

Package: nfidd (via r-universe)

September 27, 2026

Title Material to support course on nowcasting and forecasting of infectious disease dynamics

Version 1.3.1

Description Resources to support a short course on nowcasting and forecasting of infectious disease dynamics.

License MIT + file LICENSE

URL <https://github.com/nfidd/nfidd>

BugReports <https://github.com/nfidd/nfidd/issues>

Depends R (>= 4.2.0)

Imports cmdstanr, dplyr, tidyr, tidybayes

Suggests bayesplot, covr, ggplot2, knitr, lintr, lubridate, posterior, purrr, rmarkdown, scoringutils (>= 2.0.0), testthat (>= 3.0.0), rensemble (>= 0.1.0), lopensemble (>= 0.1.0), roxyglobals

Additional_repositories <https://production.r-multiverse.org/2025-12-15>,
<https://epiforecasts.r-universe.dev>

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

LazyData true

LazyDataCompression xz

Roxygen list(roclets = c("`collate", "`namespace", "`rd",
 "`roxyglobals::global_roclet"))

Config/roxyglobals/filename globals.R

Config/roxyglobals/unique FALSE

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

Date/Publication 2026-06-30 09:06:35 UTC

RemoteUrl <https://github.com/nfidd/nfidd>

RemoteRef v1.3.1

RemoteSha 5ce221ee0140a3785d4bae066c23c7b277457dce

Contents

add_delays	2
censored_delay_pmf	3
condition_onsets_by_report	4
convolve_with_delay	5
geometric_diff_ar	5
geometric_random_walk	6
infection_times	6
make_daily_infections	7
make_gen_time_pmf	7
make_ip_pmf	8
mech_forecasts	8
nfidd_cmdstan_model	9
nfidd_load_stan_functions	10
nfidd_sample	11
nfidd_stan_function_files	12
nfidd_stan_functions	12
nfidd_stan_models	13
nfidd_stan_path	13
renewal	14
rw_forecasts	14
simulate_onsets	15
stat_forecasts	16
summarise_lognormal	16
Index	18

add_delays	<i>Simulate symptom onset and hospitalization times from infection times</i>
------------	--

Description

Simulate symptom onset and hospitalization times from infection times

Usage

```
add_delays(
  infection_times,
  onset_fun = rgamma,
  onset_params = list(shape = 5, rate = 1),
  hosp_fun = rlnorm,
  hosp_params = list(meanlog = 1.75, sdlog = 0.5),
  hosp_prob = 0.3
)
```

Arguments

infection_times	A data frame containing a column of infection times
onset_fun	Function for onset delay distribution (default rgamma)
onset_params	List of parameters for onset distribution
hosp_fun	Function for hospitalization delay distribution (default rlnorm)
hosp_params	List of parameters for hospitalization distribution
hosp_prob	Probability of hospitalization (default 0.3)

Value

A data frame with columns for infection time, onset time, and hospitalization time (with (1-hosp_prob) of hospitalizations set to NA)

Examples

```
# Default parameters
delayed_infections <- add_delays(infection_times)
head(delayed_infections)

# Change delay parameters
delayed_infections_long <- add_delays(
  infection_times,
  hosp_params = list(meanlog = 2.0, sdlog = 0.5)
)

# Use different distributions
delayed_infections_gamma <- add_delays(
  infection_times,
  hosp_fun = rgamma,
  hosp_params = list(shape = 2, rate = 0.5)
)
```

censored_delay_pmf *Discretise a Continuous Delay Distribution*

Description

This function discretises a continuous delay distribution into a probability mass function (PMF) over discrete days.

Usage

```
censored_delay_pmf(rngen, max, n = 1e+06, ...)
```

Arguments

rgen	A function that generates random delays, e.g., 'rgamma', 'rlnorm'.
max	The maximum delay.
n	The number of replicates to simulate. Defaults to '1e+6'.
...	Additional parameters of the delay distribution.

Value

A vector of probabilities corresponding to discrete indices from '0' to 'max', representing the discretised delay distribution.

Examples

```
censored_delay_pmf(rgen = rgamma, max = 14, shape = 5, rate = 1)
```

condition_onsets_by_report

Reduces the number of symptom onsets by a delay distribution.

Description

This function assumes that symptom onsets are reported with a delay. It reduces the number of symptom onsets by this delay, assuming that the last element of the symptom onset vector is the day of reporting.

Usage

```
condition_onsets_by_report(onsets, delay)
```

Arguments

onsets	vector of symptom onsets
delay	vector representing the cumulative distribution function (CDF) of the (discrete) delay distribution. The first element corresponds to a delay of 0, i.e. symptom onsets reported on the same day.

Value

A vector of symptom onsets reduced by the delay distribution.

convolve_with_delay	<i>Convolve a Time Series with a Delay Distribution</i>
---------------------	---

Description

This function convolves a time series with a delay distribution given as a probability mass function (PMF).

