Package ‘bdrc’

May 7, 2026

Title Bayesian Discharge Rating Curves

Version 2.0.1

Maintainer Rafael Daníel Vias <raffidv@gmail.com>

Description Fits a discharge rating curve based on the power-law and the generalized power-law from data on paired stage and discharge measurements in a given river using a Bayesian hierarchical model as described in Hrafnkelsson et al. (2022) <doi:10.1002/env.2711>.

Depends R (>= 3.5.0)

LazyData true

License MIT + file LICENSE

RoxygenNote 7.3.2

Imports ggplot2, grid, gridExtra, Rcpp (>= 0.11.0), rlang, scales

Suggests testthat, knitr, rmarkdown, covr, vdiffr

VignetteBuilder knitr

URL https://sor16.github.io/bdrc/

BugReports https://github.com/sor16/bdrc/issues/

Encoding UTF-8

LinkingTo Rcpp (>= 0.11.0), RcppArmadillo (>= 0.2.35)

NeedsCompilation yes

Author Birgir Hrafnkelsson [aut, cph] (ORCID: <https://orcid.org/0000-0003-1864-9652>), Solvi Rognvaldsson [aut] (ORCID: <https://orcid.org/0000-0002-4376-3361>), Rafael Daníel Vias [aut, cre] (ORCID: <https://orcid.org/0009-0007-2601-6800>), Axel Orn Jansson [aut]

Repository CRAN

Date/Publication 2026-02-17 23:10:21 UTC

Contents

`autoplot.plm0` *Autoplot method for discharge rating curves*

Description

Visualize discharge rating curve model objects

Usage

```
## S3 method for class 'plm0'
autoplot(
  object,
  ...,
  type = "rating_curve",
  param = NULL,
```

```
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL
)

## S3 method for class 'plm'
autoplot(
  object,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)

## S3 method for class 'gplm0'
autoplot(
  object,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)

## S3 method for class 'gplm'
autoplot(
  object,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

`object` An object of class "plm0", "plm", "gplm0" or "gplm".

`...` Not used in this function

`type` A character denoting what type of plot should be drawn. Defaults to "rating_curve". Possible types are:

rating_curve Plots the rating curve.

rating_curve_mean Plots the posterior mean of the rating curve.

f Plots the power-law exponent.

beta Plots the random effect in the power-law exponent.

sigma_eps Plots the standard deviation on the data level.

residuals Plots the log residuals.

trace Plots trace plots of parameters given in param.

histogram Plots histograms of parameters given in param.

param	A character vector with the parameters to plot. Defaults to NULL and is only used if type is "trace" or "histogram". Allowed values are the parameters given in the model summary of x as well as "hyperparameters" or "latent_parameters" for specific groups of parameters.
transformed	A logical value indicating whether the quantity should be plotted on a transformed scale used during the Bayesian inference. Defaults to FALSE.
title	A character denoting the title of the plot.
xlim	A numeric vector of length 2, denoting the limits on the x axis of the plot. Applicable for types "rating_curve", "rating_curve_mean", "f", "beta", "sigma_eps", "residuals".
ylim	A numeric vector of length 2, denoting the limits on the y axis of the plot. Applicable for types "rating_curve", "rating_curve_mean", "f", "beta", "sigma_eps", "residuals".

Value

Returns an object of class "ggplot2".

Functions

- autoplot(plm0): Autoplot method for plm0
- autoplot(plm): Autoplot method for plm
- autoplot(gplm0): Autoplot method for gplm0
- autoplot(gplm): Autoplot method for gplm

See Also

plm0, plm, gplm0 and gplm for fitting a discharge rating curve and summary.plm0, summary.plm, summary.gplm0 and summary.gplm for summaries. It is also useful to look at spread_draws and gather_draws to work directly with the MCMC samples.

Examples

```
library(ggplot2)
data(krokfors)
set.seed(1)
plm0.fit <- plm0(Q~W,krokfors,num_cores=2)
autoplot(plm0.fit)
```

```
autoplot(plm0.fit,transformed=TRUE)
autoplot(plm0.fit,type='histogram',param='c')
autoplot(plm0.fit,type='histogram',param='c',transformed=TRUE)
autoplot(plm0.fit,type='histogram',param='hyperparameters')
autoplot(plm0.fit,type='histogram',param='latent_parameters')
autoplot(plm0.fit,type='residuals')
autoplot(plm0.fit,type='f')
autoplot(plm0.fit,type='sigma_eps')
```

---

autoplot.tournament *Autoplot method for discharge rating curve tournament*

Description

Compare the four discharge rating curves from the tournament object in different ways

Usage

```
## S3 method for class 'tournament'
autoplot(object, type = "boxplot", ...)
```

Arguments

object	An object of class "tournament"
type	A character denoting what type of plot should be drawn. Possible types are
...	Not used in this function
	boxplot Creates a boxplot of the posterior log-likelihood values transformed to the deviance scale.

Value

Returns an object of class "ggplot2".

See Also

tournament to run a discharge rating curve tournament and summary.tournament for summaries.

Examples

```
library(ggplot2)
data(krokfors)
set.seed(1)
t_obj <- tournament(formula = Q ~ W, data = krokfors, num_cores = 2)
autoplot(t_obj)
```

gather_draws — *Gather MCMC chain draws to data.frame on a long format*

Description

Useful to convert MCMC chain draws of particular parameters or output from the model object to a long format for further data wrangling

Usage

```
gather_draws(mod, ..., transformed = F)
```

Arguments

`mod`	An object of class "plm0", "plm", "gplm0" or "gplm".
`...`	Any number of character vectors containing valid names of parameters in the model or "rating_curve" and "rating_curve_mean". Also accepts "latent_parameters" and "hyperparameters".
`transformed`	A boolean value determining whether the parameter is to be represented on the transformed scale used for sampling in the MCMC chain or the original scale. Defaults to FALSE.

Value

A data frame with columns:

`chain` The chain number.

`iter` The iteration number.

`param` The parameter name.

`value` The parameter value.

References

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

See Also

`plm0`, `plm`, `gplm0`, `gplm` for further information on parameters

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,num_cores=2)
hyp_samples <- gather_draws(plm0.fit,'hyperparameters')
head(hyp_samples)
rating_curve_samples <- gather_draws(plm0.fit,'rating_curve','rating_curve_mean')
head(rating_curve_samples)
```

---

get_report — *Report for a discharge rating curve or tournament*

Description

Save a pdf file with a report of a discharge rating curve object or tournament.

