Package: bmstdr (via r-universe)

October 29, 2026

Type Package

Title Bayesian Modeling of Spatio-Temporal Data with R

Version 0.8.2

URL https://www.sujitsahu.com

Description Fits, validates and compares a number of Bayesian models for spatial and space time point referenced and areal unit data. Model fitting is done using several packages: 'rstan', 'INLA', 'spBayes', 'spTimer', 'spTDyn', 'CARBayes' and 'CARBayesST'. Model comparison is performed using the DIC and WAIC, and K-fold cross-validation where the user is free to select their own subset of data rows for validation. Sahu (2022) <doi:10.1201/9780429318443> describes the methods in detail.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports methods, spTimer, ggplot2, rstan, rstantools, spBayes, CARBayes, CARBayesST, spTDyn, MCMCpack, Rdpack, mnormt, inlabru, ggpubr

RdMacros Rdpack

Depends R (>= 3.4.0), Rcpp (>= 0.12.0)

Suggests INLA (>= 22.5.7), coda, extraDistr, maps, xtable, MASS, loo, cowplot, ggrepel, sf, knitr, rmarkdown, testthat, ggspatial, interp, tidyr, RColorBrewer, magick, markdown

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

VignetteBuilder knitr

SystemRequirements GNU make

Additional_repositories https://inla.r-inla-download.org/R/stable

BugReports https://github.com/sujit-sahu/bmstdr/issues

NeedsCompilation yes

Author Sujit K. Sahu [aut, cre]
(<https://orcid.org/0000-0003-2315-3598>), Duncan P. Lee [aut],
K. Shuvo Bakar [aut]

Maintainer Sujit K. Sahu <S.K.Sahu@soton.ac.uk>

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2025-03-31 17:30:06 UTC

RemoteUrl https://github.com/cran/bmstdr

RemoteRef HEAD

RemoteSha 1b3328e14e3a85b4880d96b7d2523ac35f095e58

Contents

argo_floats_atlantic_2003 . . . 3
Bcartime . . . 4
BCauchy . . . 16
Bmchoice . . . 17
Bmoving_sptime . . . 18
bmstdr_variogram . . . 22
Bnormal . . . 23
Bspatial . . . 25
Bsptime . . . 31
calculate_dic . . . 39
calculate_validation_statistics . . . 40
calculate_waic . . . 41
coef.bmstdr . . . 41
colpalette . . . 42
engdeaths . . . 42
engtotals . . . 44
fig11.13.plot . . . 45
fitted.bmstdr . . . 46
fnc.delete.map.XYZ . . . 46
get_parameter_estimates . . . 47
get_validation_summaries . . . 47
getvalidrows . . . 48
gridnyspatial . . . 49
gridnysptime . . . 50
hitandfalsealarm . . . 52
is.bmstdr . . . 52
materncov . . . 53
maternfun . . . 53
nyspatial . . . 54
nysptime . . . 55

obs_v_pred_plot 56
phichoice_sp 57
phichoicep 58
plot.bmstdr 59
print.bmstdr 60
resid.bmstdr 61
residuals.bmstdr 61
summary.bmstdr 62
terms.bmstdr 62
Weng 63
ydata 63

Index **65**

`argo_floats_atlantic_2003`

Temperature and salinity data from Argo floats in the North Atlantic Ocean at three layers of depth: surface (less than 50 meters), mid-layer (between 475-525 meters) and deep (975 to 1025 meters) during 2003.

Description

Temperature and salinity data from Argo floats in the North Atlantic Ocean at three layers of depth: surface (less than 50 meters), mid-layer (between 475-525 meters) and deep (975 to 1025 meters) during 2003.

Usage

```
argo_floats_atlantic_2003
```

Format

An object of class `data.frame` with 6978 rows and 11 columns.

Source

Sahu and Challenor (2008) @format A data frame with 6978 rows and 11 columns:

lon Longitude of the argo float

lat Latitude of the argo float

time Cumulative day of the year in 2003

day Day within each month in 2003

month Month in 2003

temp Temperature recorded by the Argo float in degree Celsius

sali Salinity in practical salinity units

xlon Centered and scaled values of longitude at each depth

xlat Centered and scaled values of latitude at each depth

xinter Centered and scaled values of longitude times latitude at each depth

References

Sahu SK, Challenor P (2008). "A space-time model for joint modeling of ocean temperature and salinity levels as measured by Argo floats." *Environmetrics*, **19**, 509-528.

Examples

```
head(argo_floats_atlantic_2003)
# Data for the surface layer
surface <- argo_floats_atlantic_2003[argo_floats_atlantic_2003$depth==1, ]
# Data for the mid-layer
midlayer <- argo_floats_atlantic_2003[argo_floats_atlantic_2003$depth==2, ]
# Data at the deep ocean
deep <- argo_floats_atlantic_2003[argo_floats_atlantic_2003$depth==3, ]
```

`Bcartime` *Bayesian regression model fitting for areal and areal spatio-temporal data. Calculates parameter estimates, validation statistics, and estimated values of several Bayesian model choice criteria.*

Description

Bayesian regression model fitting for areal and areal spatio-temporal data. Calculates parameter estimates, validation statistics, and estimated values of several Bayesian model choice criteria.

Usage

```
Bcartime(
  formula,
  data,
  family,
  link = NULL,
  trials = NULL,
  offsetcol = NULL,
  formula.omega = NULL,
  scol = NULL,
  tcol = NULL,
  package = "CARBayes",
  model = "glm",
  AR = 1,
  W = NULL,
  adj.graph = NULL,
  residtype = "response",
  interaction = TRUE,
  Z = NULL,
```

```
  W.binary = NULL,
  changepoint = NULL,
  knots = NULL,
  validrows = NULL,
  prior.mean.delta = NULL,
  prior.mean.beta = NULL,
  prior.var.beta = NULL,
  prior.mean.gamma = NULL,
  prior.var.gamma = NULL,
  prior.sigma2 = NULL,
  prior.tau2 = c(2, 1),
  prior.delta = NULL,
  prior.var.delta = NULL,
  prior.lambda = NULL,
  prior.nu2 = c(2, 1),
  epsilon = 0,
  G = NULL,
  ind.area = NULL,
  trends = NULL,
  rho.T = NULL,
  rho.S = NULL,
  rho = NULL,
  rho.slo = NULL,
  rho.int = NULL,
  MALA = FALSE,
  N = 2000,
  burn.in = 1000,
  thin = 10,
  rseed = 44,
  Nchains = 4,
  verbose = TRUE,
  plotit = TRUE
)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the regression model to be fitted.
data	The data frame for which the model formula is to be fitted. The data frame should be in long format having one row for each location and time combination. The data frame must be ordered by time within each site, and should optionally have a column, named s.index, providing the site indices. Thus the data, with n sites and T times within each site, should be organized in the order: (s1, t1), (s1, t2), ... (s1, T), ... (sn, t1), ... (sn, T). The data frame should also contain two columns giving the coordinates of the locations for spatio temporal model fitting.
family	One of either "gaussian", "binomial","poisson" or "zip", which respectively specify a Gaussian, binomial likelihood model with the logistic link function, a Pois-

son likelihood model with a log link function, or a zero-inflated Poisson model with a log link function.

`link` The link function to use for INLA based model fitting. This is ignored for the CARBayes and CARBayesST models.

`trials` Only used if family="binomial". Either the name (character) or number of the column in the supplied data frame containing the total number of trials The program will try to access data[, trials] to get the binomial denominators.

`offsetcol` Only used in INLA based modeling. The column name or number in the data frame that should be used as the offset.

`formula.omega` A one-sided formula object with no response variable (left side of the "~") needed, specifying the covariates in the logistic regression model for modelling the probability of an observation being a structural zero. Each covariate (or an offset) needs to be a vector of length K*1. Only required for zero-inflated Poisson models.

`scol` Either the name (character) or number of the column in the supplied data frame identifying the spatial units. The program will try to access data[, scol] to identify the spatial units. If this is omitted, no spatial modeling will be performed.

`tcol` Like the `scol` argument for the time identifier. Either the name (character) or number of the column in the supplied data frame identifying the time indices. The program will try to access data[, tcol] to identify the time points. If this is omitted, no temporal modeling will be performed.

`package` Which package is to be used in model fitting? Currently available packages are:

- "inla": INLA model fitting for areal data. See Gómez-Rubio (2020).
- "CARBayes": All possible models in this package can be fitted. See Lee (2021).
- "CARBayesST": All possible models in this package can be fitted. See Lee et al. (2018). Further details and more examples are provided in Chapters 10 and 11 of the book Sahu (2022).

`model` The specific spatio temporal model to be fitted. If the package is "INLA" then the model argument should be a vector with two elements giving the spatial model as the first component. The alternatives for the spatial model are: "bym", "bym2", "besag", "besag2", "besagproper", "besagproper2", "iid" and "none". The temporal model as the second second component can be one of "iid", "rw1", "rw2", ar1" or "none". In case the model component is "none" then no spatial/temporal random effects will be fitted. No temporal random effects will be fitted in case `model` is supplied as a scalar, i.e. not a vector of two values.

`AR` The order of the autoregressive time series process that must be either 1 or 2.

`W` A non-negative K by K neighborhood matrix (where K is the number of spatial units). Typically a binary specification is used, where the jkth element equals one if areas (j, k) are spatially close (e.g. share a common border) and is zero otherwise. The matrix can be non-binary, but each row must contain at least one non-zero entry. This argument may not need to be specified if `adj.graph` is specified instead.

`adj.graph` Adjacency graph which may be specified instead of the adjacency matrix matrix. This argument is used if `W` has not been supplied. The argument `W` is used in case both `W` and `adj.graph` are supplied.

`residtype` Residual type, either "response" or "pearson", in GLM fitting with the packages CARBayes and CARBayesST. Default is "response" type observed minus fitted. The other option "pearson" is for Pearson residuals in GLM. For INLA based model fitting only the default response residuals are calculated.

`interaction` TRUE or FALSE indicating whether the spatio-temporal interaction random effects should be included. Defaults to TRUE unless family="gaussian" in which case interactions are not allowed.

`Z` A list, where each element is a K by K matrix of non-negative dissimilarity metrics.

`W.binary` Logical, should the estimated neighbourhood matrix have only binary (0,1) values.

`changepoint` A scalar indicating the position of the change point should one of the change point trend functions be included in the trends vector, i.e. if "CP" or "CT" is specified.

`knots` A scalar indicating the number of knots to use should one of the monotonic cubic splines trend functions be included in the trends vector, i.e. if "MD" or "MI" is specified.

`validrows` A vector providing the rows of the data frame which should be used for validation. The default NULL value instructs that validation will not be performed.

`prior.mean.delta`
A vector of prior means for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector of zeros.

`prior.mean.beta`
A vector of prior means for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.beta` A vector of prior variances for the regression parameters beta (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.mean.gamma`
A vector of prior means for the temporal trend parameters (Gaussian priors are assumed). Defaults to a vector of zeros.

`prior.var.gamma`
A vector of prior variances for the temporal trend parameters (Gaussian priors are assumed). Defaults to a vector with values 100,000.

`prior.sigma2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for sigma2. Defaults to c(1, 0.01).

`prior.tau2` The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for tau2. Defaults to c(1, 0.01).

`prior.delta` The prior maximum for the cluster smoothing parameter delta. Defaults to 10.

`prior.var.delta`
A vector of prior variances for the regression parameters delta (Gaussian priors are assumed) for the zero probability logistic regression component of the model. Defaults to a vector with values 100000.

prior.lambda — A vector of prior samples sizes for the Dirichlet prior controlling the probabilities that each trend function is chosen. The vector should be the same length as the trends vector and defaults to a vector of ones.

prior.nu2 — The prior shape and scale in the form of c(shape, scale) for an Inverse-Gamma(shape, scale) prior for nu2. Defaults to c(1, 0.01) and only used if family="Gaussian".

epsilon — Diagonal ridge parameter to add to the random effects prior precision matrix, only required when rho = 1, and the prior precision is improper. Defaults to 0. Only used for adaptive model fitting in CARBayesST.

G — The maximum number of distinct intercept terms (groups) to allow in the localised model.

ind.area — A vector of integers the same length as the number of data points (individuals) giving which spatial unit (nunmbered from 1 to K to align with the rows of the W matrix) each individual belongs to.

trends — A vector containing the temporal trend functions to include in the model, which include: constant ("Constant""); linear decreasing ("LD"); linear increasing ("LI"); Known change point, where the trend can increase towards the change point before subsequently decreasing ("CP"); or decrease towards the change point before subsequently increasing ("CT"); and monotonic cubic splines which are decreasing ("MD") or increasing ("MI"). At least two trends have to be selected, with the constant trend always included. To avoid identifiability problems only one of "LI" or "MI" can be included at a given time (similarily for "LD" and "MD").

rho.T — The value in the interval [0, 1] that the temporal dependence parameter rho.T is fixed at if it should not be estimated. If this argument is NULL then rho.T is estimated in the model.

rho.S — The value in the interval [0, 1] that the spatial dependence parameter rho.S is fixed at if it should not be estimated. If this argument is NULL then rho.S is estimated in the model.

rho — The value in the interval [0, 1] that the spatial dependence parameter rho is fixed at if it should not be estimated. If this argument is NULL then rho is estimated in the model. Setting rho=1, reduces the random effects prior to the intrinsic CAR model but does require epsilon>0.

rho.slo — The value in the interval [0, 1] that the spatial dependence parameter for the slope of the linear time trend, rho.slo, is fixed at if it should not be estimated. If this argument is NULL then rho.slo is estimated in the model.

rho.int — The value in the interval [0, 1] that the spatial dependence parameter for the intercept of the linear time trend, rho.int, is fixed at if it should not be estimated. If this argument is NULL then rho.int is estimated in the model.

MALA — Logical, should the function use Metropolis adjusted Langevin algorithm (MALA) updates (TRUE) or simple random walk (FALSE, default) updates for the regression parameters and random effects.

N — MCMC sample size.

burn.in — How many initial iterations to discard. Only relevant for MCMC based model fitting, i.e., when package is spBayes or Stan.

thin	The level of thinning to apply to the MCMC samples to reduce their temporal autocorrelation. Defaults to 1 (no thinning).
rseed	Random number seed that controls the starting point for the random number stream. A set value is required to help reproduce the results.
Nchains	The number of parallel Markov chains to be used in the Metropolis coupled Markov chain Monte Carlo (MCMCMC) simulations. Defaults to 4.
verbose	Logical scalar value: whether to print various estimates and statistics.
plotit	Logical scalar value: whether to plot the predictions against the observed values.

Value

A list containing:

- params - A table of parameter estimates
- fit - The fitted model object.
- fitteds - A vector of fitted values.
- mchoice - Calculated model choice statistics if those have been requested by the input argument mchoice=T. Not all model fits will contain all the model choice statistics.
- residuals - A vector of residual values.
- stats - The four validation statistics: rmse, mae, crps and coverage. This is present only if model validation has been performed.
- yobs_preds - A data frame containing the validation rows of the model fitting data frame. The last five columns of this data frame contains the validation prediction summaries: mean, sd, median, and 95% prediction interval. This is present only if model validation has been performed.
- valpreds - A matrix containing the MCMC samples of the validation predictions. The dimension of this matrix is the number of validations times the number of retained MCMC samples. This is present only if model validation has been performed.
- validationplots - Present only if validation has been performed. Contains three validation plots with or without segment and an ordinary plot. See obs_v_pred_plot for more.
- sn - The number of areal units used in fitting.
- tn - The number of time points used in fitting.
- formula - The input formula for the regression part of the model.
- scale.transform - It is there for compatibility with Bsptime output.
- package - The name of the package used for model fitting.
- model - The name of the fitted model.
- call - The command used to call the model fitting function.
- computation.time - Computation time required to run the model fitting.

References

Gómez-Rubio V (2020). *Bayesian inference with INLA*. Boca Raton:Chapman and Hall/CRC,.

Lee D (2021). "CARBayes version 5.2.3: An R Package for Spatial Areal Unit Modelling with Conditional Autoregressive Priors." University of Glasgow. https://cran.r-project.org/package=CARBayes.

Lee D, Rushworth A, Napier G (2018). "Spatio-Temporal Areal Unit Modeling in R with Conditional Autoregressive Priors Using the CARBayesST Package." *Journal of Statistical Software*, **84**(9), 10.18637/jss.v084.i09.

Sahu SK (2022). *Bayesian Modeling of Spatio Temporal Data with R*, 1st edition. Chapman and Hall, Boca Raton. https://doi.org/10.1201/9780429318443.

Examples

