

# Package: epidist (via r-universe)

January 21, 2025

**Title** Estimate Epidemiological Delay Distributions With brms

**Version** 0.0.0.9000

**Description** Understanding and accurately estimating epidemiological delay distributions is important for public health policy. These estimates directly influence epidemic situational awareness, control strategies, and resource allocation. In this package, we provide methods to address the key challenges in estimating these distributions, including truncation, interval censoring, and dynamical biases. Despite their importance, these issues are frequently overlooked, often resulting in biased conclusions.

**License** MIT + file LICENSE

**URL** <https://epidist.epinowcast.org/>,  
<https://github.com/epinowcast/epidist/>

**BugReports** <https://github.com/epinowcast/epidist/issues/>

**Depends** R (>= 3.5.0)

**Imports** brms, ggplot2, purrr, stats, cli, checkmate, rstan (>= 2.26.0), dplyr, tibble, lubridate, primarycensored

**Suggests** bookdown, testthat (>= 3.0.0), readxl, janitor, gt, knitr, roxyglobals, bayesplot, tidyr, posterior, fs, sf, tidybayes, modelr, patchwork, cmdstanr, priorsense, usethis, pkgdown, CodeDepends, BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0)

**Additional\_repositories** <https://stan-dev.r-universe.dev>

**Config/Needs/website** r-lib/pkgdown, epinowcast/enwtheme

**Config/testthat/edition** 3

**Config/roxyglobals/filename** globals.R

**Config/roxyglobals/unique** FALSE

**LazyData** true

**VignetteBuilder** knitr

**Encoding** UTF-8

**Roxygen** `list(markdown = TRUE, roclets = c("`collate", "`namespace",  
`rd", "`roxygenals::global_roclet"))`

**RoxygenNote** 7.3.2

**Config/pak/sysreqs** `make libicu-dev`

**Repository** `https://epinowcast.r-universe.dev`

**RemoteUrl** `https://github.com/epinowcast/epidist`

**RemoteRef** `v0.1.0`

**RemoteSha** `dc5779afe52d076782748708f89bb9981af204d5`

## Contents

<code>add_mean_sd</code> . . . . .	3
<code>add_mean_sd.default</code> . . . . .	4
<code>add_mean_sd.gamma_samples</code> . . . . .	4
<code>add_mean_sd.lognormal_samples</code> . . . . .	5
<code>assert_epidist</code> . . . . .	5
<code>assert_epidist.epidist_linelist_data</code> . . . . .	6
<code>as_epidist_latent_model</code> . . . . .	6
<code>as_epidist_latent_model.epidist_linelist_data</code> . . . . .	7
<code>as_epidist_linelist_data</code> . . . . .	7
<code>as_epidist_linelist_data.data.frame</code> . . . . .	8
<code>as_epidist_linelist_data.default</code> . . . . .	9
<code>as_epidist_naive_model</code> . . . . .	9
<code>as_epidist_naive_model.epidist_linelist_data</code> . . . . .	10
<code>epidist</code> . . . . .	10
<code>epidist.default</code> . . . . .	11
<code>epidist_diagnostics</code> . . . . .	12
<code>epidist_family</code> . . . . .	13
<code>epidist_family_model</code> . . . . .	14
<code>epidist_family_model.default</code> . . . . .	14
<code>epidist_family_model.epidist_latent_model</code> . . . . .	15
<code>epidist_family_prior</code> . . . . .	16
<code>epidist_family_prior.default</code> . . . . .	16
<code>epidist_family_prior.lognormal</code> . . . . .	17
<code>epidist_family_reparam</code> . . . . .	18
<code>epidist_family_reparam.default</code> . . . . .	18
<code>epidist_family_reparam.gamma</code> . . . . .	19
<code>epidist_formula</code> . . . . .	19
<code>epidist_formula_model.default</code> . . . . .	20
<code>epidist_formula_model.epidist_latent_model</code> . . . . .	21
<code>epidist_gen_log_lik_latent</code> . . . . .	21
<code>epidist_gen_posterior_epred</code> . . . . .	22
<code>epidist_gen_posterior_predict</code> . . . . .	23
<code>epidist_model_prior</code> . . . . .	24

<code>add_mean_sd</code>	3
<code>epidist_model_prior.default</code>	24
<code>epidist_prior</code>	25
<code>epidist_stancode</code>	26
<code>epidist_stancode.default</code>	26
<code>is_epidist_latent_model</code>	27
<code>is_epidist_linelist_data</code>	27
<code>is_epidist_naive_model</code>	28
<code>new_epidist_latent_model</code>	28
<code>new_epidist_linelist_data</code>	29
<code>new_epidist_naive_model</code>	29
<code>predict_delay_parameters</code>	30
<code>sierra_leone_ebola_data</code>	30
<code>simulate_exponential_cases</code>	31
<code>simulate_gillespie</code>	32
<code>simulate_secondary</code>	32
<code>simulate_uniform_cases</code>	33
<b>Index</b>	<b>34</b>

---

<code>add_mean_sd</code>	<i>Add natural scale mean and standard deviation parameters</i>
--------------------------	---

---

### Description

Add natural scale mean and standard deviation parameters

### Usage

```
add_mean_sd(data, ...)
```

### Arguments

<code>data</code>	A dataframe of distributional parameters
<code>...</code>	Additional arguments for method.

### See Also

Other postprocess: `add_mean_sd.default()`, `add_mean_sd.gamma_samples()`, `add_mean_sd.lognormal_samples()`, `predict_delay_parameters()`

---

add\_mean\_sd.default     *Default method for add natural scale parameters*

---

### Description

Default method for add natural scale parameters

### Usage

```
## Default S3 method:
add_mean_sd(data, ...)
```

### Arguments

data	A dataframe of distributional parameters
...	Additional arguments for method.

### See Also

Other postprocess: [add\\_mean\\_sd\(\)](#), [add\\_mean\\_sd.gamma\\_samples\(\)](#), [add\\_mean\\_sd.lognormal\\_samples\(\)](#), [predict\\_delay\\_parameters\(\)](#)

---

add\_mean\_sd.gamma\_samples  
                                   *Add natural scale mean and standard deviation parameters for a latent gamma model*

---

### Description

Again, mu and shape here are the distributional parameters of brms.

### Usage

```
## S3 method for class 'gamma_samples'
add_mean_sd(data, ...)
```

### Arguments

data	A dataframe of distributional parameters
...	Additional arguments for method.

