Package: idbrms (via r-universe)

July 26, 2024

Title Infectious Disease Modelling Using brms

Version 0.0.0.9000

Description Infectious disease modelling using brms.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports brms, rstan, data.table, purrr

Suggests testthat, covr, spelling

Language en-US

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

RemoteUrl https://github.com/epiforecasts/idbrms

RemoteRef HEAD

RemoteSha f5a1c25fdedd95d5c6eb5d955df2d041687e534f

Contents

`expose_idbrms_stan_fns` *Expose package stan functions in R*

Description

This function exposes internal stan functions in R from a user supplied list of target files. Allows for testing of stan functions in R and potentially user use in R code.

Usage

```
expose_idbrms_stan_fns(files, dir, ...)
```

Arguments

`files`	A character vector indicating the target files
`dir`	A character string indicating the directory for the file
`...`	Additional arguments passed to `rstan::expose_stan_functions`.

`idbrm` *Interface for infectious disease modelling using brms.*

Description

Interface for infectious disease modelling using brms.

Usage

```
idbrm(data, formula, family, priors, custom_stancode, dry = FALSE, ...)
```

Arguments

`data`	A data frame as prepared for modelling using `prepare` with a class associated with the model prepared for.
`formula`	A formula as defined using `id_formula` or as supported by `brms::brm`.
`family`	A observation model family as defined in `brms`.
`priors`	A list of priors as defined using `brms` or `id_priors`. Defaults to the the `id_priors` defined for the model class being fit.
`custom_stancode`	A list of `stanvars` used to define custom stancode in `brms`. By default uses the code designed for the model class being fit (as specified using `id_stancode`).
`dry`	Logical, defaults to TRUE. For testing purposes should just the `stan` code be output with not fitting done.
`...`	Additional arguments to pass to `brms::brm`.

Author(s)

Sam Abbott

Examples

```
# define some example data
library(data.table)
dt <- data.table(
   region = "France", cases = seq(10, 500, by = 10),
   date = seq(as.Date("2020-10-01"), by = "days", length.out = 50)
   )
dt[, deaths := as.integer(shift(cases, 5) * 0.1)]
dt[is.na(deaths), deaths := 0]

dt <- prepare(
  dt, model = "convolution", location = "region",
  primary = "cases", secondary = "deaths",
  )

# fit the convolution model using a Poisson observation model
fit <- idbrm(data = dt, family = poisson(link = "identity"))
```

`idbrm.idbrms_convolution` *Delay Convolution Model*

Description

A model that assumes that a secondary observations can be predicted using a convolution of a primary observation multipled by some scaling factor. An example use case of this model is to estimate the case fatality rate (with the primary observation being cases and the secondary observation being deaths) and then explore factors that influence it.

Usage

```
## S3 method for class 'idbrms_convolution'
idbrm(
  data,
  formula = id_formula(data),
  family = negbinomial(link = "identity"),
  priors = id_priors(data),
  custom_stancode = id_stancode(data),
  dry = FALSE,
  ...
)
```

Arguments

`data`	A data.frame as produced by `prepare` that must contain the date, location (as loc), primary (the data that the outcome is a convolution of) and secondary (the observation of interest. Should have class "idbrms_convolution".
`formula`	A formula as defined using `id_formula` or as supported by `brms::brm`.
`family`	A observation model family as defined in `brms`.
`priors`	A list of priors as defined using `brms` or `id_priors`. Defaults to the the `id_priors` defined for the model class being fit.
`custom_stancode`	A list of `stanvars` used to define custom stancode in `brms`. By default uses the code designed for the model class being fit (as specified using `id_stancode`).
`dry`	Logical, defaults to TRUE. For testing purposes should just the `stan` code be output with not fitting done.
`...`	Additional parameters passed to `brms::brm`.

Value

A "brmsfit" object or stan code (if `dry = TRUE`).

Author(s)

Sam Abbott

Examples

```
# define some example data
library(data.table)
dt <- data.table(
   region = "France", cases = seq(10, 500, by = 10),
   date = seq(as.Date("2020-10-01"), by = "days", length.out = 50)
   )
dt[, deaths := as.integer(shift(cases, 5) * 0.1)]
dt[is.na(deaths), deaths := 0]

dt <- prepare(
  dt, model = "convolution", location = "region",
  primary = "cases", secondary = "deaths",
  )

# fit the convolution model using a Poisson observation model
fit <- idbrm(data = dt, family = poisson(link = "identity"))
```

`idbrmfit` *Infectious disease modelling wrapper for brm*

Description

Infectious disease modelling wrapper for brm

Usage

```
idbrmfit(formula, data, family, priors, custom_stancode, dry = FALSE, ...)
```

Arguments

`formula`	A `brms` model formula.
`data`	A data frame.
`family`	An observation model family as defined in `brms`.
`priors`	A list of priors as defined using `brms`.
`custom_stancode`	A list of `stanvars` used to define custom stancode in `brms`.
`dry`	Logical, defaults to TRUE. For testing purposes should just the `stan` code be output with not fitting done.
`...`	Additional arguments to pass to `brms::brm`.

Author(s)

Sam Abbott

`idbrms_stan_chunk` *Read in a idbrms Stan code chunk*

Description

Read in a idbrms Stan code chunk

Usage

```
idbrms_stan_chunk(path)
```

Arguments

`path`	The path within the "stan" folder of the installed idbrms package to the stan code chunk of interest.

Value

A character string containing the stan code chunk of interest.