```
# Set the validation row numbers
vs <- sample(nrow(engtotals), 5)
# Total number of iterations
N <- 60
# Number of burn in iterations
burn.in <- 10
# Number of thinning iterations
thin <- 1

# Set the model formula for binomial distribution based modeling
f1 <- noofhighweeks ~ jsa + log10(houseprice) + log(popdensity) + sqrt(no2)
## Independent error logistic regression
M1 <- Bcartime(formula = f1, data = engtotals, family = "binomial",
    trials = "nweek", N = N, burn.in = burn.in, thin = thin,
    verbose = TRUE)
summary(M1)
# Leroux model
M1.leroux <- Bcartime(formula = f1, data = engtotals, scol = "spaceid",
    model = "leroux", W = Weng, family = "binomial", trials = "nweek",
    N = N, burn.in = burn.in, thin = thin)
summary(M1.leroux)
# BYM model
M1.bym <- Bcartime(formula = f1, data = engtotals, scol = "spaceid",
    model = "bym", W = Weng, family = "binomial", trials = "nweek",
    N = N, burn.in = burn.in, thin = thin, verbose = FALSE)
summary(M1.bym)

# Validation for the Leroux model
M1.leroux.v <- Bcartime(formula = f1, data = engtotals, scol = "spaceid",
    model = "leroux", W = Weng, family = "binomial", trials = "nweek",
    validrows = vs, N = N, burn.in = burn.in, thin = thin, verbose = FALSE)
summary(M1.leroux.v)
```

```
## Poisson Distribution based models ###################################
# Model formula
f2 <- covid ~ offset(logEdeaths) + jsa + log10(houseprice) + log(popdensity) +
    sqrt(no2)

# Independent error Poisson regression
M2 <- Bcartime(formula = f2, data = engtotals, family = "poisson", N = N,
    burn.in = burn.in, thin = thin, verbose = FALSE)
summary(M2)
## Poisson regression with Leroux Model
M2.leroux <- Bcartime(formula = f2, data = engtotals, scol = "spaceid",
    model = "leroux", family = "poisson", W = Weng, N = N, burn.in = burn.in,
    thin = thin, verbose = FALSE)
summary(M2.leroux)
# Poisson regression with BYM Model
M2.bym <- Bcartime(formula = f2, data = engtotals, scol = "spaceid",
    model = "bym", family = "poisson", W = Weng, N = N, burn.in = burn.in,
    thin = thin)
summary(M2.bym)


## Gaussian distribution based models  ###############
f3 <- sqrt(no2) ~ jsa + log10(houseprice) + log(popdensity)

# Independent error model
M3 <- Bcartime(formula = f3, data = engtotals, family = "gaussian", N = N,
    burn.in = burn.in, thin = thin, verbose = FALSE)
summary(M3)
# Leroux model
M3.leroux <- Bcartime(formula = f3, data = engtotals, scol = "spaceid",
    model = "leroux", family = "gaussian", W = Weng, N = N, burn.in = burn.in,
    thin = thin, verbose = FALSE)
summary(M3.leroux)

## Validation
M3.leroux.v <- Bcartime(formula = f3, data = engtotals, scol = "spaceid",
    model = "leroux", family = "gaussian", W = Weng, N = N, burn.in = burn.in,
    thin = thin, validrows = vs, verbose = FALSE)
summary(M3.leroux.v)



## Spatio-temporal modeling ############################################
head(engdeaths)
dim(engdeaths)
colnames(engdeaths)
vs <- sample(nrow(engdeaths), 5)


## Binomial distribution
engdeaths$nweek <- rep(1, nrow(engdeaths))
f1 <- highdeathsmr ~ jsa + log10(houseprice) + log(popdensity)
```

```
M1st_linear <- Bcartime(formula = f1, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", trials = "nweek", W = Weng, model = "linear",
    family = "binomial", package = "CARBayesST", N = N, burn.in = burn.in,
    thin = thin, verbose = TRUE)
summary(M1st_linear)
M1st_sepspat <- Bcartime(formula = f1, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", trials = "nweek", W = Weng, model = "sepspatial",
    family = "binomial", package = "CARBayesST", N = N, burn.in = burn.in,
    thin = thin, verbose = FALSE)
summary(M1st_sepspat)
M1st_ar <- Bcartime(formula = f1, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", trials = "nweek", W = Weng, model = "ar", AR = 1,
    family = "binomial", package = "CARBayesST", N = N, burn.in = burn.in,
    thin = thin, verbose = FALSE)
summary(M1st_ar)
# Model validation
M1st_ar.v <- Bcartime(formula = f1, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", trials = "nweek", W = Weng, model = "ar", AR = 1,
    family = "binomial", package = "CARBayesST", N = N, burn.in = burn.in,
    thin = thin, validrows = vs, verbose = FALSE)
summary(M1st_ar.v)


## Spatio temporal Poisson models###################################
colnames(engdeaths)
f2 <- covid ~ offset(logEdeaths) + jsa + log10(houseprice) + log(popdensity) +
    n0

M2st_linear <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "linear", family = "poisson",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M2st_linear)
M2st_anova <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "anova", family = "poisson",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M2st_anova)
M2st_anova_nointer <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "anova", interaction = FALSE,
    family = "poisson", package = "CARBayesST", N = N, burn.in = burn.in,
    thin = thin, verbose = FALSE)
summary(M2st_anova_nointer)
M2st_sepspat <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "sepspatial", family = "poisson",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M2st_sepspat)
M2st_ar <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "ar", AR = 1, family = "poisson",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M2st_ar)
```

```
M2st_ar.v <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "ar", family = "poisson",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    validrows = vs, verbose = FALSE)
M2st_anova.v <- Bcartime(formula = f2, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "anova", family = "poisson",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    validrows = vs, verbose = FALSE)
summary(M2st_ar.v)
summary(M2st_anova.v)


## Spatio-temporal Normal models ################################
colnames(engdeaths)
f3 <- sqrt(no2) ~ jsa + log10(houseprice) + log(popdensity)

M3st_linear <- Bcartime(formula = f3, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "linear", family = "gaussian",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M3st_linear)
M3st_anova <- Bcartime(formula = f3, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "anova", family = "gaussian",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M3st_anova)
M3st_anova_nointer <- Bcartime(formula = f3, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "anova", interaction = FALSE,
    family = "gaussian", package = "CARBayesST", N = N, burn.in = burn.in,
    thin = thin, verbose = FALSE)
summary(M3st_anova_nointer)
M3st_ar <- Bcartime(formula = f3, data = engdeaths, scol = "spaceid",
    tcol = "Weeknumber", W = Weng, model = "ar", AR = 2, family = "gaussian",
    package = "CARBayesST", N = N, burn.in = burn.in, thin = thin,
    verbose = FALSE)
summary(M3st_ar)

# Execute the following examples if INLA is available
if (require(INLA)) {
    N <- 55
    burn.in <- 5
    thin <- 1
    vs <- sample(nrow(engtotals), 5)


    # Spatial Binomial GLM

    f1 <- noofhighweeks ~ jsa + log10(houseprice) + log(popdensity) +  sqrt(no2)

     M1.inla.bym <- Bcartime(data = engtotals, formula = f1,
        W = Weng, scol = "spaceid", model = c("bym"), package = "inla",
        family = "binomial", link="logit",  trials = "nweek", N = N, burn.in = burn.in,
        thin = thin)
```

```
 summary(M1.inla.bym)



## Spatial only Poisson

f2inla <- covid ~ jsa + log10(houseprice) + log(popdensity) +
    sqrt(no2)

M2.inla.bym <- Bcartime(data = engtotals, formula = f2inla,
    W = Weng, scol = "spaceid", offsetcol = "logEdeaths",
    model = c("bym"), package = "inla", link = "log",
    family = "poisson", N = N, burn.in = burn.in, thin = thin)
summary(M2.inla.bym)



## Normal models

f3 <- sqrt(no2) ~ jsa + log10(houseprice) + log(popdensity)

## Fit BYM with iid random effect
M3.inla.bym <- Bcartime(data = engtotals, formula = f3,
    W = Weng, scol = "spaceid", model = c("bym"), package = "inla",
    family = "gaussian", N = N, burn.in = burn.in, thin = thin)
summary(M3.inla.bym)

# validation
M3.inla.bym.v <- Bcartime(data = engtotals, formula = f3,
    W = Weng, scol = "spaceid", model = c("bym"), package = "inla",
    family = "gaussian",  validrows = vs, N = N, burn.in = burn.in,
    thin = thin)

summary(M3.inla.bym.v)


###### Spatio-temporal INLA models


f1 <- highdeathsmr ~ jsa + log10(houseprice) + log(popdensity)
nweek <- rep(1, nrow(engdeaths))
engdeaths$nweek <- rep(1, nrow(engdeaths))

## INLA Binomial

model <- c("bym", "ar1")
M1st_inla.bym <- Bcartime(data = engdeaths, formula = f1,
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    model = model, trials = "nweek", family = "binomial", link="logit",
    package = "inla", N = N, burn.in = burn.in, thin = thin)
summary(M1st_inla.bym)

