Package: multipleITScontrol (via r-universe)

October 31, 2026

Type Package

Title Interrupted Time Series with a Control and Multiple Interventions

Version 0.1.0

Description Provides tools to perform interrupted-time series through a generalised least squares (GLS) framework on linear outcomes. Allows for multiple interventions and a control with ARMA (autoregressive and moving-average) correction. For more details see Lopez Bernal, Cummins, and Gasparrini (2017) <doi:10.1093/ije/dyw098>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggpubr, DT, forcats, lubridate, sjPlot, stringi, zoo, tidyr

Config/testthat/edition 3

RoxygenNote 7.3.3

VignetteBuilder knitr

Depends R (>= 3.5)

Imports magrittr, dplyr, tibble, rlang, purrr, nlme, AICcmodavg, ggplot2, scales, grid, stats, stringr

URL https://herts-phei.github.io/multipleITScontrol/

Config/pak/sysreqs make libicu-dev

Repository https://herts-phei.r-universe.dev

Date/Publication 2026-03-31 11:29:56 UTC

RemoteUrl https://github.com/herts-phei/multipleitscontrol

RemoteRef HEAD

RemoteSha 6764a9c47d013c0322aa6db4e4b264d502e41617

Contents

fit_its_model 2
generate_predictions 3
its_data_bike_programme 4
its_data_gp 5
its_data_school 5
its_plot 6
slope_difference 7
summary_its 8
transform_data 9

Index **10**

`fit_its_model` *fit_its_model*

Description

Fits an interrupted time series model using the **nlme** package, defaulting to an autocorrelation-moving average correlation structure of order (p, q)

Usage

```
fit_its_model(
  transformed_data,
  impact_model,
  num_interventions,
  method = "REML",
  grid_search = TRUE,
  p = NULL,
  q = NULL,
  return_grid_search = FALSE,
  ...
)
```

Arguments

`transformed_data`	Am unmodified data frame created from ‘transform_data()‘.
`impact_model`	The hypothesized impact model from interventions. Available options are ’level’, ’slope’, or ’levelslope’.
`num_interventions`	The number of interventions in your transformed data. Should be the vector length of ‘intervention_dates‘ passed in ‘transform_data()‘.
`method`	The estimation method for ‘nlme::gls()‘, either "REML" (default) or "ML".

grid_search	logical for whether to perform a grid search for determining lag parameters (p = AR, q = MA). By default, a grid up to values of 5 for each parameter is searched.
p	The order of the autoregressive component. Defaults to 'NULL'. If 'grid_search is enabled', this argument is ignored.
q	The order of the moving average component. Defaults to 'NULL'. If 'grid_search is enabled', this argument is ignored.
return_grid_search	Logical flag returns the result of the grid search instead of the model. 'FALSE' by default.
...	Additional arguments passed to 'nlme::gls()'.

Value

A 'gls' object of the fitted model.

---

generate_predictions *Generate Predictions from a Transformed Dataset and a Model*

Description

This function generates predictions and standard errors for a given transformed dataset using the provided generalized least squares (GLS) model. It also calculates pre-intervention predictions based on model coefficients.

Usage

```
generate_predictions(transformed_data, model)
```

Arguments

transformed_data	A data frame containing the transformed dataset. Output from 'transform_data()'.
model	A GLS model object used to make predictions. The model should include coefficients used to generate pre-intervention predictions.

Details

The function first computes pre-intervention predictions using model coefficients, specifically the intercept, 'time_index', and interaction terms. It ensures that predictions for the pre-intervention period ('level_pre_intervention == 1') are set to 'NA'. Then, it uses the 'AICcmodavg::predictSE.gls()' function to calculate predictions and standard errors from the provided GLS model.

Value

A data frame with the following additional columns:

pre_intervention_predictions Predicted values for the pre-intervention period, calculated using model coefficients.

predictions Predicted values for the entire dataset using the GLS model.

se Standard errors for the predictions.

Examples

```
## Not run:
  # Assuming `transformed_data` is a prepared data frame and `model` is a GLS model:
  predictions_df <- generate_predictions(transformed_data, model)
  head(predictions_df)

## End(Not run)
```

---

its_data_bike_programme

Dummy Bike Programme Data

Description

A dummy data set documenting a two stage level intervention programme of the school scenario in the vignette. Report ...

Usage

```
its_data_bike_programme
```

Format

'its_data_bike_programme' A data frame with 104 rows and 4 columns:

Date Weekly dates

group_var Variable to indicate either treatment or control row of interest

Period Description of intervention phase

score The outcome metric of the scenario - weekly total maintenance incidents" ...

its_data_gp *GP programme Data*

Description

A dummy data set documenting a two stage level intervention programme of the medical GP practice scenario in the vignette. Report ...

Usage

```
its_data_gp
```

Format

'its_data_gp' A data frame with 104 rows and 4 columns:

Date Weekly dates

group_var Variable to indicate either treatment or control row of interest

Period Description of intervention phase

score The outcome metric of the scenario - self-reported wellbeing score" ...

its_data_school *Dummy School Data*

Description

A dummy data set documenting a two stage intervention programme of the school scenario in the vignette. Report ...

Usage

```
its_data_school
```

Format

'its_data_school' A data frame with 104 rows and 4 columns:

Date Weekly dates

group_var Variable to indicate either treatment or control row of interest

Period Description of intervention phase

score The outcome metric of the senario - reading comprehension score" ...

its_plot *its_plot*

Description

Generates a ggplot2 with the values used in the ITS model along with predicted values.