Examples

```
idbrms_stan_chunk("functions/idbrms_convolve.stan")
```

`idbrms_version_stanvar`

Label a idbrms stan model with a version indicator

Description

Label a idbrms stan model with a version indicator

Usage

```
idbrms_version_stanvar()
```

Value

A brms stanvar chunk containing the package version used to build the stan code.

`id_formula.idbrms_convolution`

Define a formula for the convolution model

Description

Define a formula for the convolution model

Define a model specific formula

Usage

```
## S3 method for class 'idbrms_convolution'
id_formula(data, scale = ~1, cmean = ~1, lcsd = ~1, ...)

id_formula(data, ...)
```

Arguments

`data`	A data.frame as produced by `prepare` that must contain the date, location (as loc), primary (the data that the outcome is a convolution of) and secondary (the observation of interest. Should have class "idbrms_convolution".
`scale`	Formula for the scaling of primary observations to secondary observations.
`cmean`	Formula for the convolution mean. Defaults to intercept only.
`lcsd`	Formula for the logged convolution standard deviation. Defaults to intercept only.
`...`	Additional arguments for method.

Author(s)

Sam Abbott

Sam Abbott

`id_priors` *Define model specific priors*

Description

Define model specific priors

Usage

```
id_priors(data, ...)
```

Arguments

`data`	A data frame as prepared for modelling using `prepare` with a class associated with the model prepared for.
`...`	Additional arguments for method.

Author(s)

Sam Abbott

`id_priors.idbrms_convolution` *Define priors for the delay convolution model*

Description

Define priors for the delay convolution model

Usage

```
## S3 method for class 'idbrms_convolution'
id_priors(
  data,
  scale = c(round(log(0.1), 2), 1),
  cmean = c(2, 1),
  lcsd = c(-0.5, 0.25),
  ...
)
```

Arguments

`data`	A data.frame as produced by `prepare` that must contain the date, location (as loc), primary (the data that the outcome is a convolution of) and secondary (the observation of interest. Should have class "idbrms_convolution".
`scale`	Vector of length two defining the mean and the standard deviation of the normal prior for the scaling factor.
`cmean`	Vector of length two defining the mean and standard deviation of the log mean of the delay distribution.
`lcsd`	Vector of length two defining the mean and standard deviation of the log standard deviation logged. the standard deviation to be greater than 0 on the unconstrained scale.
`...`	Additional arguments for method.

Author(s)

Sam Abbott

`id_stancode` *Define model specific stancode*

Description

Define model specific stancode

Usage

```
id_stancode(data, ...)
```

Arguments

`data`	A data frame as prepared for modelling using `prepare` with a class associated with the model prepared for.
`...`	Additional arguments for method.

Author(s)

Sam Abbott

`id_stancode.idbrms_convolution` *Define stan code for a delay convolution model*

Description

Define stan code for a delay convolution model

Usage

```
## S3 method for class 'idbrms_convolution'
id_stancode(data, ...)
```

Arguments

`data`	A data.frame as produced by `prepare` that must contain the date, location (as loc), primary (the data that the outcome is a convolution of) and secondary (the observation of interest. Should have class "idbrms_convolution".
`...`	Additional arguments for method.

Author(s)

Sam Abbott

`prepare` *Prepare data for modelling with idbrm*

Description

Prepare data for modelling with idbrm

Default method used when preparing data

Usage

```
prepare(data, ...)

## Default S3 method:
prepare(data, model, ...)
```

Arguments

`data`	A dataframe to be used for modelling
`...`	Additional arguments passed to model specific prepare functions
`model`	Character string, model type to prepare to use. Supported options are "convolution".

Author(s)

Sam Abbott

Sam Abbott

Examples

```
# define some example data
library(data.table)
dt <- data.table(
   region = "France", cases = seq(10, 500, by = 10),
   date = seq(as.Date("2020-10-01"), by = "days", length.out = 50)
   )
dt <- dt[, deaths := as.integer(shift(cases, 5) * 0.1)]
dt <- dt[is.na(deaths), deaths := 0]

dt <- prepare(
  dt, model = "convolution", location = "region",
  primary = "cases", secondary = "deaths",
  )
dt[]
```

`prepare.idbrms_convolution` *Prepare data for fitting using a convolution model*

Description

Prepare data for fitting using a convolution model

Usage

```
## S3 method for class 'idbrms_convolution'
prepare(
  data,
  location,
  primary,
  secondary,
  initial_obs = 14,
  max_convolution = 30,
  ...
)
```

Arguments

`data`	A data frame containing at least two integer observations and a date variable.
`location`	Character string, variable to use as the spatial location.
`primary`	Character string, variable to use as the primary observation.
`secondary`	Character string, variable to use as the secondary observation.
`initial_obs`	Integer, number of observations to hold out from the likelihood. This is useful as initially the outcome will depend on primary data outside of the range of the training set and including this could bias the estimated delay distribution. Defaults to 14 days.
`max_convolution`	Integer defining the maximum index to use for the convolution. Defaults to 30 days.
`...`	Additional arguments passed to model specific prepare functions

Author(s)

Sam Abbott

Examples

```
# define some example data
library(data.table)
dt <- data.table(
   region = "France", cases = seq(10, 500, by = 10),
```

```
    date = seq(as.Date("2020-10-01"), by = "days", length.out = 50)
    )
dt <- dt[, deaths := as.integer(shift(cases, 5) * 0.1)]
dt <- dt[is.na(deaths), deaths := 0]

dt <- prepare(
  dt, model = "convolution", location = "region",
  primary = "cases", secondary = "deaths",
  )
dt[]
```

Index