Usage

```
get_report(x, path = NULL, type = 1, ...)

## S3 method for class 'plm0'
get_report(x, path = NULL, type = 1, ...)

## S3 method for class 'plm'
get_report(x, path = NULL, type = 1, ...)

## S3 method for class 'gplm0'
get_report(x, path = NULL, type = 1, ...)

## S3 method for class 'gplm'
get_report(x, path = NULL, type = 1, ...)

## S3 method for class 'tournament'
get_report(x, path = NULL, type = 1, ...)
```

Arguments

`x`	An object of class "tournament", "plm0", "plm", "gplm0" or "gplm".
`path`	A file path to which the pdf file of the report is saved. If NULL, the current working directory is used.
`type`	An integer denoting what type of report is to be produced. Defaults to type 1. Only type 1 is permissible for an object of class "plm0", "plm", "gplm0" or "gplm". Possible types are:

1 Produces a report displaying the results of the model (winning model if a tournament provided). The first page contains a panel of four plots and a summary of the posterior distributions of the parameters. On the second page a tabular prediction of discharge on an equally spaced grid of stages is displayed. This prediction table can span multiple pages.

2 Produces a ten page report and is only permissible for objects of class "tournament". The first four pages contain a panel of four plots and a summary of the posterior distributions of the parameters for each of the four models in the tournament, the fifth page shows a summary of the tournament model comparison, the sixth page convergence diagnostics plots, and the final four pages shows the histograms of the parameters in each of the four models.

`...` Further arguments passed to other methods (currently unused).

Details

This function can only be used in an interactive R session as it asks permission from the user to write to their file system.

Value

No return value, called for side effects.

Methods (by class)

- `get_report(plm0)`: Get report for plm0 model object
- `get_report(plm)`: Get report for plm model object
- `get_report(gplm0)`: Get report for gplm0 model object
- `get_report(gplm)`: Get report for gplm
- `get_report(tournament)`: Get report for discharge rating curve tournament

See Also

`get_report` for generating and saving a report.

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,num_cores=2)

## Not run:
get_report(plm0.fit)

## End(Not run)
```

`get_report_pages` *Report pages for a discharge rating curve or tournament*

Description

Get a list of the pages of a report on a discharge rating curve model or tournament

Usage

```
get_report_pages(x, type = 1)

## S3 method for class 'plm0'
get_report_pages(x, type = 1)

## S3 method for class 'plm'
get_report_pages(x, type = 1)

## S3 method for class 'gplm0'
get_report_pages(x, type = 1)

## S3 method for class 'gplm'
get_report_pages(x, type = 1)

## S3 method for class 'tournament'
get_report_pages(x, type = 1)
```

Arguments

`x` An object of class "tournament", "plm0", "plm", "gplm0" or "gplm".

`type` An integer denoting what type of report is to be produced. Defaults to type 1. Possible types are:

1 Produces a report displaying the results of the model (winning model if a tournament provided). The first page contains a panel of four plots and a summary of the posterior distributions of the parameters. On the second page a tabular prediction of discharge on an equally spaced grid of stages is displayed. This prediction table can span multiple pages.

2 Produces a ten page report and is only permissible for objects of class "tournament". The first four pages contain a panel of four plots and a summary of the posterior distributions of the parameters for each of the four models in the tournament, the fifth page shows a summary of the tournament model comparison, the sixth page convergence diagnostics plots, and the final four pages shows the histograms of the parameters in each of the four models.

Value

A list of objects of type "grob" that correspond to the pages in a rating curve report.

Methods (by class)

- `get_report_pages(plm0)`: Get report pages for plm0 model object
- `get_report_pages(plm)`: Get report pages for plm model object
- `get_report_pages(gplm0)`: Get report pages for gplm0 model object
- `get_report_pages(gplm)`: Get report pages for gplm model object
- `get_report_pages(tournament)`: Get report pages for discharge rating curve tournament model object

See Also

`tournament` for running a tournament, `summary.tournament` for summaries and `get_report` for generating and saving a report of a tournament object.

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,num_cores=2)
plm0_pages <- get_report_pages(plm0.fit)
```

---

gplm *Generalized power-law model with variance that varies with stage.*

Description

gplm is used to fit a discharge rating curve for paired measurements of stage and discharge using a generalized power-law model with variance that varies with stage as described in Hrafnkelsson et al. (2022). See "Details" for a more elaborate description of the model.

Usage

```
gplm(
  formula,
  data,
  c_param = NULL,
  h_max = NULL,
  parallel = TRUE,
  num_cores = NULL,
  forcepoint = rep(FALSE, nrow(data)),
  verbose = TRUE
)
```

Arguments

- `formula` — An object of class "formula", with discharge column name as response and stage column name as a covariate, i.e. of the form `y~x` where `y` is discharge in m^3/s and `x` is stage in m (it is very important that the data is in the correct units).
- `data` — A data.frame containing the variables specified in formula.
- `c_param` — The largest stage value for which there is zero discharge. If NULL, it is treated as unknown in the model and inferred from the data.
- `h_max` — The maximum stage to which the rating curve should extrapolate to. If NULL, the maximum stage value in the data is selected as an upper bound.
- `parallel` — A logical value indicating whether to run the MCMC in parallel or not. Defaults to TRUE.
- `num_cores` — An integer between 1 and 4 (number of MCMC chains) indicating how many cores to use. Only used if parallel=TRUE. If NULL, the number of cores available on the device is detected automatically.
- `forcepoint` — A logical vector of the same length as the number of rows in data. If an element at index i is TRUE it indicates that the rating curve should be forced through the i-th measurement. Use with care, as this will strongly influence the resulting rating curve.
- `verbose` — A logical value indicating whether to print progress and diagnostic information. If ‘TRUE‘, the function will print messages as it runs. If ‘FALSE‘, the function will run silently. Default is ‘TRUE‘.

Details

The generalized power-law model is of the form

$$Q = a(h-c)^{f(h)}$$

where Q is discharge, h is stage, a and c are unknown constants and f is a function of h, referred to as the generalized power-law exponent.

The generalized power-law model is here inferred by using a Bayesian hierarchical model. The function f is modeled at the latent level as a fixed constant b plus a continuous stochastic process, $\beta(h)$, which is assumed to be twice differentiable. The model is on a logarithmic scale

$$\log(Q_i) = \log(a) + (b + \beta(h_i))\log(h_i - c) + \varepsilon_i, i = 1, ..., n$$

where ε_i follows a normal distribution with mean zero and variance $\sigma_\varepsilon(h_i)^2$ that varies with stage. The stochastic process $\beta(h)$ is assumed a priori to be a Gaussian process governed by a Matern covariance function with smoothness parameter $\nu = 2.5$. The error variance, $\sigma_\varepsilon^2(h)$, of the log-discharge data is modeled as an exponential of a B-spline curve, that is, a linear combination of six B-spline basis functions that are defined over the range of the stage observations. An efficient posterior simulation is achieved by sampling from the joint posterior density of the hyperparameters of the model, and then sampling from the density of the latent parameters conditional on the hyperparameters.

Bayesian inference is based on the posterior density and summary statistics such as the posterior mean and 95% posterior intervals are based on the posterior density. Analytical formulas for these summary statistics are intractable in most cases and thus they are computed by generating samples from the posterior density using a Markov chain Monte Carlo simulation.

