Package: EpiSoon (via r-universe)

July 25, 2024

Type Package

Title Forecast Cases Using Reproduction Numbers

Version 0.3.1

Description To forecast the time-varying reproduction number and use this to forecast reported case counts. Includes tools to evaluate a range of models across samples and time series using proper scoring rules.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 3.3.0)

Imports scoringutils, scoringRules, purrr, furrr, magrittr, dplyr, tidyr, rlang, tibble, data.table, lubridate, ggplot2, cowplot, HDInterval, R.utils

Remotes nikosbosse/stackr

Suggests knitr, rmarkdown, fable, fabletools, tsibble, forecastHybrid, forecast, fable, feasts, future.apply, stackr, bsts, brms, tidybayes, testthat, covr

VignetteBuilder knitr

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

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

RemoteRef HEAD

RemoteSha d8eb427000f403ca09543ce9eab54c35855256a1

Contents

brms_model 2
bsts_model 4

compare_models 5
compare_timeseries 7
draw_from_si_prob 9
evaluate_model 10
example_obs_cases 12
example_obs_rts 13
example_serial_interval 13
fable_model 13
forecastHybrid_model 14
forecast_cases 16
forecast_cases_directly 17
forecast_model 18
forecast_rt 20
iterative_case_forecast 21
iterative_direct_case_forecast 22
iterative_rt_forecast 23
plot_compare_timeseries 25
plot_forecast 26
plot_forecast_evaluation 28
plot_scores 29
predict_cases 29
predict_current_cases 31
score_case_forecast 32
score_forecast 33
stackr_model 34
summarise_case_forecast 36
summarise_forecast 37
summarise_scores 38

Index **40**

`brms_model` *brms Model Wrapper*

Description

Allows users to specify a model using the `brms::bf()` wrapper from `brms` Note that `brms` and `tidybayes` must both be installed for this model wrapper to be functional.

Usage

```
brms_model(
  y = NULL,
  samples = NULL,
  horizon = NULL,
  model = NULL,
  n_cores = 1,
  n_chains = 4,
```

```
  n_iter = 2000,
  ...
)
```

Arguments

y	Numeric vector of time points to forecast
samples	Numeric, number of samples to take.
horizon	Numeric, the time horizon over which to predict.
model	A brms model wrapped in the brms::bf() function
n_cores	Numeric, the number of cores to use, default of 1
n_chains	Numeric, the number of chains to use, default of 4
n_iter	Numeric, the number of iterations in the sampler to use, default of 4000
...	additional arguments passed to brms (e.g. priors or family)

Value

A dataframe of predictions (with columns representing the time horizon and rows representing samples).

Examples

```
## Not run:

## Used on its own
## Note: More iterations and chains should be used
library(brms)
brms_model(
  y = EpiSoon::example_obs_rts[1:10, ]$rt,
  model = brms::bf(y ~ gp(time)),
  samples = 10, horizon = 7, n_iter = 40, n_chains = 1, refresh = 0
)

## Used for forecasting
## Note that the timeout parameter has been increased to allow
## for the time for the code to be compiled
## Note: More iterations and chains should be used

forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    brms_model(model = brms::bf(y ~ gp(time)), n_iter = 40, n_chains = 1, ...)
  },
  horizon = 7, samples = 10, timeout = 300
)

## End(Not run)
```

bsts_model *bsts Model Wrapper*

Description

bsts Model Wrapper

Usage

```
bsts_model(y = NULL, samples = NULL, horizon = NULL, model = NULL)
```

Arguments

y	Numeric vector of time points to forecast
samples	Numeric, number of samples to take.
horizon	Numeric, the time horizon over which to predict.
model	A bsts model object wrapped in a function with an ss and y argument.

Value

A dataframe of predictions (with columns representing the time horizon and rows representing samples).

Examples

```
## Not run:

library(bsts)

## Used on its own
bsts_model(
  y = EpiSoon::example_obs_rts[1:10, ]$rt,
  model = function(ss, y) {
    bsts::AddAr(ss, y = y, lags = 2)
  },
  samples = 10, horizon = 7
)

## Used for forecasting
forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddAr(ss, y = y, lags = 3)
        }, ...
    )
  },
```

```
  horizon = 7, samples = 10
)

## End(Not run)
```

`compare_models` *Compare forecasting models*

Description

Compare forecasting models

Usage

```
compare_models(
  obs_rts = NULL,
  obs_cases = NULL,
  models = NULL,
  horizon = 7,
  samples = 1000,
  bound_rt = TRUE,
  timeout = 30,
  serial_interval = NULL,
  min_points = 3,
  rdist = NULL,
  return_raw = FALSE
)
```

Arguments

`obs_rts`	Dataframe of Rt observations to forecast with and score against. Should contain a `date` and `rt` column. If multiple samples are included this should be denoted using a numeric `sample` variable.
`obs_cases`	Dataframe of case observations to use for case prediction and scoring. Should contain a `date` and `cases` column. If multiple samples are included this should be denoted using a numeric `sample` variable.
`models`	A list of models. A configuration is given in the examples. Each model needs to be wrapped in a function that takes a `...` argument and returns a dataframe of samples with each column representing a time horizon. Example: `function(...) {EpiSoon::bsts_model(model = function(ss, y){bsts::AddAr(ss, y = y, lags = 3)}, ...)}`.
`horizon`	Numeric, the time horizon over which to predict.
`samples`	Numeric, number of samples to take.
`bound_rt`	Logical, defaults to `TRUE`. Should Rt values be bounded to be greater than or equal to 0.
`timeout`	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.

serial_interval
: A numeric vector describing the probability distribution the serial interval. See `EpiNow::covid_serial_interval` for an example of the format.

min_points
: Numeric, defaults to 3. The minimum number of time points at which to begin iteratively evaluating the forecast.

rdist
: A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to `rpois` if not supplied

return_raw
: Logical, should raw cases and rt forecasts be returned. Defaults to `FALSE`.