M1st_inla_v <- Bcartime(data = engdeaths, formula = f1,
```

```
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    offsetcol = "logEdeaths", model = model, trials = "nweek",
    family = "binomial",  link="logit", package = "inla", validrow = vs,
    N = N, burn.in = burn.in, thin = thin)
summary(M1st_inla_v)


model <- c("bym", "none")
M1st_inla.bym.none <- Bcartime(data = engdeaths, formula = f1,
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    model = model, trials = "nweek", family = "binomial", link="logit",
    package = "inla", N = N, burn.in = burn.in, thin = thin)
summary(M1st_inla.bym.none)


model <- c("bym")
M1st_inla.bym.none <- Bcartime(data = engdeaths, formula = f1,
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    model = model, trials = "nweek", family = "binomial", link="logit",
    package = "inla", N = N, burn.in = burn.in, thin = thin)
summary(M1st_inla.bym.none)


## Poisson models
f2inla <- covid ~ jsa + log10(houseprice) + log(popdensity) +
    n0 + n1 + n2 + n3

model <- c("bym", "ar1")
M2stinla <- Bcartime(data = engdeaths, formula = f2inla,
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    offsetcol = "logEdeaths", model = model, link = "log",
    family = "poisson", package = "inla", N = N, burn.in = burn.in,
    thin = thin)

summary(M2stinla)

M2stinla.v <- Bcartime(data = engdeaths, formula = f2inla,
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    offsetcol = "logEdeaths", model = model, link = "log",
    family = "poisson", package = "inla", validrows = vs,
    N = N, burn.in = burn.in, thin = thin)

summary(M2stinla.v)

## Normal models

f3 <- sqrt(no2) ~ jsa + log10(houseprice) + log(popdensity)

model <- c("bym", "iid")
M3inla.bym.iid <- Bcartime(data = engdeaths, formula = f3,
    W = Weng, scol = "spaceid", tcol = "Weeknumber",
    model = model, family = "gaussian", package = "inla",
    validrows = vs, N = N, burn.in = burn.in, thin = thin)
```

```
    summary(M3inla.bym.iid)

    model <- c("bym", "ar1")
    M3inla.bym.ar1 <- Bcartime(data = engdeaths, formula = f3,
        W = Weng, scol = "spaceid", tcol = "Weeknumber",
        model = model, family = "gaussian", package = "inla",
        validrows = vs, N = N, burn.in = burn.in, thin = thin)
    summary(M3inla.bym.ar1)

}
```

BCauchy *Cauchy prior simulation example.*

Description

Cauchy prior simulation example.

Usage

```
BCauchy(
  method = "exact",
  true.theta = 1,
  n = 25,
  N = 10000,
  rseed = 44,
  tuning.sd = 1
)
```

Arguments

method	Which method or package to use. Possibilities are: • "exact": Use exact numerical integration. • "importance": Use importance sampling with the prior distribution as the importance sampling distribution. • "rejection": Use rejection sampling with the prior distribution as the importance sampling distribution. • "independence": Use the Metropolis-Hastings independence sampler with the prior distribution as the proposal distribution. • "randomwalk": Use the Metropolis-Hastings random-walk sampler with normal distribution with mean 0 and variance (tuning.sd)^2 as the increment distribution.
true.theta	True value of theta with a default value of 5.
n	Data sample size; defaults to 100.

N	is the number of Monte Carlo samples.
rseed	is the random number seed for drawing data samples.
tuning.sd	is the standard deviation of the proposal increment distribution for the random walk sampler.

Value

A list containing the estimated posterior mean, ybar (the data mean) and the values of the numerator and the denominator integrals The routine simulates n observations from N(theta, 1). Mean of the simulated data values are returned as ybar.

Examples

```
BCauchy(true.theta = 1, n=25)
BCauchy(true.theta = 5, n=100)
BCauchy(method="importance", true.theta = 1, n=25)
BCauchy(method="importance", true.theta = 1, n=25, N=20000)
BCauchy(method="rejection", true.theta = 1, n=25)
BCauchy(method="independence", true.theta = 1, n=25)
BCauchy(method="randomwalk", true.theta = 1, n=25, tuning.sd =1)
```

Bmchoice *Model choice criteria calculation for univariate normal model for both known and unknown sigma^2*

Description

Model choice criteria calculation for univariate normal model for both known and unknown sigma^2

Usage

```
Bmchoice(
  case = "Exact.sigma2.known",
  y = ydata,
  mu0 = mean(y),
  sigma2 = 22,
  kprior = 1,
  prior.M = 1,
  prior.sigma2 = c(2, 1),
  N = 10000,
  rseed = 44
)
```

Arguments

case	One of the three cases: • "Exact.sigma2.known": Use exact theoretical calculation. • "MC.sigma2.known": Use Monte Carlo methods for drawing samples from the posterior assuming known sigma2. • "MC.sigma2.unknown": Use the Gibbs sampler to generate samples from the joint posterior distribution of theta and sigma^2.
y	A vector of data values. Default is 28 ydata values from the package bmstdr
mu0	The value of the prior mean if kprior=0. Default is the data mean.
sigma2	Value of the known data variance; defaults to sample variance of the data. This is ignored in the third case when sigma2 is assumed to be unknown.
kprior	A scalar providing how many data standard deviation the prior mean is from the data mean. Default value is 0.
prior.M	Prior sample size, defaults to 10^(-4).
prior.sigma2	Shape and scale parameter value for the gamma prior on 1/sigma^2, the precision.
N	The number of samples to generate.
rseed	The random number seed. Defaults to 44 to reproduce the results in the book Sahu (2022).

Value

A list containing the exact values of pdic, dic, pdicalt, dicalt, pwaic1, waic1, pwaic2, waic2, gof, penalty and pmcc. Also prints out the posterior mean and variance. @references Sahu SK (2022). *Bayesian Modeling of Spatio Temporal Data with R*, 1st edition. Chapman and Hall, Boca Raton. https://doi.org/10.1201/9780429318443.

Examples

```
Bmchoice()
b1 <- Bmchoice(case="Exact.sigma2.known")
b2 <- Bmchoice(case="MC.sigma2.known")
d1 <- Bmchoice(case="MC.sigma2.unknown")
d2 <- Bmchoice(y=rt(100, df=8),  kprior=1, prior.M=1)
```

Bmoving_sptime *Model fitting and validation for spatio-temporal data from moving sensors in time.*

Description

Model fitting and validation for spatio-temporal data from moving sensors in time.

Usage

```
Bmoving_sptime(
  formula,
  data,
  coordtype,
  coords,
  prior.sigma2 = c(2, 1),
  prior.tau2 = c(2, 1),
  prior.phi = NULL,
  prior.phi.param = NULL,
  scale.transform = "NONE",
  ad.delta = 0.8,
  t.depth = 12,
  s.size = 0.01,
  N = 2500,
  burn.in = 1000,
  no.chains = 1,
  validrows = 10,
  predspace = FALSE,
  newdata = NULL,
  mchoice = TRUE,
  plotit = FALSE,
  rseed = 44,
  verbose = TRUE,
  knots.coords = NULL,
  g_size = 5
)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	The data frame for which the model formula is to be fitted. The data frame should be in long format having one row for each location and time combination. The data frame must be ordered by time within each site, and should optionally have a column, named s.index, providing the site indices. Thus the data, with n sites and T times within each site, should be organized in the order: (s1, t1), (s1, t2), ... (s1, T), ... (sn, t1), ... (sn, T). The data frame should also contain two columns giving the coordinates of the locations for spatio temporal model fitting.
coordtype	Type of coordinates: utm, lonlat or plain with utm (supplied in meters) as the default. Distance will be calculated in units of kilometer if this argument is either utm or lonlat. Euclidean distance will be calculated if this is given as the third type plain. If distance in meter is to be calculated then coordtype should be passed on as plain although the coords are supplied in UTM.
coords	A vector of size 2 giving the column numbers of the data frame which contain the coordinates of the data locations. Here the supplied data frame must contain

a column named ‘time‘ which should indicate the time index of the data row. The values in the column ‘time‘ should be positive integers starting from 1.

`prior.sigma2` Shape and scale parameter value for the gamma prior on 1/sigma^2, the precision.

`prior.tau2` Shape and scale parameter value for the gamma prior on tau^2, the nugget effect.

`prior.phi` Specifies the prior distribution for ϕ only when package is one of Stan, spTimer or spTDyn. Distribution options uniform specified by "Unif" and gamma specified by "Gamm" have been implemented in both Stan and spTimer. Additionally a half-Cauchy prior distribution specified as "Cauchy" has been implemented in Stan. In the case of spTimer the uniform distribution is discrete while in the case of Stan the uniform distribution is continuous. In the case of spTimer the option "FIXED" can be used to keep the value fixed. In that case the fixed value can be given by by a scalar value as the argument `prior.phi.param` below or it can be left unspecified in which case the fixed value of ϕ is chosen as 3/maximum distance between the data locations. The "FIXED" option is not available for the Stan package.

`prior.phi.param`

Lower and upper limits of the uniform prior distribution for phi the spatial decay parameter. For the default uniform distribution the values correspond to an effective range that is between 1% and 100% of the maximum distance between the data locations. For the Gamma distribution the default values are 2 and 1 and for the Cauchy distribution the default values are 0, 1 which specifies a half-Cauchy distribution in $(0, \infty)$.

`scale.transform`

Transformation of the response variable. It can take three values: SQRT, LOG or NONE. Default value is "NONE".

`ad.delta` Adaptive delta controlling the behavior of Stan during fitting.

`t.depth` Maximum allowed tree depth in the fitting process of Stan.

`s.size` step size in the fitting process of Stan.

`N` MCMC sample size.

`burn.in` How many initial iterations to discard. Only relevant for MCMC based model fitting, i.e., when package is spBayes or Stan.

`no.chains` Number of parallel chains to run in Stan.

`validrows` Either a number of randomly selected data rows to validate or a vector giving the row numbers of the data set for validation.

`predspace` A 0-1 flag indicating whether spatial predictions are to be made.

`newdata` A new data frame with the same column structure as the model fitting data set.

`mchoice` Logical scalar value: whether model choice statistics should be calculated.

`plotit` Logical scalar value: whether to plot the predictions against the observed values.

`rseed` Random number seed that controls the starting point for the random number stream. A set value is required to help reproduce the results.

`verbose` Logical scalar value: whether to print various estimates and statistics.

knots.coords — Only relevant for GPP models fitted by either spTimer or spTDyn. Optional two column matrix of UTM-X and UTM-Y coordinates of the knots in kilometers. It is preferable to specify the g_size parameter instead.

g_size — Only relevant for GPP models fitted by either spTimer or spTDyn. The grid size c(m, n) for the knots for the GPP model. A square grid is assumed if this is passed on as a scalar. This does not need to be given if knots.coords is given instead.

Value

A list containing:

- params - A table of parameter estimates
- fit - The fitted model object.
- datatostan - A list containing all the information sent to the rstan package.
- prior.phi.param - This contains the values of the hyperparameters of the prior distribution for the spatial decay parameter phi.
- prior.phi - This contains the name of of the prior distribution for the spatial decay parameter phi.
- validationplots - Present only if validation has been performed. Contains three validation plots with or without segment and an ordinary plot. See obs_v_pred_plot for more.
- fitteds - A vector of fitted values.
- residuals - A vector of residual values.
- package - The name of the package used for model fitting. This is always stan for this function.
- model - The name of the fitted model.
- call - The command used to call the model fitting function.
- formula - The input formula for the regression part of the model.
- scale.transform - The transformation adopted by the input argument with the same name.
- sn - The number of data locations used in fitting.
- tn - The number of time points used in fitting for each location.
- computation.time - Computation time required to run the model fitting.

See Also

Bsptime for spatio-temporal model fitting.

Examples

```
deep <- argo_floats_atlantic_2003[argo_floats_atlantic_2003$depth==3, ]
deep$x2inter <- deep$xinter*deep$xinter
deep$month <- factor(deep$month)
deep$lat2 <- (deep$lat)^2
deep$sin1 <- round(sin(deep$time*2*pi/365), 4)
deep$cos1 <- round(cos(deep$time*2*pi/365), 4)
deep$sin2 <- round(sin(deep$time*4*pi/365), 4)
```

```
deep$cos2 <- round(cos(deep$time*4*pi/365), 4)
deep[, c( "xlat2", "xsin1", "xcos1", "xsin2", "xcos2")] <-
scale(deep[,c("lat2", "sin1", "cos1", "sin2", "cos2")])
f2 <- temp ~ xlon + xlat + xlat2+ xinter + x2inter
M2 <- Bmoving_sptime(formula=f2, data = deep, coordtype="lonlat",
coords = 1:2, N=11, burn.in=6, validrows =NULL, mchoice = FALSE)
summary(M2)
plot(M2)
names(M2)
# Testing for smaller data sets with different data pattern
d2 <- deep[1:25, ]
d2$time <- 1:25
# Now there is no missing times
M1 <- Bmoving_sptime(formula=f2, data = d2, coordtype="lonlat", coords = 1:2,
N=11, burn.in=6,  mchoice = FALSE)
summary(M1)
d2[26, ] <- d2[25, ]
# With multiple observation at the same location and time
M1 <- Bmoving_sptime(formula=f2, data = d2, coordtype="lonlat", coords = 1:2,
N=11, burn.in=6,  mchoice = FALSE)
summary(M1)
d2[27, ] <- d2[24, ]
d2[27, 3] <- 25
# With previous location re-sampled
M1 <- Bmoving_sptime(formula=f2, data = d2, coordtype="lonlat", coords = 1:2,
N=11, burn.in=6,  mchoice = FALSE)
summary(M1)
```

`bmstdr_variogram` *Calculates and plots the variogram cloud and an estimated variogram.*

Description

Calculates and plots the variogram cloud and an estimated variogram.

