

Package ‘joinpointR’

October 8, 2026

Title Tidy Tools for Joinpoint Regression Models

Version 2.0.0

Description Provides tools to fit joinpoint regression models with a log-linear specification by levels of one or two categorical variable(s) using the grid-search method. It includes functions to estimate the Annual Percent Change (APC) and the Average Annual Percent Change (AAPC), along with their 95% confidence intervals, and to generate formatted summary tables and plots of results.

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/datos-ine/joinpointR>

Imports dplyr, flextable, ggplot2, grDevices, lubridate, purrr, readr, rlang, stats, stringr, tibble, tidyr, utils

Config/roxygen2/version 8.1.0

Depends R (>= 4.1)

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Tamara Ricardo [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-0921-2611>>)

Maintainer Tamara Ricardo <tamararicardo83@gmail.com>

Repository CRAN

Date/Publication 2026-10-08 16:00:02 UTC

Contents

bic_jp	2
get_summary	3
gg_jpoint	5

hiv_data	7
model_jp	8
plot_cbpal	10
scale_cbpal	11
summary.model_jp	12

Index	13
--------------	-----------

bic_jp	<i>Bayesian Information Criteria (BIC) for joinpoint regression models</i>
--------	--

Description

Computes the Bayesian Information Criterion (BIC), the penalized BIC (BIC3), and the weighted BIC (WBIC) for models of class `model_jp`.

Usage

```
bic_jp(models)
```

Arguments

`models` Models of class `model_jp`.

Details

The Bayesian Information Criterion (BIC) for k joinpoints is estimated as:

$$BIC = \log(MSE) + \frac{param(k)}{n} * \log n$$

where MSE is the mean squared error of the fitted model, $param(k)$ is the number of parameters in the model, and n is the number of observations.

The number of parameters for the standard BIC are calculated as:

$$param(k) = 2k + 2$$

For BIC3, the penalty parameter is defined as:

$$param(k) = 3k + 2$$

The weighted Bayesian Information Criterion (WBIC) combines BIC and BIC3 using a weighted penalty term based on the data:

$$WBIC = BIC * (1 - wt) + BIC3 * wt$$

where wt is the calculated weighted penalty.

Value

A tibble containing the values for the BIC, BIC3, WBIC and the weights used to estimate the WBIC.

Examples

```
# Load packages
library(dplyr)

# Create a reduced dataset
data <- hiv_data |>
  filter(admin == "ARG")

# Fit the joinpoint models
mods <- model_jp_grid(data = data, rate = hiv_rate, time = year, group = "sex")

# Check the BIC, BIC3, WBIC values for a single model
bic_jp(mods[1])
```

get_summary

*Summary statistics for joinpoint regression models***Description**

Calculates the annual percent change (APC) and the average annual percent change (AAPC) for joinpoint regression models.

Usage

```
get_summary(
  models,
  stats = c("both", "apc", "aapc"),
  hide = c("none", "ci", "sig"),
  level.ci = 0.95,
  as.ft = FALSE,
  dec = c(".", "", "")
)

get_apc(models, stats = "apc", ...)

get_aapc(models, stats = "aapc", ...)
```

Arguments

models	A model or a list of models of class <code>model_jp</code> .
stats	Character. Summary statistics to display in the table. One of "both", "apc", or "aapc". Defaults to "both".

hide	Character. Which elements to hide: the confidence interval ("ci"), significance stars ("sig"), or none ("none"). Defaults to "none".
level.ci	Numeric. Confidence level used to calculate confidence intervals. Must be between 0 and 1. Defaults to 0.95.
as.ft	Logical. Whether to return the summary as a flextable object instead of a tibble. Defaults to FALSE.
dec	Character. Decimal separator to use, either a point (".") or a comma (","). Defaults to ".".
...	Additional arguments passed to get_summary().

Details

For each segment, the Annual Percent Change (APC) is calculated from the estimated slope $\hat{\beta}$ of the log-linear model as:

$$APC = 100[\exp(\hat{\beta}) - 1].$$

Confidence intervals for the APC are calculated using the Student's t distribution and the standard error of the estimated slope. The resulting confidence limits are then back-transformed to the APC scale:

$$APC_{lower} = 100[\exp \hat{\beta} - t_{1-\alpha/2, df} SE(\hat{\beta}) - 1],$$

$$APC_{upper} = 100[\exp \hat{\beta} + t_{1-\alpha/2, df} SE(\hat{\beta}) - 1].$$

The standard errors used for the segment-specific slopes are obtained from the variance-covariance matrix of the fitted joinpoint model.