### See Also

Other postprocess: [add\\_mean\\_sd\(\)](#), [add\\_mean\\_sd.default\(\)](#), [add\\_mean\\_sd.lognormal\\_samples\(\)](#), [predict\\_delay\\_parameters\(\)](#)

---

 add\_mean\_sd.lognormal\_samples

*Add natural scale mean and standard deviation parameters for a latent lognormal model*

---

### Description

Note that the input parameters here are mu and sigma, corresponding to the distributional parameters used by brms for the lognormal family.

### Usage

```
## S3 method for class 'lognormal_samples'
add_mean_sd(data, ...)
```

### Arguments

data	A dataframe of distributional parameters
...	Additional arguments for method.

### See Also

Other postprocess: [add\\_mean\\_sd\(\)](#), [add\\_mean\\_sd.default\(\)](#), [add\\_mean\\_sd.gamma\\_samples\(\)](#), [predict\\_delay\\_parameters\(\)](#)

---

 assert\_epidist

*Validation for epidist objects*

---

### Description

Validation for epidist objects

### Usage

```
assert_epidist(data, ...)
```

### Arguments

data	Object to validate
...	Additional arguments

### Value

NULL invisibly

---

assert\_epidist.epidist\_linelist\_data  
*Assert validity of epidist\_linelist\_data objects*

---

**Description**

Assert validity of epidist\_linelist\_data objects

**Usage**

```
## S3 method for class 'epidist_linelist_data'
assert_epidist(data, ...)
```

**Arguments**

data	An object to check
...	Additional arguments

**See Also**

Other linelist\_data: [as\\_epidist\\_linelist\\_data\(\)](#), [as\\_epidist\\_linelist\\_data.data.frame\(\)](#), [as\\_epidist\\_linelist\\_data.default\(\)](#), [is\\_epidist\\_linelist\\_data\(\)](#), [new\\_epidist\\_linelist\\_data\(\)](#)

---

as\_epidist\_latent\_model  
*Convert an object to an epidist\_latent\_model object*

---

**Description**

Convert an object to an epidist\_latent\_model object

**Usage**

```
as_epidist_latent_model(data)
```

**Arguments**

data	An object to be converted to the class epidist_latent_model
------	---

**See Also**

Other latent\_model: [as\\_epidist\\_latent\\_model.epidist\\_linelist\\_data\(\)](#), [epidist\\_family\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_formula\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_gen\\_log\\_lik\\_latent\(\)](#), [is\\_epidist\\_latent\\_model\(\)](#), [new\\_epidist\\_latent\\_model\(\)](#)

---

`as_epidist_latent_model.epidist_linelist_data`*The latent model method for epidist\_linelist\_data objects*

---

**Description**

The latent model method for epidist\_linelist\_data objects

**Usage**

```
## S3 method for class 'epidist_linelist_data'  
as_epidist_latent_model(data)
```

**Arguments**

data            An epidist\_linelist\_data object

**See Also**

Other latent\_model: [as\\_epidist\\_latent\\_model\(\)](#), [epidist\\_family\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_formula\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_gen\\_log\\_lik\\_latent\(\)](#), [is\\_epidist\\_latent\\_model\(\)](#), [new\\_epidist\\_latent\\_model\(\)](#)

---

`as_epidist_linelist_data`*Create an epidist\_linelist\_data object*

---

**Description**

Create an epidist\_linelist\_data object

**Usage**

```
as_epidist_linelist_data(data, ...)
```

**Arguments**

data            The data to convert  
...             Additional arguments passed to methods

**See Also**

Other linelist\_data: [as\\_epidist\\_linelist\\_data.data.frame\(\)](#), [as\\_epidist\\_linelist\\_data.default\(\)](#), [assert\\_epidist.epidist\\_linelist\\_data\(\)](#), [is\\_epidist\\_linelist\\_data\(\)](#), [new\\_epidist\\_linelist\\_data\(\)](#)

---

```
as_epidist_linelist_data.data.frame
```

*Create an epidist\_linelist\_data object from a data frame with event dates*

---

## Description

Create an epidist\_linelist\_data object from a data frame with event dates

## Usage

```
## S3 method for class 'data.frame'
as_epidist_linelist_data(
  data,
  pdate_lwr = NULL,
  sdate_lwr = NULL,
  pdate_upr = NULL,
  sdate_upr = NULL,
  obs_date = NULL,
  ...
)
```

## Arguments

data	A data.frame containing line list data
pdate_lwr	A string giving the column of data containing the primary event lower bound as a datetime. Defaults to NULL which assumes that the variable pdate_lwr is present.
sdate_lwr	A string giving the column of data containing the secondary event lower bound as a datetime. Defaults to NULL which assumes that the variable sdate_lwr is present.
pdate_upr	A string giving the column of data containing the primary event upper bound as a datetime. If this column exists in the data it will be used, otherwise if not supplied then the value of pdate_lwr + 1 day is used.
sdate_upr	A string giving the column of data containing the secondary event upper bound as a datetime. If this column exists in the data it will be used, otherwise if not supplied then the value of sdate_lwr + 1 day is used.
obs_date	A string giving the column of data containing the observation time as a date-time. Optional, if not supplied then the maximum of sdate_upr is used.
...	Additional arguments passed to methods

## See Also

Other linelist\_data: [as\\_epidist\\_linelist\\_data\(\)](#), [as\\_epidist\\_linelist\\_data.default\(\)](#), [assert\\_epidist.epidist\\_linelist\\_data\(\)](#), [is\\_epidist\\_linelist\\_data\(\)](#), [new\\_epidist\\_linelist\\_data\(\)](#)



---

```
as_epidist_linelist_data.default
```

*Create an epidist\_linelist\_data object from vectors of event times*

---

### Description

Create an epidist\_linelist\_data object from vectors of event times

### Usage

```
## Default S3 method:
as_epidist_linelist_data(
  data,
  ptime_upr = NULL,
  stime_lwr = NULL,
  stime_upr = NULL,
  obs_time = NULL,
  ...
)
```

### Arguments

data	Numeric vector giving lower bounds for primary times
ptime_upr	Numeric vector giving upper bounds for primary times
stime_lwr, stime_upr	Numeric vectors giving lower and upper bounds for secondary times
obs_time	Numeric vector giving observation times
...	Additional columns to add to the epidist_linelist_data object

