef](#) when the argument `nrep` is supplied.

Usage

```
sim_pred_gamma_pois(theta_sim_vals, data, nrep)
```

Arguments

`theta_sim_vals` A numeric matrix with `nrow(data)` columns. Each row of `theta_sim_vals` contains binomial success probabilities simulated from their posterior distribution.

`data` A 2-column numeric matrix: the numbers of successes in column 1 and the corresponding numbers of trials in column 2.

`nrep` A numeric scalar. The number of replications of the original dataset simulated from the posterior predictive distribution. If `nrep` is greater than `nrow(theta_sim_vals)` then `nrep` is set equal to `nrow(theta_sim_vals)`.

Value

A numeric matrix with `nrep` columns. Each column contains a draw from the posterior predictive distribution of the number of successes.

Examples

```
pump_res <- hef(model = "gamma_pois", data = pump)
pump_sim_pred <- sim_pred_gamma_pois(pump_res$theta_sim_vals, pump, 50)
```

sim_pred_hanova1 *Simulate from a one-way hierarchical ANOVA posterior predictive distribution*

Description

Simulates `nrep` draws from the posterior predictive distribution of the one-way hierarchical ANOVA model described in [hanova1](#). This function is called within [hanova1](#) when the argument `nrep` is supplied.

Usage

```
sim_pred_hanova1(theta_sim_vals, sim_vals, fac, nrep)
```

Arguments

theta_sim_vals	A numeric matrix with length(fac) columns. Each row of theta_sim_vals contains normal means simulated from their posterior distribution.
sim_vals	A numeric matrix with length(fac) columns. Each row of sim_vals contains normal standard deviations σ simulated from their posterior distribution.
fac	The argument fac to hanova1 , that is, a vector of class factor indicating group membership.
nrep	A numeric scalar. The number of replications of the original dataset simulated from the posterior predictive distribution. If nrep is greater than nrow(theta_sim_vals) then nrep is set equal to nrow(theta_sim_vals).

Value

A numeric matrix with nrep columns. Each column contains a draw from the posterior predictive distribution of the number of successes.

Examples

```
RCP26_2 <- temp2[temp2$RCP == "rcp26", ]
temp_res <- hanova1(resp = RCP26_2[, 1], fac = RCP26_2[, 2])
sim_pred <- sim_pred_hanova1(temp_res$theta_sim_vals, temp_res$sim_vals,
                             RCP26_2[, 2], 50)
```

summary.hef

Summarizing hef objects

Description

summary method for class "hef".

Usage

```
## S3 method for class 'hef'
summary(
  object,
  ...,
  params = c("hyper", "pop"),
  which_pop = 1:ncol(object$theta_sim_vals)
)
```

Arguments

object	an object of class "hef", a result of a call to hef .
...	Additional arguments passed on to summary.ru .

params	<p>A character scalar.</p> <p>If <code>params = "hyper"</code> then the posterior samples of all hyperparameter values in ϕ are summarized using summary.ru.</p> <p>If <code>params = "pop"</code> then only posterior samples of the populations specified in <code>which_pop</code> are summarized.</p>
which_pop	<p>An integer vector. If <code>params = "pop"</code> then <code>which_pop</code> indicates which populations, i.e. which columns of <code>object\$theta_sim_vals</code> to summarize, using summary. The default is all populations.</p>

Examples

```
# Beta-binomial model, rat data
rat_res <- hef(model = "beta_binom", data = rat)

# Posterior summaries of the hyperparameters alpha and beta
summary(rat_res)

# Posterior summaries of the binomial probability for rats 1 to 3
summary(rat_res, params = "pop", which_pop = 1:3)
```

temp1	<i>Mid 21st Century Global Temperature Projection Data</i>
-------	--

Description

Indices of global temperature change from late 20th century (1970-1999) to mid 21st century (2020-2049) based on data produced by the Fifth Coupled Model Intercomparison Project (CMIP5).

Usage

```
temp1
```

Format

A data frame with 270 rows and 4 columns.

- Column 1, index: anomaly of 2020-2049 mean relative to the 1970-1999 mean.
- Column 2, GCM: Abbreviated name of General Circulation Model.
- Column 3, RCP: Representative Concentration Pathway. One of `rcp26`, `rcp45`, `rcp60`, `rcp85`.
- Column 4, run: Simulation run number.

Details

The data frame temp1 data frame has 270 rows and 4 columns. Each row relates to a climate projection run from one of 38 different General Circulation Models (GCMs) under a particular Representative Concentration Pathway (RCP). Use `table(temp1[, c("GCM", "RCP")])` to see the numbers of runs under each RCP for each GCM. See Van Vuuren et al (2011) for an overview of RCPs and Northrop and Chandler (2014) for analyses of a similar older dataset (CMIP3). Column 1 contains the anomaly of the mean global temperature over the time period 2020-2049 relative to the mean global temperature over 1970-1999, i.e. the latter subtracted from the former. Column 2 contains an abbreviation for the name of the climate modelling research group and the GCM. Column 3 contains the RCP in the format rcpxx where xx is a radiative forcing level resulting from an anticipated future greenhouse gas emissions. Column 4 is the simulation run number.