Value

gplm returns an object of class "gplm". An object of class "gplm" is a list containing the following components:

`rating_curve` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior predictive distribution of the rating curve.

`rating_curve_mean` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of the mean of the rating curve.

`param_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of latent- and hyperparameters. Additionally contains columns with r_hat and the effective number of samples for each parameter as defined in Gelman et al. (2013).

`f_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of $f(h)$.

`beta_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of $\beta(h)$.

`sigma_eps_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of $\sigma_\varepsilon(h)$.

`posterior_log_likelihood_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior log-likelihood values.

`rating_curve_posterior` A matrix containing the full thinned posterior samples of the posterior predictive distribution of the rating curve excluding burn-in samples.

`rating_curve_mean_posterior` A matrix containing the full thinned posterior samples of the posterior distribution of the mean of the rating curve excluding burn-in samples.

`a_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of a excluding burn-in samples.

`b_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of b excluding burn-in samples.

`c_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of c excluding burn-in samples.

`sigma_beta_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_β excluding burn-in samples.

`phi_beta_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of ϕ_β excluding burn-in samples.

`sigma_eta_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_η excluding burn-in samples.

`eta_1_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_1 excluding burn-in samples.

`eta_2_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_2 excluding burn-in samples.

`eta_3_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_3 excluding burn-in samples.

`eta_4_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_4 excluding burn-in samples.

`eta_5_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_5 excluding burn-in samples.

`eta_6_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_6 excluding burn-in samples.

`f_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of $f(h)$ excluding burn-in samples.

`beta_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of $\beta(h)$ excluding burn-in samples.

`sigma_eps_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of $\sigma_\varepsilon(h)$ excluding burn-in samples.

`posterior_log_likelihood` A numeric vector containing the full thinned posterior log-likelihood values, excluding burn-in samples.

`D_hat` A statistic defined as -2 times the log-likelihood evaluated at the median value of the parameters.

`effective_num_param_DIC` The effective number of parameters, which is calculated as median(-2*posterior_log_likelihood) minus D_hat.

`DIC` The Deviance Information Criterion for the model, calculated as D_hat plus 2*effective_num_parameters_DIC.

`lppd` The log pointwise predictive density of the observed data under the model.

`WAIC` The Widely Applicable Information Criterion for the model, defined as -2*(lppd - effective_num_param_WAIC).

`WAIC_i` The pointwise WAIC values, where WAIC := sum(WAIC_i).

`effective_num_param_WAIC` The effective number of parameters, which is calculated by summing up the posterior variance of the log predictive density for each data point.

`autocorrelation` A data frame with the autocorrelation of each parameter for different lags.

`acceptance_rate` The proportion of accepted samples in the thinned MCMC chain (excluding burn-in).

`formula` An object of type "formula" provided by the user.

`data` The data provided by the user, ordered by stage.

`run_info` The information about the input arguments and the specific parameters used in the MCMC chain.

References

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. doi: https://doi.org/10.1201/b16018

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

Spiegelhalter, D., Best, N., Carlin, B., Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(4), 583–639. doi: https://doi.org/10.1111/1467-9868.00353

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. J. Mach. Learn. Res. 11, 3571–3594.

See Also

`summary.gplm` for summaries, `predict.gplm` for prediction and `plot.gplm` for plots. `spread_draws` and `gather_draws` are also useful to aid further visualization of the full posterior distributions.

Examples

```
data(norn)
set.seed(1)
gplm.fit <- gplm(formula=Q~W,data=norn,num_cores=2)
summary(gplm.fit)
```

---

gplm0 — *Generalized power-law model with a constant variance*

Description

gplm0 is used to fit a discharge rating curve for paired measurements of stage and discharge using a generalized power-law model with a constant variance as described in Hrafnkelsson et al. (2022). See "Details" for a more elaborate description of the model.

Usage

```
gplm0(
  formula,
  data,
  c_param = NULL,
  h_max = NULL,
  parallel = TRUE,
  num_cores = NULL,
  forcepoint = rep(FALSE, nrow(data)),
  verbose = TRUE
)
```

Arguments

`formula`	An object of class "formula", with discharge column name as response and stage column name as a covariate, i.e. of the form `y~x` where `y` is discharge in m^3/s and `x` is stage in m (it is very important that the data is in the correct units).
`data`	A data.frame containing the variables specified in formula.

`c_param` The largest stage value for which there is zero discharge. If NULL, it is treated as unknown in the model and inferred from the data.

`h_max` The maximum stage to which the rating curve should extrapolate to. If NULL, the maximum stage value in the data is selected as an upper bound.

`parallel` A logical value indicating whether to run the MCMC in parallel or not. Defaults to TRUE.

`num_cores` An integer between 1 and 4 (number of MCMC chains) indicating how many cores to use. Only used if parallel=TRUE. If NULL, the number of cores available on the device is detected automatically.

`forcepoint` A logical vector of the same length as the number of rows in data. If an element at index i is TRUE it indicates that the rating curve should be forced through the i-th measurement. Use with care, as this will strongly influence the resulting rating curve.

`verbose` A logical value indicating whether to print progress and diagnostic information. If 'TRUE', the function will print messages as it runs. If 'FALSE', the function will run silently. Default is 'TRUE'.

Details

The generalized power-law model is of the form

$$Q = a(h-c)^{f(h)}$$

where Q is discharge, h is stage, a and c are unknown constants and f is a function of h referred to as the generalized power-law exponent.

The generalized power-law model is here inferred by using a Bayesian hierarchical model. The function f is modeled at the latent level as a fixed constant b plus a continuous stochastic process,$\beta(h)$, which is assumed to be twice differentiable. The model is on a logarithmic scale

$$\log(Q_i) = \log(a) + (b + \beta(h_i))\log(h_i - c) + \varepsilon, i = 1, ..., n$$

where ε follows a normal distribution with mean zero and variance σ_ε^2, independent of stage. The stochastic process $\beta(h)$ is assumed a priori to be a Gaussian process governed by a Matern covariance function with smoothness parameter $\nu = 2.5$. An efficient posterior simulation is achieved by sampling from the joint posterior density of the hyperparameters of the model, and then sampling from the density of the latent parameters conditional on the hyperparameters.

Bayesian inference is based on the posterior density and summary statistics such as the posterior mean and 95% posterior intervals are based on the posterior density. Analytical formulas for these summary statistics are intractable in most cases and thus they are computed by generating samples from the posterior density using a Markov chain Monte Carlo simulation.