Value

A list of dataframes as produced by `evaluate model` but with an additional model column.

Examples

```
## Not run:
## List of forecasting bsts models wrapped in functions.
models <- list(
  "AR 3" =
    function(...) {
      EpiSoon::bsts_model(
        model =
          function(ss, y) {
            bsts::AddAr(ss, y = y, lags = 3)
          }, ...
      )
    },
  "Semi-local linear trend" =
    function(...) {
      EpiSoon::bsts_model(
        model =
          function(ss, y) {
            bsts::AddSemilocalLinearTrend(ss, y = y)
          }, ...
      )
    },
  "ARIMA" =
    function(...) {
      fable_model(model = fable::ARIMA(y ~ time), ...)
    }
)



## Compare models
evaluations <- compare_models(EpiSoon::example_obs_rts,
  EpiSoon::example_obs_cases, models,
  horizon = 7, samples = 10,
  serial_interval = example_serial_interval
)
```

```
## Example evaluation plot for comparing forecasts
## with actuals for a range of models and time horizons.
plot_forecast_evaluation(evaluations$forecast_rts, EpiSoon::example_obs_rts, c(1, 3, 7)) +
  ggplot2::facet_grid(model ~ horizon) +
  cowplot::panel_border()

## Hack to plot observed cases vs predicted
plot_forecast_evaluation(
  evaluations$forecast_cases,
  EpiSoon::example_obs_cases, c(1, 3, 7)
) +
  ggplot2::facet_wrap(model ~ horizon, scales = "free") +
  cowplot::panel_border()

## End(Not run)
```

`compare_timeseries` *Compare timeseries and forecast models*

Description

Compare timeseries and forecast models

Usage

```
compare_timeseries(
  obs_rts = NULL,
  obs_cases = NULL,
  models = NULL,
  horizon = 7,
  samples = 1000,
  bound_rt = TRUE,
  min_points = 3,
  timeout = 30,
  serial_interval = NULL,
  rdist = NULL,
  return_raw = FALSE
)
```

Arguments

`obs_rts`	A dataframe of observed Rts including a `timeseries` variable to denote each timeseris and a `rt` vector (to forecast) and a `date` vector (to denote time). Optionally this dataframe can contain samples for each timeseries in which case this should be denoted using a `sample` variable.

obs_cases	A dataframe of observed cases including a timeseries variable to denote each timeseris and a cases vector (to forecast) and a date vector (to denote time). Optionally this dataframe can contain samples for each timeseries in which case this should be denoted using a sample variable.
models	A list of models. A configuration is given in the examples. Each model needs to be wrapped in a function that takes a ... argument and returns a dataframe of samples with each column representing a time horizon. Example: function(...) {EpiSoon::bsts_model(model = function(ss, y){bsts::AddAr(ss, y = y, lags = 3)}, ...)}.
horizon	Numeric, the time horizon over which to predict.
samples	Numeric, number of samples to take.
bound_rt	Logical, defaults to TRUE. Should Rt values be bounded to be greater than or equal to 0.
min_points	Numeric, defaults to 3. The minimum number of time points at which to begin iteratively evaluating the forecast.
timeout	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.
serial_interval	A numeric vector describing the probability distribution the serial interval. See EpiNow::covid_serial_interval for an example of the format.
rdist	A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to rpois if not supplied
return_raw	Logical, should raw cases and rt forecasts be returned. Defaults to FALSE.

Value

A list of dataframes as produced by evaluate model but with an additional model column.

Examples

```
## Not run:
## Example data
obs_rts <- EpiSoon::example_obs_rts %>%
  dplyr::mutate(timeseries = "Region 1") %>%
  dplyr::bind_rows(EpiSoon::example_obs_rts %>%
    dplyr::mutate(timeseries = "Region 2"))

obs_cases <- EpiSoon::example_obs_cases %>%
  dplyr::mutate(timeseries = "Region 1") %>%
  dplyr::bind_rows(EpiSoon::example_obs_cases %>%
    dplyr::mutate(timeseries = "Region 2"))

## List of forecasting bsts models wrapped in functions.
models <- list(
  "AR 3" =
    function(...) {
      EpiSoon::bsts_model(
        model =
```

```
          function(ss, y) {
            bsts::AddAr(ss, y = y, lags = 3)
          }, ...
      )
    },
  "Semi-local linear trend" =
    function(...) {
      EpiSoon::bsts_model(
        model =
          function(ss, y) {
            bsts::AddSemilocalLinearTrend(ss, y = y)
          }, ...
      )
    },
  "ARIMA" =
    function(...) {
      fable_model(model = fable::ARIMA(y ~ time), ...)
    }
)


## Compare models
evaluations <- compare_timeseries(obs_rts, obs_cases, models,
  horizon = 7, samples = 10,
  serial_interval = EpiSoon::example_serial_interval
)

evaluations

## Example evaluation plot for comparing forecasts
## with actuals for a range of models and timeseries.
plot_forecast_evaluation(evaluations$forecast_rts, obs_rts, c(7)) +
  ggplot2::facet_grid(model ~ timeseries) +
  cowplot::panel_border()

## Hack to plot observed cases vs predicted
plot_forecast_evaluation(
  evaluations$forecast_cases,
  obs_cases, c(7)
) +
  ggplot2::facet_grid(model ~ timeseries, scales = "free") +
  cowplot::panel_border()

## End(Not run)
```

`draw_from_si_prob` *Draw from the Serial Interval Probability Distribution*

Description

Draw from the Serial Interval Probability Distribution

Usage

```
draw_from_si_prob(days_ago = NULL, serial_interval = NULL)
```

Arguments

days_ago	Numeric vector of days in the past. Defaults to NULL.
serial_interval	A numeric vector describing the probability distribution the serial interval. See EpiNow::covid_serial_interval for an example of the format.