Usage

```
convolve_with_delay(ts, delay_pmf)
```

Arguments

ts	A vector of the time series to convolve.
delay_pmf	The probability mass function of the delay, given as a vector of probabilities, corresponding to discrete indices 0, 1, 2 of the discretised delay distribution.

Value

A vector of the convolved time series.

Examples

```
convolve_with_delay(ts = c(10, 14, 10, 10), delay_pmf = c(0.1, 0.6, 0.3))
```

geometric_diff_ar	<i>Geometric Differenced Autoregressive Process</i>
-------------------	---

Description

This function generates a geometric differenced autoregressive process.

Usage

```
geometric_diff_ar(init, noise, std, damp)
```

Arguments

init	The initial value.
noise	A vector of steps (on the standard normal scale).
std	The step size of the random walk.
damp	A damping parameter.

Value

A vector of the generated geometric differenced autoregressive process.

Examples

```
geometric_diff_ar(init = 1, noise = rnorm(100), std = 0.1, damp = 0.1)
```

```
geometric_random_walk
```

Geometric Random Walk

Description

This function generates a geometric random walk.

Usage

```
geometric_random_walk(init, noise, std)
```

Arguments

<code>init</code>	The initial value.
<code>noise</code>	A vector of steps (on the standard normal scale).
<code>std</code>	The step size of the random walk.

Value

A vector of the generated geometric random walk.

Examples

```
geometric_random_walk(init = 1, noise = rnorm(100), std = 0.1)
```

```
infection_times
```

Infection times

Description

A dataset containing random infection times from a branching process model, generated using the code in `data-raw/epicurve.r`

Usage

```
infection_times
```

Format

A data frame with a single column

infection_time the times at which individuals were infected (and became infectious)

make_daily_infections *Convert infection times to a daily time series*

Description

Convert infection times to a daily time series

Usage

```
make_daily_infections(infection_times)
```

Arguments

infection_times
A data frame containing a column of infection times

Value

A data frame with columns infection_day and infections, containing the daily count of infections

Examples

```
make_daily_infections(infection_times)
```

make_gen_time_pmf *Generate a probability mass function for the generation time*

Description

Generate a probability mass function for the generation time

Usage

```
make_gen_time_pmf(max = 14, shape = 4, rate = 1)
```

Arguments

max	Maximum delay to consider
shape	Shape parameter for the gamma distribution
rate	Rate parameter for the gamma distribution

Value

A vector of probabilities representing the generation time PMF

make_ip_pmf	<i>Generate a probability mass function for the incubation period</i>
-------------	---

Description

Generate a probability mass function for the incubation period

Usage

```
make_ip_pmf(max = 14, shape = 5, rate = 1)
```

Arguments

max	Maximum delay to consider
shape	Shape parameter for the gamma distribution
rate	Rate parameter for the gamma distribution

Value

A vector of probabilities representing the incubation period PMF

mech_forecasts	<i>Forecasts from a mechanistic model</i>
----------------	---

Description

A dataset containing forecasts from a mechanistic model, generated using the code in data-raw/generate-example-forecasts.r

Usage

```
mech_forecasts
```

Format

A [tibble::tibble()] with a 7 columns

day the day for which the forecast was made

.draw an ID of the posterior predictive sample

.variable name of the variable

.value predicted value

.horizon the forecast horizon in days

origin_day the day on which the forecast was made (using data up to this day)

model the name of the model

`nfidd_cmdstan_model` *Create a CmdStanModel with NFIDD Stan functions*

Description

This function creates a `CmdStanModel` object using either a specified Stan model from the NFIDD package or a custom Stan file provided by the user.

Usage

```
nfidd_cmdstan_model(  
  model_name = NULL,  
  model_file = NULL,  
  include_paths = getOption("nfidd.stan_path", nfidd::nfidd_stan_path()),  
  ...  
)
```

Arguments

<code>model_name</code>	Character string specifying which Stan model to use from the NFIDD package. Ignored if <code>model_file</code> is provided.
<code>model_file</code>	Character string specifying the path to a custom Stan file. If provided, this takes precedence over <code>model_name</code> .
<code>include_paths</code>	Character vector of paths to include for Stan compilation. Defaults to the result of <code>'nfidd_stan_path()'</code> or can be overridden using the R option "nfidd.stan_path".
<code>...</code>	Additional arguments passed to <code>cmdstanr::cmdstan_model()</code> .

Value

A `CmdStanModel` object.

