

Package ‘rpanelauto’

August 5, 2026

Title Perform Automatic Estimation on Time Series in Multidimensional Panels

Version 1.0.0

Description Provides a framework for automatic estimation of time-series models for univariate and multidimensional panel data. A user-supplied estimation function is applied independently to each time series, with optional processing before and after estimation. The function returns the transformed data together with the fitted models.

Language en-US

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.3)

Suggests forecast, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/econcz/rpanelauto>

BugReports <https://github.com/econcz/rpanelauto/issues>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Ilya Bolotov [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-1148-7144>>)

Maintainer Ilya Bolotov <ilya.bolotov@vse.cz>

Repository CRAN

Date/Publication 2026-08-05 07:10:02 UTC

Contents

panelauto	2
Index	7

panelauto	<i>Perform Automatic Estimation on Time Series in Multidimensional Panels</i>
-----------	---

Description

Applies a user-supplied automatic estimation function independently to the selected time-series variables in a data frame. Estimation is performed separately for every combination of the preceding dimensions, while the final dimension determines the ordering of observations within each time series.

Usage

```
panelauto(
  data,
  vars,
  dimensions,
  estimate,
  frequency = 1,
  preestimation = NULL,
  postestimation = NULL,
  ...
)
```

Arguments

data	data frame Input data containing the dimension variables and the time-series variables to be modelled.
vars	character vector Names of the numeric variables for which models are to be estimated. A separate model is estimated for every selected variable and every combination of the grouping dimensions.
dimensions	character vector Names of the variables that uniquely identify observations. The last variable specifies the dimension along which observations are ordered, while all preceding variables define independent groups. When only one dimension is supplied, the selected variables are treated as ungrouped time series.
estimate	function Automatic estimation function applied independently to each time series. The function must accept a ts object as its first argument and return a fitted model object. Additional arguments required by the estimation function can be supplied through <code>...</code> . The function may be provided by another package or defined by the user.
frequency	numeric scalar, default = 1 Frequency used when converting each ordered variable to a ts object. For example, use 1 for annual, 4 for quarterly, and 12 for monthly observations.
preestimation	function or NULL, default = NULL Function evaluated after each panel group has been ordered and before estimation. The function must accept the ordered group data frame as its first argument and return a data frame with the same number

of observations and unchanged dimension variables. The returned data frame is subsequently used to estimate all selected variables in that group.

`postestimation` function or NULL, default = NULL Function evaluated after each model has been estimated. The function must accept the ordered group data frame as its first argument and the fitted model as its second argument. It may additionally accept a named argument `variable`, identifying the variable associated with the current model. It must return the final result as a named list containing components `data` and `model`. The function may add fitted values, residuals, forecasts, standard errors, or other post-estimation quantities to the group data frame.

... Optional. Additional arguments passed unchanged to estimate.

Details

When only one dimension is supplied, each selected variable is treated as a single ungrouped time series. Optional pre-estimation and post-estimation functions can be evaluated before and after each automatic estimation.

The final variable in `dimensions` determines the ordering of observations within each time series. All preceding dimension variables define independent groups.

For example, with

```
dimensions = c("country", "sector", "year")
```

observations are ordered by year, and separate models are estimated for every variable-country-sector combination.

With

```
dimensions = "year"
```

each variable listed in `vars` is treated as one ungrouped time series.

The dimension variables themselves are not modified, and the original row order of data is preserved.

Missing values in variables listed in `vars` are passed unchanged to the estimation function. Their admissibility and treatment therefore depend on the selected estimator.

Missing values in dimension variables are not permitted. Each combination of the dimension variables must uniquely identify one observation.

The order of operations within each panel group is:

1. Order observations by the final dimension.
2. Apply preestimation to the ordered group data.
3. Convert each selected variable to a `ts` object.
4. Apply estimate independently to each time series.
5. Apply postestimation to the group data and fitted model.
6. Store the fitted model and copy the transformed data back to the original row positions.

The preestimation and postestimation functions must preserve the number of observations and the values of all dimension variables within each group.

When several variables are modelled, postestimation should generally use variable-specific output names to avoid overwriting quantities created for another model. When the function declares a variable argument, panelauto supplies the name of the current variable automatically.

Any estimation function can be used, provided that it accepts the time series as its first argument. Functions with different interfaces can be supplied through a user-defined wrapper.

Value

A named list containing at least the following components:

data The complete data frame after application of the optional preestimation and postestimation functions. The original row order is preserved.

models A named list containing one model record for every selected variable and every combination of the grouping dimensions. Each model record contains the following components:

variable Name of the modelled variable.

group Named list identifying the values of the grouping dimensions. This component is an empty list when no grouping dimensions are supplied.

model The fitted model returned by `estimate`, or the model component returned by `postestimation`.

See Also

[arima](#), [ts](#), [auto.arima](#), [ets](#)

Examples

```
## Example: user-defined automatic autoregressive estimation

set.seed(123456789)

data <- data.frame(
  country = rep(c("A", "B"), each = 20L),
  year    = rep(2001:2020, 2L),
  y       = c(
    cumsum(stats::rnorm(20L)),
    cumsum(stats::rnorm(20L))
  )
)

automatic_ar <- function(series, orders = 1:3) {
  models <- lapply(
    orders,
    function(p) {
      stats::arima(
        series,
        order = c(p, 0L, 0L)
      )
    }
  )
}
```

```

    aic <- vapply(
      models,
      stats::AIC,
      numeric(1L)
    )

    models[[which.min(aic)]]
  }

result <- panelauto(
  data      = data,
  vars      = "y",
  dimensions = c("country", "year"),
  estimate  = automatic_ar,
  orders    = 1:2
)

print(result$data)
print(result$models[[1L]]$model)

## Example: automatic ARIMA estimation

if (requireNamespace("forecast", quietly = TRUE)) {
  result <- panelauto(
    data      = data,
    vars      = "y",
    dimensions = c("country", "year"),
    estimate  = forecast::auto.arima,
    seasonal  = FALSE
  )

  print(result$models[[1L]]$model)
}

## Pre-estimation, estimation, and fitted values

result <- panelauto(
  data      = data,
  vars      = "y",
  dimensions = c("country", "year"),
  preestimation = function(data) {
    data$y <- data$y - mean(data$y)
    data
  },
  estimate = automatic_ar,
  orders   = 1:2,
  postestimation = function(data, model, variable) {
    fitted_name <- paste0(variable, "_fitted")
    residual_name <- paste0(variable, "_residual")

    data[[residual_name]] <- as.numeric(stats::residuals(model))
    data[[fitted_name]] <- data[[variable]] -

```

```
as.numeric(stats::residuals(model))

      list(
        data = data,
        model = model
      )
)

print(result$data)
```

Index

arima, [4](#)
auto.arima, [4](#)

ets, [4](#)

panelauto, [2](#)

ts, [2](#), [4](#)