Value

gplm0 returns an object of class "gplm0". An object of class "gplm0" is a list containing the following components:

rating_curve A data frame with 2.5%, 50% and 97.5% percentiles of the posterior predictive distribution of the rating curve.

rating_curve_mean A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of the mean of the rating curve.

param_summary A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of latent- and hyperparameters. Additionally contains columns with r_hat and the effective number of samples for each parameter as defined in Gelman et al. (2013).

beta_summary A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of β.

posterior_log_likelihood_summary A data frame with 2.5%, 50% and 97.5% percentiles of the posterior log-likelihood values.

rating_curve_posterior A matrix containing the full thinned posterior samples of the posterior predictive distribution of the rating curve (excluding burn-in).

rating_curve_mean_posterior A matrix containing the full thinned posterior samples of the posterior distribution of the mean of the rating curve (excluding burn-in).

a_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of a.

b_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of b.

c_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of c.

sigma_eps_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_{ε}.

sigma_beta_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_{β}.

phi_beta_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of ϕ_{β}.

sigma_eta_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_{η}.

beta_posterior A numeric vector containing the full thinned posterior samples of the posterior distribution of β.

posterior_log_likelihood A numeric vector containing the full thinned posterior log-likelihood values, excluding burn-in samples.

D_hat A statistic defined as -2 times the log-likelihood evaluated at the median value of the parameters.

effective_num_param_DIC The effective number of parameters, which is calculated as median(-2*posterior_log_likelihood) minus D_hat.

DIC The Deviance Information Criterion for the model, calculated as D_hat plus 2*effective_num_parameters_DIC.

lppd The log pointwise predictive density of the observed data under the model.

WAIC The Widely Applicable Information Criterion for the model, defined as -2*(lppd - effective_num_param_WAIC).

WAIC_i The pointwise WAIC values, where WAIC := sum(WAIC_i).

`effective_num_param_WAIC` The effective number of parameters, which is calculated by summing up the posterior variance of the log predictive density for each data point.

`autocorrelation` A data frame with the autocorrelation of each parameter for different lags.

`acceptance_rate` The proportion of accepted samples in the thinned MCMC chain (excluding burn-in).

`formula` An object of type "formula" provided by the user.

`data` The data provided by the user, ordered by stage.

`run_info` The information about the input arguments and the specific parameters used in the MCMC chain.

References

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. doi: https://doi.org/10.1201/b16018

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

Spiegelhalter, D., Best, N., Carlin, B., Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(4), 583–639. doi: https://doi.org/10.1111/1467-9868.00353

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. J. Mach. Learn. Res. 11, 3571–3594.

See Also

`summary.gplm0` for summaries, `predict.gplm0` for prediction. It is also useful to look at `spread_draws` and `plot.gplm0` to help visualize the full posterior distributions.

Examples

```
data(krokfors)
set.seed(1)
gplm0.fit <- gplm0(formula=Q~W,data=krokfors,num_cores=2)
summary(gplm0.fit)
```

`jokdal` *Jokulsa a Dal gauging station in Iceland*

Description

Data on discharge and stage from Jokulsa a Dal gauging station in Iceland

Usage

```
jokdal
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Icelandic Meteorological Office, Landsvirkjun - the National Power Company of Iceland, and the Icelandic Road and Coastal Administration.

`jokfjoll` *Jokulsa a Fjollum gauging station in Iceland*

Description

Data on discharge and stage from Jokulsa a Fjollum gauging station in Iceland

Usage

```
jokfjoll
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Icelandic Meteorological Office, Landsvirkjun - the National Power Company of Iceland, and the Icelandic Road and Coastal Administration.

`kallstorp` *Kallstorp gauging station in Sweden*

Description

Data on discharge and stage from Kallstorp gauging station in Sweden

Usage

```
kallstorp
```

Format

A data frame with columns:

`W` Measurements of water stage in meters

`Q` Measurements of water discharge in cubic meters per second

Source

Swedish Meteorological and Hydrological Institute.

`krokfors` *Krokfors gauging station in Sweden*

Description

Data on discharge and stage from Krokfors gauging station in Sweden.

Usage

```
krokfors
```

Format

A data frame with columns:

`W` Measurements of water stage in meters

`Q` Measurements of water discharge in cubic meters per second

Source

Swedish Meteorological and Hydrological Institute.

`melby` *Melby gauging station in Sweden*

Description

Data on discharge and stage from Melby gauging station in Sweden

Usage

```
melby
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Swedish Meteorological and Hydrological Institute.

`nordura` *Nordura gauging station in Iceland*

Description

Data on discharge and stage from Nordura gauging station in Iceland

Usage

```
nordura
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Icelandic Meteorological Office, Landsvirkjun - the National Power Company of Iceland, and the Icelandic Road and Coastal Administration.

`norn` *Norn gauging station in Sweden*

Description

Data on discharge and stage from Norn gauging station in Sweden.

Usage

```
norn
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Swedish Meteorological and Hydrological Institute.

`plm` *Power-law model with variance that varies with stage.*

Description

plm is used to fit a discharge rating curve for paired measurements of stage and discharge using a power-law model with variance that varies with stage as described in Hrafnkelsson et al. (2022). See "Details" for a more elaborate description of the model.

Usage