### See Also

Other linelist\_data: [as\\_epidist\\_linelist\\_data\(\)](#), [as\\_epidist\\_linelist\\_data.data.frame\(\)](#), [assert\\_epidist.epidist\\_linelist\\_data\(\)](#), [is\\_epidist\\_linelist\\_data\(\)](#), [new\\_epidist\\_linelist\\_data\(\)](#)

---

```
as_epidist_naive_model
```

*Prepare naive model to pass through to brms*

---

### Description

Prepare naive model to pass through to brms

### Usage

```
as_epidist_naive_model(data)
```

**Arguments**

data                    An object to be converted to the class `epidist_naive_model`

**See Also**

Other `naive_model`: [as\\_epidist\\_naive\\_model.epidist\\_linelist\\_data\(\)](#), [is\\_epidist\\_naive\\_model\(\)](#), [new\\_epidist\\_naive\\_model\(\)](#)

---

`as_epidist_naive_model.epidist_linelist_data`

*The naive model method for `epidist_linelist_data` objects*

---

**Description**

The naive model method for `epidist_linelist_data` objects

**Usage**

```
## S3 method for class 'epidist_linelist_data'
as_epidist_naive_model(data)
```

**Arguments**

data                    An `epidist_linelist_data` object

**See Also**

Other `naive_model`: [as\\_epidist\\_naive\\_model\(\)](#), [is\\_epidist\\_naive\\_model\(\)](#), [new\\_epidist\\_naive\\_model\(\)](#)

---

`epidist`

*Fit epidemiological delay distributions using a brms interface*

---

**Description**

Fit epidemiological delay distributions using a brms interface

**Usage**

```
epidist(data, formula, family, prior, fn, ...)
```

**Arguments**

data	An object with class corresponding to an implemented model.
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter mu, and may optionally be provided for other distributional parameters.
family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <code>brmsfamily()</code> . Commonly used, such as <code>lognormal()</code> , are also reexported as part of <code>epidist</code> .
prior	One or more <code>brmsprior</code> objects created by <code>brms::set_prior()</code> or related functions. These priors are passed to <code>epidist_prior()</code> in the prior argument.
fn	The internal function to be called. By default this is <code>brms::brm()</code> which performs inference for the specified model. Other options are <code>brms::make_stancode()</code> which returns the Stan code for the specified model, or <code>brms::make_standata()</code> which returns the data passed to Stan. These two later options may be useful for model debugging and extensions.
...	Additional arguments passed to fn method.

**See Also**

Other fit: `epidist.default()`

---

`epidist.default`      *Default method used for interface using brms*

---

**Description**

Default method used for interface using brms

**Usage**

```
## Default S3 method:
epidist(
  data,
  formula = mu ~ 1,
  family = lognormal(),
  prior = NULL,
  fn = brms::brm,
  ...
)
```

**Arguments**

data	An object with class corresponding to an implemented model.
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter <code>mu</code> , and may optionally be provided for other distributional parameters.
family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <code>brmsfamily()</code> . Commonly used, such as <code>lognormal()</code> , are also reexported as part of <code>epidist</code> .
prior	One or more <code>brmsprior</code> objects created by <code>brms::set_prior()</code> or related functions. These priors are passed to <code>epidist_prior()</code> in the prior argument.
fn	The internal function to be called. By default this is <code>brms::brm()</code> which performs inference for the specified model. Other options are <code>brms::make_stancode()</code> which returns the Stan code for the specified model, or <code>brms::make_standata()</code> which returns the data passed to Stan. These two later options may be useful for model debugging and extensions.
...	Additional arguments passed to <code>fn</code> method.

**See Also**

Other fit: `epidist()`

---

epidist\_diagnostics     *Diagnostics for epidist\_fit models*

---

**Description**

This function computes diagnostics to assess the quality of a fitted model. When the fitting algorithm used is "sampling" (HMC) then the output of `epidist_diagnostics` is a `data.frame` containing:

- `time`: the total time taken to fit all chains
- `samples`: the total number of samples across all chains
- `max_rhat`: the highest value of the Gelman-Rubin statistic
- `divergent_transitions`: the total number of divergent transitions
- `per_divergent_transitions`: the proportion of samples which had divergent transitions
- `max_treedepth`: the highest value of the treedepth HMC parameter
- `no_at_max_treedepth`: the number of samples which attained the `max_treedepth`
- `per_at_max_treedepth`: the proportion of samples which attained the `max_treedepth`

**Usage**

```
epidist_diagnostics(fit)
```

**Arguments**

`fit`                    A fitted model of class `epidist_fit`

**Details**

When the fitting algorithm is not "sampling" (see `brms::brm()` for other possible algorithms) then diagnostics are yet to be implemented.

---

<code>epidist_family</code>	<i>Define epidist family</i>
-----------------------------	------------------------------

---

**Description**

This function is used within `epidist()` to create a model specific custom brms family object. This custom family is passed to brms. It is unlikely that as a user you will need this function, but we export it nonetheless to be transparent about what happens inside of a call to `epidist()`.

**Usage**

```
epidist_family(data, family = lognormal(), ...)
```

**Arguments**

`data`                    An object with class corresponding to an implemented model.

`family`                  A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see `brmsfamily()`. Commonly used, such as `lognormal()`, are also reexported as part of `epidist`.

`...`                    Additional arguments passed to fn method.

**See Also**

Other family: `epidist_family_model()`, `epidist_family_model.default()`, `epidist_family_reparam()`, `epidist_family_reparam.default()`, `epidist_family_reparam.gamma()`

---

epidist\_family\_model *The model-specific parts of an epidist\_family() call*

---

### Description

The model-specific parts of an epidist\_family() call

The model-specific parts of an epidist\_formula() call

### Usage

```
epidist_family_model(data, family, ...)
```

```
epidist_formula_model(data, formula, ...)
```

### Arguments

data	An object with class corresponding to an implemented model.
family	Output of a call to <code>brms::brmsfamily()</code> with additional information as provided by <code>.add_dpar_info()</code>
...	Additional arguments passed to fn method.
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter <code>mu</code> , and may optionally be provided for other distributional parameters.