Value

A draw from the probability distribution the serial interval.

Examples

```
## Draw
draw_from_si_prob(rev(c(1, 2, 4, 10, 1:100)), EpiSoon::example_serial_interval)
```

---

evaluate_model *Evaluate a Model for Forecasting Rts*

Description

Evaluate a Model for Forecasting Rts

Usage

```
evaluate_model(
  obs_rts = NULL,
  obs_cases = NULL,
  model = NULL,
  horizon = 7,
  samples = 1000,
  timeout = 30,
  bound_rt = TRUE,
  min_points = 3,
  serial_interval = NULL,
  rdist = NULL,
  return_raw = FALSE
)
```

Arguments

obs_rts	Dataframe of Rt observations to forecast with and score against. Should contain a date and rt column. If multiple samples are included this should be denoted using a numeric sample variable.
obs_cases	Dataframe of case observations to use for case prediction and scoring. Should contain a date and cases column. If multiple samples are included this should be denoted using a numeric sample variable.
model	A model object in the format of bsts_model or fable_model. See the corresponding help files for details.
horizon	Numeric, the time horizon over which to predict.
samples	Numeric, number of samples to take.
timeout	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.
bound_rt	Logical, defaults to TRUE. Should Rt values be bounded to be greater than or equal to 0.
min_points	Numeric, defaults to 3. The minimum number of time points at which to begin iteratively evaluating the forecast.
serial_interval	A numeric vector describing the probability distribution the serial interval. See EpiNow::covid_serial_interval for an example of the format.
rdist	A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to rpois if not supplied
return_raw	Logical, should raw cases and rt forecasts be returned. Defaults to FALSE.

Value

a list of tibbles containing the predicted Rt values (forecast_rts), their scores (rt_scores), as well as predicted cases (forecast_cases) and their scores (case_scores).

Examples

```
## Not run:
## Evaluate a model based on a single sample of input cases
evaluate_model(EpiSoon::example_obs_rts,
  EpiSoon::example_obs_cases,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10,
  serial_interval = example_serial_interval
)
```

```
## Samples of observed data
sampled_obs <- EpiSoon::example_obs_rts %>%
  dplyr::mutate(sample = 1) %>%
  dplyr::bind_rows(EpiSoon::example_obs_rts %>%
    dplyr::mutate(sample = 2))

sampled_cases <- EpiSoon::example_obs_cases %>%
  dplyr::mutate(sample = 1) %>%
  dplyr::bind_rows(EpiSoon::example_obs_cases %>%
    dplyr::mutate(sample = 2))


## Evaluate a model across samples
evaluate_model(sampled_obs,
  sampled_cases,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10,
  serial_interval = EpiSoon::example_serial_interval
)

## End(Not run)
```

`example_obs_cases` *Example Observed Cases*

Description

An example data frame of observed cases

Usage

```
example_obs_cases
```

Format

A data frame containing cases reported on each date.

example_obs_rts *Example Observed Rts*

Description

An example data frame of observed Reproduction numbers

Usage

```
example_obs_rts
```

Format

A data frame containing Rts estimated for each date.

example_serial_interval *Example Serial Interval*

Description

An example serial interval probability vector

Usage

```
example_serial_interval
```

Format

A vector giviing the probability for each day

fable_model *fable Model Wrapper*

Description

Provides an interface for models from the `fable` package. Note the `feasts::ARIMA` model requires the `feast` package. If `future` is being used `fable` will require `future.apply` in order to not silently fail.

Usage

```
fable_model(y = NULL, samples = NULL, horizon = NULL, model = NULL)
```

Arguments

y	Numeric vector of time points to forecast
samples	Numeric, number of samples to take.
horizon	Numeric, the time horizon over which to predict.
model	A fable model object. For models that use a formula interface time can be accessed using time.

Value

A dataframe of predictions (with columns representing the time horizon and rows representing samples).

Examples

```
## Not run:
## Used on its own
fable_model(
  y = EpiSoon::example_obs_rts[1:10, ]$rt,
  model = fable::ARIMA(y ~ time),
  samples = 10, horizon = 7
)


forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    fable_model(model = fable::ARIMA(y ~ time), ...)
  },
  horizon = 7, samples = 10
)

## End(Not run)
```

---

forecastHybrid_model *forecastHybrid Model Wrapper*

Description

Allows users to forecast using ensembles from the forecastHybrid package. Note that whilst weighted ensembles can be created this is not advised when samples > 1 as currently samples are derived assuming a normal distribution using the upper and lower confidence intervals of the ensemble. These confidence intervals are themselves either based on the unweighted mean of the ensembled models or the maximum/minimum from the candiate models. Note that forecastHybrid must be installed for this model wrapper to be functional.

