

Package ‘epidemia’

October 25, 2021

Title Modeling of Epidemics using Hierarchical Bayesian Models

Version 1.0.0

Description Flexibly specify and fit Bayesian statistical models for epidemics. ‘epidemia’ leverages R’s formula interface so that users can parameterize reproduction numbers and ascertainment rates in terms of predictors. Infections are propagated over time using self-exciting point processes. Multiple regions can be modeled simultaneously with multilevel models. The models and framework behind the package are described in Bhatt et al. (2021) <[arXiv:2012.00394](https://arxiv.org/abs/2012.00394)>. The design of the package has been inspired by, and has borrowed from, ‘rstanarm’ (Goodrich et al., 2020) <<https://mc-stan.org/rstanarm/>>. ‘rstan’ (Stan Development Team, 2020) <<https://mc-stan.org/>> is used as the back end for fitting models.

License GPL-3

URL <https://github.com/ImperialCollegeLondon/epidemia/>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Biarch true

Depends R (>= 3.5.0)

Imports dplyr (>= 0.8.5), methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.0.0), magrittr, rstanarm, lme4, ggplot2, Matrix, scales, Rdpack, stats, zoo, tidyr, rlang, bayesplot, hrbrthemes

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

SystemRequirements GNU make

Suggests testthat, knitr, gridExtra, bookdown, EpiEstim, lubridate, kableExtra, extrafont

RdMacros Rdpack

NeedsCompilation yes

Author James Scott [aut, cre] (<<https://orcid.org/0000-0003-0106-5618>>),
 Axel Gandy [aut] (<<https://orcid.org/0000-0002-6777-0451>>),
 Swapnil Mishra [aut] (<<https://orcid.org/0000-0002-8759-5902>>),
 Juliette Unwin [aut] (<<https://orcid.org/0000-0002-9120-4003>>),
 Seth Flaxman [aut] (<<https://orcid.org/0000-0002-2477-4217>>),
 Samir Bhatt [aut] (<<https://orcid.org/0000-0002-0891-4611>>),
 Jonathan Ish-Horowicz [ctb] (<<https://orcid.org/0000-0002-5386-6620>>),
 Leon Wu [ctb],
 Ben Goodrich [ctb],
 Jonah Gabry [ctb],
 Trustees of Columbia University [cph] (R/helpers.R),
 Paul-Christian Burkner [cph] (R/misc.R),
 Triad Sou [cph] (R/geom_stepribbon.R),
 Kengo Nagashima [cph] (R/geom_stepribbon.R)

Maintainer James Scott <james.scott15@imperial.ac.uk>

Repository CRAN

Date/Publication 2021-10-25 07:50:05 UTC

R topics documented:

epidemia-package	3
all_obs_types	4
as.matrix.epimodel	4
EnglandNewCases	6
epiinf	6
epim	8
epimodel-objects	10
epiobs	11
epirt	12
EuropeCovid	15
EuropeCovid2	16
evaluate_forecast	16
formula.epimodel	17
get_samps	18
get_x	19
hexp	19
model.frame.epimodel	20
ngrps.mixed	20
pairs.epimodel	21
plot.epimodel	22
plot_coverage	23
plot_infections	24
plot_infectious	27
plot_metrics	29
plot_obs	30
plot_rt	33
posterior_coverage	35

posterior_infections	36
posterior_infectious	37
posterior_latent	37
posterior_linpred	38
posterior_metrics	39
posterior_predict.epimodel	40
posterior_rt	41
posterior_sample_size	42
posterior_sample_size.epimodel	42
print.epimodel	43
print.prior_summary.epimodel	44
print.prior_summary_reg.epimodel	44
prior_summary.epimodel	45
rw	46
scaled_logit	47
shifted_gamma	47
summary.epimodel	48
terms.epimodel	49
terms_rw	49

Index	50
--------------	-----------

epidemia-package	<i>Flexible Epidemic Modeling with epidemia</i>
------------------	---

Description

The **epidemia** package allows researchers to flexibly specify and fit Bayesian epidemiological models in the style of Flaxman et al. (2020). The package leverages R’s formula interface to parameterize the reproduction rate in terms of covariates, and allows pooling of parameters. The design of the package has been inspired by, and borrowed from, the **rstanarm** package (Goodrich et al. 2020). **epidemia** uses **rstan** (Stan Development Team 2020) as the backend for fitting the models. The primary model fitting function in **epidemia** is **epim**.

References

Flaxman S, Mishra S, Gandy A, Unwin HJT, Mellan TA, Coupland H, Whittaker C, Zhu H, Berrah T, Eaton JW, Monod M, Perez-Guzman PN, Schmit N, Cilloni L, Ainslie KEC, Baguelin M, Boonyasiri A, Boyd O, Cattarino L, Cooper LV, Cucunubá Z, Cuomo-Dannenburg G, Dighe A, Djaafara B, Dorigatti I, van Elsland SL, FitzJohn RG, Gaythorpe KAM, Geidelberg L, Grassly NC, Green WD, Hallett T, Hamlet A, Hinsley W, Jeffrey B, Knock E, Laydon DJ, Nedjati-Gilani G, Nouvellet P, Parag KV, Siveroni I, Thompson HA, Verity R, Volz E, Walters CE, Wang H, Wang Y, Watson OJ, Winskill P, Xi X, Walker PGT, Ghani AC, Donnelly CA, Riley SM, Vollmer MAC, Ferguson NM, Okell LC, Bhatt S, Team ICCR (2020). “Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe.” *Nature*. ISSN 1476-4687, doi: [10.1038/s4158602024057](https://doi.org/10.1038/s4158602024057).

Goodrich B, Gabry J, Ali I, Brilleman S (2020). “rstanarm: Bayesian applied regression modeling via Stan.” <https://mc-stan.org/rstanarm/>.

Stan Development Team (2020). "RStan: the R interface to Stan." [https://mc-stan.org/.](https://mc-stan.org/)(.)

all_obs_types *Get a list of all observation types used in a model*

Description

Get a list of all observation types used in a model

Usage

```
all_obs_types(object)
```

Arguments

object An object of class epimodel.

Value

Character vector giving names of all observation types

as.matrix.epimodel *Extract posterior samples*

Description

Get parameter samples from a fitted model object of class epimodel.

Usage

```
## S3 method for class 'epimodel'
as.matrix(
  x,
  ...,
  pars = NULL,
  regex_pars = NULL,
  par_models = NULL,
  par_types = NULL,
  par_groups = NULL
)
```

```
## S3 method for class 'epimodel'
as.array(
  x,
  ...,
```

```

    pars = NULL,
    regex_pars = NULL,
    par_models = NULL,
    par_types = NULL,
    par_groups = NULL
  )

## S3 method for class 'epimodel'
as.data.frame(
  x,
  ...,
  pars = NULL,
  regex_pars = NULL,
  par_models = NULL,
  par_types = NULL,
  par_groups = NULL
)

```

Arguments

x	A fitted model object returned by epim . See epimodel-objects .
...	Not used.
pars	Character vector of parameter names to return. Same as as.matrix.stanreg
regex_pars	Character vector of regular expressions against which to match parameter names. Same as as.matrix.stanreg
par_models	A character vector that restricts parameters to a subset of model components. For example, "R" only uses parameters in the transmission model, "inf" uses parameters in infection model. Strings giving the name of the response in an observation model (i.e. LHS of the formula in <code>epiobs</code>) can also be used. If NULL (the default), all components are used.
par_types	A character vector that restricts parameters based on their type. The vector can include any of "fixed", "autocor", "random", "aux", "latent", or "seeds". The default is <code>c("fixed", "aux", "seeds")</code> , to avoid printing a very large number of parameters. If NULL, all types are used.
par_groups	A character vector restricting parameters to those used for a subset of regions in which the epidemic is modeled. Defaults to NULL in which case all regions are used.

Details

`as.matrix`, `as.array` and `as.data.frame` each return a sample of parameter draws from objects of class `epimodel`. The returned parameters include those in the regression for R_{tm} , but also other parameters in the model. These methods closely resemble those for `stanreg` objects in **rstanarm**. Please see [as.matrix.stanreg](#) for a general explanation of these methods.

Value

A `matrix`, `array` or `data.frame` object.

EnglandNewCases *Covid-19 Case Counts for England*

Description

Contains case counts of SARS-CoV-2 in England from 30/01/2020 until 30/05/2021. Case counts correspond to 'New Cases by Specimen Date', as defined by Public Health England. The data was downloaded from Public Health England (2020) on 01/06/2021. Case counts in the last few days of May may be underreported as not all cases have been counted as of the download date.

Usage

```
EnglandNewCases
```

Format

A dataframe with three columns, date, region and cases. Each row gives case counts for a given date in England.

References

Public Health England (2020). "Official UK Coronavirus Dashboard." <https://coronavirus.data.gov.uk/details/cases>.

epiinf *Model Latent Infections*

Description

`epiinf` defines a model for latent infections. For the basic version of the model, this defines the generation distribution of the disease, the number of days for which to seed infections, and the prior distribution on the parameter τ , as described in the [model description](#) vignette. Recall that τ is the prior mean on daily seeded infections. These three parameters are controlled by the arguments `gen`, `seed_days` and `prior_seeds` respectively.