### See Also

Other family: [epidist\\_family\(\)](#), [epidist\\_family\\_model.default\(\)](#), [epidist\\_family\\_reparam\(\)](#), [epidist\\_family\\_reparam.default\(\)](#), [epidist\\_family\\_reparam.gamma\(\)](#)

Other formula: [epidist\\_formula\(\)](#), [epidist\\_formula\\_model.default\(\)](#)

---

epidist\_family\_model.default

*Default method for defining a model specific family*

---

### Description

Default method for defining a model specific family

### Usage

```
## Default S3 method:
epidist_family_model(data, family, ...)
```

**Arguments**

data	An object with class corresponding to an implemented model.
family	Output of a call to <code>brms::brmsfamily()</code> with additional information as provided by <code>.add_dpar_info()</code>
...	Additional arguments passed to <code>fn</code> method.

**See Also**

Other family: [epidist\\_family\(\)](#), [epidist\\_family\\_model\(\)](#), [epidist\\_family\\_reparam\(\)](#), [epidist\\_family\\_reparam.gamma\(\)](#)

---

```
epidist_family_model.epidist_latent_model
```

*Create the model-specific component of an epidist custom family*

---

**Description**

Create the model-specific component of an epidist custom family

**Usage**

```
## S3 method for class 'epidist_latent_model'
epidist_family_model(data, family, ...)
```

**Arguments**

data	An object with class corresponding to an implemented model.
family	Output of a call to <code>brms::brmsfamily()</code> with additional information as provided by <code>.add_dpar_info()</code>
...	Additional arguments passed to method.

**See Also**

Other latent\_model: [as\\_epidist\\_latent\\_model\(\)](#), [as\\_epidist\\_latent\\_model.epidist\\_linelist\\_data\(\)](#), [epidist\\_formula\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_gen\\_log\\_lik\\_latent\(\)](#), [is\\_epidist\\_latent\\_model\(\)](#), [new\\_epidist\\_latent\\_model\(\)](#)

---

epidist\_family\_prior *Family specific prior distributions*

---

### Description

This function contains brms prior distributions which are specific to particular likelihood families e.g. `brms::lognormal()`.

### Usage

```
epidist_family_prior(family, ...)
```

### Arguments

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <code>brmsfamily()</code> . Commonly used, such as <code>lognormal()</code> , are also reexported as part of <code>epidist</code> .
...	Additional arguments passed to fn method.

### See Also

Other prior: `epidist_family_prior.default()`, `epidist_family_prior.lognormal()`, `epidist_model_prior()`, `epidist_model_prior.default()`, `epidist_prior()`

---

epidist\_family\_prior.default  
*Default family specific prior distributions*

---

### Description

By default, we do not return any family specific prior distributions.

### Usage

```
## Default S3 method:  
epidist_family_prior(family, formula, ...)
```



**Arguments**

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <a href="#">brmsfamily()</a> . Commonly used, such as <a href="#">lognormal()</a> , are also reexported as part of epidist.
formula	An object of class <a href="#">stats::formula</a> or <a href="#">brms::brmsformula</a> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter mu, and may optionally be provided for other distributional parameters.
...	Additional arguments passed to fn method.

**See Also**

Other prior: [epidist\\_family\\_prior\(\)](#), [epidist\\_family\\_prior.lognormal\(\)](#), [epidist\\_model\\_prior\(\)](#), [epidist\\_model\\_prior.default\(\)](#), [epidist\\_prior\(\)](#)

---

`epidist_family_prior.lognormal`

*Family specific prior distributions for the lognormal family*

---

**Description**

We suggest priors to overwrite the brms defaults for the lognormal family.

**Usage**

```
## S3 method for class 'lognormal'
epidist_family_prior(family, formula, ...)
```

**Arguments**

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <a href="#">brmsfamily()</a> . Commonly used, such as <a href="#">lognormal()</a> , are also reexported as part of epidist.
formula	An object of class <a href="#">stats::formula</a> or <a href="#">brms::brmsformula</a> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter mu, and may optionally be provided for other distributional parameters.
...	Additional arguments passed to fn method.

**See Also**

Other prior: [epidist\\_family\\_prior\(\)](#), [epidist\\_family\\_prior.default\(\)](#), [epidist\\_model\\_prior\(\)](#), [epidist\\_model\\_prior.default\(\)](#), [epidist\\_prior\(\)](#)

---

 epidist\_family\_reparam

*Reparameterise an epidist family to align brms and Stan*


---

### Description

Reparameterise an epidist family to align brms and Stan

### Usage

```
epidist_family_reparam(family, ...)
```

### Arguments

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <a href="#">brmsfamily()</a> . Commonly used, such as <a href="#">lognormal()</a> , are also reexported as part of epidist.
...	Additional arguments passed to fn method.

### See Also

Other family: [epidist\\_family\(\)](#), [epidist\\_family\\_model\(\)](#), [epidist\\_family\\_model.default\(\)](#), [epidist\\_family\\_reparam.default\(\)](#), [epidist\\_family\\_reparam.gamma\(\)](#)

---

 epidist\_family\_reparam.default

*Default method for families which do not require a reparameterisation*


---

### Description

Default method for families which do not require a reparameterisation

### Usage

```
## Default S3 method:
epidist_family_reparam(family, ...)
```

### Arguments

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <a href="#">brmsfamily()</a> . Commonly used, such as <a href="#">lognormal()</a> , are also reexported as part of epidist.
...	Additional arguments passed to fn method.

**See Also**

Other family: [epidist\\_family\(\)](#), [epidist\\_family\\_model\(\)](#), [epidist\\_family\\_model.default\(\)](#), [epidist\\_family\\_reparam\(\)](#), [epidist\\_family\\_reparam.gamma\(\)](#)

---

epidist\_family\_reparam.gamma

*Reparameterisation for the gamma family*

---

**Description**

Reparameterisation for the gamma family

**Usage**

```
## S3 method for class 'gamma'
epidist_family_reparam(family, ...)
```

**Arguments**

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <a href="#">brmsfamily()</a> . Commonly used, such as <a href="#">lognormal()</a> , are also reexported as part of <a href="#">epidist</a> .
...	Additional arguments passed to fn method.

**See Also**

Other family: [epidist\\_family\(\)](#), [epidist\\_family\\_model\(\)](#), [epidist\\_family\\_model.default\(\)](#), [epidist\\_family\\_reparam\(\)](#), [epidist\\_family\\_reparam.default\(\)](#)

---

epidist\_formula

*Define a model specific formula*

---

**Description**

This function is used within [epidist\(\)](#) to create the formula object passed to [brms](#). It is unlikely that as a user you will need this function, but we export it nonetheless to be transparent about what exactly is happening inside of a call to [epidist\(\)](#).