Usage

```
bmstdr_variogram(
  formula = yo3 ~ utmx + utmy,
  coordtype = "utm",
  data = nyspatial,
  nbins = 30
)
```

Arguments

`formula`	Its a formula argument for the response and the coordinates.

coordtype	Type of coordinates: utm, lonlat or plain with utm (supplied in meters) as the default. Distance will be calculated in units of kilometer if this argument is either utm or lonlat. Euclidean distance will be calculated if this is given as the third type plain. If distance in meter is to be calculated then coordtype should be passed on as plain although the coords are supplied in UTM.
data	A data frame containing the response and the co-ordinates
nbins	Number of bins for the variogram. Default is 30.

Value

A list containing:

- cloud - A data frame containing the variogram cloud. This contains pairs of all the data locations, distance between the locations and the variogram value for the pair.
- variogram A data frame containing the variogram values in each bin.
- cloudplot A ggplot2 object of the plot of the variogram cloud.
- variogramplot A ggplot2 object of the plot of the binned variogram values.

Examples

```
a <- bmstdr_variogram(data=nyspatial, formula = yo3~utmx + utmy,
coordtype="utm", nb=50)
names(a)
if (require(ggpubr)) ggarrange(a$cloudplot, a$variogramplot, nrow=1, ncol=2)
```

Bnormal *N(theta, sigma2): Using different methods.*

Description

N(theta, sigma2): Using different methods.

Usage

```
Bnormal(
  package = "exact",
  y = ydata,
  mu0 = mean(y),
  kprior = 0,
  prior.M = 1e-04,
  prior.sigma2 = c(0, 0),
  N = 2000,
  burn.in = 1000,
  rseed = 44
)
```

Arguments

package	Which package (or method) to use. Possibilities are: • "exact": Use exact theoretical calculation. • "RGibbs": Use Gibbs sampler using R code. • "stan": Use HMC by implementing in Stan. • "inla": Use the INLA package.
y	A vector of data values. Default is 28 ydata values from the package bmstdr
mu0	The value of the prior mean if kprior=0. Default is the data mean.
kprior	A scalar providing how many data standard deviation the prior mean is from the data mean. Default value is 0.
prior.M	Prior sample size, defaults to 10^(-4).#'
prior.sigma2	Shape and scale parameter value for the gamma prior on 1/sigma^2, the precision.
N	is the number of Gibbs sampling iterations
burn.in	is the number of initial iterations to discard before making inference.
rseed	is the random number seed defaults to 44.

Value

A list containing the exact posterior means and variances of theta and sigma2

Examples

```
# Find the posterior mean and variance  using `exact' methods - no sampling
# or approximation
Bnormal(kprior = 1, prior.M = 1, prior.sigma2 = c(2, 1))
# Use default non-informative prior
Bnormal(mu0 = 0)
# Start creating table
y <-  ydata
mu0 <-  mean(y)
kprior <-  1
prior.M  <-  1
prior.sigma2 <- c(2, 1)
N  <-  10000
eresults <- Bnormal(package = "exact", y = y, mu0 = mu0, kprior = kprior,
    prior.M = prior.M, prior.sigma2 = prior.sigma2)
eresults
# Run Gibbs sampling
samps <- Bnormal(package = "RGibbs", y = y, mu0 = mu0, kprior = kprior,
    prior.M = prior.M, prior.sigma2 = prior.sigma2, N = N)
gres <- list(mean_theta = mean(samps[, 1]), var_theta = var(samps[, 1]),
    mean_sigma2 = mean(samps[, 2]), var_sigma2 = var(samps[, 2]))
glow <- list(theta_low = quantile(samps[, 1], probs = 0.025), var_theta = NA,
    sigma2_low = quantile(samps[, 2], probs = 0.025), var_sigma2 = NA)
gupp <- list(theta_low = quantile(samps[, 1], probs = 0.975), var_theta = NA,
    sigma2_low = quantile(samps[, 2], probs = 0.975), var_sigma2 = NA)
```

```
a <- rbind(unlist(eresults), unlist(gres), unlist(glow), unlist(gupp))
cvsamp <- sqrt(samps[, 2])/samps[, 1]
cv <- c(NA, mean(cvsamp), quantile(cvsamp, probs = c(0.025, 0.975)))
u <- data.frame(a, cv)
rownames(u) <- c("Exact", "Estimate", "2.5%", "97.5%")
print(u)
# End create table
##
# Compute using the model fitted by Stan
u <- Bnormal(package = "stan", kprior = 1, prior.M = 1, prior.sigma = c(2, 1),
    N = 2000, burn.in = 1000)
print(u)
###
# Compute using INLA
if (require(INLA)) {
    u <- Bnormal(package = "inla", kprior = 1, prior.M = 1, prior.sigma = c(2,
        1), N = 1000)
    print(u)
}
```

`Bspatial` *Bayesian regression model fitting for point referenced spatial data. Calculates parameter estimates, validation statistics, and estimated values of several Bayesian model choice criteria.*

Description

Bayesian regression model fitting for point referenced spatial data. Calculates parameter estimates, validation statistics, and estimated values of several Bayesian model choice criteria.

Usage

```
Bspatial(
  formula,
  data,
  package = "none",
  model = "lm",
  coordtype = NULL,
  coords = NULL,
  validrows = NULL,
  scale.transform = "NONE",
  prior.beta0 = 0,
  prior.M = 1e-04,
  prior.sigma2 = c(2, 1),
  prior.tau2 = c(2, 0.1),
  phi = NULL,
  prior.phi.param = NULL,
  prior.range = c(1, 0.5),
```

```
  prior.sigma = c(1, 0.005),
  offset = c(10, 140),
  max.edge = c(50, 1000),
  cov.model = "exponential",
  N = 5000,
  burn.in = 1000,
  rseed = 44,
  n.report = 500,
  no.chains = 1,
  ad.delta = 0.99,
  s.size = 0.01,
  t.depth = 15,
  verbose = TRUE,
  plotit = TRUE,
  mchoice = FALSE,
  ...
)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	The data frame for which the model formula is to be fitted. If a spatial model is to be fitted then the data frame should contain two columns containing the locations of the coordinates. See the coords argument below.
package	Which package is to be used in model fitting? Currently available packages are: • "spBayes": The model implemented is the marginal model with nugget effect using the spLM function. • "stan": The model implemented is the full spatial random effect model with nugget effect where the decay parameter has been assumed to be fixed. • "inla": The model fitted is the spatial random effect model with the nugget effect. • "none": In this case case, the argument model must be specified either as "lm" or "spat". See below. Further details and more examples are provided in Chapter 6 of the book Sahu (2022).
model	Only used when the package has been chosen to be "none". It can take one of two values: either "lm" or "spat". The "lm" option is for an independent error regression model while the "spat" option fits a spatial model without any nugget effect.
coordtype	Type of coordinates: utm, lonlat or plain with utm (supplied in meters) as the default. Distance will be calculated in units of kilometer if this argument is either utm or lonlat. Euclidean distance will be calculated if this is given as the third type plain. If distance in meter is to be calculated then coordtype should be passed on as plain although the coords are supplied in UTM.

`coords` A vector of size two identifying the two column numbers of the data frame to take as coordinates. Or this can be given as a matrix of number of sites by 2 providing the coordinates of all the data locations.

`validrows` A vector of site indices which should be used for validation. This function does not allow some sites to be used for both fitting and validation. The remaining observations will be used for model fitting. The default NULL value instructs that validation will not be performed.

`scale.transform` Transformation of the response variable. It can take three values: SQRT, LOG or NONE.

`prior.beta0` A scalar value or a vector providing the prior mean for beta parameters.

`prior.M` Prior precision value (or matrix) for beta. Defaults to a diagonal matrix with diagonal values 10^(-4).

`prior.sigma2` Shape and scale parameter value for the gamma prior on 1/sigma^2, the precision.

`prior.tau2` Shape and scale parameter value for the gamma prior on tau^2, the nugget effect.

`phi` The spatial decay parameter for the exponential covariance function. Only used if the package is Stan or the model is a spatial model "spat" without nugget effect when the `package` is "none".

`prior.phi.param` Lower and upper limits of the uniform prior distribution for ϕ, the spatial decay parameter when the package is `spBayes`. If this is not specified the default values are chosen so that the effective range is uniformly distributed between 25% and 100% of the maximum distance between data locations.

`prior.range` A length 2 vector, with (range0, Prange) specifying that $P(\rho < \rho_0) = p_\rho$, where ρ is the spatial range of the random field. If Prange is NA, then range0 is used as a fixed range value. If this parameter is unspecified then range0=0.90 * maximum distance and Prange =0.95. If instead a single value is specified then the range is set at the single value.

`prior.sigma` A length 2 vector, with (sigma0, Psigma) specifying that $P(\sigma > \sigma_0) = p_\sigma$, where σ is the marginal standard deviation of the field. If Psigma is NA, then sigma0 is used as a fixed range value.

`offset` Only used in INLA based modeling. Offset parameter. See documentation for `inla.mesh.2d`.

`max.edge` Only used in INLA based modeling. See documentation for `inla.mesh.2d`.

`cov.model` Only relevant for the spBayes package. Default is the exponential model. See the documentation for spLM in the package spBayes.

`N` MCMC sample size. Default value 5000.

`burn.in` How many initial iterations to discard. Default value 1000. Only relevant for MCMC based model fitting, i.e., when package is spBayes or Stan.

`rseed` Random number seed that controls the starting point for the random number stream. A set value is required to help reproduce the results.

`n.report` How many times to report in MCMC progress. This is used only when the package is spBayes.

no.chains	Number of parallel chains to run in Stan.
ad.delta	Adaptive delta controlling the behavior of Stan during fitting.
s.size	step size in the fitting process of Stan.
t.depth	Maximum allowed tree depth in the fitting process of Stan.
verbose	Logical scalar value: whether to print various estimates and statistics.
plotit	Logical scalar value: whether to plot the predictions against the observed values.
mchoice	Logical scalar value: whether model choice statistics should be calculated.
...	Any additional arguments that may be passed on to the fitting package.

Value

A list containing:

- params - A table of parameter estimates
- fit - The fitted model object. This is present only if a named package, e.g. spBayes has been used.
- max.d - Maximum distance between data locations. This is in unit of kilometers unless the coordtype argument is set as plain.
- fitteds - A vector of fitted values.
- mchoice - Calculated model choice statistics if those have been requested by the input argument mchoice=TRUE. Not all model fits will contain all the model choice statistics.
- stats - The four validation statistics: rmse, mae, crps and coverage. This is present only if model validation has been performed.
- yobs_preds - A data frame containing the validation rows of the model fitting data frame. The last five columns of this data frame contains the validation prediction summaries: mean, sd, median, and 95% prediction interval. This is present only if model validation has been performed.
- valpreds - A matrix containing the MCMC samples of the validation predictions. The dimension of this matrix is the number of validations times the number of retained MCMC samples. This is present only if model validation has been performed.
- validationplots - Present only if validation has been performed. Contains three validation plots with or without segment and an ordinary plot. See obs_v_pred_plot for more.
- residuals - A vector of residual values.
- sn - The number of data locations used in fitting.
- tn Defaults to 1. Used for plotting purposes.
- phi - If present this contains the fixed value of the spatial decay parameter phi used to fit the model.
- prior.phi.param - If present this contains the values of the hyperparameters of the prior distribution for the spatial decay parameter phi.
- prior.range - Present only if the INLA package has been used in model fitting. This contains the values of the hyperparameters of the prior distribution for the range.
- logliks - A list containing the log-likelihood values used in calculation of the model choice statistics if those have been requested in the first place.

- formula - The input formula for the regression part of the model.
- scale.transform - The transformation adopted by the input argument with the same name.
- package - The name of the package used for model fitting.
- model - The name of the fitted model.
- call - The command used to call the model fitting function.
- computation.time - Computation time required to run the model fitting.

See Also

Bsptime for Bayesian spatio-temporal model fitting.

Examples