Usage

```
forecastHybrid_model(
  y = NULL,
  samples = NULL,
  horizon = NULL,
  model_params = NULL,
  forecast_params = NULL
)
```

Arguments

y	Numeric vector of time points to forecast
samples	Numeric, number of samples to take.
horizon	Numeric, the time horizon over which to predict.
model_params	List of parameters to pass to `forecastHybrid::hybridModel`.
forecast_params	List of parameters to pass to `forecastHybrid:::forecast.hybridModel`.

Value

A dataframe of predictions (with columns representing the time horizon and rows representing samples).

Examples

```
## Not run:

library(forecastHybrid)

## Used on its own
forecastHybrid_model(
  y = EpiSoon::example_obs_rts$rt,
  samples = 10, horizon = 7
)


## Used with non-default arguments
## Note that with the current sampling from maximal confidence intervals model
## Weighting using cross-validation will only have an impact when 1 sample is used.
forecastHybrid_model(
  y = EpiSoon::example_obs_rts$rt,
  samples = 1, horizon = 7,
  model_params = list(
    cvHorizon = 7, windowSize = 7,
    rolling = TRUE, models = "zeta"
  )
)


## Used for forecasting
```

```
forecast_rt(EpiSoon::example_obs_rts,
  model = EpiSoon::forecastHybrid_model,
  horizon = 7, samples = 1
)

## Used for forcasting with non-default arguments
forecast_rt(EpiSoon::example_obs_rts,
  model = function(...) {
    EpiSoon::forecastHybrid_model(
      model_params = list(models = "zte"),
      forecast_params = list(PI.combination = "mean"), ...
    )
  },
  horizon = 7, samples = 10
)

## End(Not run)
```

`forecast_cases` *Forecasts Cases for a Rt Forecasts*

Description

Forecasts Cases for a Rt Forecasts

Usage

```
forecast_cases(
  cases = NULL,
  fit_samples = NULL,
  serial_interval = NULL,
  forecast_date = NULL,
  horizon = NULL,
  rdist = NULL
)
```

Arguments

`cases`	A dataframe containing `date` and `cases` variables
`fit_samples`	A dataframe as produced by `EpiSoon::forecast`.
`serial_interval`	A numeric vector describing the probability distribution the serial interval. See `EpiNow::covid_serial_interval` for an example of the format.
`forecast_date`	A character string date (format "yyyy-mm-dd") indicating the forecast date. Defaults to `NULL` in which case it will be assumed that the forecast date is the day before the first date present in the `fit_samples` dataframe
`horizon`	Numeric, the time horizon over which to predict.

rdist	A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to rpois if not supplied

Value

Forecast cases for over a future forecast horizon

Examples

```
## Not run:
## Rt forecast
forecast <- forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(model = function(ss, y) {
      bsts::AddAutoAr(ss, y = y, lags = 10)
    }, ...)
  },
  horizon = 7, samples = 10
)


forecast_cases(EpiSoon::example_obs_cases,
  fit_samples = forecast,
  serial_interval = EpiSoon::example_serial_interval
)

## End(Not run)
```

---

forecast_cases_directly

Forecasts Cases Directly

Description

Forecasts Cases Directly

Usage

```
forecast_cases_directly(
  cases = NULL,
  model,
  horizon = 7,
  samples = 1000,
  bound_rt = TRUE,
  timeout = 100
)
```

Arguments

cases	A dataframe containing date and cases variables
model	A model object in the format of bsts_model or fable_model. See the corresponding help files for details.
horizon	Numeric, the time horizon over which to predict.
samples	Numeric, number of samples to take.
bound_rt	Logical, defaults to TRUE. Should Rt values be bounded to be greater than or equal to 0.
timeout	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.

Value

Forecast cases over a future forecast horizon

Examples

```
## Not run:
forecast_cases_directly(EpiSoon::example_obs_cases,
  model = function(...) {
    EpiSoon::bsts_model(model = function(ss, y) {
      bsts::AddAutoAr(ss, y = y, lags = 10)
    }, ...)
  },
  horizon = 7, samples = 10
)

## End(Not run)
```

---

forecast_model *forecast Model Wrapper*

Description

Allows users to forecast using models from the forecast package. Note that forecast must be installed for this model wrapper to be functional.

Usage

```
forecast_model(y = NULL, samples = NULL, horizon = NULL, model = NULL, ...)
```

Arguments

y	Numeric vector of time points to forecast
samples	Numeric, number of samples to take.
horizon	Numeric, the time horizon over which to predict.
model	A forecast model object.
...	pass further arguments to the forecast models

Value

A dataframe of predictions (with columns representing the time horizon and rows representing samples).

Examples

```
## Not run:

## Used on its own
forecast_model(
  y = EpiSoon::example_obs_rts[1:10, ]$rt,
  model = forecast::auto.arima,
  samples = 10, horizon = 7
)

## Used for forecasting
forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    forecast_model(model = forecast::ets, ...)
  },
  horizon = 7, samples = 10
)

# run with non-default arguments
forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    forecast_model(
      model = forecast::ets,
      damped = TRUE, ...
    )
  },
  horizon = 7, samples = 10
)

models <- list(
  "ARIMA" = function(...) {
    forecast_model(model = forecast::auto.arima, ...)
  },
  "ETS" = function(...) {
    forecast_model(model = forecast::ets, ...)
  },
  "TBATS" = function(...) {
    forecast_model(model = forecast::tbats, ...)
  }
)

## Compare models
evaluations <- compare_models(EpiSoon::example_obs_rts,
  EpiSoon::example_obs_cases, models,
  horizon = 7, samples = 10,
  serial_interval = example_serial_interval
)
```

```
plot_forecast_evaluation(evaluations$forecast_rts,
  EpiSoon::example_obs_rts,
  horizon_to_plot = 7
) +
  ggplot2::facet_grid(~model) +
  cowplot::panel_border()

## End(Not run)
```