**Usage**

```
epidist_formula(data, family, formula, ...)
```

**Arguments**

data	An object with class corresponding to an implemented model.
family	A description of the response distribution and link function to be used in the model created using <code>epidist_family()</code> .
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter $\mu$ , and may optionally be provided for other distributional parameters.
...	Additional arguments passed to <code>fn</code> method.

**See Also**

Other formula: `epidist_family_model()`, `epidist_formula_model.default()`

---

`epidist_formula_model.default`

*Default method for defining a model specific formula*

---

**Description**

Default method for defining a model specific formula

**Usage**

```
## Default S3 method:
epidist_formula_model(data, formula, ...)
```

**Arguments**

data	An object with class corresponding to an implemented model.
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter $\mu$ , and may optionally be provided for other distributional parameters.
...	Additional arguments passed to <code>fn</code> method.

**See Also**

Other formula: `epidist_family_model()`, `epidist_formula()`

---

```
epidist_formula_model.epidist_latent_model
```

*Define the model-specific component of an epidist custom formula*

---

### Description

Define the model-specific component of an epidist custom formula

### Usage

```
## S3 method for class 'epidist_latent_model'
epidist_formula_model(data, formula, ...)
```

### Arguments

data	An object with class corresponding to an implemented model.
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter $\mu$ , and may optionally be provided for other distributional parameters.
...	Additional arguments passed to method.

### See Also

Other latent\_model: `as_epidist_latent_model()`, `as_epidist_latent_model.epidist_linelist_data()`, `epidist_family_model.epidist_latent_model()`, `epidist_gen_log_lik_latent()`, `is_epidist_latent_model()`, `new_epidist_latent_model()`

---

```
epidist_gen_log_lik_latent
```

*Create a function to calculate the pointwise log likelihood of the latent model*

---

### Description

This function creates a log likelihood function that accounts for the latent variables in the model, including primary and secondary event windows and their overlap. The returned function calculates the log likelihood for a single observation by augmenting the data with the latent variables and using the underlying brms log likelihood function.

### Usage

```
epidist_gen_log_lik_latent(family)
```

**Arguments**

family A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see [brmsfamily\(\)](#). Commonly used, such as [lognormal\(\)](#), are also reexported as part of `epidist`.

**Value**

A function that calculates the log likelihood for a single observation. The prep object must have the following variables:

- `vreal1`: relative observation time
- `vreal2`: primary event window
- `vreal3`: secondary event window

**See Also**

[brms::log\\_lik\(\)](#) for details on the brms log likelihood interface.

Other latent\_model: [as\\_epidist\\_latent\\_model\(\)](#), [as\\_epidist\\_latent\\_model.epidist\\_linelist\\_data\(\)](#), [epidist\\_family\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_formula\\_model.epidist\\_latent\\_model\(\)](#), [is\\_epidist\\_latent\\_model\(\)](#), [new\\_epidist\\_latent\\_model\(\)](#)

---

epidist\_gen\_posterior\_epred

*Create a function to draw from the expected value of the posterior predictive distribution for a latent model*

---

**Description**

This function creates a function that calculates the expected value of the posterior predictive distribution for a latent model. The returned function takes a prep argument (from `brms`) and returns posterior expected values. This is used internally by [brms::posterior\\_epred\(\)](#) to calculate expected values for latent models.

**Usage**

```
epidist_gen_posterior_epred(family)
```

**Arguments**

family A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see [brmsfamily\(\)](#). Commonly used, such as [lognormal\(\)](#), are also reexported as part of `epidist`.

**Value**

A function that takes a prep argument from brms and returns a matrix of posterior expected values, with one row per posterior draw and one column per observation.

**See Also**

[brms::posterior\\_epred\(\)](#) for details on how this is used within brms.

Other gen: [epidist\\_gen\\_posterior\\_predict\(\)](#)

epidist\_gen\_posterior\_predict

*Create a function to draw from the posterior predictive distribution for a latent model*

**Description**

This function creates a function that draws from the posterior predictive distribution for a latent model using [primarycensored::rpcens\(\)](#) to handle censoring and truncation. The returned function takes a prep argument from brms and returns posterior predictions. This is used internally by [brms::posterior\\_predict\(\)](#) to generate predictions for latent models.

**Usage**

```
epidist_gen_posterior_predict(family)
```

**Arguments**

family	A description of the response distribution and link function to be used in the model. Every family function has a link argument allowing users to specify the link function to be applied on the response variable. If not specified, default links are used. For details of all supported families see <a href="#">brmsfamily()</a> . Commonly used, such as <a href="#">lognormal()</a> , are also reexported as part of epidist.
--------	---

**Value**

A function that takes a prep argument from brms and returns a matrix of posterior predictions, with one row per posterior draw and one column per observation. The prep object must have the following variables:

- vreal1: relative observation time
- vreal2: primary event window
- vreal3: secondary event window

**See Also**

[brms::posterior\\_predict\(\)](#) for details on how this is used within brms, [primarycensored::rpcens\(\)](#) for details on the censoring approach

Other gen: [epidist\\_gen\\_posterior\\_epred\(\)](#)

---

epidist\_model\_prior    *Model specific prior distributions*

---

### Description

This function contains brms prior distributions which are specific to particular epidist models e.g. the latent\_lognormal model.

### Usage

```
epidist_model_prior(data, ...)
```

### Arguments

data	An object with class corresponding to an implemented model.
...	Additional arguments passed to fn method.

### See Also

Other prior: [epidist\\_family\\_prior\(\)](#), [epidist\\_family\\_prior.default\(\)](#), [epidist\\_family\\_prior.lognormal\(\)](#), [epidist\\_model\\_prior.default\(\)](#), [epidist\\_prior\(\)](#)

---

epidist\_model\_prior.default  
*Default model specific prior distributions*

---

### Description

By default, we do not return any model specific prior distributions.

### Usage

```
## Default S3 method:
epidist_model_prior(data, formula, ...)
```

### Arguments

data	An object with class corresponding to an implemented model.
formula	An object of class <code>stats::formula</code> or <code>brms::brmsformula</code> (or one that can be coerced to those classes). A symbolic description of the model to be fitted. A formula must be provided for the distributional parameter $\mu$ , and may optionally be provided for other distributional parameters.
...	Additional arguments passed to fn method.