```
N <- 10
burn.in <- 5
n.report <- 2
a <- Bspatial(formula = mpg ~ wt, data = mtcars, package = "none", model = "lm",
    N = N)
summary(a)
plot(a)
print(a)
b <- Bspatial(formula = mpg ~ disp + wt + qsec + drat, data = mtcars,
    validrows = c(8, 11, 12, 14, 18, 21, 24, 28), N = N)
#' print(b)
summary(b)
## Illustration with the nyspatial data set
head(nyspatial)
## Linear regression model fitting
M1 <- Bspatial(formula = yo3 ~ xmaxtemp + xwdsp + xrh, data = nyspatial,
    mchoice = TRUE, N = N)
print(M1)
plot(M1)
a <- residuals(M1)
summary(M1)
## Spatial model fitting
M2 <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp +
    xrh, data = nyspatial, coordtype = "utm", coords = 4:5, phi = 0.4,
    mchoice = TRUE, N = N)
names(M2)
print(M2)
plot(M2)
a <- residuals(M2)
summary(M2)
## Fit model 2 on the square root scale
M2root <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, scale.transform = "SQRT")
summary(M2root)
## Spatial model fitting using spBayes
M3 <- Bspatial(package = "spBayes", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, prior.phi = c(0.005,
        2), mchoice = TRUE, N = N, burn.in = burn.in, n.report = n.report)
```

```
summary(M3)

# Spatial model fitting using stan (with a small number of iterations)
M4 <- Bspatial(package = "stan", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, phi = 0.4, N = N,
    burn.in = burn.in, mchoice = TRUE)
summary(M4)


## K fold cross-validation for M2 only
set.seed(44)
x <- runif(n = 28)
u <- order(x)
# Here are the four folds
s1 <- u[1:7]
s2 <- u[8:14]
s3 <- u[15:21]
s4 <- u[22:28]
summary((1:28) - sort(c(s1, s2, s3, s4)))  ## check
v1 <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s1,
    phi = 0.4, N = N)
v2 <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s2,
    phi = 0.4, N = N)
v3 <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s3,
    phi = 0.4, N = N)
v4 <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s4,
    phi = 0.4, N = N)
M2.val.table <- cbind(unlist(v1$stats), unlist(v2$stats), unlist(v3$stats),
    unlist(v4$stats))
dimnames(M2.val.table)[[2]] <- paste("Fold", 1:4, sep = "")
round(M2.val.table, 3)

## Model validation
s <- c(1, 5, 10)
M1.v <- Bspatial(model = "lm", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s, N = N,
    burn.in = burn.in)
M2.v <- Bspatial(model = "spat", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
    data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s, phi = 0.4,
    N = N, burn.in = burn.in)
M3.v <- Bspatial(package = "spBayes", formula = yo3 ~ xmaxtemp + xwdsp +
    xrh, data = nyspatial, coordtype = "utm", coords = 4:5, validrows = s,
    prior.phi = c(0.005, 2), n.report = 2, N = N, burn.in = burn.in)
# Collect all the results
Mall.table <- cbind(unlist(M1.v$stats), unlist(M2.v$stats), unlist(M3.v$stats))
colnames(Mall.table) <- paste("M", c(1:3), sep = "")
round(Mall.table, 3)
```

```
if (require(INLA) & require(inlabru)) {
    N <- 10
    burn.in <- 5
    # Spatial model fitting using INLA
    M5 <- Bspatial(package = "inla", formula = yo3 ~ xmaxtemp + xwdsp + xrh,
        data = nyspatial, coordtype = "utm", coords = 4:5, mchoice = TRUE,
        N = N, burn.in = burn.in)
    summary(M5)
}
```

Bsptime *Bayesian regression model fitting for point referenced spatio-temporal data. Calculates parameter estimates, validation statistics, and estimated values of several Bayesian model choice criteria.*

Description

Bayesian regression model fitting for point referenced spatio-temporal data. Calculates parameter estimates, validation statistics, and estimated values of several Bayesian model choice criteria.

Usage

```
Bsptime(
  formula,
  data,
  package = "none",
  model = "GP",
  coordtype = NULL,
  coords = NULL,
  validrows = NULL,
  scale.transform = "NONE",
  prior.beta0 = 0,
  prior.M = 1e-04,
  prior.sigma2 = c(2, 1),
  prior.tau2 = c(2, 0.1),
  prior.sigma.eta = c(2, 0.001),
  phi.s = NULL,
  phi.t = NULL,
  prior.phi = "Gamm",
  prior.phi.param = NULL,
  phi.tuning = NULL,
  phi.npoints = NULL,
  prior.range = c(1, 0.5),
  prior.sigma = c(1, 0.005),
  offset = c(10, 140),
  max.edge = c(50, 1000),
  rhotp = 0,
```

```
  time.data = NULL,
  truncation.para = list(at = 0, lambda = 2),
  newcoords = NULL,
  newdata = NULL,
  annual.aggrn = "NONE",
  cov.model = "exponential",
  g_size = NULL,
  knots.coords = NULL,
  tol.dist = 0.005,
  N = 2000,
  burn.in = 1000,
  rseed = 44,
  n.report = 2,
  no.chains = 1,
  ad.delta = 0.8,
  t.depth = 15,
  s.size = 0.01,
  verbose = FALSE,
  plotit = TRUE,
  mchoice = FALSE,
  ...
)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	The data frame for which the model formula is to be fitted. The data frame should be in long format having one row for each location and time combination. The data frame must be ordered by time within each site, and should optionally have a column, named s.index, providing the site indices. Thus the data, with n sites and T times within each site, should be organized in the order: (s1, t1), (s1, t2), ... (s1, T), ... (sn, t1), ... (sn, T). The data frame should also contain two columns giving the coordinates of the locations for spatio temporal model fitting.
package	Which package is to be used in model fitting? Currently available packages are: • "spBayes": The model implemented is the dynamic spatio-temporal model fitted using the spDynLM function in the spBayes package. • "stan": The model implemented is the marginal independent GP model. • "inla" The only model implemented is the AR model. • "spTimer": All possible models in this package can be fitted. • "sptDyn": All possible models in this package can be fitted. • "none".: In this case case, the argument model must be specified either as "lm" or "separable". See below. Further details and more examples are provided in Chapters 7-9 of the book Sahu (2022).

`model` The model to be fitted. This argument is passed to the fitting package. In case the package is none, then it can be either "lm" or "separable". The "lm" option is for an independent error regression model while the other option fits a separable model without any nugget effect. The separable model fitting method cannot handle missing data. All missing data points in the response variable will be replaced by the grand mean of the available observations.

`coordtype` Type of coordinates: utm, lonlat or plain with utm (supplied in meters) as the default. Distance will be calculated in units of kilometer if this argument is either utm or lonlat. Euclidean distance will be calculated if this is given as the third type plain. If distance in meter is to be calculated then coordtype should be passed on as plain although the coords are supplied in UTM.

`coords` A vector of size two identifying the two column numbers of the data frame to take as coordinates. Or this can be given as a matrix of number of sites by 2 providing the coordinates of all the data locations.

`validrows` A vector of row numbers of the supplied data frame which should be used for validation. When the model is "separable" this argument must include all the time points for the sites to be validated. Otherwise, the user is allowed to select the row numbers of the data frame validation as they wish. The default NULL value instructs that validation will not be performed.

- lm model: If package is "none" and the model is "lm", this argument is a vector of row indices of the data frame which should be used for validation.
- separable model: If package is "none" and the model is "separable" this argument is a vector of site indices which should be used for validation. The "separable" model does not allow some sites to be used for both fitting and validation. Thus it is not possible to validate at selected time points using the separable model. Further details are provided in Chapter 7 of Sahu (2022).

`scale.transform` Transformation of the response variable. It can take three values: SQRT, LOG or NONE. Default value is "NONE".

`prior.beta0` A scalar value or a vector providing the prior mean for beta parameters.

`prior.M` Prior precision value (or matrix) for beta. Defaults to a diagonal matrix with diagonal values 10^(-4).

`prior.sigma2` Shape and scale parameter value for the gamma prior on 1/sigma^2, the precision.

`prior.tau2` Shape and scale parameter value for the gamma prior on tau^2, the nugget effect.

`prior.sigma.eta` Shape and scale parameter value for the inverse gamma prior distribution for sigma^2 eta; only used in the spBayes package.

`phi.s` Only used if the model is "separable". The value of the fixed spatial decay parameter for the exponential covariance function. If this is not provided then a value is chosen which corresponds to an effective range which is the maximum distance between the data locations.

`phi.t` Only used if the model is "separable". The fixed decay parameter for the exponential covariance function in the temporal domain. If this is not provided then

a value is chosen which corresponds to an effective temporal range which is the maximum time of the data set.

`prior.phi` Specifies the prior distribution for ϕ only when package is one of Stan, spTimer or spTDyn. Distribution options uniform specified by "Unif" and gamma specified by "Gamm" have been implemented in both Stan and spTimer. Additionally a half-Cauchy prior distribution specified as "Cauchy" has been implemented in Stan. In the case of spTimer the uniform distribution is discrete while in the case of Stan the uniform distribution is continuous. In the case of spTimer the option "FIXED" can be used to keep the value fixed. In that case the fixed value can be given by by a scalar value as the argument `prior.phi.param` below or it can be left unspecified in which case the fixed value of ϕ is chosen as 3/maximum distance between the data locations. The "FIXED" option is not available for the Stan package.

`prior.phi.param`
Lower and upper limits of the uniform prior distribution for phi the spatial decay parameter. For the default uniform distribution the values correspond to an effective range that is between 1% and 100% of the maximum distance between the data locations. For the Gamma distribution the default values are 2 and 1 and for the Cauchy distribution the default values are 0, 1 which specifies a half-Cauchy distribution in $(0, \infty)$.

`phi.tuning` Only relevant for spTimer and spTDyn models. Tuning parameter fo sampling phi. See the help file for spT.Gibbs

`phi.npoints` Only relevant for spTimer and spTDyn models. Number of points for the discrete uniform prior distribution on phi. See the help file for spT.Gibbs

`prior.range` A length 2 vector, with (range0, Prange) specifying that $P(\rho < \rho_0) = p_\rho$, where ρ is the spatial range of the random field. If Prange is NA, then range0 is used as a fixed range value. If this parameter is unspecified then range0=0.90 * maximum distance and Prange =0.95. If instead a single value is specified then the range is set at the single value.

`prior.sigma` A length 2 vector, with (sigma0, Psigma) specifying that $P(\sigma > \sigma_0) = p_\sigma$, where σ is the marginal standard deviation of the field. If Psigma is NA, then sigma0 is taken as the fixed value of this parameter.

`offset` Only used in INLA based modeling. Offset parameter. See documentation for `inla.mesh.2d`.

`max.edge` Only used in INLA based modeling. See documentation for `inla.mesh.2d`.

`rhotp` Only relevant for models fitted by spTDyn. Initial value for the rho parameters in the temporal dynamic model. The default is rhotp=0 for which parameters are sampled from the full conditional distribution via MCMC with initial value 0. If rhotp=1,parameters are not sampled and fixed at value 1.

`time.data` Defining the segments of the time-series set up using the function spT.time in the spTimer package. Only used with the spTimer package.

`truncation.para`
Provides truncation parameter lambda and truncation point "at" using list. Only used with the spTimer package for a truncated model.

newcoords	The locations of the prediction sites in similar format to the coords argument, only required if fit and predictions are to be performed simultaneously. If omitted, no predictions will be performed.
newdata	The covariate values at the prediction sites specified by newcoords. This should have same space-time structure as the original data frame.
annual.aggrn	This provides the options for calculating annual summary statistics by aggregating different time segments (e.g., annual mean). Currently implemented options are: "NONE", "ave" and "an4th", where "ave" = annual average, "an4th"= annual 4th highest. Only applicable if spT.time inputs more than one segment and when fit and predict are done simultaneously. Only used in the spTimer package.
cov.model	Model for the covariance function. Only relevant for the spBayes, spTimer and the spTDyn packages. Default is the exponential model. See the documentation for spLM in the package spBayes.
g_size	Only relevant for GPP models fitted by either spTimer or spTDyn. The grid size c(m, n) for the knots for the GPP model. A square grid is assumed if this is passed on as a scalar. This does not need to be given if knots.coords is given instead.
knots.coords	Only relevant for GPP models fitted by either spTimer or spTDyn. Optional two column matrix of UTM-X and UTM-Y coordinates of the knots in kilometers. It is preferable to specify the g_size parameter instead.
tol.dist	Minimum separation distance between any two locations out of those specified by coords, knots.coords and pred.coords. The default is 0.005. The program will exit if the minimum distance is less than the non-zero specified value. This will ensure non-singularity of the covariance matrices.
N	MCMC sample size.
burn.in	How many initial iterations to discard. Only relevant for MCMC based model fitting, i.e., when package is spBayes or Stan.
rseed	Random number seed that controls the starting point for the random number stream. A set value is required to help reproduce the results.
n.report	How many times to report in MCMC progress. This is only used when the package is spBayes or spTimer.
no.chains	Number of parallel chains to run in Stan.
ad.delta	Adaptive delta controlling the behavior of Stan during fitting.
t.depth	Maximum allowed tree depth in the fitting process of Stan.
s.size	step size in the fitting process of Stan.
verbose	Logical scalar value: whether to print various estimates and statistics.
plotit	Logical scalar value: whether to plot the predictions against the observed values.
mchoice	Logical scalar value: whether model choice statistics should be calculated.
...	Any additional arguments that may be passed on to the fitting package.