For each fitted model, the Annual Average Percent Change (AAPC) is calculated as the exponential transformation of the weighted average of the segment-specific slopes on the log scale:

$$AAPC = 100 \left[\exp \left(\sum_j w_j \hat{\beta}_j \right) - 1 \right],$$

where $\hat{\beta}_j$ is the estimated slope for segment j , and w_j is the length of segment j divided by the total length of the selected time interval. Thus, the weights sum to one. For models without joinpoints, the AAPC is equivalent to the APC.

The confidence interval for the AAPC is calculated on the log scale. The variance of the weighted slope is obtained from the variance-covariance matrix of the segment-specific slopes. The confidence limits are calculated using the Student's t distribution and the residual degrees of freedom of the fitted model, and are then back-transformed to the AAPC scale:

$$AAPC_{lower} = 100 \left[\exp \left\{ \hat{\beta}_{AAPC} - t_{1-\alpha/2, df} SE(\hat{\beta}_{AAPC}) \right\} - 1 \right].$$

$$AAPC_{upper} = 100 \left[\exp \left\{ \hat{\beta}_{AAPC} + t_{1-\alpha/2, df} SE(\hat{\beta}_{AAPC}) \right\} - 1 \right].$$

The confidence intervals described above are based on the variance-covariance matrix and residual degrees of freedom of the fitted joinpoint model and may therefore differ from confidence intervals reported by other joinpoint regression software using different inferential procedures.

Value

For `get_summary()`, a [tibble](#) or a [flextable](#) containing the APC and/or AAPC for each model, along with their confidence intervals and significance stars according to the values of `stats` and `hide`.

For `get_apc()`, a [tibble](#) or a [flextable](#) containing the APC for each model, along with its confidence intervals and significance stars according to the value of `hide`.

For `get_aapc()`, a [tibble](#) or a [flextable](#) containing the AAPC for each model, along with its confidence intervals and significance stars according to the value of `hide`.

Examples

```
# Create an example dataset
data <- hiv_data |>
  dplyr::filter(admin == "ARG")

# Fit the joinpoint models
mods <- model_jp_grid(data = data, rate = hiv_rate, time = year, group = "sex")

# Obtain the model summary
get_summary(mods = mods)

# Same output calling summary(mods)
summary(mods, as.ft = TRUE)

# Obtain the APC with 95% CI
get_apc(mods = mods)

# Obtain the AAPC with 95% CI
get_aapc(mods = mods)
```

Description

Creates a `ggplot2` visualization of joinpoint regression models, showing observed values, fitted regression lines, and optionally the estimated joinpoints and Average Annual Percent Change (AAPC).

Usage

```
gg_jpoint(
  models,
  geom = c("linepoint", "line", "area"),
  facets = c("wrap", "grid", "grid2"),
  color.by = NULL,
  jp = TRUE,
  aapc = FALSE,
  time.var = c("year", "yearmon"),
  ...
)

gg_jpoint_area(models, ...)

gg_jpoint_line(models, ...)
```

Arguments

<code>models</code>	A model or a list of models of class <code>model_jp</code> , returned by <code>model_jp_grid()</code> .
<code>geom</code>	Character. Determines how the model results are displayed. <code>geom = "line"</code> displays the fitted regression lines; <code>geom = "linepoint"</code> displays the fitted regression lines together with the observed data points; and <code>geom = "area"</code> displays the fitted lines with background color depending on the grouping variable, time period or trend. Defaults to <code>"linepoint"</code> .
<code>facets</code>	Character. Determines the facet layout. <code>facets = "wrap"</code> displays one facet for each grouping level; <code>facets = "grid"</code> displays groups in rows and subgroups in columns; and <code>facets = "grid2"</code> displays subgroups in rows and groups in columns. This argument is ignored when only one model is provided.
<code>color.by</code>	Character. Determines whether the data should be colored using the grouping variable (<code>"group"</code>), the time periods (<code>"period"</code>), the time segments (<code>"segment"</code>), or the APC trend (<code>"trend"</code>). Defaults to <code>"group"</code> for <code>geom = "linepoint"</code> , to <code>"period"</code> for <code>geom = "area"</code> , and to <code>"segment"</code> for <code>geom = "line"</code> .
<code>jp</code>	Logical. Whether to display the estimated joinpoint(s) as vertical lines. Defaults to <code>TRUE</code> .
<code>aapc</code>	Logical. Whether to display a label with the Average Annual Percent Change (AAPC) and its statistical significance. Defaults to <code>FALSE</code> .
<code>time.var</code>	Character. Determines the class of the time variable, among <code>"year"</code> and <code>"yearmon"</code> . Defaults to <code>"year"</code> .
<code>...</code>	Additional arguments passed to <code>ggplot2</code> .