Source

The raw data from which the indices are calculated are monthly CMIP5 scenario runs for global surface air temperature (tas) downloaded from the KNMI Climate Explorer (<https://climexp.knmi.nl/>) on 4/3/2015.

References

- Northrop, P.J. and R.E. Chandler (2014). Quantifying Sources of Uncertainty in Projections of Future Climate. *Journal of Climate*, **27**, 8793-8808. doi:10.1175/JCLI1400265.1
- Van Vuuren, D. P., Edmonds, J., Kainuma, M., Riahi, K. Thomson, A., Hibbard, K., Hurtt, G. C., Kram, T., Krey, V., Lamarque, J.-F. (2011). The representative concentration pathways: an overview. *Climatic change*, **109**, 5-31. doi:10.1007/s105840110148z

temp2

Late 21st Century Global Temperature Projection Data

Description

Indices of global temperature change from late 20th century (1970-1999) to late 21st century (2069-2098) based on data produced by the Fifth Coupled Model Intercomparison Project (CMIP5).

Usage

temp2

Format

A data frame with 270 rows and 4 columns.

- Column 1, index: anomaly of 2069-2098 mean relative to the 1970-1999 mean.
- Column 2, GCM: Abbreviated name of General Circulation Model.
- Column 3, RCP: Representative Concentration Pathway. One of rcp26, rcp45, rcp60, rcp85.
- Column 4, run: Simulation run number.

Details

The data frame temp2 data frame has 270 rows and 4 columns. Each row relates to a climate projection run from one of 38 different General Circulation Models (GCMs) under a particular Representative Concentration Pathway (RCP). Use `table(temp2[, c("GCM", "RCP")])` to see the numbers of runs under each RCP for each GCM. See Van Vuuren et al (2011) for an overview of RCPs and Northrop and Chandler (2014) for analyses of a similar older dataset (CMIP3). Column 1 contains the anomaly of the mean global temperature over the time period 2069-2098 relative to the mean global temperature over 1970-1999, i.e. the latter subtracted from the former. Column 2 contains an abbreviation for the name of the climate modelling research group and the GCM. Column 3 contains the RCP in the format rcp xx where xx is a radiative forcing level resulting from an anticipated future greenhouse gas emissions. Column 4 is the simulation run number.

Source

The raw data from which the indices are calculated are monthly CMIP5 scenario runs for global surface air temperature (tas) downloaded from the KNMI Climate Explorer (<https://climexp.knmi.nl/>) on 4/3/2015.

References

- Northrop, P.J. and R.E. Chandler (2014). Quantifying Sources of Uncertainty in Projections of Future Climate. *Journal of Climate*, **27**, 8793-8808. doi:10.1175/JCLI1400265.1
- Van Vuuren, D. P., Edmonds, J., Kainuma, M., Riahi, K. Thomson, A., Hibbard, K., Hurtt, G. C., Kram, T., Krey, V., Lamarque, J.-F. (2011). The representative concentration pathways: an overview. *Climatic change*, **109**, 5-31. doi:10.1007/s105840110148z

weight_gain

Weight Gained by Rats

Description

Data from an experiment to study weight gained by 10 rats fed on four different diets, defined by a combination of the amount of protein (low and high) and by the source of protein (beef and cereal).

Usage

weight_gain

Format

A data frame with 40 rows and 3 columns.

- Column 1, source: source of protein, a factor with levels Beef and Cereal.
- Column 2, type: amount of protein, a factor with levels High and Low.
- Column 3, weightgain: weight gained, in grams.

Source

D. J. Hand, A. D. Lunn, K. J. McConway, and E. Ostrowski (1994). *A Handbook of Small Datasets*, Chapman and Hall/CRC, London.

Index

* datasets

- coagulation, 3
 - pump, 16
 - rat, 17
 - temp1, 22
 - temp2, 23
 - weight_gain, 24
- bang (bang-package), 2
- bang-package, 2
- coagulation, 3
- factor, 4, 21
- hanova1, 2, 3, 4, 8, 9, 12–16, 18, 20, 21
- hef, 2, 3, 6, 7, 12–16, 18–21
- hist, 12
- legend, 12
- pairs, 12
- plot.hef, 11
- plot.ru, 12
- pp_check, 14
- pp_check (pp_check.hef), 13
- pp_check.hef, 13
- PPC-distributions, 14
- PPC-intervals, 14
- PPC-overview, 13, 14
- PPC-test-statistics, 14
- print, 15, 16
- print.hef, 15
- print.summary.hef, 16
- pump, 16
- rat, 17
- ru, 4–6, 8, 9, 11, 12
- rust, 5, 6, 8, 9
- set_user_prior, 3–6, 8, 9, 18
- sim_pred_beta_binom, 19
- sim_pred_gamma_pois, 20
- sim_pred_hanova1, 20
- summary, 22
- summary.hef, 16, 21
- summary.ru, 21, 22
- temp1, 22
- temp2, 23
- weight_gain, 24