`forecast_rt` *Fit and Forecast using a Model*

Description

Fit and Forecast using a Model

Usage

```
forecast_rt(
  rts,
  model,
  horizon = 7,
  samples = 1000,
  bound_rt = TRUE,
  timeout = 100
)
```

Arguments

`rts`	A dataframe of containing two variables `rt` and `date` with `rt` being numeric and `date` being a date.
`model`	A model object in the format of `bsts_model` or `fable_model`. See the corresponding help files for details.
`horizon`	Numeric, the time horizon over which to predict.
`samples`	Numeric, number of samples to take.
`bound_rt`	Logical, defaults to `TRUE`. Should Rt values be bounded to be greater than or equal to 0.
`timeout`	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.

Value

A dataframe of samples containing the following variables: `sample`, `date`, `rt`, and `horizon`.

Examples

```
## Not run:
forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(model = function(ss, y) {
      bsts::AddAutoAr(ss, y = y, lags = 10)
    }, ...)
  },
  horizon = 7, samples = 10
)

## End(Not run)
```

---

iterative_case_forecast

Iteratively Forecast Cases Using an Iterative Rt Forecast

Description

Iteratively Forecast Cases Using an Iterative Rt Forecast

Usage

```
iterative_case_forecast(
  it_fit_samples = NULL,
  cases = NULL,
  serial_interval,
  rdist = NULL
)
```

Arguments

it_fit_samples	Dataframe of iterative forecasts as produced by iterative_rt_forecast.
cases	A dataframe containing date and cases variables
serial_interval	A numeric vector describing the probability distribution the serial interval. See EpiNow::covid_serial_interval for an example of the format.
rdist	A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to rpois if not supplied

Value

A dataframe of iterative case forecasts

Examples

```
## Not run:
## Iterative Rt forecast
it_forecast <-
  iterative_rt_forecast(EpiSoon::example_obs_rts,
    model = function(...) {
      EpiSoon::bsts_model(
        model =
          function(ss, y) {
            bsts::AddSemilocalLinearTrend(ss, y = y)
          }, ...
      )
    },
    horizon = 7, samples = 10
  )


## Iterative case forecast
iterative_case_forecast(
  it_fit_samples = it_forecast,
  cases = EpiSoon::example_obs_cases,
  serial_interval = EpiSoon::example_serial_interval
)

## End(Not run)
```

---

iterative_direct_case_forecast

Iteratively forecast directly on cases

Description

Iteratively forecast directly on cases

Usage

```
iterative_direct_case_forecast(
  cases,
  model = NULL,
  horizon = 7,
  samples = 1000,
  timeout = 30,
  bound_rt = TRUE,
  min_points = 3
)
```

Arguments

cases	A dataframe containing date and cases variables
model	A model object in the format of bsts_model or fable_model. See the corresponding help files for details.
horizon	Numeric, the time horizon over which to predict.
samples	Numeric, number of samples to take.
timeout	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.
bound_rt	Logical, defaults to TRUE. Should Rt values be bounded to be greater than or equal to 0.
min_points	Numeric, defaults to 3. The minimum number of time points at which to begin iteratively evaluating the forecast.

Value

A tibble of iterative forecasts

Examples

```
## Not run:
iterative_direct_case_forecast(EpiSoon::example_obs_cases,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10, min_points = 4
)

## End(Not run)
```

---

iterative_rt_forecast *Iteratively Forecast*

Description

Iteratively Forecast

Usage

```
iterative_rt_forecast(
  rts,
  model = NULL,
  horizon = 7,
```

```
  samples = 1000,
  timeout = 30,
  bound_rt = TRUE,
  min_points = 3
)
```

Arguments

rts	A dataframe of containing two variables rt and date with rt being numeric and date being a date.
model	A model object in the format of bsts_model or fable_model. See the corresponding help files for details.
horizon	Numeric, the time horizon over which to predict.
samples	Numeric, number of samples to take.
timeout	Numeric, timeout of model fitting in seconds. Defaults to 30 seconds.
bound_rt	Logical, defaults to TRUE. Should Rt values be bounded to be greater than or equal to 0.
min_points	Numeric, defaults to 3. The minimum number of time points at which to begin iteratively evaluating the forecast.

Value

A tibble of iterative forecasts

Examples

```
## Not run:
iterative_rt_forecast(EpiSoon::example_obs_rts,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10, min_points = 4
) -> tmp

## End(Not run)
```

plot_compare_timeseries *Summary plots to compare timeseries and forecast models*

Description

Summary plots to compare timeseries and forecast models

Usage

```
plot_compare_timeseries(
  compare_timeseries_output,
  type = c("summary_score", "horizon_score", "region_score"),
  score = c("Bias", "CRPS", "Dispersion", "AE (median)", "SE (mean)")
)
```

Arguments

compare_timeseries_output	A named list of dataframes produced by `compare_timeseries`
type	Type(s) of summary plot to be produced for Rt and case observations for each model in n `compare_timeseries_output`. `"summary_score"` provides a plot of model fit scores for the 0-7 and 8-14 day horizons. If desired, a subset of scores can be specified using the `score` argument. `"horizon_score"` provides a plot of scores (CRPS, Calibration, Sharpness, Median, IQR, CI, Bias) across horizons. `"region_score"` provides a plot of scores (CRPS, Calibration, Sharpness, Bias, Median, IQR) by region for the 0-7 and 8-14 day horizons.
score	(Optional) One or more of `c("Bias","CRPS","Sharpness","Calibration","Median","IQR","CI")` when `type="summary_score"` is used.