**See Also**

Other prior: [epidist\\_family\\_prior\(\)](#), [epidist\\_family\\_prior.default\(\)](#), [epidist\\_family\\_prior.lognormal\(\)](#), [epidist\\_model\\_prior\(\)](#), [epidist\\_prior\(\)](#)

---

epidist_prior	<i>Define prior distributions using brms defaults, model specific priors, family specific priors, and user provided priors</i>
---------------	--

---

**Description**

This function obtains the brms default prior distributions for a particular model, then replaces these prior distributions using:

1. Model specific prior distributions from [epidist\\_model\\_prior\(\)](#)
2. Family specific prior distributions from [epidist\\_family\\_prior\(\)](#)
3. User provided prior distributions Each element of this list overwrites previous elements, such that user provided prior distribution have the highest priority. At the third stage, if a prior distribution is provided which is not included in the model, then a warning will be shown. To prevent this warning, do not pass prior distributions for parameters which are not in the model.

**Usage**

```
epidist_prior(data, family, formula, prior)
```

**Arguments**

data	An object with class corresponding to an implemented model.
family	A description of the response distribution and link function to be used in the model created using <a href="#">epidist_family()</a> .
formula	A symbolic description of the model to be fitted created using <a href="#">epidist_formula()</a> .
prior	One or more brmsprior objects created by <a href="#">brms::set_prior()</a> or related functions. These priors are passed to <a href="#">epidist_prior()</a> in the prior argument.

**See Also**

Other prior: [epidist\\_family\\_prior\(\)](#), [epidist\\_family\\_prior.default\(\)](#), [epidist\\_family\\_prior.lognormal\(\)](#), [epidist\\_model\\_prior\(\)](#), [epidist\\_model\\_prior.default\(\)](#)

---

epidist\_stancode      *Define model specific Stan code*

---

### Description

This function is used within `epidist()` to create any custom Stan code which is injected into brms via the `stanvars` argument. It is unlikely that as a user you will need this function, but we export it nonetheless to be transparent about what exactly is happening inside of a call to `epidist()`.

### Usage

```
epidist_stancode(data, ...)
```

### Arguments

<code>data</code>	An object with class corresponding to an implemented model.
<code>...</code>	Additional arguments passed to fn method.

### See Also

Other stan: `epidist_stancode.default()`

---

epidist\_stancode.default  
*Default method for defining model specific Stan code*

---

### Description

Default method for defining model specific Stan code

### Usage

```
## Default S3 method:
epidist_stancode(data, ...)
```

### Arguments

<code>data</code>	An object with class corresponding to an implemented model.
<code>...</code>	Additional arguments passed to fn method.

### See Also

Other stan: `epidist_stancode()`

---

`is_epidist_latent_model`*Check if data has the epidist\_latent\_model class*

---

**Description**

Check if data has the `epidist_latent_model` class

**Usage**

```
is_epidist_latent_model(data)
```

**Arguments**

`data`            An object

**See Also**

Other `latent_model`: [as\\_epidist\\_latent\\_model\(\)](#), [as\\_epidist\\_latent\\_model.epidist\\_linelist\\_data\(\)](#), [epidist\\_family\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_formula\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_gen\\_log\\_lik\\_latent\(\)](#), [new\\_epidist\\_latent\\_model\(\)](#)

---

`is_epidist_linelist_data`*Check if data has the epidist\_linelist\_data class*

---

**Description**

Check if data has the `epidist_linelist_data` class

**Usage**

```
is_epidist_linelist_data(data, ...)
```

**Arguments**

`data`            The data to convert  
`...`            Additional arguments

**See Also**

Other `linelist_data`: [as\\_epidist\\_linelist\\_data\(\)](#), [as\\_epidist\\_linelist\\_data.data.frame\(\)](#), [as\\_epidist\\_linelist\\_data.default\(\)](#), [assert\\_epidist.epidist\\_linelist\\_data\(\)](#), [new\\_epidist\\_linelist\\_data\(\)](#)

---

is\_epidist\_naive\_model

*Check if data has the epidist\_naive\_model class*

---

**Description**

Check if data has the epidist\_naive\_model class

**Usage**

```
is_epidist_naive_model(data)
```

**Arguments**

data            An object

**See Also**

Other naive\_model: [as\\_epidist\\_naive\\_model\(\)](#), [as\\_epidist\\_naive\\_model.epidist\\_linelist\\_data\(\)](#), [new\\_epidist\\_naive\\_model\(\)](#)

---

new\_epidist\_latent\_model

*Class constructor for epidist\_latent\_model objects*

---

**Description**

Class constructor for epidist\_latent\_model objects

**Usage**

```
new_epidist_latent_model(data)
```

**Arguments**

data            An object to be set with the class epidist\_latent\_model

**Value**

An object of class epidist\_latent\_model

**See Also**

Other latent\_model: [as\\_epidist\\_latent\\_model\(\)](#), [as\\_epidist\\_latent\\_model.epidist\\_linelist\\_data\(\)](#), [epidist\\_family\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_formula\\_model.epidist\\_latent\\_model\(\)](#), [epidist\\_gen\\_log\\_lik\\_latent\(\)](#), [is\\_epidist\\_latent\\_model\(\)](#)

---

`new_epidist_linelist_data`*Class constructor for epidist\_linelist\_data objects*

---

**Description**

Class constructor for epidist\_linelist\_data objects

**Usage**

```
new_epidist_linelist_data(data)
```

**Arguments**

`data`            A data.frame to convert

**Value**

An object of class epidist\_linelist\_data

**See Also**

Other linelist\_data: [as\\_epidist\\_linelist\\_data\(\)](#), [as\\_epidist\\_linelist\\_data.data.frame\(\)](#),  
[as\\_epidist\\_linelist\\_data.default\(\)](#), [assert\\_epidist.epidist\\_linelist\\_data\(\)](#), [is\\_epidist\\_linelist\\_data\(\)](#)

---

`new_epidist_naive_model`*Class constructor for epidist\_naive\_model objects*

---

**Description**

Class constructor for epidist\_naive\_model objects

**Usage**

```
new_epidist_naive_model(data)
```

**Arguments**

`data`            An object to be set with the class epidist\_naive\_model

**Value**

An object of class epidist\_naive\_model

**See Also**

Other naive\_model: [as\\_epidist\\_naive\\_model\(\)](#), [as\\_epidist\\_naive\\_model.epidist\\_linelist\\_data\(\)](#), [is\\_epidist\\_naive\\_model\(\)](#)