```
plm(
  formula,
  data,
  c_param = NULL,
  h_max = NULL,
  parallel = TRUE,
  num_cores = NULL,
  forcepoint = rep(FALSE, nrow(data)),
  verbose = TRUE
)
```

Arguments

`formula`	An object of class "formula", with discharge column name as response and stage column name as a covariate, i.e. of the form `y~x` where `y` is discharge in m^3/s and `x` is stage in `m` (it is very important that the data is in the correct units).
`data`	A data.frame containing the variables specified in formula.
`c_param`	The largest stage value for which there is zero discharge. If NULL, it is treated as unknown in the model and inferred from the data.
`h_max`	The maximum stage to which the rating curve should extrapolate to. If NULL, the maximum stage value in the data is selected as an upper bound.
`parallel`	A logical value indicating whether to run the MCMC in parallel or not. Defaults to TRUE.
`num_cores`	An integer between 1 and 4 (number of MCMC chains) indicating how many cores to use. Only used if parallel=TRUE. If NULL, the number of cores available on the device is detected automatically.
`forcepoint`	A logical vector of the same length as the number of rows in data. If an element at index i is TRUE it indicates that the rating curve should be forced through the i-th measurement. Use with care, as this will strongly influence the resulting rating curve.
`verbose`	A logical value indicating whether to print progress and diagnostic information. If 'TRUE', the function will print messages as it runs. If 'FALSE', the function will run silently. Default is 'TRUE'.

Details

The power-law model, which is commonly used in hydraulic practice, is of the form

$$Q = a(h-c)^b$$

where Q is discharge, h is stage and a, b and c are unknown constants.

The power-law model is here inferred by using a Bayesian hierarchical model. The model is on a logarithmic scale

$$\log(Q_i) = \log(a) + b\log(h_i - c) + \varepsilon_i, i = 1, ..., n$$

where ε_i follows a normal distribution with mean zero and variance $\sigma_\varepsilon(h_i)^2$ that varies with stage. The error variance, $\sigma_\varepsilon^2(h)$, of the log-discharge data is modeled as an exponential of a B-spline curve, that is, a linear combination of six B-spline basis functions that are defined over the range of the stage observations. An efficient posterior simulation is achieved by sampling from the joint posterior density of the hyperparameters of the model, and then sampling from the density of the latent parameters conditional on the hyperparameters.

Bayesian inference is based on the posterior density and summary statistics such as the posterior mean and 95% posterior intervals are based on the posterior density. Analytical formulas for these summary statistics are intractable in most cases and thus they are computed by generating samples from the posterior density using a Markov chain Monte Carlo simulation.

Value

plm returns an object of class "plm". An object of class "plm" is a list containing the following components:

`rating_curve` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior predictive distribution of the rating curve.

`rating_curve_mean` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of the mean of the rating curve. Additionally contains columns with r_hat and the effective number of samples for each parameter as defined in Gelman et al. (2013).

`param_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of latent- and hyperparameters.

`sigma_eps_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior of σ_ε.

`posterior_log_likelihood_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior log-likelihood values.

`rating_curve_posterior` A matrix containing the full thinned posterior samples of the posterior predictive distribution of the rating curve (excluding burn-in).

`rating_curve_mean_posterior` A matrix containing the full thinned posterior samples of the posterior distribution of the mean of the rating curve (excluding burn-in).

`a_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of a.

`b_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of b.

`c_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of c.

`sigma_eps_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_ε.

`eta_1_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_1.

`eta_2_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_2.

`eta_3_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_3.

`eta_4_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_4.

`eta_5_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_5.

`eta_6_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of η_6.

`posterior_log_likelihood` A numeric vector containing the full thinned posterior log-likelihood values, excluding burn-in samples.

`D_hat` A statistic defined as -2 times the log-likelihood evaluated at the median value of the parameters.

`effective_num_param_DIC` The effective number of parameters, which is calculated as median(-2*posterior_log_likelihood) minus D_hat.

`DIC` The Deviance Information Criterion for the model, calculated as D_hat plus 2*effective_num_parameters_DIC.

`lppd` The log pointwise predictive density of the observed data under the model.

`WAIC` The Widely Applicable Information Criterion for the model, defined as -2*(lppd - effective_num_param_WAIC).

`WAIC_i` The pointwise WAIC values, where WAIC := sum(WAIC_i).

`effective_num_param_WAIC` The effective number of parameters, which is calculated by summing up the posterior variance of the log predictive density for each data point.

`autocorrelation` A data frame with the autocorrelation of each parameter for different lags.

`acceptance_rate` The proportion of accepted samples in the thinned MCMC chain (excluding burn-in).

`formula` An object of type "formula" provided by the user.

`data` The data provided by the user, ordered by stage.

`run_info` The information about the input arguments and the specific parameters used in the MCMC chain.

References

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. doi: https://doi.org/10.1201/b16018

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

Spiegelhalter, D., Best, N., Carlin, B., Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(4), 583–639. doi: https://doi.org/10.1111/1467-9868.00353

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. J. Mach. Learn. Res. 11, 3571–3594.

See Also

`summary.plm` for summaries, `predict.plm` for prediction. It is also useful to look at `spread_draws` and `plot.plm` to help visualize the full posterior distributions.

Examples

```
data(spanga)
set.seed(1)
plm.fit <- plm(formula=Q~W,data=spanga,num_cores=2)
summary(plm.fit)
```

plm0 *Power-law model with a constant variance*

Description

plm0 is used to fit a discharge rating curve for paired measurements of stage and discharge using a power-law model with a constant variance as described in Hrafnkelsson et al. (2022). See "Details" for a more elaborate description of the model.

Usage

```
plm0(
  formula,
  data,
  c_param = NULL,
  h_max = NULL,
  parallel = TRUE,
  num_cores = NULL,
  forcepoint = rep(FALSE, nrow(data)),
  verbose = TRUE
)
```

Arguments

`formula` An object of class "formula", with discharge column name as response and stage column name as a covariate, i.e. of the form `y~x` where `y` is discharge in m^3/s and `x` is stage in m (it is very important that the data is in the correct units).

`data` A data.frame containing the variables specified in formula.

`c_param` The largest stage value for which there is zero discharge. If NULL, it is treated as unknown in the model and inferred from the data.

`h_max` The maximum stage to which the rating curve should extrapolate to. If NULL, the maximum stage value in the data is selected as an upper bound.

`parallel` A logical value indicating whether to run the MCMC in parallel or not. Defaults to TRUE.

`num_cores` An integer between 1 and 4 (number of MCMC chains) indicating how many cores to use. Only used if parallel=TRUE. If NULL, the number of cores available on the device is detected automatically.

`forcepoint` A logical vector of the same length as the number of rows in data. If an element at index i is TRUE it indicates that the rating curve should be forced through the i-th measurement. Use with care, as this will strongly influence the resulting rating curve.

`verbose` A logical value indicating whether to print progress and diagnostic information. If ‘TRUE‘, the function will print messages as it runs. If ‘FALSE‘, the function will run silently. Default is ‘TRUE‘.

Details

The power-law model, which is commonly used in hydraulic practice, is of the form

$$Q = a(h - c)^b$$

where Q is discharge, h is stage and a, b and c are unknown constants.

The power-law model is here inferred by using a Bayesian hierarchical model. The model is on a logarithmic scale

$$\log(Q_i) = \log(a) + b\log(h_i - c) + \varepsilon, i = 1, ..., n$$

where ε follows a normal distribution with mean zero and variance σ_ε^2, independent of stage. An efficient posterior simulation is achieved by sampling from the joint posterior density of the hyperparameters of the model, and then sampling from the density of the latent parameters conditional on the hyperparameters.

Bayesian inference is based on the posterior density and summary statistics such as the posterior mean and 95% posterior intervals are based on the posterior density. Analytical formulas for these summary statistics are intractable in most cases and thus they are computed by generating samples from the posterior density using a Markov chain Monte Carlo simulation.

Value

plm0 returns an object of class "plm0". An object of class "plm0" is a list containing the following components:

`rating_curve` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior predictive distribution of the rating curve.

`rating_curve_mean` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of the mean of the rating curve.

`param_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior distribution of latent- and hyperparameters. Additionally contains columns with r_hat and the effective number of samples for each parameter as defined in Gelman et al. (2013).

`posterior_log_likelihood_summary` A data frame with 2.5%, 50% and 97.5% percentiles of the posterior log-likelihood values.

`rating_curve_posterior` A matrix containing the full thinned posterior samples of the posterior predictive distribution of the rating curve (excluding burn-in).

`rating_curve_mean_posterior` A matrix containing the full thinned posterior samples of the posterior distribution of the mean of the rating curve (excluding burn-in).

`a_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of a.

`b_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of b.

`c_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of c.

`sigma_eps_posterior` A numeric vector containing the full thinned posterior samples of the posterior distribution of σ_ε.

`posterior_log_likelihood` A numeric vector containing the full thinned posterior log-likelihood values, excluding burn-in samples.

`D_hat` A statistic defined as -2 times the log-likelihood evaluated at the median value of the parameters.

`effective_num_param_DIC` The effective number of parameters, which is calculated as median(-2*posterior_log_likelihood) minus D_hat.

`DIC` The Deviance Information Criterion for the model, calculated as D_hat plus 2*effective_num_parameters_DIC.

`lppd` The log pointwise predictive density of the observed data under the model.

`WAIC` The Widely Applicable Information Criterion for the model, defined as -2*(lppd - effective_num_param_WAIC).

`WAIC_i` The pointwise WAIC values, where WAIC := sum(WAIC_i).

`effective_num_param_WAIC` The effective number of parameters, which is calculated by summing up the posterior variance of the log predictive density for each data point.

`autocorrelation` A data frame with the autocorrelation of each parameter for different lags.

`acceptance_rate` The proportion of accepted samples in the thinned MCMC chain (excluding burn-in).

`formula` An object of type "formula" provided by the user.

`data` The data provided by the user, ordered by stage.

`run_info` The information about the input arguments and the specific parameters used in the MCMC chain.

References

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. doi: https://doi.org/10.1201/b16018

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

Spiegelhalter, D., Best, N., Carlin, B., Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(4), 583–639. doi: https://doi.org/10.1111/1467-9868.00353

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. J. Mach. Learn. Res. 11, 3571–3594.

See Also

`summary.plm0` for summaries, `predict.plm0` for prediction. It is also useful to look at `spread_draws` and `plot.plm0` to help visualize the full posterior distributions.

Examples