Value

A named list of ggplot2 objects

Examples

```
## Not run:
obs_rts <- EpiSoon::example_obs_rts %>%
  dplyr::mutate(timeseries = "Region 1") %>%
  dplyr::bind_rows(EpiSoon::example_obs_rts %>%
    dplyr::mutate(timeseries = "Region 2"))

obs_cases <- EpiSoon::example_obs_cases %>%
  dplyr::mutate(timeseries = "Region 1") %>%
  dplyr::bind_rows(EpiSoon::example_obs_cases %>%
    dplyr::mutate(timeseries = "Region 2"))

models <- list(
```

```
  "AR 3" = function(...) {
    EpiSoon::bsts_model(model = function(ss, y) {
      bsts::AddAr(ss, y = y, lags = 3)
    }, ...)
  },
  "Semi-local linear trend" = function(...) {
    EpiSoon::bsts_model(model = function(ss, y) {
      bsts::AddSemilocalLinearTrend(ss, y = y)
    }, ...)
  }
)

forecast_eval <-
  compare_timeseries(obs_rts, obs_cases, models,
    horizon = 10, samples = 10,
    serial_interval = EpiSoon::example_serial_interval
  )

## Produce all plots
plot_compare_timeseries(forecast_eval)

## Produce subsets of plots
plot_compare_timeseries(forecast_eval, type = "summary_score")
plot_compare_timeseries(forecast_eval,
  type = "summary_score",
  score = "Bias"
)
plot_compare_timeseries(forecast_eval, type = "horizon_score")
plot_compare_timeseries(forecast_eval, type = "region_score")
plot_compare_timeseries(forecast_eval,
  type = c("horizon_score", "region_score")
)

## End(Not run)
```

`plot_forecast` *Plot a Forecast*

Description

Plot a Forecast

Usage

```
plot_forecast(
  forecast = NULL,
  observations = NULL,
  horizon_cutoff = NULL,
  obs_cutoff_at_forecast = TRUE
)
```

Arguments

forecast	A dataframe with summarised forecasts as produced by summarise_forecast or summarise_case_forecast .
observations	A dataframe of observations containing the following variables: • either rt or cases • and date.
horizon_cutoff	Numeric, defaults to NULL. Forecast horizon to plot up to.
obs_cutoff_at_forecast	Logical defaults to TRUE. Should observations only be shown up to the date of the forecast.

Value

A ggplot2 object

Examples

```
## Not run:
## Forecast an Rt sample
samples <- forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 21, samples = 10
)

## Summarise forecast
summarised_forecast <- summarise_forecast(samples)

## Plot forecast_cases
plot_forecast(summarised_forecast, EpiSoon::example_obs_rts)

## Forecast a case sample
pred_cases <- forecast_cases(EpiSoon::example_obs_cases, samples,
  serial_interval = EpiSoon::example_serial_interval
)

summarised_case_forecast <- summarise_case_forecast(pred_cases)

plot_forecast(summarised_case_forecast, EpiSoon::example_obs_cases)

## End(Not run)
```

`plot_forecast_evaluation` *Plot a Forecast*

Description

Plot a Forecast

Usage

```
plot_forecast_evaluation(
  forecasts = NULL,
  observations = NULL,
  horizon_to_plot = 1
)
```

Arguments

`forecasts`	A dataframe as produced by `forecast_rt` or `forecast_cases`
`observations`	A dataframe of observations containing the following variables: • either `rt` or `cases` • and `date`.
`horizon_to_plot`	Numeric vector, the forecast horizon to plot.

Value

A `ggplot2` object

Examples

```
## Not run:
## Evaluate a model
forecast_eval <- evaluate_model(EpiSoon::example_obs_rts,
  EpiSoon::example_obs_cases,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  serial_interval = EpiSoon::example_serial_interval,
  horizon = 7, samples = 10
)

## Plot Rt forecast
```

```
plot_forecast_evaluation(forecast_eval$forecast_rts,
  EpiSoon::example_obs_rts,
  horizon_to_plot = 7
)


## Plot case forecast
plot_forecast_evaluation(forecast_eval$forecast_cases,
  EpiSoon::example_obs_cases,
  horizon_to_plot = 7
)

## End(Not run)
```

`plot_scores` *Plot forecast scores*

Description

Plot forecast scores

Usage

```
plot_scores()
```

Value

A dataframe of summarised scores in a tidy format.

`predict_cases` *Predict cases for a single Rt sample forecasts*

Description

Predict cases for a single Rt sample forecasts

Usage

```
predict_cases(
  cases = NULL,
  rts = NULL,
  serial_interval = NULL,
  forecast_date = NULL,
  horizon = NULL,
  rdist = NULL
)
```

Arguments

cases	A dataframe containing date and cases variables
rts	A dataframe of containing two variables rt and date with rt being numeric and date being a date.
serial_interval	A numeric vector describing the probability distribution the serial interval. See EpiNow::covid_serial_interval for an example of the format.
forecast_date	A character string date (format "yyyy-mm-dd") indicating the forecast date. Defaults to NULL in which case it will be assumed that the forecast date is the last data present in the cases dataframe
horizon	Numeric, the time horizon over which to predict.
rdist	A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to rpois if not supplied