Value

A list containing:

- params - A table of parameter estimates
- fit - The fitted model object. This is present only if a named package, e.g. `spTimer` has been used.
- max.d - Maximum distance between data locations. This is in unit of kilometers unless the `coordtype` argument is set as `plain`.
- fitteds - A vector of fitted values.
- mchoice - Calculated model choice statistics if those have been requested by the input argument `mchoice=TRUE`. Not all model fits will contain all the model choice statistics.
- stats - The four validation statistics: rmse, mae, crps and coverage. This is present only if model validation has been performed.
- yobs_preds - A data frame containing the validation rows of the model fitting data frame. The last five columns of this data frame contains the validation prediction summaries: mean, sd, median, and 95% prediction interval. This is present only if model validation has been performed.
- valpreds - A matrix containing the MCMC samples of the validation predictions. The dimension of this matrix is the number of validations times the number of retained MCMC samples. This is present only if model validation has been performed.
- validationplots - Present only if validation has been performed. Contains three validation plots with or without segment and an ordinary plot. See `obs_v_pred_plot` for more.
- residuals - A vector of residual values.
- sn - The number of data locations used in fitting.
- tn - The number of time points used in fitting.
- phi.s, phi.t - Adopted value of the spatial and temporal decay parameters if those were fixed during model fitting.
- prior.phi - If present this contains the name of the prior distribution for the spatial decay parameter phi used to fit the model.
- prior.phi.param - If present this contains the values of the hyperparameters of the prior distribution for the spatial decay parameter phi.
- prior.range - Present only if the `INLA` package has been used in model fitting. This contains the values of the hyperparameters of the prior distribution for the range.
- logliks - A list containing the log-likelihood values used in calculation of the model choice statistics if those have been requested in the first place.
- knots.coords - The locations of the knots if the model has been fitted using the GPP method.
- formula - The input formula for the regression part of the model.
- scale.transform - The transformation adopted by the input argument with the same name.
- package - The name of the package used for model fitting.
- model - The name of the fitted model.
- call - The command used to call the model fitting function.
- computation.time - Computation time required to run the model fitting.

References

Sahu SK (2022). *Bayesian Modeling of Spatio Temporal Data with R*, 1st edition. Chapman and Hall, Boca Raton. https://doi.org/10.1201/9780429318443.

Examples

```
# Set the total number of iterations
N <- 45
# Set the total number of burn-in iterations
burn.in <- 5
# How many times to report progress
n.report <- 2
# Model formula used in most model fitting
f2 <- y8hrmax ~ xmaxtemp + xwdsp + xrh
# Check out the data set
head(nysptime)
## Fit linear regression model
M1 <- Bsptime(model = "lm", data = nysptime, formula = f2,
    scale.transform = "SQRT", N = N, burn.in = burn.in, mchoice = TRUE)
names(M1)
plot(M1)
print(M1)
summary(M1)
a <- residuals(M1, numbers = list(sn = 28, tn = 62))
M2 <- Bsptime(model = "separable", data = nysptime, formula = f2,
    coordtype = "utm", coords = 4:5, mchoice = TRUE, scale.transform = "SQRT",
    N = N, burn.in = burn.in)
names(M2)
plot(M2)
print(M2)
summary(M2)
b <- residuals(M2)
# Spatio-temporal model fitting and validation
valids <- c(8, 11)
vrows <- which(nysptime$s.index %in% valids)
## Fit separable spatio-temporal model
M2.1 <- Bsptime(model = "separable", formula = f2, data = nysptime,
    validrows = vrows, coordtype = "utm", coords = 4:5, phi.s = 0.005,
    phi.t = 0.05, scale.transform = "SQRT", N = N)
summary(M2.1)
plot(M2.1)
# Use spTimer to fit independent GP model
M3 <- Bsptime(package = "spTimer", formula = f2, data = nysptime,
    coordtype = "utm", coords = 4:5, scale.transform = "SQRT", mchoice = TRUE,
    N = N, burn.in = burn.in, n.report = 2)
summary(M3)

valids <- c(1, 5, 10)
validt <- sort(sample(1:62, size = 31))
vrows <- getvalidrows(sn = 28, tn = 62, valids = valids, validt = validt)
ymat <- matrix(nysptime$y8hrmax, byrow = TRUE, ncol = 62)
yholdout <- ymat[valids, validt]
```

```
# Perform validation
M31 <- Bsptime(package = "spTimer", formula = f2, data = nysptime,
    coordtype = "utm", coords = 4:5, validrows = vrows, model = "GP",
    scale.transform = "NONE", N = N, burn.in = burn.in, n.report = 2)
summary(M31)
modfit <- M31$fit
## Extract the fits for the validation sites
fitall <- data.frame(modfit$fitted)
head(fitall)
tn <- 62
fitall$s.index <- rep(1:28, each = tn)
library(spTimer)
vdat <- spT.subset(data = nysptime, var.name = c("s.index"), s = valids)
fitvalid <- spT.subset(data = fitall, var.name = c("s.index"), s = valids)
head(fitvalid)
fitvalid$low <- fitvalid$Mean - 1.96 * fitvalid$SD
fitvalid$up <- fitvalid$Mean + 1.96 * fitvalid$SD
fitvalid$yobs <- sqrt(vdat$y8hrmax)
fitvalid$yobs <- vdat$y8hrmax
yobs <- matrix(fitvalid$yobs, byrow = TRUE, ncol = tn)
y.valids.low <- matrix(fitvalid$low, byrow = TRUE, ncol = tn)
y.valids.med <- matrix(fitvalid$Mean, byrow = TRUE, ncol = tn)
y.valids.up <- matrix(fitvalid$up, byrow = TRUE, ncol = tn)
library(ggplot2)
p1 <- fig11.13.plot(yobs[1, ], y.valids.low[1, ], y.valids.med[1, ],
    y.valids.up[1, ], misst = validt)
p1 <- p1 + ggtitle("Validation for Site 1")
p1
p2 <- fig11.13.plot(yobs[2, ], y.valids.low[2, ], y.valids.med[2, ],
    y.valids.up[2, ], misst = validt)
p2 <- p2 + ggtitle("Validation for Site 5")
p2
p3 <- fig11.13.plot(yobs[3, ], y.valids.low[3, ], y.valids.med[3, ],
    y.valids.up[3, ], misst = validt)
p3 <- p3 + ggtitle("Validation for Site 10")
p3

## Independent marginal GP model fitting using rstan

M4 <- Bsptime(package = "stan", formula = f2, data = nysptime,
    coordtype = "utm", coords = 4:5, N = N, burn.in = burn.in,
    verbose = FALSE)
summary(M4)

# Spatio-temporal hierarchical auto-regressive modeling useing spTimer
M5 <- Bsptime(package = "spTimer", model = "AR", formula = f2, data = nysptime,
    coordtype = "utm", coords = 4:5, scale.transform = "SQRT", mchoice = TRUE,
    n.report = n.report, N = N, burn.in = burn.in)
summary(M5)
a <- residuals(M5)

## Spatio-temporal dynamic model fitting using spTDyn
library(spTDyn)
```

```
f3 <- y8hrmax ~ xmaxtemp + sp(xmaxtemp) + tp(xwdsp) + xrh
M7 <- Bsptime(package = "sptDyn", model = "GP", formula = f3, data = nysptime,
    coordtype = "utm", coords = 4:5, scale.transform = "SQRT", mchoice = TRUE,
    N = N, burn.in = burn.in, n.report = n.report)
summary(M7)

# Dynamic Model fitting using spBayes
M8 <- Bsptime(package = "spBayes", formula = f2, data = nysptime,
    prior.sigma2 = c(2, 25), prior.tau2 = c(2, 25), prior.sigma.eta = c(2,
        0.001), coordtype = "utm", coords = 4:5, scale.transform = "SQRT",
    N = N, burn.in = burn.in, n.report = n.report)
summary(M8)

## Gussian Predictive Process based model fitting using spTimer
M9 <- Bsptime(package = "spTimer", model = "GPP", g_size = 5, formula = f2,
    data = nysptime, coordtype = "utm", coords = 4:5, scale.transform = "SQRT",
    N = N, burn.in = burn.in, n.report = n.report)
summary(M9)

# This INLA run may take a long time
if (require(INLA) & require(inlabru)) {
    f2 <- y8hrmax ~ xmaxtemp + xwdsp + xrh
    M6 <- Bsptime(package = "inla", model = "AR", formula = f2, data = nysptime,
        coordtype = "utm", coords = 4:5, scale.transform = "SQRT",
        offset = c(100, 200), max.edge = c(500, 10000),
        mchoice = TRUE, plotit=TRUE)
    # Takes a minute
    summary(M6)
}
```

`calculate_dic` *Calculate DIC function. Has two arguments: (1) log full likelihood at thetahat and (2) vector of log-likelihood at the theta samples Calculate the DIC criteria values*

Description

Calculate DIC function. Has two arguments: (1) log full likelihood at thetahat and (2) vector of log-likelihood at the theta samples Calculate the DIC criteria values

Usage

```
calculate_dic(loglikatthetahat, logliks)
```

Arguments

loglikatthetahat	Log of the likelihood function at theta hat (Bayes). It is a scalar value.
logliks	A vector of log likelihood values at the theta samples

Value

a list containing four values pdic, pdicalt, dic and dicalt

---

calculate_validation_statistics

Calculates the four validation statistics: RMSE, MAE, CRPS and coverage given the observed values and MCMC iterates.

Description

Calculates the four validation statistics: RMSE, MAE, CRPS and coverage given the observed values and MCMC iterates.

Usage

```
calculate_validation_statistics(yval, yits, level = 95, summarystat = "mean")
```

Arguments

yval	A vector containing n observed values of the response variable.
yits	A n by N matrix of predictive samples from the n observations contained in yval.
level	The nominal coverage level, defaults to 95%.
summarystat	Summary statistics to use to calculate the validation predictions from the samples. It should be a function like mean or median which can be calculated by R. The default is mean.

Value

A list giving the rmse, mae, crps and coverage.

Examples

```
set.seed(4)
vrows <- sample(nrow(nysptime), 100)
M1 <- Bsptime(model="lm", formula=y8hrmax~xmaxtemp+xwdsp+xrh, data=nysptime,
validrows=vrows, scale.transform = "SQRT")
valstats <- calculate_validation_statistics(M1$yobs_preds$y8hrmax,
yits=t(M1$valpreds))
unlist(valstats)
```

calculate_waic *Calculate WAIC function. Has the sole argument component wise likelihood at thetahat samples. v must be a matrix Calculate the WAIC criteria values*

Description

Calculate WAIC function. Has the sole argument component wise likelihood at thetahat samples. v must be a matrix Calculate the WAIC criteria values

Usage

```
calculate_waic(v)
```

Arguments

v	must be a N (MCMC) by n (data) sample size matrix of the log-likelihood evaluations

Value

a list containing four values p_waic, p_waic alt, waic and waic_alt

coef.bmstdr *Prints and returns the estimates of the coefficients*

Description

Prints and returns the estimates of the coefficients

Usage

```
## S3 method for class 'bmstdr'
coef(object, digits = 3, ...)
```

Arguments

object	A bmstdr model fit object.
digits	How many significant digits after the decimal to print, defaults to 3.
...	Any other additional arguments

Value

Coefficients are returned as a data frame preserving the names of the covariates

colpalette — *The color palette used to draw maps to illustrate the package bmstdr, see Sahu (2022) It has the values in order: dodgerblue4, dodgerblue2, firebrick2, firebrick4 and purple.*

Description

The color palette used to draw maps to illustrate the package bmstdr, see Sahu (2022) It has the values in order: dodgerblue4, dodgerblue2, firebrick2, firebrick4 and purple.

Usage

```
colpalette
```

Format

An object of class `character` of length 5.

References

Sahu SK (2022). *Bayesian Modeling of Spatio Temporal Data with R*, 1st edition. Chapman and Hall, Boca Raton. https://doi.org/10.1201/9780429318443.

Examples

```
colpalette
```

engdeaths — *Number of weekly Covid-19 deaths and cases in the 313 local Local Authority Districts, Counties and Unitary Authorities (LADCUA) in England during the 20 peaks in the first peak from March 13 to July 31, 2020.*

Description

Number of weekly Covid-19 deaths and cases in the 313 local Local Authority Districts, Counties and Unitary Authorities (LADCUA) in England during the 20 peaks in the first peak from March 13 to July 31, 2020.

Usage

```
engdeaths
```

Format

An object of class `data.frame` with 6260 rows and 24 columns.

Source

Sahu and Böhning (2021). @format A data frame with 6260 rows and 24 columns:

Areacode Areacode identifier of the 313 Local Authority Districts, Counties and Unitary Authorities (LADCUA)

mapid A numeric column identifying the map area needed for plotting

spaceid A numeric variable taking value 1 to 313 identifying the LADCUA's

Region Identifies one of the 9 English regions

popn Population number in mid-2019

jsa Percentage of the working age population receiving job-seekers allowance during January 2020

houseprice Median house price in March 2020

popdensity Population density in mid-2019

no2 Estimated average value of NO2 at the centroid of the LADCUA

covid Number of Covid-19 deaths within 28 days of a positive test

allcause Number deaths

noofcases Number of cases

n0 Log of the standardized case morbidity during the current week

n1 Log of the standardized case morbidity during the week before

n2 Log of the standardized case morbidity during the second week before

n3 Log of the standardized case morbidity during the third week before

n4 Log of the standardized case morbidity during the fourth week before

Edeaths Expected number of Covid-19 deaths. See Sahu and Bohning (2021) for methodology.