Usage

```
epiinf(
  gen,
  seed_days = 6L,
  prior_seeds = hexp(prior_aux = rstanarm::exponential(0.03)),
  latent = FALSE,
  family = "normal",
  prior_aux = rstanarm::normal(10, 5),
  fixed_vtm = 1,
  pop_adjust = FALSE,
```

```

    pops = NULL,
    rm = NULL,
    prior_susc = NULL,
    prior_rm_noise = NULL
  )

```

Arguments

gen	A numeric vector representing the probability mass function for the generation time of the disease (must be a probability vector).
seed_days	An integer giving the number of days for which to seed infections. Defaults to 6L.
prior_seeds	The prior distribution on seeded infections. This may be a call to normal , student_t , exponential , or to hexp . The latter distribution allows hierarchical modeling of seeded infections.
latent	If TRUE, treat infections as latent parameters using the extensions described in Section 5.2 here .
family	Specifies the family for the prior distribution on daily infections. Only used if latent = TRUE, and currently restricted to normal.
prior_aux	Prior distribution for the auxiliary variable in the distribution for latent infections. Only used if latent = TRUE. If fixed_vtm = TRUE, then this refers to the variance-to-mean ratio. If fixed_vtm = FALSE, it is instead the coefficient of variation. Can be a call to exponential , normal , student_t or cauchy . These result in half-normal, half-t and half-cauchy priors.
fixed_vtm	If TRUE, then the prior variance-to-mean ratio for latent infections is fixed, i.e. the auxiliary variable refers to the coefficient of dispersion. If FALSE, then the prior ratio of standard deviation to mean is fixed instead, and the auxiliary variable refers to the coefficient of variation.
pop_adjust	If TRUE, applies a population adjustment to the infection process. Defaults to FALSE.
pops	A character vector giving the name of the column in the dataframe corresponding to the population of each group.
rm	A character vector giving a column name in data (see epim). Each entry should be the proportion of the susceptible population in that group that are removed at that point by some means other than infection; i.e. vaccination. Only used if pop_adjust=TRUE.
prior_susc	Prior distribution on the initial susceptible population at time 0, expressed as a proportion of the total population size. This quantity lies between 0 and 1, and is useful when the first modeled date is after the true beginning of an epidemic. Only used when pop_adjust = TRUE. If unspecified, then the entire population is assumed to be susceptible at time 0. If specified, should be a call to normal .
prior_rm_noise	Removal from the susceptible population (to account for vaccinations) can be applied using the rm argument. However, in practice, it is difficult to specify the proportion of the susceptible class removed at any point in time. prior_rm_noise helps to model noise around this. If specified, should be a call to normal .

Details

`epiinf` has additional arguments which allow the user to extend the basic model. Using `latent=TRUE` replaces the renewal process with a model that treats latent infections as unknown parameters that are sampled along with other parameters. The `family` argument then gives the distribution family for latent infections, while `prior_aux` defines the prior on the coefficient of dispersion d of this distribution.

Recall that one can adjust the infection process to explicitly model changes in infection rates as the remaining susceptible population is depleted. In particular, the adjustment ensures that cumulative infections never breaches the initial susceptible population. The adjustment was described in Section 5.3 of the [model description](#) article. It can be employed by setting `pop_adjust = TRUE` and using the `susceptibles` argument to point towards a variable in the dataframe which gives the susceptible population at each point in time.

Value

An object of class `epiinf`.

Examples

```
data(EuropeCovid)
inf <- epiinf(
  gen = EuropeCovid$si,
  seed_days = 6L,
  prior_seeds = hexp(rstanarm::exponential(0.02))
)
```

epim

Fit a Bayesian epidemiological model with epidemia

Description

`epim` is the only model fitting function in **epidemia**. It takes a model description, a dataframe, and additional arguments relating to the fitting algorithm, and translates this to data that is then passed to a precompiled **Stan** program which is used to fit the model. This allows model fitting to begin immediately as opposed to requiring compilation each time `epim` is called.

Usage

```
epim(
  rt,
  inf,
  obs,
  data,
  algorithm = c("sampling", "meanfield", "fullrank"),
  group_subset = NULL,
  prior_PD = FALSE,
  ...
)
```


Arguments

rt	An object of class <code>epirt</code> . This defines the model for the time-varying reproduction rates. See epirt for more details.
inf	An object of class <code>epiinf</code> . This defines the model for latent infections. See epiinf for more details.
obs	Either an <code>epiobs</code> object, or a list of such objects. Each element in the list defines a model for the specified observation vector. See epiobs for more details.
data	A dataframe with all data required for fitting the model. This includes all observation variables and covariates specified in the models for the reproduction number and ascertainment rates.
algorithm	One of "sampling", "meanfield" or "fullrank". This specifies which rstan function to use for fitting the model. For "sampling" this is sampling , otherwise this is vb .
group_subset	If specified, a character vector naming a subset of regions to include in the model.
prior_PD	Same as in stan_glm . If TRUE, samples all parameters from the joint prior distribution. This is useful for prior predictive checks. Defaults to FALSE.
...	Additional arguments to pass to the rstan function used to fit the model.

Details

This is similar to the workflow for fitting Bayesian regression models with **rstanarm**. A key difference, however, is that the models fit by **epidemia** are much more complex, and are therefore inherently more difficult to specify. **epidemia** aims to simplify this process by modularizing the model definition into three distinct parts: transmission, infections and observations. These components of the model are defined with the functions [epirt](#), [epiinf](#) and [epiobs](#) respectively.

[epim](#) has arguments `rt`, `inf` and `obs` which expect a description of the transmission model, infection model and observational models respectively. Together, these fully define the joint distribution of data and parameters. Each of these model components are described in terms of variables that are expected to live in a single dataframe, `data`. This dataframe must be compatible with the model components, in the sense that it holds all variables defined in these models.

In addition to taking a model description and a dataframe, [epim](#) has various additional arguments which specify how the model should be fit. If `algorithm = "sampling"` then the model will be fit using **Stan**'s adaptive Hamiltonian Monte Carlo sampler. This is done internally by calling [sampling](#). If `algorithm = "meanfield"` or `algorithm = "fullrank"`, then **Stan**'s variational Bayes algorithms are used instead, by calling [vb](#). Any unnamed arguments in the call to [epim](#) are passed directly on to the **rstan** sampling function. [epim](#) returns a fitted model object of class `epimodel`, which contains posterior samples from the model along with other useful objects.

In general, the adaptive Hamiltonian Monte Carlo sampler should be used for final inference. Nonetheless, fitting these models using HMC is often computationally demanding, and variational Bayes can often be fruitful for quickly iterating models.

Value

An object of class `epimodel`.

Examples

```
library(EpiEstim)
data("Flu1918")

date <- as.Date("1918-01-01") + seq(0, along.with = c(NA, Flu1918$incidence))
data <- data.frame(
  city = "Baltimore",
  cases = c(NA, Flu1918$incidence),
  date = date,
  week = lubridate::week(date)
)

rt <- epirt(
  formula = R(city, date) ~ rw(time = week, prior_scale = 0.1),
  prior_intercept = rstanarm::normal(log(2), 0.2),
  link = 'log'
)

obs <- epiobs(
  formula = cases ~ 1,
  prior_intercept = rstanarm::normal(location=1, scale=0.01),
  link = "identity",
  i2o = rep(.25,4)
)

args <- list(
  rt = rt,
  inf = epiinf(gen = Flu1918$si_distr),
  obs = obs,
  data = data,
  algorithm = "fullrank",
  iter = 1e4,
  seed = 12345
)

fm <- do.call(epim, args)
```

Description

An S3 class representing a fitted epidemiological model.

Details

The workhorse function `epim` of the **epidemia** package returns an object of class `epimodel`. This is heavily based on the `stanreg` class in **rstanarm** (see [stanreg-objects](#)). The internals are unimportant, but it is helpful to read the documentation to understand how to use methods operating on `epimodel` objects.

epiobs

Define Observational Models

Description

`epiobs` defines a model for an observation vector. These observations are taken to be a function of the latent infections in the population. Examples include daily death or hospitalization rates. For more details on the model assumptions please refer to the [model description](#) vignette.

Usage

```
epiobs(
  formula,
  i2o,
  family = "neg_binom",
  link = "logit",
  center = FALSE,
  prior = rstanarm::normal(scale = 0.2),
  prior_intercept = rstanarm::normal(scale = 0.2),
  prior_aux = rstanarm::normal(location = 10, scale = 5),
  ...
)
```

Arguments

<code>formula</code>	An object of class <code>formula</code> which determines the linear predictor for the ascertainment rate. The left hand side must define the response that is being modeled (i.e. the actual observations, not the latent ascertainment rates) in a given country on a given date.
<code>i2o</code>	A numeric (simplex) vector defining the probability mass function of the time distribution from infection to observation (i.e. a single death or hospitalization etc.). The n th element represents the probability that the individual was infected exactly n days prior to this.
<code>family</code>	A string representing the family of the sampling distribution. Can be "poisson", "neg_binom", "quasi_poisson", "normal" or "log_normal".
<code>link</code>	A string representing the link function used to transform the linear predictor. Can be one of "logit", "probit", "cauchit", "cloglog", "identity". Defaults to "logit".
<code>center</code>	If TRUE then the covariates are centered to have mean zero. All of the priors are then interpreted as priors on the centered covariates. Defaults to FALSE.

prior	Same as in stan_glm . Note: If <code>autoscale=TRUE</code> in the call to the prior distribution then automatic rescaling of the prior may take place.
prior_intercept	Same as in stan_glm . Prior for the regression intercept, if one has been specified.
prior_aux	The prior distribution for the auxiliary parameter, if it exists. Only used if family is "neg_binom" (reciprocal dispersion), "quasi_poisson" (dispersion), "normal" (standard deviation) or "log_normal" (sigma parameter). Can be a call to exponential , normal , student_t or cauchy . These result in half-normal, half-t and half-cauchy priors.
...	Additional arguments for <code>model.frame</code>

Details

Each observational model is given by a call to [epiobs](#). In particular, this must define the model for ascertainment rates and the time distribution from infection to observation. [epiobs](#) has a `formula` argument. The left hand side must define the observation vector to be modeled, while the right hand side defines a linear predictor for the ascertainment rate. The argument `i2o` plays a similar role to the `gen` argument in [epiinf](#), however it instead defines the probability mass function for the time between infection and observation.