---

`predict_delay_parameters`

*Extract samples of the delay distribution parameters*

---

**Description**

Extract samples of the delay distribution parameters

**Usage**

```
predict_delay_parameters(fit, newdata = NULL, ...)
```

```
predict_dpar(fit, newdata = NULL, ...)
```

**Arguments**

<code>fit</code>	A model fit with <a href="#">epidist()</a>
<code>newdata</code>	An optional data.frame for which to evaluate predictions. If NULL (default), the original data of the model is used. NA values within factors (excluding grouping variables) are interpreted as if all dummy variables of this factor are zero. This allows, for instance, to make predictions of the grand mean when using sum coding. NA values within grouping variables are treated as a new level.
<code>...</code>	Additional arguments passed to <a href="#">brms::prepare_predictions()</a>

**See Also**

Other postprocess: [add\\_mean\\_sd\(\)](#), [add\\_mean\\_sd.default\(\)](#), [add\\_mean\\_sd.gamma\\_samples\(\)](#), [add\\_mean\\_sd.lognormal\\_samples\(\)](#)

---

`sierra_leoneEbola_data`

*Ebola linelist data from Fang et al. (2016)*

---

**Description**

Linelist data for the Ebola virus collected in Sierra Leone. If you use this data in your work, please cite the corresponding paper.

**Usage**

```
sierra_leoneEbola_data
```

**Format**

A tibble with 8,358 rows and 8 columns:

**id** Unique identification number for the case  
**name** Name as character, omitted  
**age** Age as numeric  
**sex** Sex as character, either "F", "M" or NA  
**date\_of\_symptom\_onset** The date symptoms began  
**date\_of\_sample\_tested** The date the sample was tested  
**district** The district (ADM2)  
**chiefdom** The chiefdom (ADM3)

**Source**

<https://www.pnas.org/doi/full/10.1073/pnas.1518587113>

---

simulate\_exponential\_cases  
*Simulate exponential cases*

---

**Description**

This function simulates cases from an exponential distribution. The user may specify the rate parameter  $r$ , the sample size, and the upper bound of the survival time. If the rate parameter is 0, then this function defaults to the uniform distribution.

**Usage**

```
simulate_exponential_cases(r = 0.2, sample_size = 10000, seed, t = 30)
```

**Arguments**

<code>r</code>	The exponential growth rate parameter. Defaults to 0.2.
<code>sample_size</code>	The number of cases to simulate. Defaults to 10000.
<code>seed</code>	The random seed to be used in the simulation process.
<code>t</code>	Upper bound of the survival time. Defaults to 30.

**Value**

A data.frame with two columns: case (case number) and ptime (primary event time).

**See Also**

Other simulate: [simulate\\_gillespie\(\)](#), [simulate\\_secondary\(\)](#), [simulate\\_uniform\\_cases\(\)](#)

---

simulate\_gillespie      *Simulate cases from a stochastic SIR model*

---

### Description

This function simulates cases from an stochastic SIR model. The user may specify the initial epidemic growth rate  $r$ , the rate of recovery  $\gamma$ , the initial number of infected cases  $I_0$ , and the total population size  $N$ .

### Usage

```
simulate_gillespie(r = 0.2, gamma = 1/7, I0 = 50, N = 10000, seed)
```

### Arguments

<code>r</code>	The initial epidemic growth rate. Defaults to 0.2.
<code>gamma</code>	The rate of recovery. Defaults to 1/7.
<code>I0</code>	The initial number of infected people. Defaults to 50.
<code>N</code>	The total population size. Defaults to 10000.
<code>seed</code>	The random seed to be used in the simulation process.

### Value

A data.frame with two columns: case (case number) and ptime (primary event time).

### See Also

Other simulate: [simulate\\_exponential\\_cases\(\)](#), [simulate\\_secondary\(\)](#), [simulate\\_uniform\\_cases\(\)](#)

---

simulate\_secondary      *Simulate secondary events based on a delay distribution*

---

### Description

This function simulates secondary events based on a given delay distribution. The input dataset should have the primary event times in a column named ptime.

### Usage

```
simulate_secondary(data, dist = rlnorm, ...)
```

### Arguments

<code>data</code>	A data frame with the primary event times.
<code>dist</code>	The delay distribution to be used. Defaults to <a href="#">rlnorm()</a> .
<code>...</code>	Arguments to be passed to the delay distribution function.



**Value**

A `data.frame` that augments data with two new columns: `delay` (secondary event latency) and `stime` (the time of the secondary event).

**See Also**

Other simulate: [simulate\\_exponential\\_cases\(\)](#), [simulate\\_gillespie\(\)](#), [simulate\\_uniform\\_cases\(\)](#)

---

`simulate_uniform_cases`

*Simulate cases from a uniform distribution*

---

**Description**

This function simulates cases from a uniform distribution, where the primary event times are uniformly distributed between 0 and `t`.

**Usage**

```
simulate_uniform_cases(sample_size = 1000, t = 60)
```

**Arguments**

<code>sample_size</code>	The number of cases to simulate.
<code>t</code>	Upper bound of the uniform distribution to generate primary event times.

**Value**

A `data.frame` with two columns: `case` (case number) and `ptime` (primary event time).

**See Also**

Other simulate: [simulate\\_exponential\\_cases\(\)](#), [simulate\\_gillespie\(\)](#), [simulate\\_secondary\(\)](#)