```
data(skogsliden)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=skogsliden,num_cores=2)
summary(plm0.fit)
```

---

plot.plm0 *Plot method for discharge rating curves*

Description

Visualize discharge rating curve model objects

Usage

```
## S3 method for class 'plm0'
plot(
  x,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)

## S3 method for class 'plm'
plot(
  x,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)

## S3 method for class 'gplm0'
plot(
  x,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
```

```
  xlim = NULL,
  ylim = NULL,
  ...
)

## S3 method for class 'gplm'
plot(
  x,
  type = "rating_curve",
  param = NULL,
  transformed = FALSE,
  title = NULL,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

`x` An object of class "plm0", "plm", "gplm0" or "gplm".

`type` A character denoting what type of plot should be drawn. Defaults to "rating_curve". Possible types are:

- `rating_curve` Plots the rating curve.
- `rating_curve_mean` Plots the posterior mean of the rating curve.
- `f` Plots the power-law exponent.
- `beta` Plots the random effect in the power-law exponent.
- `sigma_eps` Plots the standard deviation on the data level.
- `residuals` Plots the log residuals.
- `trace` Plots trace plots of parameters given in param.
- `histogram` Plots histograms of parameters given in param.
- `panel` Plots a 2x2 panel of plots: "rating curve", "residuals", "f" and "sigma_eps".

`param` A character vector with the parameters to plot. Defaults to NULL and is only used if type is "trace" or "histogram". Allowed values are the parameters given in the model summary of x as well as "hyperparameters" or "latent_parameters" for specific groups of parameters.

`transformed` A logical value indicating whether the quantity should be plotted on a transformed scale used during the Bayesian inference. Defaults to FALSE.

`title` A character denoting the title of the plot.

`xlim` A numeric vector of length 2, denoting the limits on the x axis of the plot. Applicable for types "rating_curve", "rating_curve_mean", "f", "beta", "sigma_eps", "residuals".

`ylim` A numeric vector of length 2, denoting the limits on the y axis of the plot. Applicable for types "rating_curve", "rating_curve_mean", "f", "beta", "sigma_eps", "residuals".

`...` Not used in this function

Value

No return value, called for side effects.

Functions

- `plot(plm0)`: Plot method for plm0
- `plot(plm)`: Plot method for plm
- `plot(gplm0)`: Plot method for gplm0
- `plot(gplm)`: Plot method for gplm

See Also

`plm0`, `plm`, `gplm0` and `gplm` for fitting a discharge rating curve and `summary.plm0`, `summary.plm`, `summary.gplm0` and `summary.gplm` for summaries. It is also useful to look at `spread_draws` and `gather_draws` to work directly with the MCMC samples.

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,num_cores=2)

plot(plm0.fit)
plot(plm0.fit,transformed=TRUE)
plot(plm0.fit,type='histogram',param='c')
plot(plm0.fit,type='histogram',param='c',transformed=TRUE)
plot(plm0.fit,type='histogram',param='hyperparameters')
plot(plm0.fit,type='histogram',param='latent_parameters')
plot(plm0.fit,type='residuals')
plot(plm0.fit,type='f')
plot(plm0.fit,type='sigma_eps')
```

`plot.tournament` *Plot method for a discharge rating curve tournament*

Description

Compare the four models from the tournament object in multiple ways

Usage

```
## S3 method for class 'tournament'
plot(x, type = "tournament_results", transformed = FALSE, ...)
```

Arguments

- `x` — An object of class "tournament"
- `type` — A character denoting what type of plot should be drawn. Possible types are:
  - `boxplot` Creates a boxplot of the posterior log-likelihood values, on the deviance scale.
  - `rating_curve` Plots the rating curve.
  - `rating_curve_mean` Plots the posterior mean of the rating curve.
  - `f` Plots the power-law exponent.
  - `sigma_eps` Plots the standard deviation on the data level.
  - `residuals` Plots the log residuals.
  - `tournament_results` Plots a diagram showing the tournament results.
- `transformed` — A logical value indicating whether the quantity should be plotted on a transformed scale used during the Bayesian inference. Defaults to FALSE.
- `...` — Not used in this function

Value

No return value, called for side effects

See Also

`tournament` to run a discharge rating curve tournament and `summary.tournament` for summaries.

Examples

```
data(krokfors)
set.seed(1)
t_obj <- tournament(formula = Q ~ W, data = krokfors, num_cores = 2)
plot(t_obj)
plot(t_obj, transformed = TRUE)
plot(t_obj, type = 'boxplot')
plot(t_obj, type = 'f')
plot(t_obj, type = 'sigma_eps')
plot(t_obj, type = 'residuals')
plot(t_obj, type = 'tournament_results')
```

`predict.plm0` *Predict method for discharge rating curves*

Description

Predict the discharge for given stage values based on a discharge rating curve model object.

Usage

```
## S3 method for class 'plm0'
predict(object, newdata = NULL, wide = FALSE, ...)