Value

Forecast cases for over a future forecast horizon

Examples

```
forecast <- forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 1
)


purrr::map_dfr(1:100, ~ predict_cases(
  cases = EpiSoon::example_obs_cases,
  rts = forecast,
  forecast_date = as.Date("2020-03-10"),
  serial_interval = example_serial_interval
)) %>%
  dplyr::group_by(date) %>%
  dplyr::summarise(cases = mean(cases))
```

`predict_current_cases` *Predict cases for Rts based on observed data*

Description

Predict cases for Rts based on observed data

Usage

```
predict_current_cases(
  cases = NULL,
  rts = NULL,
  serial_interval = NULL,
  rdist = NULL
)
```

Arguments

`cases`	A dataframe containing `date` and `cases` variables
`rts`	A dataframe of containing two variables `rt` and `date` with `rt` being numeric and `date` being a date.
`serial_interval`	A numeric vector describing the probability distribution the serial interval. See `EpiNow::covid_serial_interval` for an example of the format.
`rdist`	A function to be used to sample the number of cases. Must take two arguments with the first specfying the number of samples and the second the mean. Defaults to `rpois` if not supplied

Value

Forecast cases for the current timestep

Examples

```
purrr::map_dfr(1:100, ~ predict_current_cases(
  cases = EpiSoon::example_obs_cases,
  rts = EpiSoon::example_obs_rts,
  serial_interval = EpiSoon::example_serial_interval
)) %>%
  dplyr::group_by(date) %>%
  dplyr::summarise(cases = mean(cases))
```

`score_case_forecast` *Score a case forecast*

Description

Score a case forecast

Usage

```
score_case_forecast(pred_cases, obs_cases, scores = "all")
```

Arguments

`pred_cases`	Dataframe of predicted cases with the following variables: `sample`, `date`, `cases` and forecast horizon. As produced by `forecast_cases`.
`obs_cases`	Dataframe of observed cases with the following variables: `date` and `cases`.
`scores`	Character vector defaulting to "all". Select which scores to return, default is all scores but any subset can be returned.

Value

A dataframe containing the following scores per forecast timepoint: dss, crps, logs, bias, and sharpness as well as the forecast date and time horizon.

Examples

```
## Not run:
## Fit a model (using a subset of observations)
samples <- forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10
)

pred_cases <- forecast_cases(
  EpiSoon::example_obs_cases,
  samples, EpiSoon::example_serial_interval
)

## Score the model fit (with observations during the time horizon of the forecast)
score_case_forecast(pred_cases, EpiSoon::example_obs_cases)
```

```
## Score the model fit (with observations during the time horizon of the forecast)
score_case_forecast(pred_cases, EpiSoon::example_obs_cases, scores = c("crps", "sharpness", "bias"))

## End(Not run)
```

---

score_forecast *Score a Model Fit*

Description

Score a Model Fit

Usage

```
score_forecast(fit_samples, observations, scores = "all")
```

Arguments

fit_samples	A dataframe as produced by EpiSoon::forecast.
observations	A dataframe of observations against which to score. Should contain a date and rt column.
scores	Character vector defaulting to "all". Select which scores to return, default is all scores but any subset can be returned.

Value

A dataframe containing the following scores per forecast timepoint: dss, crps, logs, bias, and sharpness as well as the forecast date and time horizon.

Examples

```
## Not run:
## Fit a model (using a subset of observations)
samples <- forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10
)

## Score the model fit (with observations during the time horizon of the forecast)
score_forecast(samples, EpiSoon::example_obs_rts)

## Return just CRPS, bias and sharpness
```

```
score_forecast(samples, EpiSoon::example_obs_rts, scores = c("crps", "sharpness", "bias"))

## Return just the CRPS
score_forecast(samples, EpiSoon::example_obs_rts, scores = "crps")

## End(Not run)
```

`stackr_model` *Stack Models According to CRPS*

Description

Provides a wrapper for different EpiSoon model wrappers and generates a mixture model of these models based on the (Continuous) Rank Probability Score

A list of models is supplied. These models are fit to the data up until a period of observations of size `weighting_period`. Forecasts are generated from all the models for all time points in the `weighting_period`. Predictive samples generated by the individual models are then used to create model weights in an ensemble based on CRPS. All models are then refitted for the entire timeseries and predictions are generated from these models. Draws from the individual model predictive samples are then used to generate a mixture model with the weights obtained in the previous step.

The weights are computed using `crps_weights` from the package `stackr` to minimise CRPS. The function `mixture_from_samples` from the same package is used to draw samples from the individual models to form the mixture models.

Usage

```
stackr_model(
  y = NULL,
  models = NULL,
  samples = NULL,
  horizon = NULL,
  weighting_period = 5,
  verbose = TRUE
)
```

Arguments

`y`	Numeric vector of time points to forecast
`models`	A list of models. Models must be analogous to the form `function(...){EpiSoon::fable_model(model = , ...)}` or `function(...){EpiSoon::bsts_model(model = , ...)}`.
`samples`	Numeric, number of samples to take.
`horizon`	Numeric, the time horizon over which to predict.
`weighting_period`	The number of most recent timepoints to hold out to generate the weights for the mixture model
`verbose`	if TRUE, gives a message if number of observations is too small to do crps weighting

Value

A dataframe of predictions (with columns representing the time horizon and rows representing samples).