Value

An object of class `epiobs`.

Examples

```
data(EuropeCovid)
# constant ascertainment rate (intercept model)
# link ensures ascertainment is between 0 and 2%
deaths <- epiobs(
  deaths ~ 1,
  i2o = EuropeCovid$inf2death,
  link = scaled_logit(0.02)
)
```

epirt

Model Reproduction Rates

Description

[epirt](#) defines a model for reproduction rates. For more details on the model assumptions, please read the [model description](#) vignette.

Usage

```
epirt(
  formula,
  link = "log",
  center = FALSE,
  prior = rstanarm::normal(scale = 0.5),
  prior_intercept = rstanarm::normal(scale = 0.5),
  prior_covariance = rstanarm::decov(scale = 0.5),
  ...
)
```

Arguments

formula	An object of class <code>formula</code> which determines the linear predictor for the reproduction rates. The left hand side must take the form $R(\text{group}, \text{date})$, where <code>group</code> and <code>date</code> variables. <code>group</code> must be a factor vector indicating group membership (i.e. country, state, age cohort), and <code>date</code> must be a vector of class <code>Date</code> . This is syntactic sugar for the reproduction number in the given group at the given date.
link	The link function. This must be either <code>"identity"</code> , <code>"log"</code> , or a call to scaled_logit .
center	If <code>TRUE</code> then the covariates for the regression are centered to have mean zero. All of the priors are then interpreted as prior on the centered covariates. Defaults to <code>FALSE</code> .
prior	Same as in stan_glm . In addition to the <code>rstanarm</code> provided priors, a shifted_gamma can be used. Note: If <code>autoscale=TRUE</code> in the call to the prior distribution then automatic rescaling of the prior may take place.
prior_intercept	Same as in stan_glm . Prior for the regression intercept (if it exists).
prior_covariance	Same as in stan_glm . Only used if the <code>formula</code> argument specifies random effects.
...	Additional arguments for model.frame

Details

[epirt](#) has a `formula` argument which defines the linear predictor, an argument `link` defining the link function, and additional arguments to specify priors on parameters making up the linear predictor.

A general R formula gives a symbolic description of a model. It takes the form $y \sim \text{model}$, where `y` is the response and `model` is a collection of terms separated by the `+` operator. `model` fully defines a linear predictor used to predict `y`. In this case, the “response” being modeled are reproduction numbers which are unobserved. [epirt](#) therefore requires that the left hand side of the formula takes the form $R(\text{group}, \text{date})$, where `group` and `date` refer to variables representing the region and date respectively. The right hand side can consist of fixed effects, random effects, and autocorrelation terms.

Value

An object of class epirt.

Examples

```

library(epidemia)
library(ggplot2)
data("EuropeCovid")
options(mc.cores = 2)

data <- EuropeCovid$data
data$week <- lubridate::week(data$date)

# collect arguments for epim
args <- list(
  inf = epiinf(gen = EuropeCovid$si),
  obs = epiobs(deaths ~ 1, i2o = EuropeCovid$inf2death, link = scaled_logit(0.02)),
  data = data,
  algorithm = "fullrank", # For speed - should generally use "sampling"
  iter = 2e4,
  group_subset = "France",
  seed = 12345,
  refresh = 0
)

# a simple random walk model for R
args$rt <- epirt(
  formula = R(country, date) ~ rw(time = week),
  link = scaled_logit(7)
)

fm1 <- do.call(epim, args)
plot_rt(fm1) + theme_bw()

# Modeling effects of NPIs
args$rt <- epirt(
  formula = R(country, date) ~ 1 + lockdown + public_events,
  link = scaled_logit(7)
)

fm2 <- do.call(epim, args)
plot_rt(fm2) + theme_bw()

# shifted gamma prior for NPI effects
args$rt <- epirt(
  formula = R(country, date) ~ 1 + lockdown + public_events,
  link = scaled_logit(7),
  prior = shifted_gamma(shape = 1/2, scale = 1, shift = log(1.05)/2)
)

```

```
# How does the implied prior look?  
args$prior_PD <- TRUE  
fm3 <- do.call(epim, args)  
plot_rt(fm3) + theme_bw()
```

EuropeCovid

Covid-19 data for European countries

Description

Contains a dataframe with recorded daily deaths from Covid-19 in 11 European countries up until 05/05/2020. The dataframe includes variables representing different non-pharmaceutical interventions implemented by the countries considered. The data matches that used in Flaxman et al. (2020). Also includes empirical distributions for the serial interval and the time from infection to death.

Usage

EuropeCovid

Format

A named list. The fields are:

data A data frame giving indicators of certain non-pharmaceutical interventions in each country, along with death data and populations. The earliest date for each country in the dataframe is exactly 30 days before 10 cumulative deaths were observed in the country.

inf2death A numeric vector representing the time distribution from infection to death assumed in Flaxman et al. (2020).

si The serial interval of covid-19 assumed in Flaxman et al. (2020).

References

Flaxman S, Mishra S, Gandy A, Unwin HJT, Mellan TA, Coupland H, Whittaker C, Zhu H, Be-rah T, Eaton JW, Monod M, Perez-Guzman PN, Schmit N, Cilloni L, Ainslie KEC, Baguelin M, Boonyasiri A, Boyd O, Cattarino L, Cooper LV, Cucunubá Z, Cuomo-Dannenburg G, Dighe A, Djaafara B, Dorigatti I, van Elsland SL, FitzJohn RG, Gaythorpe KAM, Geidelberg L, Grassly NC, Green WD, Hallett T, Hamlet A, Hinsley W, Jeffrey B, Knock E, Laydon DJ, Nedjati-Gilani G, Nouvellet P, Parag KV, Siveroni I, Thompson HA, Verity R, Volz E, Walters CE, Wang H, Wang Y, Watson OJ, Winskill P, Xi X, Walker PGT, Ghani AC, Donnelly CA, Riley SM, Vollmer MAC, Ferguson NM, Okell LC, Bhatt S, Team ICCR (2020). “Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe.” *Nature*. ISSN 1476-4687, doi: [10.1038/s4158602024057](https://doi.org/10.1038/s4158602024057).

 EuropeCovid2

Covid-19 data for European countries

Description

Similar to EuropeCovid, with the following exceptions. Daily death data is obtained from the WHO COVID-19 Explorer as of 05/01/2021. This differs from the data used in Flaxman et al. (2020), because counts were updated retrospectively by the WHO as new information came to light. Daily case data is also included from the same source. This data runs from 03/01/2020 until 30/06/2020.

Usage

EuropeCovid2

Format

A named list. The fields are:

data A data frame giving indicators of certain non-pharmaceutical interventions in each country, along with death data and populations.

inf2death A numeric vector representing the time distribution from infection to death.

si The serial interval of covid-19 assumed in Flaxman et al. (2020).

References

Flaxman S, Mishra S, Gandy A, Unwin HJT, Mellan TA, Coupland H, Whittaker C, Zhu H, Be-rah T, Eaton JW, Monod M, Perez-Guzman PN, Schmit N, Cilloni L, Ainslie KEC, Baguelin M, Boonyasiri A, Boyd O, Cattarino L, Cooper LV, Cucunubá Z, Cuomo-Dannenburg G, Dighe A, Djaafara B, Dorigatti I, van Elsland SL, FitzJohn RG, Gaythorpe KAM, Geidelberg L, Grassly NC, Green WD, Hallett T, Hamlet A, Hinsley W, Jeffrey B, Knock E, Laydon DJ, Nedjati-Gilani G, Nouvellet P, Parag KV, Siveroni I, Thompson HA, Verity R, Volz E, Walters CE, Wang H, Wang Y, Watson OJ, Winskill P, Xi X, Walker PGT, Ghani AC, Donnelly CA, Riley SM, Vollmer MAC, Ferguson NM, Okell LC, Bhatt S, Team ICCR (2020). “Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe.” *Nature*. ISSN 1476-4687, doi: [10.1038/s4158602024057](https://doi.org/10.1038/s4158602024057).

 evaluate_forecast

Posterior model evaluations

Description

Calculate daily error using one of three metrics, and also return coverage of credible intervals. Uses continuous ranked probability score (CRPS), mean absolute error and median absolute error.