## S3 method for class 'plm'
predict(object, newdata = NULL, wide = FALSE, ...)

## S3 method for class 'gplm0'
predict(object, newdata = NULL, wide = FALSE, ...)

## S3 method for class 'gplm'
predict(object, newdata = NULL, wide = FALSE, ...)
```

Arguments

`object`	An object of class "plm0", "plm", "gplm0" or "gplm".
`newdata`	A numeric vector of stage values for which to predict. If omitted, the stage values in the data are used.
`wide`	A logical value denoting whether to produce a wide prediction output. If TRUE, then the output is a table with median prediction values for an equally spaced grid of stages with 1 cm increments, each row containing predictions in a decimeter range of stages.
`...`	Not used in this function

Value

An object of class "data.frame" with four columns:

`h` The stage.

`lower` The 2.5% posterior predictive quantile.

`median` The 50% posterior predictive quantile.

`upper` The 97.5% posterior predictive quantile.

If wide=TRUE, a matrix as described above (see wide parameter) is returned.

Functions

- `predict(plm0)`: Predict method for plm0
- `predict(plm)`: Predict method for plm
- `predict(gplm0)`: Predict method for gplm0
- `predict(gplm)`: Predict method for gplm

See Also

`plm0`, `plm`, `gplm0` and `gplm` for fitting a discharge rating curve and `summary.plm0`, `summary.plm`, `summary.gplm0` and `summary.gplm` for summaries. It is also useful to look at `plot.plm0`, `plot.plm`, `plot.gplm0` and `plot.gplm` to help visualize all aspects of the fitted discharge rating curve. Additionally, `spread_draws` and `spread_draws` help working directly with the MCMC samples.

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,h_max=10,num_cores=2)
#predict rating curve on a equally 10 cm spaced grid from 9 to 10 meters
predict(plm0.fit,newdata=seq(9,10,by=0.1))
```

---

print.plm0 *Print method for discharge rating curves*

Description

Print a discharge rating curve model object

Usage

```
## S3 method for class 'plm0'
print(x, ...)

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

## S3 method for class 'gplm0'
print(x, ...)

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

Arguments

x	an object of class "plm0", "plm", "gplm0" or "gplm".
...	Not used in this function

Functions

- `print(plm0)`: Print method for plm0
- `print(plm)`: Print method for plm
- `print(gplm0)`: Print method for gplm0
- `print(gplm)`: Print method for gplm

See Also

`plm0`, `plm`, `gplm0`, `gplm` for fitting a discharge rating curve and `summary.plm0`, `summary.plm`, `summary.gplm0` and `summary.gplm` for summaries. It is also useful to look at `plot.plm0`, `plot.plm`, `plot.gplm0` and `plot.gplm` to help visualize all aspects of the fitted discharge rating curve. Additionally, `spread_draws` and `spread_draws` help working directly with the MCMC samples.

print.tournament *Print method for discharge rating curve tournament*

Description

Print the results of a tournament of discharge rating curve model comparisons

Usage

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

Arguments

`x`	An object of class "tournament"
`...`	Not used in this function

See Also

`tournament` to run a discharge rating curve tournament, `summary.tournament` for summaries and `plot.tournament` for visualizing the mode comparison.

skjalf *Skjalfandafljot gauging station in Iceland*

Description

Data on discharge and stage from Skjalfandafljot gauging station in Iceland

Usage

```
skjalf
```

Format

A data frame with columns:

`W` Measurements of water stage in meters

`Q` Measurements of water discharge in cubic meters per second

Source

Icelandic Meteorological Office, Landsvirkjun - the National Power Company of Iceland, and the Icelandic Road and Coastal Administration.

skogsliden — *Skogsliden gauging station in Sweden*

Description

Data on discharge and stage from Skogsliden gauging station in Sweden

Usage

```
skogsliden
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Swedish Meteorological and Hydrological Institute.

spanga — *Spanga gauging station in Sweden*

Description

Data on discharge and stage from Spanga gauging station in Sweden.

Usage

```
spanga
```

Format

A data frame with columns:

- `W` Measurements of water stage in meters
- `Q` Measurements of water discharge in cubic meters per second

Source

Swedish Meteorological and Hydrological Institute.

`spread_draws` *Spread MCMC chain draws to data.frame on a wide format*

Description

Useful to convert MCMC chain draws of particular parameters or output from the model object to a wide format for further data wrangling

Usage

```
spread_draws(mod, ..., transformed = FALSE)
```

Arguments

`mod`	An object of class "plm0", "plm", "gplm0" or "gplm".
`...`	Any number of character vectors containing valid names of parameters in the model or "rating_curve" and "rating_curve_mean". Also accepts "latent_parameters" and "hyperparameters".
`transformed`	A boolean value determining whether the output is to be represented on the transformed scale used for sampling in the MCMC chain or the original scale. Defaults to FALSE.

Value

A data frame with columns:

`chain` The chain number.

`iter` The iteration number.

`param` The parameter name.

`value` The parameter value.

References

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

See Also

`plm0`, `plm`, `gplm0`, `gplm` for further information on parameters

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,num_cores=2)
hyp_samples <- spread_draws(plm0.fit,'hyperparameters')
head(hyp_samples)
rating_curve_samples <- spread_draws(plm0.fit,'rating_curve','rating_curve_mean')
head(rating_curve_samples)
```

---

summary.plm0 *Summary method for discharge rating curves*

Description

Summarize a discharge rating curve model object

Usage

```
## S3 method for class 'plm0'
summary(object, ...)

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

## S3 method for class 'gplm0'
summary(object, ...)

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

Arguments

`object`	an object of class "plm0", "plm", "gplm0" or "gplm".
`...`	Not used in this function

Functions

- `summary(plm0)`: Summary method for plm0
- `summary(plm)`: Summary method for plm
- `summary(gplm0)`: Summary method for gplm0
- `summary(gplm)`: Summary method for gplm

See Also

plm0, plm, gplm0 and gplm for fitting a discharge rating curve. It is also useful to look at plot.plm0, plot.plm, plot.gplm0 and plot.gplm to help visualize all aspects of the fitted discharge rating curve. Additionally, spread_draws and spread_draws help working directly with the MCMC samples.

Examples

```
data(krokfors)
set.seed(1)
plm0.fit <- plm0(formula=Q~W,data=krokfors,num_cores=2)
summary(plm0.fit)
```

`summary.tournament` *Summary method for a discharge rating curve tournament*

Description

Print the summary of a tournament of model comparisons. This function allows for an efficient and fast re-run of the tournament with different methods or winning criteria.

Usage

```
## S3 method for class 'tournament'
summary(object, method = NULL, winning_criteria = NULL, ...)
```

Arguments

- `object` An object of class "tournament"
- `method` Optional; a string specifying the method to use for the summary. If NULL, uses the method from the original tournament. Options are "WAIC", "DIC", or "PMP".
- `winning_criteria` Optional; specifies new winning criteria for the summary. If NULL, uses the criteria from the original tournament. See Details in tournament for proper formatting.
- `...` Not used in this function

Details

If either `method` or `winning_criteria` is provided, the function re-runs the tournament with the new parameters using the fitted models.

Value

Prints the summary to the console.