Usage

```
its_plot(
  model,
  data_with_predictions,
  time_var,
  intervention_dates,
  project_pre_intervention_trend = TRUE,
  colours,
  se = TRUE,
  point_shape = 3,
  point_size = 1,
  linetype = 1,
  caption = waiver(),
  title = waiver(),
  subtitle = waiver(),
  x_axis = waiver(),
  y_axis = waiver()
)
```

Arguments

model	model output from ‘multipleITScontrol::summary_its()‘
data_with_predictions	A data frame containing the initial time series data along with predicts created from ‘generate_predictions()‘
time_var	A variable indicating the time index in the data frame. It must be a sequential time-series of equal intervals in numeric or a date/POSIXct/POSIXlt class.
intervention_dates	A vector of time points (matching ‘time_var‘ type) when interventions start. These time points are mutually exclusive and should not overlap. Should match ‘intervention_dates‘ argument used in ‘fit_its_model()‘.
project_pre_intervention_trend	Logical value whether to include a projection of the pre-intervention predicted values. Defaults to ‘TRUE‘.
colours	Colours passed to the ‘values‘ argument in ‘scale_color_manual()‘ and ‘scale_fill_manual()‘. If no colours are given, defaults to ‘c("#3969B5", "#46C3AE")‘.
se	Logical value whether to include standard error values of the predictions. Defaults to ‘TRUE‘.

point_shape	Parameter passed to ‘shape‘ in ‘geom_point‘ to represent the shape of the treatment data points. Defaults to ‘3‘.
point_size	Parameter passed to ‘size‘ in ‘geom_point‘ to represent the size of the treatment data points. Defaults to ‘1‘.
linetype	Parameter passed to ‘linetype‘ in ‘geom_vline‘ to represent the line type of the vertical intervention break points. Defaults to ‘1‘.
caption	Optional argument passed to caption in ‘labs()‘. If no argument is given, defaults to a few descriptive sentences on the lines shown in the plot.
title	Optional argument passed to title in ‘labs()‘.
subtitle	Optional argument passed to subtitle in ‘labs()‘.
x_axis	Optional argument passed to x in ‘labs()‘.
y_axis	Optional argument passed to y in ‘labs()‘.

Value

A ggplot object

`slope_difference` *slope_difference*

Description

Ascertains whether there is a statistically significant difference in a slope or level during an intervention period between the pilot group and the control group. The estimated difference is relative to the control group. I.e. a positive coefficient means the slope of the pilot group is higher than the control group.

The result is functionally equivalent to a two sided t-test.

Usage

```
slope_difference(model, intervention, return = TRUE)
```

Arguments

model	model output object from `multipleITScontrol::fit_its_model`
intervention	which intervention to test significance in difference. Integer value of 1, 2, or 3.
return	Logical for returning pretty output in console. TRUE by default

Value

A transformed data frame to be passed to `transformed_df` in fit_its_model.

summary_its *Summarize and Rename Coefficients for an ITS Model*

Description

This function takes a `gls` model object generated by 'fit_its_model()' and modifies the summary output by renaming the coefficients, variable names, and other model-related terms to make them easier to interpret in the context of interrupted time series (ITS) analysis.

Usage

```
summary_its(model)
```

Arguments

`model`	A `gls` model object that was generated by another custom function in the package.

Details

The function modifies various components of the `gls` object, including:

- Coefficients
- Variable names in matrices (e.g., variance-covariance)
- Terms and factors used in the model
- Predicted variables

The renamed terms in the model output are specifically tailored to better describe the parameters of ITS models, such as control and pilot group slopes before and after interventions.

The function provides more meaningful names for ITS models by replacing default coefficient names with:

- `(Intercept)` becomes "Control y-axis intercept"
- `x` becomes "Pilot y-axis intercept difference to control"
- `time_index` becomes "Control pre-intervention slope"
- `x:time_index` becomes "Pilot pre-intervention slope difference to control"

Additional terms for up to 3 interventions are automatically renamed, reflecting intervention-related slopes in both control and pilot groups.

Value

A modified summary of the `gls` model object, where the coefficient names and other relevant model attributes have been renamed to be more interpretable.

transform_data *transform_data*

Description

Transforms a data frame ready for input into fit_its_model, creating relevant columns for slope and level effects of interventions.

Requires a minimum of three time points in each intervention period and pre-intervention period.

Usage

```
transform_data(df, time_var, group_var, outcome_var, intervention_dates)
```

Arguments

df	A data frame containing the initial time series data.
time_var	A variable indicating the time index in the data frame. It must be a sequential time-series of equal intervals in numeric or a date/POSIXct/POSIXlt class.
group_var	A character or factor variable indicating treatment and control groups. Only 'treatment' and 'control' are valid elements.
outcome_var	A numeric variable indicating outcome.response variable.
intervention_dates	A vector of time points (matching type of time_var) when interventions start. These time points are mutually exclusive and should not overlap. The argument accepts up to three values representing the intervention start times.

Value

A transformed data frame to be passed to transformed_df in fit_its_model.

Index

∗ **datasets**
    its_data_bike_programme, 4
    its_data_gp, 5
    its_data_school, 5

fit_its_model, 2, *7*, *9*

generate_predictions, 3

its_data_bike_programme, 4
its_data_gp, 5
its_data_school, 5
its_plot, 6

slope_difference, 7
summary_its, 8

transform_data, 9