See Also

Other modelhelpers: [nfidd_sample\(\)](#)

Examples

```
if (!is.null(cmdstanr::cmdstan_version(error_on_NA = FALSE))) {  
  # Using a model from the NFIDD package  
  model <- nfidd_cmdstan_model("simple-nowcast", compile = FALSE)  
  model  
  
}
```

`nfidd_load_stan_functions`*Load Stan functions as a string*

Description

Load Stan functions as a string

Usage

```
nfidd_load_stan_functions(  
  functions = NULL,  
  stan_path = nfidd::nfidd_stan_path(),  
  wrap_in_block = FALSE,  
  write_to_file = FALSE,  
  output_file = "nfidd_functions.stan"  
)
```

Arguments

<code>functions</code>	Character vector of function names to load. Defaults to all functions.
<code>stan_path</code>	Character string, the path to the Stan code. Defaults to the path to the Stan code in the <code>nfidd</code> package.
<code>wrap_in_block</code>	Logical, whether to wrap the functions in a ‘functions’ block. Default is FALSE.
<code>write_to_file</code>	Logical, whether to write the output to a file. Default is FALSE.
<code>output_file</code>	Character string, the path to write the output file if <code>write_to_file</code> is TRUE. Defaults to "nfidd_functions.stan".

Value

A character string containing the requested Stan functions

See Also

Other stantools: [nfidd_stan_function_files\(\)](#), [nfidd_stan_functions\(\)](#), [nfidd_stan_path\(\)](#)

nfidd_sample	<i>Sample from a CmdStanModel with NFIDD course defaults</i>
--------------	--

Description

This function wraps the cmdstanr sample method with optimized defaults for course use to speed up model fitting. All cmdstanr sample arguments can still be overridden for experimentation.

Usage

```
nfidd_sample(
  model,
  iter_warmup = 500,
  iter_sampling = 500,
  parallel_chains = 4,
  save_warmup = FALSE,
  ...
)
```

Arguments

model	A CmdStanModel object to sample from.
iter_warmup	Integer, number of warmup iterations per chain. Defaults to 500 (reduced from cmdstanr default of 1000) for course speed.
iter_sampling	Integer, number of sampling iterations per chain. Defaults to 500 (reduced from cmdstanr default of 1000) for course speed.
parallel_chains	Integer, number of chains to run in parallel. Defaults to 4 for course speed.
save_warmup	Logical, whether to save warmup samples. Defaults to FALSE for course speed.
...	Additional arguments passed to the model's sample method. All cmdstanr sample arguments are supported.

Value

A CmdStanMCMC object containing the posterior samples.

See Also

Other modelhelpers: [nfidd_cmdstan_model\(\)](#)

nfidd_stan_function_files

Get Stan files containing specified functions

Description

This function retrieves Stan files from a specified directory, optionally filtering for files that contain specific functions.

Usage

```
nfidd_stan_function_files(
  functions = NULL,
  stan_path = nfidd::nfidd_stan_path()
)
```

Arguments

functions	Character vector of function names to search for. If NULL, all Stan files are returned.
stan_path	Character string specifying the path to the directory containing Stan files. Defaults to the Stan path of the nfidd package.

Value

A character vector of file paths to Stan files.

See Also

Other stantools: [nfidd_load_stan_functions\(\)](#), [nfidd_stan_functions\(\)](#), [nfidd_stan_path\(\)](#)

nfidd_stan_functions *Get Stan function names from Stan files*

Description

This function reads all Stan files in the specified directory and extracts the names of all functions defined in those files.

Usage

```
nfidd_stan_functions(stan_path = nfidd::nfidd_stan_path())
```

Arguments

stan_path	Character string specifying the path to the directory containing Stan files. Defaults to the Stan path of the nfidd package.
-----------	--

Value

A character vector containing unique names of all functions found in the Stan files.

See Also

Other stantools: [nfidd_load_stan_functions\(\)](#), [nfidd_stan_function_files\(\)](#), [nfidd_stan_path\(\)](#)

nfidd_stan_models

List Available Stan Models in NFIDD

Description

This function finds all available Stan models in the NFIDD package and returns their names without the .stan extension.

Usage

```
nfidd_stan_models(stan_path = nfidd::nfidd_stan_path())
```

Arguments

stan_path Character string specifying the path to Stan files. Defaults to the result of 'nfidd_stan_path()'.

Value

A character vector of available Stan model names.

Examples

```
nfidd_stan_models()
```

nfidd_stan_path

Get the path to Stan code

Description

Get the path to Stan code

Usage

```
nfidd_stan_path()
```

Value

A character string with the path to the Stan code

See Also

Other stantools: [nfidd_load_stan_functions\(\)](#), [nfidd_stan_function_files\(\)](#), [nfidd_stan_functions\(\)](#)

renewal	<i>Simulate Infections using the Renewal Equation</i>
---------	---

Description

This function simulates infections using the renewal equation.

Usage

```
renewal(I0, R, gen_time)
```

Arguments

- I0 The initial number of infections.
- R The reproduction number, given as a vector with one entry per time point.
- gen_time The generation time distribution, given as a vector with one entry per day after infection (the first element corresponding to one day after infection).