Ecases Expected number of cases.

logEdeaths Log of the column `Edeaths`

logEcases Log of the column Ecases

highdeathsmr A binary (0-1) random variable taking the value 1 if the SMR of Covid-19 death is higher than 1

References

Sahu SK, Böhning D (2021). "Bayesian spatio-temporal joint disease mapping of Covid-19 cases and deaths in local authorities of England." *Spatial Statistics*. doi:10.1016/j.spasta.2021.100519.

Examples

```
colnames(engdeaths)
dim(engdeaths)
summary(engdeaths[, 11:24])
```

`engtotals` *Total number of weekly Covid-19 deaths and cases in the 313 local Local Authority Districts, Counties and Unitary Authorities (LADCUA) in England during the first peak from March 13 to July 31, 2020.*

Description

Total number of weekly Covid-19 deaths and cases in the 313 local Local Authority Districts, Counties and Unitary Authorities (LADCUA) in England during the first peak from March 13 to July 31, 2020.

Usage

```
engtotals
```

Format

An object of class `data.frame` with 313 rows and 19 columns.

Source

Sahu and Böhning (2021). @format A data frame with 313 rows and 19 columns:

Areacode Areacode identifier of the 313 Local Authority Districts, Counties and Unitary Authorities (LADCUA)

mapid A numeric column identifying the map area needed for plotting

spaceid A numeric variable taking value 1 to 313 identifying the LADCUA's

Region Identifies one of the 9 English regions

popn Population number in mid-2019

jsa Percentage of the working age population receiving job-seekers allowance during January 2020

houseprice Median house price in March 2020

popdensity Population density in mid-2019

startdate Start date of the week

Weeknumber Week numbers 11 to 30

no2 Estimated average value of NO2 at the centroid of the LADCUA in that week

covid Number of Covid-19 deaths within 28 days of a positive test

allcause Total number deaths

noofcases Total number of cases

Edeaths Expected number of Covid-19 deaths. See Sahu and Bohning (2021) for methodology.

Ecases Expected number of cases.

logEdeaths Log of the column `Edeaths`

logEcases Log of the column `Ecases`

casesmr Standaridised morbidity rate for the number of cases, `noofcases/Ecases`

nweek Number of weeks during March 13 to July 31, 2020. All values are 20.

noofhighweeks Number of weeks out of 20 when the `casesmr` was greater than 1

References

Sahu SK, Böhning D (2021). "Bayesian spatio-temporal joint disease mapping of Covid-19 cases and deaths in local authorities of England." *Spatial Statistics*. doi:10.1016/j.spasta.2021.100519.

Examples

```
colnames(engtotals)
dim(engtotals)
summary(engtotals[, 5:14])
```

`fig11.13.plot` *Draws a time series (ribbon) plot by combining fitted and predicted values*

Description

Draws a time series (ribbon) plot by combining fitted and predicted values

Usage

```
fig11.13.plot(yobs, ylow, ymed, yup, misst)
```

Arguments

`yobs`	A vector of the observed values
`ylow`	A vector of the lower limits of the fitted or predicted values
`ymed`	A vector of fitted or predicted values
`yup`	A vector of the upper limits of the fitted or predicted values
`misst`	An integer vector which contains the indices of the predicted values

Value

A ribbon plot, ggplot2 object, which shows observed values in red color and open circle, predicted values in blue color and filled circle.

fitted.bmstdr *Extract fitted values from bmstdr objects.*

Description

Extract fitted values from bmstdr objects.

Usage

```
## S3 method for class 'bmstdr'
fitted(object, ...)
```

Arguments

object	A bmstdr model fit object.
...	Any other additional arguments.

Value

Returns a vector of fitted values.

fnc.delete.map.XYZ *This function is used to delete values outside the state of New York*

Description

This function is used to delete values outside the state of New York

Usage

```
fnc.delete.map.XYZ(xyz)
```

Arguments

xyz	A list containing the x values, y values and interpolated z values at each x and y pair.

Value

Returns the input but with NA placed in z values corresponding to the locations whose x-y coordinates are outside the land boundary of the USA.

get_parameter_estimates *Obtains parameter estimates from MCMC samples*

Description

Obtains parameter estimates from MCMC samples

Usage

```
get_parameter_estimates(samps, level = 95)
```

Arguments

samps	A matrix of N by p samples for the p parameters
level	Desired confidence level - defaults to 95%.

Value

A data frame containing four columns: mean, sd, low (er limit), and up (per limit) for the p parameters.

Examples

```
samps <- matrix(rnorm(10000), ncol= 10 )
dim(samps)
a <- get_parameter_estimates(samps)
a
b <- get_parameter_estimates(samps, level=98)
b
```

get_validation_summaries *Obtains suitable validation summary statistics from MCMC samples obtained for validation.*

Description

Obtains suitable validation summary statistics from MCMC samples obtained for validation.

Usage

```
get_validation_summaries(samps, level = 95)
```

Arguments

samps	A matrix of N by p samples for the p parameters
level	Desired confidence level - defaults to 95%.

Value

A data frame containing five columns: meanpred, medianpred, sdpred, low (er limit), and up (per limit) for the p parameters.

Examples

```
set.seed(4)
vrows <- sample(nrow(nysptime), 100)
M1 <- Bsptime(model="lm", formula=y8hrmax~xmaxtemp+xwdsp+xrh, data=nysptime,
validrows=vrows, scale.transform = "SQRT")
samps<- M1$valpreds
valsums <- get_validation_summaries(samps)
head(valsums)
```

---

getvalidrows *Returns a vector of row numbers for validation.*

Description

Returns a vector of row numbers for validation.

Usage

```
getvalidrows(sn, tn, valids, validt = NULL, allt = FALSE)
```

Arguments

sn	The total number of spatial locations.
tn	The total number of time points in each location.
valids	A vector of site numbers in (1:sn) to be used for validation.
validt	A vector of time points in (1:tn) to be used for validation.
allt	Whether all the time points should be used for validation.

Value

Integer vector providing the row numbers of the data frame for validation. Output of this function is suitable as the argument validrows for the bmstdr model fitting functions Bsptime, Bcartime.

Examples

```
{
# To validate at site numbers 1, 5, and 10 at 31 randomly selected
# time points for the nysptime data set we issue the following commands
set.seed(44)
vt <- sample(62, 31)
vrows <- getvalidrows(sn=28, tn=62, valids=c(1, 5, 10), validt=vt)
# To validate at sites 1 and 2 at all time points
vrows <- getvalidrows(sn=28, tn=62, valids=c(1, 2), allt=TRUE)
}
```

---

gridnyspatial *Values of three covariates for 100 grid locations in New York averaged over the 62 days during the months of July and August, 2006.*

Description

Values of three covariates for 100 grid locations in New York averaged over the 62 days during the months of July and August, 2006.

Usage

```
gridnyspatial
```

Format

An object of class `data.frame` with 100 rows and 8 columns.

Source

See the NYgrid data set in spTimer package, Bakar and Sahu (2015). Each data row is the mean of the available covariate values at the 100 grid locations during the months of July and August in 2006.

@format A data frame with 100 rows and 8 columns:

s.index site index (1 to 28)

Longitude Longitude of the site

Latitude Latitude of the site

utmx UTM X-coordinate of the site

utmy UTM Y-coordinate of the site

xmaxtemp Average maximum temperature (degree Celsius) at the site over 62 days in July and August, 2006

xwdsp Average wind speed (nautical mile per hour) over 62 days in July and August, 2006

xwdsp Average relative humidity over 62 days in July and August, 2006

References

Bakar KS, Sahu SK (2015). "spTimer: Spatio-Temporal Bayesian Modeling Using R." *Journal of Statistical Software, Articles*, **63**(15), 1–32. ISSN 1548-7660, doi:10.18637/jss.v063.i15, https://www.jstatsoft.org/v063/i15.

Examples

```
summary(gridnyspatial)
```

`gridnysptime` *Values of three covariates for 100 grid locations in New York for the 62 days during the months of July and August, 2006.*

Description

Values of three covariates for 100 grid locations in New York for the 62 days during the months of July and August, 2006.

Values of three covariates for 100 grid locations in New York for the 62 days during the months of July and August, 2006.

Usage

```
gridnysptime

gridnysptime
```

Format

An object of class `data.frame` with 6200 rows and 11 columns.

An object of class `data.frame` with 6200 rows and 11 columns.

Source

It is the same data set as NYgrid data set in the spTimer package, Bakar and Sahu (2015). with two added columns providing the UTM X- and Y- coordinates. Each data row is for a particular grid site and day as detailed below.

@format A data frame with 6200 rows and 11 columns:

s.index site index (1 to 100)

Longitude Longitude of the site

Latitude Latitude of the site

utmx UTM X-coordinate of the site

utmy UTM Y-coordinate of the site

Year This is 2006 for all the rows

Month Month taking values 7 for July and 8 for August

Day Day taking values 1 to 31

xmaxtemp Maximum temperature (degree Celsius)

xwdsp wind speed (nautical mile per hour)

xrh Relative humidity

It is the same data set as NYgrid data set in the spTimer package,

with two added columns providing the UTM X- and Y- coordinates. Each data row is for a particular grid site and day as detailed below.

@format A data frame with 6200 rows and 11 columns:

s.index site index (1 to 100)

Longitude Longitude of the site

Latitude Latitude of the site

utmx UTM X-coordinate of the site

utmy UTM Y-coordinate of the site

Year This is 2006 for all the rows

Month Month taking values 7 for July and 8 for August

Day Day taking values 1 to 31

xmaxtemp Maximum temperature (degree Celsius)

xwdsp wind speed (nautical mile per hour)

xrh Relative humidity

References

Bakar KS, Sahu SK (2015). "spTimer: Spatio-Temporal Bayesian Modeling Using R." *Journal of Statistical Software, Articles*, **63**(15), 1–32. ISSN 1548-7660, doi:10.18637/jss.v063.i15, https://www.jstatsoft.org/v063/i15.

Bakar KS, Sahu SK (2015). "spTimer: Spatio-Temporal Bayesian Modeling Using R." *Journal of Statistical Software, Articles*, **63**(15), 1–32. ISSN 1548-7660, doi:10.18637/jss.v063.i15, https://www.jstatsoft.org/v063/i15.

Examples

```
head(gridnysptime)
summary(gridnysptime[, 9:11])
summary(gridnysptime[, 9:11])
```

hitandfalsealarm *Calculate the hit and false alarm rates*

Description

Calculate the hit and false alarm rates

Usage

```
hitandfalsealarm(thresh, yobs, ypred)
```

Arguments

thresh	Threshold value
yobs	A vector of observations, may include missing values
ypred	Predictions

Value

A list containing the calculated hit and false alarm rates

is.bmstdr *Is it a bmstdr model fitted object?*

Description

Is it a bmstdr model fitted object?

Usage

```
is.bmstdr(x)
```

Arguments

x	Any R object.

Value

A TRUE/FALSE logical output.

maternncov *Banerjee, Carlin and Gelfand (2015) Mat'ern covariance function*

Description

Banerjee, Carlin and Gelfand (2015) Mat'ern covariance function

Usage

```
maternncov(d, phi, nu)
```

Arguments

d	is the distance
phi	is the rate of decay
nu	is the smoothness parameter

Value

Returns the Mat'ern covariance for distance object d

maternfun *Banerjee et al Mat'ern covariance function*

Description

Banerjee et al Mat'ern covariance function

Usage

```
maternfun(d, phi, nu)
```

Arguments

d	is the distance
phi	is the rate of decay
nu	is the smoothness parameter

Value

Returns the Mat'ern correlation function for distance object d

`nyspatial` *Average ozone concentration values and three covariates from 28 sites in New York.*

Description

Average ozone concentration values and three covariates from 28 sites in New York.

Usage

```
nyspatial
```

Format

An object of class `data.frame` with 28 rows and 9 columns.

Source

See the NYdata set in spTimer package, Bakar and Sahu (2015). Each data row is the mean of the available daily 8-hour maximum average ozone concentrations in parts per billion (ppb) at each of the 28 sites. The daily values are for the months of July and August in 2006. @format A data frame with 28 rows and 9 columns:

s.index site index (1 to 28)

Longitude Longitude of the site

Latitude Latitude of the site

utmx UTM X-coordinate of the site

utmy UTM Y-coordinate of the site

yo3 Average ozone concentration value (ppb) at the site over 62 days in July and August, 2006

xmaxtemp Average maximum temperature (degree Celsius) at the site over 62 days in July and August, 2006

xwdsp Average wind speed (nautical mile per hour) over 62 days in July and August, 2006

xrh Average relative humidity over 62 days in July and August, 2006

References

Bakar KS, Sahu SK (2015). "spTimer: Spatio-Temporal Bayesian Modeling Using R." *Journal of Statistical Software, Articles*, **63**(15), 1–32. ISSN 1548-7660, doi:10.18637/jss.v063.i15, https://www.jstatsoft.org/v063/i15.