Usage

```
evaluate_forecast(
  object,
  newdata = NULL,
  type,
  groups = NULL,
  metrics = NULL,
  levels = c(50, 95)
)
```

Arguments

object	A fitted model object returned by <code>epim</code> . See <code>epimodel-objects</code> .
newdata	If provided, the original data used in object is overridden. Useful for forecasting
type	A string specifying the name of the observations to plot. This should match one of the names of the response variables in the <code>obs</code> argument used in the call to <code>epim</code> .
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
metrics	A string or character vector specifying the plotted forecast error metrics. One of NULL, "crps", "mean_abs_error"
levels	A numeric vector defining the levels of the plotted credible intervals.

Value

A named list with dataframes giving metrics and coverage.

formula.epimodel	<i>Formula method for epimodel objects</i>
------------------	--

Description

Formula method for epimodel objects

Usage

```
## S3 method for class 'epimodel'
formula(x, ...)
```

Arguments

x	An epimodel object.
...	Can contain <code>fixed.only</code> and <code>random.only</code> arguments that both default to FALSE.

Value

An object of class formula.

get_samps

Retrieve final states from sampled Markov chains

Description

`get_samps` can be used to randomly select states from a fitted model object of class `epimodel`. The object must have been fit using Markov chain Monte Carlo, i.e. using `algorithm = "sampling"` in the call to `epim`. The states are sampled uniformly at random without replacement, across all chains and not including the warmup period.

Usage

```
get_samps(prefit, n)
```

Arguments

<code>prefit</code>	An object of class <code>epimodel</code> . This object must have been fit using <code>algorithm = "sampling"</code> .
<code>n</code>	A positive integer. This specifies the number of states to sample.

Details

This function can be used to specify the initial state for sampling based on states from another sampling run. This is particularly useful, for example, when you wish to fit a model using `pop_adjust = T`, as this makes the posterior geometry difficult to explore. Using a "prefit" run with `pop_adjust = F` is useful for finding good states that can be used as initial states for the run with the population adjustment.

Value

A list of length n . Each element in the list is itself a named list, with elements corresponding to sample parameters. The result can be passed directly as the `init` argument in `epim`.

get_x	<i>Extract X or Z from an epimodel object</i>
-------	---

Description

Extract X or Z from an epimodel object

Usage

```
get_x(object, ...)
```

```
get_z(object, ...)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Other arguments passed to methods.

Value

A matrix.

hexp	<i>A hierarchical model for seeded infections</i>
------	---

Description

This distribution assigns seeded infections in each population an exponential prior. The aux parameter refers to the mean of this distribution. This mean parameter is common to seeded infections in each group, and is given a prior distribution. This approach of assigning priors to hyperparameters is referred to as hierarchical modeling. A call to this function can be passed as the prior_seeds argument in [epiinf](#).

Usage

```
hexp(prior_aux = rstanarm::exponential(0.03))
```

Arguments

prior_aux	Specifies the prior distribution on the auxiliary parameter. This can be a call to normal , student_t or exponential .
-----------	--

Value

A named list to be parsed internally by [epim](#).

References

There are no references for Rd macro \insertAllCites on this help page.

`model.frame.epimodel` *model.frame method for epimodel objects. Please see [model.frame](#) for more details.*

Description

`model.frame` method for `epimodel` objects. Please see [model.frame](#) for more details.

Usage

```
## S3 method for class 'epimodel'
model.frame(formula, fixed.only = FALSE, ...)
```

Arguments

<code>formula</code>	A fitted model object returned by epim . See epimodel-objects .
<code>fixed.only</code>	See model.frame.merMod .
<code>...</code>	See model.frame .

Value

A [data.frame](#) containing information needed to fit the model. See [model.frame](#) for more details.

`ngrps.mixed` *Returns the levels for each grouping factor in the fitted object*

Description

Returns the levels for each grouping factor in the fitted object

Usage

```
## S3 method for class 'mixed'
ngrps(object, ...)
```

Arguments

<code>object</code>	an R object, see Details.
<code>...</code>	currently ignored.

Value

The number of levels (of a factor) or vector of number of levels for each “grouping factor” of a

pairs.epimodel *Pairs method for epimodel objects*

Description

Interface to **bayesplot**'s `mcmc_pairs` function. Closely mirrors the functionality of `pairs.stanreg`. Remember not to specify too many parameters. They will render slowly, and be difficult to interpret.

Usage

```
## S3 method for class 'epimodel'
pairs(
  x,
  pars = NULL,
  regex_pars = NULL,
  par_models = NULL,
  par_types = NULL,
  par_groups = NULL,
  condition = pairs_condition(nuts = "accept_stat__"),
  ...
)
```

Arguments

<code>x</code>	A fitted model object returned by <code>epim</code> . See epimodel-objects .
<code>pars</code>	A character vector giving parameter names.
<code>regex_pars</code>	A character vector of regular expressions to select parameters. If <code>pars</code> is also used, <code>regex_pars</code> is used in conjunction with <code>pars</code> .
<code>par_models</code>	A character vector that restricts parameters to a subset of model components. For example, "R" only uses parameters in the transmission model, "inf" uses parameters in infection model. Strings giving the name of the response in an observation model (i.e. LHS of the formula in <code>epiobs</code>) can also be used. If NULL (the default), all components are used.
<code>par_types</code>	A character vector that restricts parameters based on their type. The vector can include any of "fixed", "autocor", "random", "aux", "latent", or "seeds". The default is <code>c("fixed", "aux", "seeds")</code> , to avoid printing a very large number of parameters. If NULL, all types are used.
<code>par_groups</code>	A character vector restricting parameters to those used for a subset of regions in which the epidemic is modeled. Defaults to NULL in which case all regions are used.
<code>condition</code>	Same as <code>mcmc_pairs</code> , except that the default is <code>accept_stat__</code> , as in <code>pairs.stanreg</code> . Please see the documentation for <code>pairs.stanreg</code> for more details on this default.
<code>...</code>	Arguments passed to <code>mcmc_pairs</code> . The arguments <code>np</code> , <code>lp</code> , and <code>max_treedepth</code> are automatically handled, and therefore do not need to be specified.

Value

Multiple ggplot objects in a grid using [bayesplot_grid](#).

plot.epimodel	<i>Plot method for epimodel objects</i>
---------------	---

Description

Provides an interface to the [MCMC](#) module in the [bayesplot](#) package, and allows seamless plotting of MCMC draws along with various diagnostics. This method relies heavily on the code base for the [plot.stanreg](#) method in [rstanarm](#).

Usage

```
## S3 method for class 'epimodel'
plot(
  x,
  plotfun = "intervals",
  pars = NULL,
  regex_pars = NULL,
  par_models = NULL,
  par_types = c("fixed", "aux", "seeds"),
  par_groups = NULL,
  ...
)
```

Arguments

x	A fitted model object returned by epim . See epimodel-objects .
plotfun	Same as in plot.stanreg . A character string giving the name of the bayesplot MCMC function to use. These can be listed using available_mcmc . Defaults to "interval"
pars	A character vector giving parameter names.
regex_pars	A character vector of regular expressions to select parameters. If pars is also used, regex_pars is used in conjunction with pars.
par_models	A character vector that restricts parameters to a subset of model components. For example, "R" only uses parameters in the transmission model, "inf" uses parameters in infection model. Strings giving the name of the response in an observation model (i.e. LHS of the formula in epiobs) can also be used. If NULL (the default), all components are used.
par_types	A character vector that restricts parameters based on their type. The vector can include any of "fixed", "autocor", "random", "aux", "latent", or "seeds". The default is c("fixed", "aux", "seeds"), to avoid printing a very large number of parameters. If NULL, all types are used.

par_groups	A character vector restricting parameters to those used for a subset of regions in which the epidemic is modeled. Defaults to NULL in which case all regions are used.
...	Arguments passed on to the bayesplot function specified by plot fun.

Value

Either a ggplot object that can be further customized using the **ggplot2** package, or an object created from multiple ggplot objects (e.g. a gtable object created by [arrangeGrob](#)).

See Also

- [plot.stanreg](#).
- **bayesplot** vignettes for examples.
- [MCMC-overview](#) (**bayesplot**) for plotting function documentation.
- [color_scheme_set](#) (**bayesplot**) to set plotting color scheme.

plot_coverage	<i>Plot coverage probability of posterior credible intervals</i>
---------------	--

Description

Plots histograms showing empirical coverage of credible intervals specified using 'levels'. Can bucket by time period, by group, by whether the observation is new (not used in fitting).

Usage

```
plot_coverage(
  object,
  type,
  newdata = NULL,
  groups = NULL,
  levels = c(50, 95),
  period = NULL,
  by_group = FALSE,
  by_unseen = FALSE
)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
type	A string specifying the name of the observations to plot. This should match one of the names of the response variables in the obs argument used in the call to epim .
newdata	If provided, the original data used in object is overridden. Useful for forecasting

groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
levels	A numeric vector defining the levels of the plotted credible intervals.
period	Buckets computed empirical probabilities into time periods if specified.
by_group	Plot coverage for each group individually
by_unseen	Plot coverage separately for seen and unseen observations. Observations are 'seen' if they were used for fitting.

Value

A ggplot object.

plot_infections	<i>Plot latent infections</i>
-----------------	-------------------------------

Description

Plots posterior credible intervals and median for latent infections over time. The user can control the interval levels (i.e. 30%, 50% etc.) and the plotted group(s). This is a generic function.