# Index

- \* **assert**
    - assert\_epidist, 5
  - \* **datasets**
    - sierra\_leone Ebola\_data, 30
  - \* **data**
    - sierra\_leone Ebola\_data, 30
  - \* **diagnostics**
    - epidist\_diagnostics, 12
  - \* **family**
    - epidist\_family, 13
    - epidist\_family\_model, 14
    - epidist\_family\_model.default, 14
    - epidist\_family\_reparam, 18
    - epidist\_family\_reparam.default, 18
    - epidist\_family\_reparam.gamma, 19
  - \* **fit**
    - epidist, 10
    - epidist.default, 11
  - \* **formula**
    - epidist\_family\_model, 14
    - epidist\_formula, 19
    - epidist\_formula\_model.default, 20
  - \* **gen**
    - epidist\_gen\_posterior\_epred, 22
    - epidist\_gen\_posterior\_predict, 23
  - \* **latent\_model**
    - as\_epidist\_latent\_model, 6
    - as\_epidist\_latent\_model.epidist\_linelist\_data, 7
    - epidist\_family\_model.epidist\_latent\_model, 15
    - epidist\_formula\_model.epidist\_latent\_model, 21
    - epidist\_gen\_log\_lik\_latent, 21
    - is\_epidist\_latent\_model, 27
    - new\_epidist\_latent\_model, 28
  - \* **linelist\_data**
    - as\_epidist\_linelist\_data, 7
    - as\_epidist\_linelist\_data.data.frame, 8
    - as\_epidist\_linelist\_data.default, 9
    - assert\_epidist.epidist\_linelist\_data, 6
    - is\_epidist\_linelist\_data, 27
    - new\_epidist\_linelist\_data, 29
  - \* **naive\_model**
    - as\_epidist\_naive\_model, 9
    - as\_epidist\_naive\_model.epidist\_linelist\_data, 10
    - is\_epidist\_naive\_model, 28
    - new\_epidist\_naive\_model, 29
  - \* **postprocess**
    - add\_mean\_sd, 3
    - add\_mean\_sd.default, 4
    - add\_mean\_sd.gamma\_samples, 4
    - add\_mean\_sd.lognormal\_samples, 5
    - predict\_delay\_parameters, 30
  - \* **prior**
    - epidist\_family\_prior, 16
    - epidist\_family\_prior.default, 16
    - epidist\_family\_prior.lognormal, 17
    - epidist\_model\_prior, 24
    - epidist\_model\_prior.default, 24
    - epidist\_prior, 25
  - \* **simulate**
    - simulate\_exponential\_cases, 31
    - simulate\_gillespie, 32
    - simulate\_secondary, 32
    - simulate\_uniform\_cases, 33
  - \* **stan**
    - epidist\_stancode, 26
    - epidist\_stancode.default, 26
- add\_mean\_sd, 3, 4, 5, 30  
add\_mean\_sd.default, 3, 4, 4, 5, 30  
add\_mean\_sd.gamma\_samples, 3, 4, 4, 5, 30  
add\_mean\_sd.lognormal\_samples, 3, 4, 5, 30

- as\_epidist\_latent\_model, 6, 7, 15, 21, 22, 27, 28
- as\_epidist\_latent\_model.epidist\_linelist\_data, 6, 7, 15, 21, 22, 27, 28
- as\_epidist\_linelist\_data, 6, 7, 8, 9, 27, 29
- as\_epidist\_linelist\_data.data.frame, 6, 7, 8, 9, 27, 29
- as\_epidist\_linelist\_data.default, 6–8, 9, 27, 29
- as\_epidist\_naive\_model, 9, 10, 28, 30
- as\_epidist\_naive\_model.epidist\_linelist\_data, 10, 10, 28, 30
- assert\_epidist, 5
- assert\_epidist.epidist\_linelist\_data, 6, 7–9, 27, 29
  
- brms::brm(), 11–13
- brms::brmsformula, 11, 12, 14, 17, 20, 21, 24
- brms::log\_lik(), 22
- brms::lognormal(), 16
- brms::make\_stancode(), 11, 12
- brms::make\_standata(), 11, 12
- brms::posterior\_epred(), 22, 23
- brms::posterior\_predict(), 23
- brms::prepare\_predictions(), 30
- brms::set\_prior(), 11, 12, 25
- brmsfamily(), 11–13, 16–19, 22, 23
  
- epidist, 10, 12
- epidist(), 13, 19, 26, 30
- epidist.default, 11, 11
- epidist\_diagnostics, 12
- epidist\_family, 13, 14, 15, 18, 19
- epidist\_family(), 20, 25
- epidist\_family\_model, 13, 14, 15, 18–20
- epidist\_family\_model.default, 13, 14, 14, 18, 19
- epidist\_family\_model.epidist\_latent\_model, 6, 7, 15, 21, 22, 27, 28
- epidist\_family\_prior, 16, 17, 24, 25
- epidist\_family\_prior(), 25
- epidist\_family\_prior.default, 16, 16, 17, 24, 25
- epidist\_family\_prior.lognormal, 16, 17, 17, 24, 25
- epidist\_family\_reparam, 13–15, 18, 19
- epidist\_family\_reparam.default, 13–15, 18, 18, 19
- epidist\_family\_reparam.gamma, 13–15, 18, 19, 19
- epidist\_formula, 14, 19, 20
- epidist\_formula(), 25
- epidist\_formula\_model (epidist\_family\_model), 14
- epidist\_formula\_model.default, 14, 20, 20
- epidist\_formula\_model.epidist\_latent\_model, 6, 7, 15, 21, 22, 27, 28
- epidist\_gen\_log\_lik\_latent, 6, 7, 15, 21, 21, 27, 28
- epidist\_gen\_posterior\_epred, 22, 23
- epidist\_gen\_posterior\_predict, 23, 23
- epidist\_model\_prior, 16, 17, 24, 25
- epidist\_model\_prior(), 25
- epidist\_model\_prior.default, 16, 17, 24, 24, 25
- epidist\_prior, 16, 17, 24, 25, 25
- epidist\_prior(), 11, 12, 25
- epidist\_stancode, 26, 26
- epidist\_stancode.default, 26, 26
  
- is\_epidist\_latent\_model, 6, 7, 15, 21, 22, 27, 28
- is\_epidist\_linelist\_data, 6–9, 27, 29
- is\_epidist\_naive\_model, 10, 28, 30
  
- lognormal(), 11–13, 16–19, 22, 23
  
- new\_epidist\_latent\_model, 6, 7, 15, 21, 22, 27, 28
- new\_epidist\_linelist\_data, 6–9, 27, 29
- new\_epidist\_naive\_model, 10, 28, 29
  
- predict\_delay\_parameters, 3–5, 30
- predict\_dpar (predict\_delay\_parameters), 30
- primarycensored::rpcens(), 23
  
- rlnorm(), 32
  
- sierra\_leone\_ebola\_data, 30
- simulate\_exponential\_cases, 31, 32, 33
- simulate\_gillespie, 31, 32, 33
- simulate\_secondary, 31, 32, 32, 33
- simulate\_uniform\_cases, 31–33, 33
- stats::formula, 11, 12, 14, 17, 20, 21, 24