Package: TACforecasting (via r-universe)

October 13, 2024

Version 0.0.1

Title Forecasting Functions for the Transport Accident Commission

Description Functions to make hierarchical time series forecasts of attendant care hours easier.

Depends R (>= 4.2.0), tsibble, fabletools

Imports abind, distributional, dplyr, forcats, forecast, furrr, ggplot2, hts, janitor, lubridate, MASS, purrr, readr, stringr, tibble, tidyr, tsbox

LazyData yes

ByteCompile TRUE

BugReports https://github.com/robjhyndman/TAC_forecasting/issues

License GPL-3

URL https://pkg.robjhyndman.com/TACforecasting/, https://github.com/robjhyndman/TACforecasting/

Encoding UTF-8

RoxygenNote 7.2.3

Config/testthat/edition 3

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

RemoteUrl https://github.com/robjhyndman/TACforecasting

RemoteRef HEAD

RemoteSha 8e5ea07ad49dd25f0f58f47f2d54ce711208e520

Contents

`get_forecasts` *Generate forecasts of attendant care hours*

Description

Generate forecasts from ETS and ARIMA models, reconcile them, and combine them. Return a fable object containing the forecasts.

Usage

```
get_forecasts(data, h, nsim)
```

Arguments

`data`	Data set computed from `read_tac_data`
`h`	Forecast horizon.
`nsim`	Number of simulated future sample paths per model.

`group_costs` *Synthetic data for attendant hours by age group and injury group*

Description

Artificial Transport Accident Commission attendant care data

Format

Time series of class ‘tsibble‘

Details

`group_costs` is a daily ‘tsibble‘ with index ‘billing_period‘ and two values:

adjusted_hours:	Total attendant care hours
nclaims:	Number of active claims

The data is disaggregated using two keys:

age_group:	Age group of client at the time of accident
injury_group:	Injury sustained by client due to accident

Source

Synthetic data

Examples

```
group_costs
```

`plot_forecasts` *Plot forecasts of attendant care hours disaggregated by age or injury group.*

Description

Produce a time plot of attendant care hours per billing period for specific disaggregations.

Usage

```
plot_forecasts(
  forecasts,
  data,
  show_age_group = "<aggregated>",
  show_injury_group = "<aggregated>"
)
```

Arguments

`forecasts`	A fable object created by `get_forecasts`
`data`	The data used to construct the forecasts. This should be a tsibble object of the same form as `group_costs`.
`show_age_group`	A character string specifying either a specific age group or "<aggregated>" meaning the total across all age groups.
`show_injury_group`	A character string specifying either a specific injury group or "<aggregated>" meaning the total across all injury groups.

Author(s)

Rob J Hyndman

Examples

```
## Not run:
  group_costs |>
    get_forecasts(h=13, nsim=100) |>
    plot_forecasts(group_costs)

## End(Not run)
```

plot_total_hours *Plot attendant care hours disaggregated by age or injury group.*

Description

Produce a time plot of attendant care hours per billing period disaggregated by the given ‘variable‘

Usage

```
plot_total_hours(group_costs, variable = NULL, include_average = TRUE)
```

Arguments

group_costs	A tsibble containing costs optionally split by a variable
variable	Name of disaggregation variable. If NULL, aggregated costs are shown
include_average	Should the average cost per billing period be shown?

Author(s)

Rob J Hyndman

Examples

```
group_costs |>
  plot_total_hours(age_group)
```

read_tac_data *Read in TAC data*

Description

This function takes two csv files as inputs: one containing the claims header and the other containing the attendant hours. It returns total hours per age group and injury group by billing period.

Usage

```
read_tac_data(claims_file, costs_file)
```

Arguments

claims_file	CSV file containing claims header
costs_file	CSV file containing attendant hours

Value

A tsibble object containing total attendant care adjusted hours for each billing period, disaggregated by age group and injury group. The column ‘nclaims‘ shows the number of "active" claims in each billing period.

Author(s)

Rob J Hyndman

Examples

```
## Not run:
group_costs <- read_tac_data(
  claims_file = "T086_claim_header.csv",
  costs_file = "T086_attendant_care_hours.csv"
)

## End(Not run)
```

`tac_accuracy` *Compute accuracy statistics*

Description

Compute accuracy statistics

Usage

```
tac_accuracy(forecasts, actuals)
```

Arguments

`forecasts`	A fable object with forecasts, usually the output from `get_forecasts`
`actuals`	A tsibble with actual values. For example, the output from `read_tac_data`

Value

A tibble with accuracy statistics

`tscv_accuracy` *Compute forecasts with a rolling origin and return accuracy statistics*

Description

Compute forecasts with a rolling origin and return accuracy statistics

Usage

```
tscv_accuracy(group_costs, h, nsim, init, step)
```

Arguments

`group_costs`	A tsibble with actual values. For example, the output from `read_tac_data`
`h`	The forecast horizon
`nsim`	The number of simulations used in each forecast for each model.
`init`	The number of initial observations to use for the first fold.
`step`	The number of observations to skip between each fold.

Value

A tibble with accuracy statistics.

Index