Usage

```
plot_infections(object, ...)

## S3 method for class 'epimodel'
plot_infections(
  object,
  groups = NULL,
  dates = NULL,
  date_breaks = "2 weeks",
  date_format = "%Y-%m-%d",
  cumulative = FALSE,
  by_100k = FALSE,
  levels = c(30, 60, 90),
  log = FALSE,
  ...
)

spaghetti_infections(
  object,
  draws = min(500, posterior_sample_size(object)),
  alpha = 1/sqrt(draws),
  groups = NULL,
  dates = NULL,
  date_breaks = "2 weeks",
  date_format = "%Y-%m-%d",
```



```

    cumulative = FALSE,
    by_100k = FALSE,
    log = FALSE,
    smooth = 1,
    ...
)

```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Additional arguments for posterior_infections . Examples include <code>newdata</code> , which allows predictions or counterfactuals.
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
dates	A length 2 vector of Date objects. This defines the start and end dates of the date-range to be plotted. Must be coercible to Date if not NA. If an element of the vector is NA then the default lower/upper limit is used. See examples.
date_breaks	A string giving the distance between date tick labels. Default is "2 weeks". This is passed as the <code>date_breaks</code> argument to scale_x_date . Please see here for details.
date_format	This function attempts to coerce the <code>dates</code> argument to a vector of Date objects. <code>date_format</code> is passed as the <code>format</code> argument to as.Date . Default is "%Y-%m-%d".
cumulative	If TRUE then cumulative observations are plotted rather than daily. Defaults to FALSE.
by_100k	If TRUE, all quantities are plotted per 100k of population. Only possible if the model used a population adjustment.
levels	A numeric vector defining the levels of the plotted credible intervals.
log	If TRUE, plot quantities on a log10-scale. This argument must be logical, and defaults to FALSE.
draws	The number of sample paths to plot.
alpha	Sets transparency of sample paths.
smooth	An integer specifying the window used to smooth the reproduction rates. The default is 1, which corresponds to no smoothing.

Value

A ggplot object which can be further modified.

See Also

[plot_rt](#), [plot_obs](#), [plot_infectious](#), [posterior_infections](#)

Examples

```

data("EuropeCovid2")
data <- EuropeCovid2$data
data <- dplyr::filter(data, date > date[which(cumsum(deaths) > 10)[1] - 30])
data <- dplyr::filter(data, date < as.Date("2020-05-05"))

rt <- epi_rt(
  formula = R(country, date) ~ 0 + (1 + public_events + schools_universities +
    self_isolating_if_ill + social_distancing_encouraged + lockdown || country) +
    public_events + schools_universities + self_isolating_if_ill +
    social_distancing_encouraged + lockdown,
  prior = shifted_gamma(shape=1/6, scale = 1, shift = log(1.05)/6),
  prior_covariance = rstanarm::decov(shape = c(2, rep(0.5, 5)), scale=0.25),
  link = scaled_logit(6.5)
)

inf <- epi_inf(gen = EuropeCovid2$si, seed_days = 6)

deaths <- epi_obs(
  formula = deaths ~ 1,
  i2o = EuropeCovid2$inf2death,
  prior_intercept = rstanarm::normal(0,0.2),
  link = scaled_logit(0.02)
)

args <- list(rt=rt, inf=inf, obs=deaths, data=data, seed=12345)
args$group_subset <- c("Italy", "Austria", "Germany")
args$algorithm <- "fullrank"
args$iter <- 1e4
args$tol_rel_obj <- 1e-3

fm <- do.call(epim, args)

# different ways of using plot_rt
p <- plot_rt(fm) # default, plots all groups and dates
p <- plot_rt(fm, dates=c("2020-03-21", NA)) # plot 21 March 2020 onwards
p <- plot_rt(fm, dates=c(NA, "2020-03-20")) # plot up to 20 March 2020
p <- plot_rt(fm, dates=c("2020-03-20", "2020-04-20"))
p <- plot_rt(fm,
  dates=c("2020-03-20", "2020-04-20"),
  date_breaks="1 day") # ticks every day
p <- plot_rt(fm,
  dates=c("2020-20-03", "2020-20-04"),
  date_format="%Y-%d-%m") # (different date format)

# other plotting functions
p <- plot_obs(fm, type = "deaths")
p <- plot_infections(fm)
p <- plot_infectious(fm)

```

plot_infectious	<i>Plot total infectiousness over time.</i>
-----------------	---

Description

Plots credible intervals and the median for total infectiousness over time. This is basically a weighted sum of all infected individuals. Each infected individual is weighted by how infectious they are expected to be given how long they have been infected for. The user can control the interval levels (i.e. 30%, 50% etc.) and the plotted group(s). This is a generic function.

Usage

```
plot_infectious(object, ...)

## S3 method for class 'epimodel'
plot_infectious(
  object,
  groups = NULL,
  dates = NULL,
  date_breaks = "2 weeks",
  date_format = "%Y-%m-%d",
  levels = c(30, 60, 90),
  by_100k = FALSE,
  log = FALSE,
  ...
)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Additional arguments for posterior_infectious . Examples include <code>newdata</code> , which allows predictions or counterfactuals.
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
dates	A length 2 vector of Date objects. This defines the start and end dates of the date-range to be plotted. Must be coercible to Date if not NA. If an element of the vector is NA then the default lower/upper limit is used. See examples.
date_breaks	A string giving the distance between date tick labels. Default is "2 weeks". This is passed as the <code>date_breaks</code> argument to scale_x_date . Please see here for details.
date_format	This function attempts to coerce the <code>dates</code> argument to a vector of Date objects. <code>date_format</code> is passed as the <code>format</code> argument to as.Date . Default is "%Y-%m-%d".
levels	A numeric vector defining the levels of the plotted credible intervals.

by_100k	If TRUE, all quantities are plotted per 100k of population. Only possible if the model used a population adjustment.
log	If TRUE, plot quantities on a log10-scale. This argument must be logical, and defaults to FALSE.

Value

A ggplot object which can be further modified.

See Also

[plot_rt](#), [plot_obs](#), [plot_infections](#), [posterior_infectious](#)

Examples

```

data("EuropeCovid2")
data <- EuropeCovid2$data
data <- dplyr::filter(data, date > date[which(cumsum(deaths) > 10)[1] - 30])
data <- dplyr::filter(data, date < as.Date("2020-05-05"))

rt <- epirt(
  formula = R(country, date) ~ 0 + (1 + public_events + schools_universities +
    self_isolating_if_ill + social_distancing_encouraged + lockdown || country) +
    public_events + schools_universities + self_isolating_if_ill +
    social_distancing_encouraged + lockdown,
  prior = shifted_gamma(shape=1/6, scale = 1, shift = log(1.05)/6),
  prior_covariance = rstanarm::decov(shape = c(2, rep(0.5, 5)),scale=0.25),
  link = scaled_logit(6.5)
)

inf <- epiinf(gen = EuropeCovid2$si, seed_days = 6)

deaths <- epiobs(
  formula = deaths ~ 1,
  i2o = EuropeCovid2$inf2death,
  prior_intercept = rstanarm::normal(0,0.2),
  link = scaled_logit(0.02)
)

args <- list(rt=rt, inf=inf, obs=deaths, data=data, seed=12345)
args$group_subset <- c("Italy", "Austria", "Germany")
args$algorithm <- "fullrank"
args$iter <- 1e4
args$tol_rel_obj <- 1e-3

fm <- do.call(epim, args)

# different ways of using plot_rt
p <- plot_rt(fm) # default, plots all groups and dates
p <- plot_rt(fm, dates=c("2020-03-21", NA)) # plot 21 March 2020 onwards
p <- plot_rt(fm, dates=c(NA, "2020-03-20")) # plot up to 20 March 2020

```

```

p <- plot_rt(fm, dates=c("2020-03-20", "2020-04-20"))
p <- plot_rt(fm,
  dates=c("2020-03-20", "2020-04-20"),
  date_breaks="1 day") # ticks every day
p <- plot_rt(fm,
  dates=c("2020-20-03", "2020-20-04"),
  date_format="%Y-%d-%m") # (different date format)

# other plotting functions
p <- plot_obs(fm, type = "deaths")
p <- plot_infections(fm)
p <- plot_infectious(fm)

```

plot_metrics

Plot CRPS, Median/Mean Absolute Error

Description

Plots various metrics for evaluating probabilistic forecasts by group.

Usage

```
plot_metrics(object, groups = NULL, type, metrics = NULL, newdata = NULL)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
type	A string specifying the name of the observations to plot. This should match one of the names of the response variables in the obs argument used in the call to epim .
metrics	A string or character vector specifying the plotted forecast error metrics. One of NULL, "crps", "mean_abs_error"
newdata	If provided, the original data used in object is overridden. Useful for forecasting

Value

A ggplot object which can be further modified.

`plot_obs`*Plot posterior predictive distributions*

Description

Plots credible intervals and median for the observed data under the posterior predictive distribution, and for a specific observation type. The user can control the interval levels (i.e. 30%, 50% etc.) and the plotted group(s). This is a generic function.