See Also

tournament to run a discharge rating curve tournament and plot.tournament for visualizing the model comparison

Examples

```
data(krokfors)
set.seed(1)
t_obj <- tournament(Q ~ W, krokfors, num_cores = 2)
summary(t_obj)

# Re-run summary with different method
summary(t_obj, method = "DIC")

# Re-run summary with different winning criteria
summary(t_obj, winning_criteria = "Delta_WAIC > 3")
```

---

tournament — *Tournament - Model comparison*

Description

tournament compares four rating curve models of different complexities and determines the model that provides the best fit of the data at hand.

Usage

```
tournament(
  formula = NULL,
  data = NULL,
  model_list = NULL,
  method = "WAIC",
  winning_criteria = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

formula	An object of class "formula", with discharge column name as response and stage column name as a covariate.
data	A data.frame containing the variables specified in formula.
model_list	A list of exactly four model objects of types "plm0","plm","gplm0" and "gplm" to be used in the tournament. Note that all of the model objects are required to be run with the same data and same c_param.

method
: A string specifying the method used to estimate the predictive performance of the models. The allowed methods are "WAIC", "DIC" and "PMP".

winning_criteria
: Specifies the criteria for model selection. For "WAIC", it can be a numeric value or a string expression. For "DIC", it must be a numeric value. For "PMP", it must be a numeric value between 0 and 1. See Details section.

verbose
: A logical value indicating whether to print progress and diagnostic information. If ‘TRUE‘, the function will print messages as it runs. If ‘FALSE‘, the function will run silently. Default is ‘TRUE‘.

...
: Optional arguments passed to the model functions.

Details

Tournament is a model comparison method that uses WAIC (default method) to estimate the expected prediction error of the four models and select the most appropriate model given the data. The first round of model comparisons sets up model types, "gplm" vs. "gplm0" and "plm" vs. "plm0". The two comparisons are conducted such that if the WAIC of the more complex model ("gplm" and "plm", respectively) is smaller than the WAIC of the simpler models ("gplm0" and "plm0", respectively) by an input argument called the `winning_criteria` (default value = 2), then it is chosen as the more appropriate model. If not, the simpler model is chosen. The more appropriate models move on to the second round and are compared in the same way. The winner of the second round is chosen as the overall tournament winner and deemed the most appropriate model given the data.

The default method "WAIC", or the Widely Applicable Information Criterion (see Watanabe (2010)), is used to estimate the predictive performance of the models. This method is a fully Bayesian method that uses the full set of posterior draws to estimate of the expected log pointwise predictive density.

Method "DIC", or Deviance Information Criterion (see Spiegelhalter (2002)), is similar to the "WAIC" but instead of using the full set of posterior draws to compute the estimate of the expected log pointwise predictive density, it uses a point estimate of the posterior distribution.

Method "PMP" uses the posterior model probabilities, calculated with Bayes factor (see Jeffreys (1961) and Kass and Raftery (1995)), to compare the models, where all the models are assumed a priori to be equally likely. This method is not chosen as the default method because the Bayes factor calculations can be quite unstable.

When method "WAIC" is used, the `winning_criteria` can be either a numeric value or a string expression. If numeric, it sets the threshold which the more complex model must exceed to be declared the more appropriate model. If a string, it must be a valid R expression using Delta_WAIC and/or SE_Delta_WAIC (e.g., "Delta_WAIC > 2 & Delta_WAIC - SE_Delta_WAIC > 0"). For method "DIC", `winning_criteria` must be a numeric value. For method "PMP", the winning criteria should be a numeric value between 0 and 1 (default value = 0.75). This sets the threshold value for which the posterior probability of the more complex model, given the data, in each model comparison must exceed to be declared the more appropriate model. In all cases, the default values are selected to give the less complex models a slight advantage, which should give more or less consistent results when applying the tournament to real world data.

Value

An object of type "tournament" with the following elements:

`contestants` The model objects of types "plm0", "plm", "gplm0" and "gplm" being compared.

`winner` The model object of the tournament winner.

`info` The specifics about the tournament; the overall winner; the method used; and the winning criteria.

`summary` A data frame with information on results of the different comparisons in the power-law tournament. The contents of this data frame depend on the method used:

- For all methods:
  - round: The tournament round
  - comparison: The comparison number
  - complexity: Indicates whether a model is the "more" or "less" complex model in a comparison
  - model: The model type
  - winner: Logical value indicating if the model was selected in the corresponding comparison
- Additional columns for method "WAIC":
  - lppd: Log pointwise predictive density
  - eff_num_param: Effective number of parameters (WAIC)
  - WAIC: Widely Applicable Information Criterion
  - SE_WAIC: Standard error of WAIC
  - Delta_WAIC: Difference in WAIC
  - SE_Delta_WAIC: Standard error of the difference in WAIC
- Additional columns for method "DIC":
  - D_hat: Minus two times the log-likelihood evaluated at the median of the posterior samples
  - eff_num_param: Effective number of parameters (DIC)
  - DIC: Deviance Information Criterion
  - Delta_DIC: Difference in DIC
- Additional columns for method "PMP":
  - log_marg_lik: Logarithm of the marginal likelihood estimated, computed with the harmonic-mean estimator
  - PMP: Posterior model probability computed with Bayes factor

References

Hrafnkelsson, B., Sigurdarson, H., Rögnvaldsson, S., Jansson, A. Ö., Vias, R. D., and Gardarsson, S. M. (2022). Generalization of the power-law rating curve using hydrodynamic theory and Bayesian hierarchical modeling, Environmetrics, 33(2):e2711. doi: https://doi.org/10.1002/env.2711

Jeffreys, H. (1961). Theory of Probability, Third Edition. Oxford University Press.

Kass, R., and A. Raftery, A. (1995). Bayes Factors. Journal of the American Statistical Association, 90, 773-795. doi: https://doi.org/10.1080/01621459.1995.10476572

Spiegelhalter, D., Best, N., Carlin, B., Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(4), 583–639. doi: https://doi.org/10.1111/1467-9868.00353

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. J. Mach. Learn. Res. 11, 3571–3594.

See Also

plm0 plm, gplm0,gplm summary.tournament and plot.tournament

Examples

```
data(krokfors)
set.seed(1)
t_obj <- tournament(formula = Q ~ W, data = krokfors, num_cores = 2)
t_obj
summary(t_obj)

# Using different methods and winning criteria
t_obj_dic <- tournament(Q ~ W,
                        krokfors,
                        num_cores = 2,
                        method = "DIC",
                        winning_criteria = 3)
t_obj_pmp <- tournament(Q ~ W,
                        krokfors,
                        num_cores = 2,
                        method = "PMP",
                        winning_criteria = 0.8)
t_obj_waic_expr <- tournament(Q ~ W,
                              krokfors,
                              num_cores = 2,
                      winning_criteria = "Delta_WAIC > 2 & Delta_WAIC - SE_Delta_WAIC > 0")
```

Index