Details

Available colorblind-friendly palettes can be checked using `plot_cbpal()`.

Value

A ggplot2 object showing observed values, fitted joinpoint regression lines, and optionally the estimated joinpoints and AAPC.

Examples

```
# Create an example dataset
data <- hiv_data |>
dplyr::filter(admin %in% c("CABA", "Catamarca"))

# Fit models
mods <- model_jp_grid(data = data, rate = hiv_rate, time = year,
group = c("admin", "sex"))

# Plot results with AAPC
gg_jpoint(models = mods, aapc = TRUE)
```

hiv_data*HIV Incidence Rates in Argentina by Sex and Jurisdiction*

Description

HIV Incidence Rates in Argentina by Sex and Jurisdiction

Usage

```
hiv_data
```

Format

A data frame with 4 variables:

admin Administrative division or jurisdiction name.

year Calendar year of observation.

sex Biological sex (male, female, or both sexes).

hiv_rate Incidence rate per 100,000 inhabitants.

Details

A dataset containing HIV incidence rates for males, females, and both sexes combined, covering the national level as well as individual jurisdictions.

Source

<https://datos.gob.ar>

model_jp

*Fit Joinpoint Regression Models Using Grid Search***Description**

Fits log-linear joinpoint regression models using the grid-search method, assuming constant variance and uncorrelated errors. The best-fit model is selected according to the Bayesian Information Criterion (BIC).

Usage

```
model_jp(
  data,
  rate,
  time,
  group = NULL,
  jp = 2,
  min.dist = 2,
  method = "bic",
  silent = FALSE
)

model_jp_grid(data, ...)

## S3 method for class 'model_jp'
update(object, ...)
```

Arguments

data	A data.frame or tibble containing rates, time points, and optional grouping variables.
rate	Character string specifying the variable with the rates.
time	Character string specifying the variable with the time points.
group	Character vector specifying the name(s) of the variable(s) used to group the data. A maximum of two grouping variables is allowed. Defaults to NULL.
jp	Integer specifying the maximum number of joinpoints to test. Must be between 0 and 7 (see Details). Defaults to 2.
min.dist	Integer specifying the minimum number of time points required between consecutive joinpoints, as well as between each endpoint of the time series and the nearest joinpoint. Defaults to 2.
method	Character string specifying the method used to calculate the BIC. Options are "bic" for standard BIC, "bic3" for penalized BIC (BIC3), or "wbic" for weighted BIC (WBIC). Defaults to "bic" (see Details).
silent	Logical. Whether to display a message showing the model name, number of joinpoints detected and BIC of the best fit model. Defaults to TRUE.

... Additional parameters passed to `model_jp()`.
 object Model or list of models to update.

Details

The maximum recommended number of joinpoints is determined by the number of time points in the series, following the criteria described by Kim et al. (2000):

- 0–6 time points: 0 joinpoints.
- 7–11 time points: 1 joinpoint.
- 12–16 time points: 2 joinpoints.
- 17–21 time points: 3 joinpoints.
- 22–26 time points: 4 joinpoints.
- 27–31 time points: 5 joinpoints.
- 32–36 time points: 6 joinpoints.
- 37+ time points: 7 joinpoints.

When grouping variables are specified, all groups must contain the same number of time points.

Value

A named list with one element per group, where each element contains:

- `fit`: An `lm` object corresponding to the selected joinpoint model.
- `joinpoints`: Numeric vector with the estimated joinpoint positions of the selected model.
- `time`: Vector of time points used to fit the model.
- `log_rate`: Vector of log-transformed rates used as the response variable.
- `BIC`: BIC value of the selected model based on the specified method.
- `model_sel`: A tibble containing the number and positions of joinpoints, SSE, BIC, BIC3, Weights and WBIC for all evaluated candidate models.