Usage

```
plot_obs(object, ...)  
  
## S3 method for class 'epimodel'  
plot_obs(  
  object,  
  type,  
  groups = NULL,  
  dates = NULL,  
  date_breaks = "2 weeks",  
  date_format = "%Y-%m-%d",  
  cumulative = FALSE,  
  by_100k = FALSE,  
  bar = TRUE,  
  levels = c(30, 60, 90),  
  log = FALSE,  
  ...  
)  
  
spaghetti_obs(  
  object,  
  type,  
  draws = min(500, posterior_sample_size(object)),  
  alpha = 1/sqrt(draws),  
  groups = NULL,  
  dates = NULL,  
  date_breaks = "2 weeks",  
  date_format = "%Y-%m-%d",  
  cumulative = FALSE,  
  by_100k = FALSE,  
  bar = TRUE,  
  log = FALSE,  
  smooth = 1,  
  ...  
)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Additional arguments for posterior_predict.epimodel . Examples include newdata , which allows predictions or counterfactuals.
type	A string specifying the name of the observations to plot. This should match one of the names of the response variables in the <code>obs</code> argument used in the call to epim .
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
dates	A length 2 vector of Date objects. This defines the start and end dates of the date-range to be plotted. Must be coercible to Date if not NA. If an element of the vector is NA then the default lower/upper limit is used. See examples.
date_breaks	A string giving the distance between date tick labels. Default is "2 weeks". This is passed as the <code>date_breaks</code> argument to scale_x_date . Please see here for details.
date_format	This function attempts to coerce the <code>dates</code> argument to a vector of Date objects. <code>date_format</code> is passed as the <code>format</code> argument to as.Date . Default is "%Y-%m-%d".
cumulative	If TRUE then cumulative observations are plotted rather than daily. Defaults to FALSE.
by_100k	If TRUE, all quantities are plotted per 100k of population. Only possible if the model used a population adjustment.
bar	If TRUE, observations are plotted as a bar plot. Otherwise, a scatterplot is used. Defaults to TRUE.
levels	A numeric vector defining the levels of the plotted credible intervals.
log	If TRUE, plot quantities on a log10-scale. This argument must be logical, and defaults to FALSE.
draws	The number of sample paths to plot.
alpha	Sets transparency of sample paths.
smooth	An integer specifying the window used to smooth the reproduction rates. The default is 1, which corresponds to no smoothing.

Value

A ggplot object which can be further modified.

See Also

[plot_rt](#), [plot_infections](#), [plot_infectious](#), [posterior_predict](#)

Examples

```

data("EuropeCovid2")
data <- EuropeCovid2$data
data <- dplyr::filter(data, date > date[which(cumsum(deaths) > 10)[1] - 30])
data <- dplyr::filter(data, date < as.Date("2020-05-05"))

rt <- epi_rt(
  formula = R(country, date) ~ 0 + (1 + public_events + schools_universities +
    self_isolating_if_ill + social_distancing_encouraged + lockdown || country) +
    public_events + schools_universities + self_isolating_if_ill +
    social_distancing_encouraged + lockdown,
  prior = shifted_gamma(shape=1/6, scale = 1, shift = log(1.05)/6),
  prior_covariance = rstanarm::decov(shape = c(2, rep(0.5, 5)), scale=0.25),
  link = scaled_logit(6.5)
)

inf <- epi_inf(gen = EuropeCovid2$si, seed_days = 6)

deaths <- epi_obs(
  formula = deaths ~ 1,
  i2o = EuropeCovid2$inf2death,
  prior_intercept = rstanarm::normal(0,0.2),
  link = scaled_logit(0.02)
)

args <- list(rt=rt, inf=inf, obs=deaths, data=data, seed=12345)
args$group_subset <- c("Italy", "Austria", "Germany")
args$algorithm <- "fullrank"
args$iter <- 1e4
args$tol_rel_obj <- 1e-3

fm <- do.call(epim, args)

# different ways of using plot_rt
p <- plot_rt(fm) # default, plots all groups and dates
p <- plot_rt(fm, dates=c("2020-03-21", NA)) # plot 21 March 2020 onwards
p <- plot_rt(fm, dates=c(NA, "2020-03-20")) # plot up to 20 March 2020
p <- plot_rt(fm, dates=c("2020-03-20", "2020-04-20"))
p <- plot_rt(fm,
  dates=c("2020-03-20", "2020-04-20"),
  date_breaks="1 day") # ticks every day
p <- plot_rt(fm,
  dates=c("2020-20-03", "2020-20-04"),
  date_format="%Y-%d-%m") # (different date format)

# other plotting functions
p <- plot_obs(fm, type = "deaths")
p <- plot_infections(fm)
p <- plot_infectious(fm)

```

plot_rt	<i>Plot time-varying reproduction rates</i>
---------	---

Description

Plots credible intervals and the median from the posterior distribution for the time-varying reproduction rates. The user can control the interval levels (i.e. 30%, 50% etc.) and which groups/regions to plot for. This is a generic function.

Usage

```
plot_rt(object, ...)

## S3 method for class 'epimodel'
plot_rt(
  object,
  groups = NULL,
  step = FALSE,
  dates = NULL,
  date_breaks = "2 weeks",
  date_format = "%Y-%m-%d",
  levels = c(30, 60, 90),
  log = FALSE,
  smooth = 1,
  ...
)

spaghetti_rt(
  object,
  draws = min(500, posterior_sample_size(object)),
  alpha = 1/sqrt(draws),
  groups = NULL,
  step = FALSE,
  dates = NULL,
  date_breaks = "2 weeks",
  date_format = "%Y-%m-%d",
  log = FALSE,
  smooth = 1,
  ...
)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Additional unnamed arguments to be passed to posterior_rt . Examples include <code>newdata</code> , which allows predictions or counterfactuals. <code>adjusted=FALSE</code> prevents application of the population adjustment to the reproduction number.

groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
step	If TRUE, plot the median and credible intervals as a step function.
dates	A length 2 vector of Date objects. This defines the start and end dates of the date-range to be plotted. Must be coercible to Date if not NA. If an element of the vector is NA then the default lower/upper limit is used. See examples.
date_breaks	A string giving the distance between date tick labels. Default is "2 weeks". This is passed as the date_breaks argument to scale_x_date . Please see here for details.
date_format	This function attempts to coerce the dates argument to a vector of Date objects. date_format is passed as the format argument to as.Date . Default is "%Y-%m-%d".
levels	A numeric vector defining the levels of the plotted credible intervals.
log	If TRUE, plot quantities on a log10-scale. This argument must be logical, and defaults to FALSE.
smooth	An integer specifying the window used to smooth the reproduction rates. The default is 1, which corresponds to no smoothing.
draws	The number of sample paths to plot.
alpha	Sets transparency of sample paths.

Value

A ggplot object which can be further modified.

See Also

[plot_obs](#), [plot_infections](#), [plot_infectious](#)

Examples

```
data("EuropeCovid2")
data <- EuropeCovid2$data
data <- dplyr::filter(data, date > date[which(cumsum(deaths) > 10)[1] - 30])
data <- dplyr::filter(data, date < as.Date("2020-05-05"))

rt <- epirt(
  formula = R(country, date) ~ 0 + (1 + public_events + schools_universities +
    self_isolating_if_ill + social_distancing_encouraged + lockdown || country) +
    public_events + schools_universities + self_isolating_if_ill +
    social_distancing_encouraged + lockdown,
  prior = shifted_gamma(shape=1/6, scale = 1, shift = log(1.05)/6),
  prior_covariance = rstanarm::decov(shape = c(2, rep(0.5, 5)), scale=0.25),
  link = scaled_logit(6.5)
)

inf <- epiinf(gen = EuropeCovid$si, seed_days = 6)
```

```

deaths <- epiobs(
  formula = deaths ~ 1,
  i2o = EuropeCovid2$inf2death,
  prior_intercept = rstanarm::normal(0,0.2),
  link = scaled_logit(0.02)
)

args <- list(rt=rt, inf=inf, obs=deaths, data=data, seed=12345)
args$group_subset <- c("Italy", "Austria", "Germany")
args$algorithm <- "fullrank"
args$iter <- 1e4
args$tol_rel_obj <- 1e-3

fm <- do.call(epim, args)

# different ways of using plot_rt
p <- plot_rt(fm) # default, plots all groups and dates
p <- plot_rt(fm, dates=c("2020-03-21", NA)) # plot 21 March 2020 onwards
p <- plot_rt(fm, dates=c(NA, "2020-03-20")) # plot up to 20 March 2020
p <- plot_rt(fm, dates=c("2020-03-20", "2020-04-20"))
p <- plot_rt(fm,
  dates=c("2020-03-20", "2020-04-20"),
  date_breaks="1 day") # ticks every day
p <- plot_rt(fm,
  dates=c("2020-20-03", "2020-20-04"),
  date_format="%Y-%d-%m") # (different date format)

# other plotting functions
p <- plot_obs(fm, type = "deaths")
p <- plot_infections(fm)
p <- plot_infectious(fm)

```

posterior_coverage *Coverage of posterior credible intervals*

Description

Calculate daily error using one of three metrics, and also return coverage of credible intervals. Uses continuous ranked probability score (CRPS), mean absolute error and median absolute error.

Usage

```

posterior_coverage(
  object,
  type,
  newdata = NULL,
  groups = NULL,
  levels = c(50, 95)
)

```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
type	A string specifying the name of the observations to plot. This should match one of the names of the response variables in the obs argument used in the call to epim .
newdata	If provided, the original data used in object is overridden. Useful for forecasting.
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
levels	A numeric vector defining the levels of the plotted credible intervals.