Value

A vector of simulated infections over time.

Examples

```
renewal(  
  I0 = 5,  
  R = c(rep(3, 4), rep(0.5, 5)),  
  gen_time = c(0.1, 0.2, 0.3, 0.2, 0.1)  
)
```

rw_forecasts	<i>Forecasts from a semi-mechanistic model</i>
--------------	--

Description

A dataset containing forecasts from a semi-mechanistic model (using a geometric random walk prior on the reproduction number), generated using the code in `data-raw/generate-example-forecasts.r`

Usage

```
rw_forecasts
```

Format

A [tibble::tibble()] with a 7 columns

day the day for which the forecast was made

.draw an ID of the posterior predictive sample

.variable name of the variable

.value predicted value

.horizon the forecast horizon in days

origin_day the day on which the forecast was made (using data up to this day)

model the name of the model

simulate_onsets	<i>Simulate symptom onsets from daily infection counts</i>
-----------------	--

Description

Simulate symptom onsets from daily infection counts

Usage

```
simulate_onsets(inf_ts, ip_pmf = make_ip_pmf())
```

Arguments

inf_ts A data frame containing columns `infection_day` and `infections`, as returned by `'make_daily_infections()'`.

ip_pmf A vector of probabilities representing the incubation period PMF, as returned by `'make_ip_pmf()'`.

Value

A data frame with columns `day`, `onsets`, and `infections` containing the daily count of symptom onsets and infections

Examples

```
ip_pmf <- make_ip_pmf()
simulate_onsets(make_daily_infections(infection_times), ip_pmf)
```

stat_forecasts	<i>Forecasts from a semi-mechanistic model with additional statistical complexity</i>
----------------	---

Description

A dataset containing forecasts from a semi-mechanistic model (using an autoregressive prior for reproduction number), generated using the code in data-raw/generate-example-forecasts.r

Usage

```
stat_forecasts
```

Format

A [tibble::tibble()] with a 7 columns

day the day for which the forecast was made

.draw an ID of the posterior predictive sample

.variable name of the variable

.value predicted value

.horizon the forecast horizon in days

origin_day the day on which the forecast was made (using data up to this day)

model the name of the model

summarise_lognormal	<i>Summarise lognormal distribution parameters</i>
---------------------	--

Description

Extract mean and standard deviation from lognormal distribution parameters in a posterior or draws object.

Usage

```
summarise_lognormal(posterior_obj)
```

Arguments

posterior_obj A posterior or draws object containing lognormal parameters

Details

This function extracts lognormal parameters from a posterior object (expecting them to be called 'meanlog' and 'sdlog' and calculates the mean and standard deviation on the natural scale using:

- $\text{mean} = \exp(\text{meanlog} + 0.5 * \text{sdlog}^2)$
- $\text{sd} = \exp(\text{meanlog} + 0.5 * \text{sdlog}^2) * \sqrt{\exp(\text{sdlog}^2) - 1}$

Value

A summary of the mean and standard deviation of the lognormal distribution

Examples

```
## Not run:  
# After fitting a lognormal model with cmdstan  
res |> summarise_lognormal()  
  
## End(Not run)
```

Index

- * **datasets**
 - infection_times, [6](#)
 - mech_forecasts, [8](#)
 - rw_forecasts, [14](#)
 - stat_forecasts, [16](#)
- * **modelhelpers**
 - nfidd_cmdstan_model, [9](#)
 - nfidd_sample, [11](#)
- * **stantools**
 - nfidd_load_stan_functions, [10](#)
 - nfidd_stan_function_files, [12](#)
 - nfidd_stan_functions, [12](#)
 - nfidd_stan_path, [13](#)
- add_delays, [2](#)
- censored_delay_pmf, [3](#)
- condition_onsets_by_report, [4](#)
- convolve_with_delay, [5](#)
- geometric_diff_ar, [5](#)
- geometric_random_walk, [6](#)
- infection_times, [6](#)
- make_daily_infections, [7](#)
- make_gen_time_pmf, [7](#)
- make_ip_pmf, [8](#)
- mech_forecasts, [8](#)
- nfidd_cmdstan_model, [9](#), [11](#)
- nfidd_load_stan_functions, [10](#), [12–14](#)
- nfidd_sample, [9](#), [11](#)
- nfidd_stan_function_files, [10](#), [12](#), [13](#), [14](#)
- nfidd_stan_functions, [10](#), [12](#), [12](#), [14](#)
- nfidd_stan_models, [13](#)
- nfidd_stan_path, [10](#), [12](#), [13](#), [13](#)
- renewal, [14](#)
- rw_forecasts, [14](#)
- simulate_onsets, [15](#)
- stat_forecasts, [16](#)
- summarise_lognormal, [16](#)