References

Kim, H. J., Fay, M. P., Feuer, E. J., & Midthune, D. N. (2000). Permutation tests for joinpoint regression with applications to cancer rates. *Statistics in Medicine*, 19(3), 335–351. [https://doi.org/10.1002/\(sici\)1097-0258\(20000215\)19:3%253C335::aid-sim336%253E3.0.co;2-z](https://doi.org/10.1002/(sici)1097-0258(20000215)19:3%253C335::aid-sim336%253E3.0.co;2-z)

Examples

```
# Create a reduced dataset
data <- hiv_data |>
  dplyr::filter(admin == "ARG")

# Fit the joinpoint models
mods <- model_jp(data = data, rate = hiv_rate,
  time = year, group = "sex")
```

```
# Update model using the BIC3
mods_bic3 <- update(mods, method = "bic3")
```

plot_cbpal

Display colorblind-friendly palettes

Description

Plots a list of available colorblind-friendly palettes. Allows filtering by palette type, series, or name.

Usage

```
plot_cbpal(type = c("all", "cat", "seq", "div"), series = NULL, name = NULL)
```

Arguments

type	Character. Selects palettes based on type: "cat" for categorical, "div" for diverging, or "seq" for sequential. Defaults to "all".
series	Character vector. Selects palettes based on their series name. Defaults to NULL.
name	Character vector. Selects one or more specific palettes by name. Defaults to NULL.

Details

Available palette series:

brewer, carto, cols4all, gmt, hcl, kovesi, matplotlib, met, meteo, misc, ocean, parks, powerbi, scico, seaborn, stevens, tableau, tol.

Value

A [ggplot](#) object showing the selected palettes.

Examples

```
# Display all available palettes
plot_cbpal()

# Display specific palettes by name
plot_cbpal(name = c("viridis", "tokyo", "algae"))

# Display a specific palette series
plot_cbpal(series = "seaborn")
```

scale_cbpal	<i>Colorblind-friendly scales</i>
-------------	-----------------------------------

Description

Scales for applying colorblind-friendly palettes to ggplot2 aesthetics.

Usage

```
scale_cbpal(  
  palette = NULL,  
  reverse = FALSE,  
  discrete = TRUE,  
  aesthetics = c("fill", "color"),  
  ...  
)  
  
scale_cbpal_fill(aesthetics = "fill", ...)  
  
scale_cbpal_colour(aesthetics = "color", ...)  
  
scale_cbpal_color(aesthetics = "color", ...)
```

Arguments

palette	Character. String specifying the palette name. Defaults to "viridis".
reverse	Logical. If TRUE, reverses the order of colors in the palette. Defaults to FALSE.
discrete	Logical. Should the scale be discrete?
aesthetics	Character vector of aesthetics to apply the scale to.
...	Additional arguments passed to ggplot2 scale constructor.

Value

A ggplot2 scale.

Examples

```
library(ggplot2)  
data(iris)  
  
# Use a discrete variable for color  
iris |>  
ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +  
  geom_point() +  
  scale_cbpal_color()  
  
# Use a continuous variable for color
```

```
iris |>
ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Petal.Length)) +
  geom_point() +
  scale_cbpal_color(discrete = FALSE)

# Use a discrete variable for fill
iris |>
ggplot(aes(x = Sepal.Length, fill = Species)) +
  geom_bar() +
  scale_cbpal_fill()
```

summary.model_jp	<i>Use the shortcut summary()</i>
------------------	-----------------------------------

Description

Use the shortcut summary()

Usage

```
## S3 method for class 'model_jp'
summary(object, ...)
```

Arguments

object	Model or list of models to update.
...	Additional arguments passed to get_summary().

Index

* datasets

hiv_data, [7](#)

bic_jp, [2](#)

flextable, [5](#)

get_aapc (get_summary), [3](#)

get_apc (get_summary), [3](#)

get_summary, [3](#)

gg_jpoint, [5](#)

gg_jpoint_area (gg_jpoint), [5](#)

gg_jpoint_line (gg_jpoint), [5](#)

ggplot, [10](#)

hiv_data, [7](#)

model_jp, [8](#)

model_jp_grid (model_jp), [8](#)

plot_cbpal, [10](#)

scale_cbpal, [11](#)

scale_cbpal_color (scale_cbpal), [11](#)

scale_cbpal_colour (scale_cbpal), [11](#)

scale_cbpal_fill (scale_cbpal), [11](#)

summary.model_jp, [12](#)

tibble, [5](#)

update (model_jp), [8](#)