Value

A dataframe indicating whether observations fall within the specified credible intervals

posterior_infections *Generic function for getting posterior draws of daily infections over time*

Description

Generic function for getting posterior draws of daily infections over time

Usage

```
posterior_infections(object, ...)

## S3 method for class 'epimodel'
posterior_infections(object, newdata = NULL, draws = NULL, seed = NULL, ...)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Not used.
newdata	If provided, the original data used in object is overridden. Useful both for counterfactual and prediction analysis.
draws	Number of posterior draws to use. Defaults to the number of parameter draws in the fitted model.
seed	An optional seed.

Value

A named list with draws of the series, and vectors giving the corresponding dates and population.

posterior_infectious *Generic function for getting posterior draws of total infectiousness over time*

Description

Generic function for getting posterior draws of total infectiousness over time

Usage

```
posterior_infectious(object, ...)

## S3 method for class 'epimodel'
posterior_infectious(object, newdata = NULL, draws = NULL, seed = NULL, ...)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Not used.
newdata	If provided, the original data used in object is overridden. Useful both for counterfactual and prediction analysis
draws	Number of posterior draws to use. Defaults to the number of parameter draws in the fitted model.
seed	An optional seed.

Value

A named list with draws of the series, and vectors giving the corresponding dates and population.

posterior_latent *Generic function for getting posterior draws of a specified latent sequence*

Description

Draws samples from one of a number of unobserved time series using the posterior parameter draws from the passed object. Can retrieve posterior estimate of the reproduction number over time (unadjusted or adjusted), the daily new infections, or the total "infectiousness" of the population over time.

Usage

```
posterior_latent(object, ...)

## S3 method for class 'epimodel'
posterior_latent(
  object,
  newdata = NULL,
  series = c("Rt", "Rt_unadj", "infections"),
  draws = NULL,
  seed = NULL,
  ...
)
```

Arguments

object	A fitted model object returned by <code>epim</code> . See <code>epimodel-objects</code> .
...	Not used.
newdata	If provided, the original data used in object is overridden. Useful both for counterfactual and prediction analysis
series	Type of latent series to return.
draws	Number of posterior draws to use. Defaults to the number of parameter draws in the fitted model.
seed	An optional seed.

Value

A named list with draws of the series, and vectors giving the corresponding dates and population.

posterior_linpred	<i>Gives the posterior linear predictor for the reproduction numbers Will be extended for observations in future versions</i>
-------------------	---

Description

Gives the posterior linear predictor for the reproduction numbers Will be extended for observations in future versions

Usage

```
posterior_linpred(
  object,
  transform = FALSE,
  type = NULL,
  newdata = NULL,
  draws = NULL,
```

```

    fixed = TRUE,
    random = TRUE,
    autocor = TRUE,
    offset = TRUE,
    ...
  )

```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
transform	If TRUE, transforms the predictor by the inverse link function. Defaults to FALSE.
type	If NULL, then gives posterior linear predictor for reproduction numbers. Otherwise gives the predictor for the specified observation type.
newdata	If provided, the original data used in object is overridden. Useful both for counterfactual and prediction analysis
draws	Number of posterior draws to use. Defaults to the number of parameter draws in the fitted model.
fixed	Include fixed effects. Defaults to TRUE.
random	Include random effects. Defaults to TRUE.
autocor	Include autocorrelation terms. Defaults to TRUE.
offset	Include offsets. Defaults to TRUE.
...	Not used.

Value

A list containing the parameter draws and associated groups and dates.

posterior_metrics	<i>CRPS, Mean Absolute Error, Median Absolute Error</i>
-------------------	---

Description

Calculate daily error using one of three metrics, and also return coverage of credible intervals. Uses continuous ranked probability score (CRPS), mean absolute error and median absolute error.

Usage

```
posterior_metrics(object, type, newdata = NULL, groups = NULL, metrics = NULL)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
type	A string specifying the name of the observations to plot. This should match one of the names of the response variables in the obs argument used in the call to epim .
newdata	If provided, the original data used in object is overridden. Useful for forecasting.
groups	Either NULL or a character vector specifying the groups to plot for. Default is NULL, which plots all modeled groups.
metrics	A string or character vector specifying the plotted forecast error metrics. One of NULL, "crps", "mean_abs_error"

Value

A dataframe giving forecast error for each metric and observation

posterior_predict.epimodel

Draws samples from the posterior predictive distribution of the observations

Description

Generate data from the posterior predictive distribution. This is useful for assessing the fit of a model. Alternatively this can be used for assessing counterfactuals or for prediction using the newdata argument.

Usage

```
## S3 method for class 'epimodel'
posterior_predict(
  object,
  newdata = NULL,
  draws = NULL,
  types = NULL,
  seed = NULL,
  posterior_mean = FALSE,
  ...
)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
newdata	If provided, the original data used in object is overridden. Useful both for counterfactual and prediction analysis

draws	Number of posterior draws to use. Defaults to the number of parameter draws in the fitted model.
types	A character vector specifying the names of the outcome variables to consider. If unspecified, uses all.
seed	An optional seed.
posterior_mean	If TRUE, return samples of posterior means rather than from the posterior predictive.
...	Not used.

Value

A named list of draws from the posterior predictive. Each element corresponds to a specific outcome.

posterior_rt	<i>Generic function for getting posterior draws of the time-varying reproduction rates</i>
--------------	--

Description

Generic function for getting posterior draws of the time-varying reproduction rates

Usage

```
posterior_rt(object, ...)

## S3 method for class 'epimodel'
posterior_rt(
  object,
  newdata = NULL,
  draws = NULL,
  seed = NULL,
  adjusted = TRUE,
  ...
)
```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Not used.
newdata	If provided, the original data used in object is overridden. Useful both for counterfactual and prediction analysis
draws	Number of posterior draws to use. Defaults to the number of parameter draws in the fitted model.
seed	An optional seed.
adjusted	Flag whether to return the adjusted reproduction rates. Defaults to TRUE.

Value

A named list with draws of the series, and vectors giving the corresponding dates and population.

posterior_sample_size *Get posterior sample size from a fitted model*

Description

Get posterior sample size from a fitted model

Usage

```
posterior_sample_size(object)
```

Arguments

object An object of class epimodel

Value

Integer giving sample size of fitted model

posterior_sample_size.epimodel
Plotting the posterior linear predictor for R or ascertainment rates

Description

Plots credible intervals for the observed data under the posterior predictive distribution, and for a specific observation type. The user can control the levels of the intervals and the plotted group(s). This is a generic function.

Usage

```
## S3 method for class 'epimodel'
posterior_sample_size(object)

## S3 method for class 'epimodel'
all_obs_types(object)

plot_linpred(object, ...)

## S3 method for class 'epimodel'
plot_linpred(
  object,
```

```

    type = NULL,
    groups = NULL,
    dates = NULL,
    date_breaks = "2 weeks",
    date_format = "%Y-%m-%d",
    levels = c(30, 60, 90),
    ...
  )

```

Arguments

object	A fitted model object returned by epim . See epimodel-objects .
...	Additional arguments for posterior_predict.epimodel . Examples include <code>newdata</code> , which allows predictions or counterfactuals.
type	the name of the observations to plot. This should match one of the names of the <code>obs</code> argument to <code>epim</code> .
groups	Either <code>NULL</code> or a character vector specifying the groups to plot for. Default is <code>NULL</code> , which plots all modeled groups.
dates	A length 2 vector of Date objects. This defines the start and end dates of the date-range to be plotted. Must be coercible to Date if not NA. If an element of the vector is NA then the default lower/upper limit is used. See examples.
date_breaks	A string giving the distance between date tick labels. Default is "2 weeks". This is passed as the <code>date_breaks</code> argument to scale_x_date . Please see here for details.
date_format	This function attempts to coerce the <code>dates</code> argument to a vector of Date objects. <code>date_format</code> is passed as the <code>format</code> argument to as.Date . Default is "%Y-%m-%d".
levels	A numeric vector defining the levels of the plotted credible intervals.

Value

A ggplot object which can be further modified.

```
print.epimodel      Print fitted model details
```

Description

Prints estimated regression parameters, and other model parameters. Similar to printing of `rstan::stanreg` objects.

Usage

```
## S3 method for class 'epimodel'
print(x, digits = 1, ...)
```

Arguments

x	A fitted model object returned by epim . See epimodel-objects .
digits	Number of decimal places to print.
...	Not used.

Value

No return value.

```
print.prior_summary.epimodel
```

Print method for prior_summary.epimodel objects

Description

Print method for prior_summary.epimodel objects

Usage

```
## S3 method for class 'prior_summary.epimodel'
print(x, digits, ...)
```

Arguments

x	A fitted model object returned by epim . See epimodel-objects .
digits	Number of decimal places to print.
...	Not used.

Value

No return value.

```
print.prior_summary_reg.epimodel
```

Print method for prior_summary_reg.epimodel objects

Description

Print method for prior_summary_reg.epimodel objects

Usage

```
## S3 method for class 'prior_summary_reg.epimodel'
print(x, digits, ...)
```

Arguments

x	A fitted model object returned by <code>epim</code> . See <code>epimodel-objects</code> .
digits	Number of decimal places to print.
...	Not used.