Examples

```
head(nyspatial)
summary(nyspatial)
pairs(nyspatial[, 6:9])
```

`nysptime` *Daily 8-hour maximum ozone concentration values and three covariates from 28 sites in New York for the 62 days during the months of July and August, 2006.*

Description

Daily 8-hour maximum ozone concentration values and three covariates from 28 sites in New York for the 62 days during the months of July and August, 2006.

Usage

```
nysptime
```

Format

An object of class `data.frame` with 1736 rows and 12 columns.

Source

It is the same as the NYdata set in the spTimer package, Bakar and Sahu (2015), with two added columns providing the UTM X- and Y- coordinates. Each data row is for a particular site and a day as detailed below.

@format A data frame with 1736 rows and 12 columns:

s.index site index (1 to 28)

Longitude Longitude of the site

Latitude Latitude of the site

utmx UTM X-coordinate of the site

utmy UTM Y-coordinate of the site

Year This is 2006 for all the rows

Month Month taking values 7 for July and 8 for August

Day Day taking values 1 to 31

y8hrmax Daily 8-hour maximum ozone concentration value

xmaxtemp Maximum temperature (degree Celsius)

xwdsp wind speed (nautical mile per hour)

xrh Relative humidity

References

Bakar KS, Sahu SK (2015). "spTimer: Spatio-Temporal Bayesian Modeling Using R." *Journal of Statistical Software, Articles*, **63**(15), 1–32. ISSN 1548-7660, doi:10.18637/jss.v063.i15, https://www.jstatsoft.org/v063/i15.

Examples

```
head(nysptime)
summary(nysptime[, 9:12])
```

---

obs_v_pred_plot *Observed against predicted plot*

Description

Observed against predicted plot

Usage

```
obs_v_pred_plot(
  yobs,
  predsums,
  segments = TRUE,
  summarystat = "median",
  plotit = TRUE
)
```

Arguments

yobs	A vector containing the actual observations
predsums	A data frame containing predictive summary statistics with the same number of rows as the length of the vector yobs. The data frame must have columns named as meanpred, medianpred, sd, low and up. Ideally this argument should be the output of the command get_validation_summaries.
segments	Logical: whether to draw line segments for the prediction intervals.
summarystat	Can take one of two values "median" (default) or "mean" indicating which one to use for the plot.
plotit	Logical scalar value: whether to plot the predictions against the observed values.

Value

Draws a plot only after removing the missing observations. It also returns a list of two ggplot2 objects: (i) a plot with intervals drawn pwithseg and (ii) a plot without the segments drawn: pwithoutseg and (iii) a simple plot not showing the range of the prediction intervals.

Examples

```
set.seed(4)
vrows <- sample(nrow(nysptime), 100)
M1 <- Bsptime(model="lm", formula=y8hrmax~xmaxtemp+xwdsp+xrh, data=nysptime,
validrows=vrows, scale.transform = "SQRT")
psums <-  get_validation_summaries(M1$valpreds)
```

```
oplots <- obs_v_pred_plot(yobs=M1$yobs_preds$y8hrmax, predsum=psums)
names(oplots)
plot(oplots$pwithoutseg)
plot(oplots$pwithseg)
```

`phichoice_sp` *Grid search method for choosing phi Calculates the validation statistics using the spatial model with a given range of values of the decay parameter phi.*

Description

Grid search method for choosing phi Calculates the validation statistics using the spatial model with a given range of values of the decay parameter phi.

Usage

```
phichoice_sp(
  formula,
  data,
  coordtype,
  coords,
  phis,
  scale.transform,
  s,
  N,
  burn.in,
  verbose = TRUE,
  ...
)
```

Arguments

`formula`	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	The data frame for which the model formula is to be fitted. If a spatial model is to be fitted then the data frame should contain two columns containing the locations of the coordinates. See the coords argument below.
`coordtype`	Type of coordinates: utm, lonlat or plain with utm (supplied in meters) as the default. Distance will be calculated in units of kilometer if this argument is either utm or lonlat. Euclidean distance will be calculated if this is given as the third type plain. If distance in meter is to be calculated then coordtype should be passed on as plain although the coords are supplied in UTM.
`coords`	A vector of size two identifying the two column numbers of the data frame to take as coordinates. Or this can be given as a matrix of number of sites by 2 providing the coordinates of all the data locations.

phis	A vector values of phi
scale.transform	Transformation of the response variable. It can take three values: SQRT, LOG or NONE.
s	A vector giving the validation sites
N	MCMC sample size. Default value 5000.
burn.in	How many initial iterations to discard. Default value 1000. Only relevant for MCMC based model fitting, i.e., when package is spBayes or Stan.
verbose	Logical. Should it print progress?
...	Any additional parameter that may be passed to Bspatial

Value

A data frame giving the phi values and the corresponding validation statistics

Examples

```
a <- phichoice_sp(formula=yo3~xmaxtemp+xwdsp+xrh, data=nyspatial,
coordtype="utm", coords=4:5,
phis=seq(from=0.1, to=1, by=0.4), scale.transform="NONE",
s=c(8,11,12,14,18,21,24,28), N=20, burn.in=10, verbose=TRUE)
```

---

phichoicep *Calculates the validation statistics using the spatial model with a given range of values of the decay parameter phi.*

Description

Calculates the validation statistics using the spatial model with a given range of values of the decay parameter phi.

Usage

```
phichoicep(
  formula,
  data,
  coordtype,
  coords,
  scale.transform,
  phis,
  phit,
  valids,
  N,
  burn.in,
  verbose = TRUE
)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	The data frame for which the model formula is to be fitted. If a spatial model is to be fitted then the data frame should contain two columns containing the locations of the coordinates. See the coords argument below.
coordtype	Type of coordinates: utm, lonlat or plain with utm (supplied in meters) as the default. Distance will be calculated in units of kilometer if this argument is either utm or lonlat. Euclidean distance will be calculated if this is given as the third type plain. If distance in meter is to be calculated then coordtype should be passed on as plain although the coords are supplied in UTM.
coords	A vector of size two identifying the two column numbers of the data frame to take as coordinates. Or this can be given as a matrix of number of sites by 2 providing the coordinates of all the data locations.
scale.transform	Transformation of the response variable. It can take three values: SQRT, LOG or NONE.
phis	A vector values of phi for spatial decay
phit	A vector values of phi for temporal decay
valids	A vector giving the validation sites
N	MCMC sample size.
burn.in	How many initial iterations to discard. Only relevant for MCMC based model fitting, i.e., when package is spBayes or Stan.
verbose	Logical. Should progress be printed?

Value

A data frame giving the phi values and the corresponding validation statistics

Examples

```
a <-  phichoicep(formula=y8hrmax ~ xmaxtemp+xwdsp+xrh, data=nysptime,
coordtype="utm", coords=4:5, scale.transform = "SQRT",
phis=c(0.001,  0.005, 0.025, 0.125, 0.625), phit=c(0.05, 0.25, 1.25, 6.25),
valids=c(8,11,12,14,18,21,24,28), N=20, burn.in=10, verbose=TRUE)
```

---

plot.bmstdr *Plot method for bmstdr objects.*

Description

Plot method for bmstdr objects.

Usage

```
## S3 method for class 'bmstdr'
plot(x, segments = TRUE, ...)
```

Arguments

x	A bmstdr model fit object.
segments	TRUE or FALSE. It decides whether to draw the prediction intervals as line segments.
...	Any other additional arguments.

Value

It plots the observed values on the original scale against the predictions and the 95% prediction intervals if validation has been performed. It then plots the residuals against fitted values. It then applies plotting method to the model fitted object as returned by the chosen named package. There is no return value.

---

print.bmstdr *Provides basic information regarding the fitted model.*

Description

Provides basic information regarding the fitted model.

Usage

```
## S3 method for class 'bmstdr'
print(x, digits = 3, ...)
```

Arguments

x	A bmstdr model fit object.
digits	How many significant digits after the decimal to print, defaults to 3.
...	Any other additional arguments

Value

No return value, called for side effects.

resid.bmstdr *Extract residuals from a bmstdr fitted object.*

Description

Extract residuals from a bmstdr fitted object.

Usage

```
## S3 method for class 'bmstdr'
resid(object, ...)
```

Arguments

object	A bmstdr model fit object.
...	Any other additional arguments.

residuals.bmstdr *Extract residuals from a bmstdr fitted object.*

Description

Extract residuals from a bmstdr fitted object.

Usage

```
## S3 method for class 'bmstdr'
residuals(object, numbers = NULL, ...)
```

Arguments

object	A bmstdr model fit object.
numbers	a list with two components: sn=number of spatial locations tn=number of time points. Residuals will be assumed to follow the arrangement of the data frame - sorted by space and then time within space.
...	Any other additional arguments.

Value

Returns a vector of residuals. If appropriate, it draws a time series plot of residuals. Otherwise, it draws a plot of residuals against observation numbers.

summary.bmstdr *Provides basic summaries of model fitting.*

Description

Provides basic summaries of model fitting.

Usage

```
## S3 method for class 'bmstdr'
summary(object, digits = 3, ...)
```

Arguments

object	A bmstdr model fit object.
digits	How many significant digits after the decimal to print, defaults to 3.
...	Any other additional arguments

Value

No return value, called for side effects.

terms.bmstdr *Prints the terms*

Description

Prints the terms

Usage

```
## S3 method for class 'bmstdr'
terms(x, ...)
```

Arguments

x	A bmstdr model fit object.
...	Any other additional arguments

Value

Terms in the model formula

Weng — *A 313 by 313 proximity matrix for the 313 LADCUAS in England. Each entry is either 0 or 1 and is 1 if the corresponding row and column LADCUAs share a common boundary.*

Description

A 313 by 313 proximity matrix for the 313 LADCUAS in England. Each entry is either 0 or 1 and is 1 if the corresponding row and column LADCUAs share a common boundary.

Usage

```
Weng
```

Format

An object of class `matrix` (inherits from `array`) with 313 rows and 313 columns.

Source

Sahu and Böhning (2021).

References

Sahu SK, Böhning D (2021). "Bayesian spatio-temporal joint disease mapping of Covid-19 cases and deaths in local authorities of England." *Spatial Statistics*. doi:10.1016/j.spasta.2021.100519.

Examples

```
dim(Weng)
summary(apply(Weng, 1, sum))
```

ydata — *Average air pollution values from 28 sites in New York.*

Description

Average air pollution values from 28 sites in New York.

Usage

```
ydata
```

```
ydata
```

Format

A vector with 28 real values.

An object of class `numeric` of length 28.

Source

This is obtained by calculating site-wise averages of the NYdata set in the 'spTimer' package Bakar and Sahu (2015). Each data point is the mean of the available daily 8-hour maximum average ozone concentrations in parts per billion (ppb) at each of the 28 sites. The daily values are for the month of July and August in 2006.

References

Bakar KS, Sahu SK (2015). "spTimer: Spatio-Temporal Bayesian Modeling Using R." *Journal of Statistical Software, Articles*, **63**(15), 1–32. ISSN 1548-7660, doi:10.18637/jss.v063.i15, https://www.jstatsoft.org/v063/i15.

Examples

```
summary(ydata)
```

Index

∗ **datasets**
  argo_floats_atlantic_2003, 3
  colpalette, 42
  engdeaths, 42
  engtotals, 44
  gridnyspatial, 49
  gridnysptime, 50
  nyspatial, 54
  nysptime, 55
  Weng, 63
  ydata, 63

argo_floats_atlantic_2003, 3

Bcartime, 4
BCauchy, 16
Bmchoice, 17
Bmoving_sptime, 18
bmstdr_variogram, 22
Bnormal, 23
Bspatial, 25
Bsptime, *21*, *29*, 31

calculate_dic, 39
calculate_validation_statistics, 40
calculate_waic, 41
coef.bmstdr, 41
colpalette, 42

engdeaths, 42
engtotals, 44

fig11.13.plot, 45
fitted.bmstdr, 46
fnc.delete.map.XYZ, 46

get_parameter_estimates, 47
get_validation_summaries, 47, *56*
getvalidrows, 48
gridnyspatial, 49
gridnysptime, 50

hitandfalsealarm, 52

is.bmstdr, 52

materncov, 53
maternfun, 53

nyspatial, 54
nysptime, 55

obs_v_pred_plot, *9*, *21*, *28*, *36*, 56

phichoice_sp, 57
phichoicep, 58
plot.bmstdr, 59
print.bmstdr, 60

resid.bmstdr, 61
residuals.bmstdr, 61

summary.bmstdr, 62

terms.bmstdr, 62

Weng, 63

ydata, 63