Examples

```
## Not run:

# make list with models
models <- list(
  "ARIMA" = function(...) {
    EpiSoon::fable_model(model = fable::ARIMA(y), ...)
  },
  "ETS" = function(...) {
    EpiSoon::fable_model(model = fable::ETS(y), ...)
  },
  "Drift" = function(...) {
    EpiSoon::fable_model(model = fable::RW(y ~ drift()), ...)
  }
)

# make forecast on its own
forecast <- stackr_model(
  y = EpiSoon::example_obs_rts[1:10, ]$rt,
  models = models,
  samples = 10,
  horizon = 7,
  weighting_period = 5
)


# together with forecast_rt
fc_rt <- forecast_rt(EpiSoon::example_obs_rts[1:10, ],
  model = function(...) {
    stackr_model(
      models = models,
      weighting_period = 5,
      ...
    )
  },
  samples = 10,
  horizon = 7
)

forecast_eval <- evaluate_model(EpiSoon::example_obs_rts,
  EpiSoon::example_obs_cases,
  model = function(...) {
    stackr_model(
      models = models,
      weighting_period = 5,
      ...
    )
```

```
  },
  horizon = 7, samples = 10,
  serial_interval = example_serial_interval,
  min_points = 10
)

plot_forecast_evaluation(forecast_eval$forecast_rts,
  EpiSoon::example_obs_rts,
  horizon_to_plot = 7
)

## End(Not run)
```

`summarise_case_forecast` *Summarise Forecast Cases*

Description

Summarise Forecast Cases

Usage

```
summarise_case_forecast(pred_cases)
```

Arguments

`pred_cases`	A dataframe as produced by `EpiSoon::forecast_cases`.

Value

A summarised dataframe.

Examples

```
## Not run:
## Example forecast
forecast <- forecast_rt(EpiSoon::example_obs_rts,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10
)
```

```
## Forecast cases
case_forecast <- forecast_cases(
  EpiSoon::example_obs_cases,
  forecast,
  EpiSoon::example_serial_interval
)
## Summarise case forecast
summarise_case_forecast(case_forecast)

## End(Not run)
```

`summarise_forecast` *Summarise Forecast Rts*

Description

Summarise Forecast Rts

Usage

```
summarise_forecast(fit_samples)
```

Arguments

`fit_samples`	A dataframe as produced by `EpiSoon::forecast`.

Value

A summarised dataframe.

Examples

```
## Not run:
samples <- forecast_rt(example_obs_rts,
  model = function(...) {
    EpiSoon::bsts_model(
      model =
        function(ss, y) {
          bsts::AddSemilocalLinearTrend(ss, y = y)
        }, ...
    )
  },
  horizon = 7, samples = 10
)


summarise_forecast(samples)

## End(Not run)
```

summarise_scores *Summarise model forecasting scores*

Description

Summarise model forecasting scores

Usage

```
summarise_scores(scores, variables = NULL, sel_scores = NULL)
```

Arguments

scores	A dataframe of model scores as produced by score_model
variables	A character vector of variables names to group over. By default score type and model is grouped over if present.
sel_scores	A character vector indicating which scores to return information on. Defaults to all scores

Value

A dataframe of summarised scores in a tidy format.

Examples

```
## Not run:
## Example cases
cases <- EpiSoon::example_obs_cases %>%
  dplyr::mutate(timeseries = "Region 1") %>%
  dplyr::bind_rows(EpiSoon::example_obs_cases %>%
    dplyr::mutate(timeseries = "Region 2"))


## Example Rts
rts <- EpiSoon::example_obs_rts %>%
  dplyr::mutate(timeseries = "Region 1") %>%
  dplyr::bind_rows(EpiSoon::example_obs_rts %>%
    dplyr::mutate(timeseries = "Region 2"))

## List of forecasting bsts models wrapped in functions.
models <- list(
  "AR 3" =
    function(...) {
      EpiSoon::bsts_model(
        model =
          function(ss, y) {
            bsts::AddAr(ss, y = y, lags = 3)
          }, ...
      )
```

```
    },
  "Semi-local linear trend" =
    function(...) {
      EpiSoon::bsts_model(
        model =
          function(ss, y) {
            bsts::AddSemilocalLinearTrend(ss, y = y)
          }, ...
      )
    }
)


## Compare models
evaluations <- compare_timeseries(rts, cases, models,
  horizon = 7, samples = 10,
  serial_interval = example_serial_interval
)



## Score across the default groups
summarise_scores(evaluations$rt_scores)


## Also summarise across time horizon
summarise_scores(evaluations$rt_scores, "horizon", sel_scores = "crps")

## Instead summarise across region and summarise case scores
summarise_scores(evaluations$case_scores, "timeseries", sel_scores = "logs")

## End(Not run)
```

Index

∗ **datasets**
example_obs_cases, 12
example_obs_rts, 13
example_serial_interval, 13

brms_model, 2
bsts_model, 4

compare_models, 5
compare_timeseries, 7
crps_weights, *34*

draw_from_si_prob, 9

evaluate_model, 10
example_obs_cases, 12
example_obs_rts, 13
example_serial_interval, 13

fable_model, 13
forecast_cases, 16
forecast_cases_directly, 17
forecast_model, 18
forecast_rt, 20
forecastHybrid_model, 14

iterative_case_forecast, 21
iterative_direct_case_forecast, 22
iterative_rt_forecast, 23

mixture_from_samples, *34*

plot_compare_timeseries, 25
plot_forecast, 26
plot_forecast_evaluation, 28
plot_scores, 29
predict_cases, 29
predict_current_cases, 31

score_case_forecast, 32
score_forecast, 33
stackr_model, 34
summarise_case_forecast, 36
summarise_forecast, 37
summarise_scores, 38