Value

No return value.

prior_summary.epimodel

Returns a summary of the prior distributions used

Description

Returns a summary of the prior distributions used

Usage

```
## S3 method for class 'epimodel'
prior_summary(object, digits = 3, ...)
```

Arguments

object	The object to use.
digits	Number of digits used for rounding.
...	Arguments passed to methods. See the methods in the <code>rstanarm</code> package for examples.

Value

`prior_summary()` methods should return an object containing information about the prior distribution(s) used for the given model. The structure of this object will depend on the method.

The default method just returns `object$prior.info`, which is `NULL` if there is no `'prior.info'` element.

rw	<i>Adds random walks with independent Gaussian steps to the parameterization of the time-varying reproduction number.</i>
----	---

Description

A call to `rw` can be used in the 'formula' argument of `epim`, allowing random walks for the reproduction number. Does not evaluate arguments. Simply creates a list with the information needed for the stan data to be parsed correctly.

Usage

```
rw(time, gr, prior_scale = 0.2)
```

Arguments

time	An optional name defining the random walk time periods for each date and group. This must be a column name found in the data argument to <code>epim</code> . If not specified, determined by the dates column implied by the formula argument to <code>epim</code> is used.
gr	Same as for <code>time</code> , except this defines the grouping to use for the random walks. A separate walk is defined for each group. If not specified a common random walk is used for all groups.
prior_scale	The steps of the walks are independent zero mean normal with an unknown scale hyperparameter. This scale is given a half-normal prior. <code>prior_scale</code> sets the scale parameter of this hyperprior.

Value

A list to be parsed internally.

Examples

```
data("EuropeCovid")
args <- EuropeCovid
args$formula <- R(country, date) ~ 1 + rw(gr=country) + lockdown
```

scaled_logit	<i>Represents a 'scaled' logit link</i>
--------------	---

Description

The link function is parameterised by a value $r > 0$, and takes the form $\log(x/(K - x))$. The inverse link is then $K * \text{invlogit}(x)$. This is similar to the logit link, although x can range between $[0, K]$ rather than $[0, 1]$. The parameter K can be chosen.

Usage

```
scaled_logit(K = 6)
```

Arguments

K parameterises the link function. The inverse of which then takes values between 0 and K .

Value

A list with class "scaled_logit"

shifted_gamma	<i>A shifted gamma prior</i>
---------------	------------------------------

Description

A gamma prior distribution which can be shifted.

Usage

```
shifted_gamma(shape = 1, scale = 1, shift = 0, autoscale = TRUE)
```

Arguments

$shape, scale$ Sets the shape and scale parameters of the Gamma prior.
 $shift$ The Gamma prior can be shifted to allow for positive support.
 $autoscale$ Same as in [priors](#).

Details

rstanarm provides a set of distributions (see [priors](#)) which can be used for the priors on regression parameters. Intuitively, non-pharmaceutical interventions are unlikely to a-priori cause a large increase in the reproduction number. A shifted gamma prior can be used to model this idea, and has been used in Flaxman et al. (2020). `shifted_gamma` can be used as the prior argument to `epim`. This specified a shifted gamma prior on the negative of the regression parameters. i.e. if there is no shift, the support is on the negative half of the real line.

Value

A named list to be parsed internally by `epim`.

References

Flaxman S, Mishra S, Gandy A, Unwin HJT, Mellan TA, Coupland H, Whittaker C, Zhu H, Be-rah T, Eaton JW, Monod M, Perez-Guzman PN, Schmit N, Cilloni L, Ainslie KEC, Baguelin M, Boonyasiri A, Boyd O, Cattarino L, Cooper LV, Cucunubá Z, Cuomo-Dannenburg G, Dighe A, Djaafara B, Dorigatti I, van Elsland SL, FitzJohn RG, Gaythorpe KAM, Geidelberg L, Grassly NC, Green WD, Hallett T, Hamlet A, Hinsley W, Jeffrey B, Knock E, Laydon DJ, Nedjati-Gilani G, Nouvellet P, Parag KV, Siveroni I, Thompson HA, Verity R, Volz E, Walters CE, Wang H, Wang Y, Watson OJ, Winskill P, Xi X, Walker PGT, Ghani AC, Donnelly CA, Riley SM, Vollmer MAC, Ferguson NM, Okell LC, Bhatt S, Team ICCR (2020). “Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe.” *Nature*. ISSN 1476-4687, doi: [10.1038/s4158602024057](https://doi.org/10.1038/s4158602024057).

summary.epimodel

Summary method for epimodel objects

Description

Provides a summary of parameter estimates and MCMC diagnostics. Similar to `summary.stanreg` in `rstanarm`.

Usage

```
## S3 method for class 'epimodel'
summary(
  object,
  pars = NULL,
  regex_pars = NULL,
  probs = c(0.1, 0.5, 0.9),
  ...,
  digits = 1
)

## S3 method for class 'summary.epimodel'
print(x, digits = max(1, attr(x, "print.digits")), ...)
```

Arguments

<code>object</code>	A fitted model object returned by <code>epim</code> . See <code>epimodel-objects</code> .
<code>pars</code>	A character vector giving a subset of parameters to include. Default is <code>NULL</code> , in which case all parameters are included.
<code>regex_pars</code>	A character vector of regular expressions to select parameters. If <code>pars</code> is also used, <code>regex_pars</code> is used in conjunction with <code>pars</code> .

probs	A numeric vector of probabilities for computing quantiles of parameter estimates.
...	Not used.
digits	Number of digits to use for formatting numbers when printing.
x	An object of class "summary.epimodel".

Value

An object of class "summary.epimodel".

terms.epimodel	<i>Terms method for epimodel objects</i>
----------------	--

Description

Terms method for epimodel objects

Usage

```
## S3 method for class 'epimodel'
terms(x, fixed.only = TRUE, random.only = FALSE, ...)
```

Arguments

x, fixed.only, random.only, ...
See [terms.merMod](#)

Value

The value of attributes. See [attr](#) for more details.

terms_rw	<i>Finds random walk terms in a formula object</i>
----------	--

Description

Finds random walk terms in a formula object

Usage

```
terms_rw(x)
```

Arguments

x An object of class "formula"

Value

The value of attributes. See [attr](#) for more details.

Index

- * **datasets**
 - EnglandNewCases, 6
 - EuropeCovid, 15
 - EuropeCovid2, 16
- all_obs_types, 4
- all_obs_types.epimodel
 - (posterior_sample_size.epimodel), 42
- arrangeGrob, 23
- as.array.epimodel (as.matrix.epimodel), 4
- as.data.frame.epimodel
 - (as.matrix.epimodel), 4
- as.Date, 25, 27, 31, 34, 43
- as.matrix.epimodel, 4
- as.matrix.stanreg, 5
- attr, 49
- available_mcmc, 22

- bayesplot, 22
- bayesplot_grid, 22

- cauchy, 7, 12
- color_scheme_set, 23

- data.frame, 20

- EnglandNewCases, 6
- epidemia (epidemia-package), 3
- epidemia-package, 3
- epiinf, 6, 6, 8, 9, 19
- epim, 3, 5, 7, 8, 8, 9, 17–23, 25, 27, 29, 31, 33, 36–41, 43–45, 48
- epimodel-objects, 10
- epiobs, 9, 11, 11, 12
- epirt, 9, 12, 12, 13
- EuropeCovid, 15
- EuropeCovid2, 16
- evaluate_forecast, 16
- exponential, 7, 12, 19

- formula.epimodel, 17

- get_samps, 18, 18
- get_x, 19
- get_z (get_x), 19

- hexp, 7, 19

- MCMC, 22
- mcmc_pairs, 21
- model.frame, 12, 13, 20
- model.frame.epimodel, 20
- model.frame.merMod, 20

- ngrps.mixed, 20
- normal, 7, 12, 19

- pairs.epimodel, 21
- pairs.stanreg, 21
- plot.epimodel, 22
- plot.stanreg, 22, 23
- plot_coverage, 23
- plot_infections, 24, 28, 31, 34
- plot_infectious, 25, 27, 31, 34
- plot_linpred
 - (posterior_sample_size.epimodel), 42
- plot_metrics, 29
- plot_obs, 25, 28, 30, 34
- plot_rt, 25, 28, 31, 33
- posterior_coverage, 35
- posterior_infections, 25, 36
- posterior_infectious, 27, 28, 37
- posterior_latent, 37
- posterior_linpred, 38
- posterior_metrics, 39
- posterior_predict, 31
- posterior_predict.epimodel, 31, 40, 43
- posterior_rt, 33, 41
- posterior_sample_size, 42
- posterior_sample_size.epimodel, 42

print.epimodel, 43
print.prior_summary.epimodel, 44
print.prior_summary_reg.epimodel, 44
print.summary.epimodel
 (summary.epimodel), 48
prior_summary.epimodel, 45
priors, 13, 47

rw, 46

sampling, 9
scale_x_date, 25, 27, 31, 34, 43
scaled_logit, 13, 47
shifted_gamma, 13, 47
spaghetti_infections (plot_infections),
 24
spaghetti_obs (plot_obs), 30
spaghetti_rt (plot_rt), 33
stan_glm, 9, 12, 13
stan_glmer, 13
student_t, 7, 12, 19
summary.epimodel, 48
summary.stanreg, 48

terms.epimodel, 49
terms.merMod, 49
terms_rw, 49

vb, 9