Package: nfidd (via r-universe)

November 5, 2024

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

Version 1.0.0

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 dplyr, tidyr

Suggests bayesplot, cmdstanr, ggplot2, lubridate, posterior, purrr, rmarkdown, scoringutils (>= 2.0.0), tidybayes, qrensemble (>= 0.1.0), stackr (>= 0.1.0), roxyglobals

Additional_repositories `https://stan-dev.r-universe.dev`, `https://epiforecasts.r-universe.dev`

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.2

LazyData true

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

Config/roxyglobals/filename globals.R

Config/roxyglobals/unique FALSE

Config/pak/sysreqs libicu-dev

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

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

RemoteRef v1.0.0

RemoteSha d01678e1083a2a11901d2af4989130327d43a0db

Contents

`add_delays` *Simulate symptom onset and hospitalization times from infection times*

Description

Simulate symptom onset and hospitalization times from infection times

Usage

```
add_delays(infection_times)
```

Arguments

`infection_times`
A data frame containing a column of infection times

Value

A data frame with columns for infection time, onset time, and hospitalization time (with 70

Examples

```
delayed_infections <- add_delays(infection_times)
head(delayed_infections)
```

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(rgen, 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)
```

convolve_with_delay *Convolve a Time Series with a Delay Distribution*

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` *Geometric Differenced Autoregressive Process*

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

`init`	The initial value.
`noise`	A vector of steps (on the standard normal scale).
`std`	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` *Generate a probability mass function for the incubation period*

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` *Forecasts from a mechanistic model*

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

target_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 a specified Stan model from the NFIDD package and optionally includes additional user-specified Stan files.

Usage

```
nfidd_cmdstan_model(model_name, include_paths = nfidd::nfidd_stan_path(), ...)
```

Arguments

`model_name`	Character string specifying which Stan model to use.
`include_paths`	Character vector of paths to include for Stan compilation. Defaults to the result of ‘nfidd_stan_path()‘.
`...`	Additional arguments passed to cmdstanr::cmdstan_model().

Value

A CmdStanModel object.

Examples

```
if (!is.null(cmdstanr::cmdstan_version(error_on_NA = FALSE))) {
  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

`functions`	Character vector of function names to load. Defaults to all functions.
`stan_path`	Character string, the path to the Stan code. Defaults to the path to the Stan code in the nfidd package.
`wrap_in_block`	Logical, whether to wrap the functions in a ‘functions‘ block. Default is FALSE.
`write_to_file`	Logical, whether to write the output to a file. Default is FALSE.
`output_file`	Character string, the path to write the output file if write_to_file 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_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_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_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` *Simulate Infections using the Renewal Equation*

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` *Forecasts from a semi-mechanistic model*

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

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

model the name of the model

`simulate_onsets` *Simulate symptom onsets from daily infection counts*

Description

Simulate symptom onsets from daily infection counts

Usage

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

Arguments

`inf_ts`	A data frame containing columns infection_day and infections, as returned by 'make_daily_infections()'.
`gen_time_pmf`	A vector of probabilities representing the generation time PMF, as returned by 'make_gen_time_pmf()'.
`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

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

`stat_forecasts` *Forecasts from a semi-mechanistic model with additional statistical complexity*

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

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

model the name of the model

Index