

# Package ‘fixes’

May 8, 2026

**Type** Package

**Title** Tools for Creating and Visualizing Fixed-Effects Event Study Models

**Version** 0.7.2

**Date** 2026-05-02

**Description** Provides functions for creating, analyzing, and visualizing event study models using fixed-effects regression. Supports staggered adoption, multiple confidence intervals, flexible clustering, and panel/time transformations in a simple workflow.

**Depends** R (>= 4.1.0)

**Imports** dplyr, ggplot2, fixest, broom, tibble, rlang

**License** MIT + file LICENSE

**Encoding** UTF-8

**Suggests** knitr, rmarkdown, haven, testthat, plotly, tidyr

**VignetteBuilder** knitr

**URL** <https://github.com/yo5uke/fixes>

**BugReports** <https://github.com/yo5uke/fixes/issues>

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Yosuke Abe [aut, cre]

**Maintainer** Yosuke Abe <yosuke.abe0507@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-05-02 18:40:14 UTC

## Contents

autoplot.es_result . . . . .	2
plot_es . . . . .	2
plot_es_interactive . . . . .	3
run_es . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

---

autoplot.es\_result      *Autoplot for event-study results*

---

### Description

S3 method that plots an es\_result (from run\_es()). It forwards arguments to [plot\\_es](#).

### Usage

```
## S3 method for class 'es_result'
autoplot(object, ci_level = 0.95, type = "ribbon", ...)
```

### Arguments

object	An es_result returned by run_es().
ci_level	Confidence level (numeric, e.g., 0.95). Passed to plot_es().
type	Plot type: "ribbon" (default) or "errorbar". Passed to plot_es().
...	Additional arguments forwarded to <a href="#">plot_es</a> .

### Value

A ggplot2 ggplot object.

### Examples

```
# res <- run_es(...)
# ggplot2::autoplot(res, ci_level = 0.95, type = "ribbon")
```

---

plot\_es      *Plot event-study results with ribbons or error bars*

---

### Description

Plot event-study results with ribbons or error bars

### Usage

```
plot_es(
  data,
  ci_level = 0.95,
  type = "ribbon",
  vline_val = 0,
  vline_color = "#000",
  hline_val = 0,
```

```

hline_color = "#000",
linewidth = 1,
pointsize = 2,
alpha = 0.2,
barwidth = 0.2,
color = "#B25D91FF",
fill = "#B25D91FF",
theme_style = "bw"
)

```

### Arguments

**data** An object of class `es_result` returned by `run_es()`.

**ci\_level** Confidence level to display (e.g., 0.95).

**type** One of "ribbon" (default) or "errorbar".

**vline\_val, hline\_val** Numeric locations for vertical/horizontal reference lines (default 0).

**vline\_color, hline\_color** Colors for reference lines.

**linewidth, pointsize, alpha, barwidth** Styling parameters for lines/points/bands/bars.

**color, fill** Optional, override line/point color and ribbon fill.

**theme\_style** One of "bw", "minimal", or "classic" for ggplot theme.

### Value

A ggplot object.

### Examples

```

# Assuming `res <- run_es(...)`
# p <- plot_es(res, ci_level = 0.95, type = "ribbon")
# print(p)

```

---

plot\_es\_interactive *Interactive event-study plot with hover details*

---

### Description

Creates an interactive plotly visualization of event study results with hover-over displays showing coefficients, confidence intervals, and other details.

**Usage**

```
plot_es_interactive(
  data,
  ci_level = 0.95,
  vline_val = 0,
  hline_val = 0,
  vline_color = "#000",
  hline_color = "#000",
  color = "#B25D91FF",
  fill = "#B25D91FF",
  alpha = 0.2,
  linewidth = 2,
  markersize = 8,
  show_ribbon = TRUE,
  height = NULL,
  width = NULL
)
```

**Arguments**

<code>data</code>	An object of class <code>es_result</code> returned by <code>run_es()</code> .
<code>ci_level</code>	Confidence level to display (e.g., 0.95). Default is 0.95.
<code>vline_val</code>	Numeric location for vertical reference line (default 0).
<code>hline_val</code>	Numeric location for horizontal reference line (default 0).
<code>vline_color</code>	Color for vertical reference line (default "#000").
<code>hline_color</code>	Color for horizontal reference line (default "#000").
<code>color</code>	Point and line color (default "#B25D91FF").
<code>fill</code>	Ribbon/band fill color (default "#B25D91FF").
<code>alpha</code>	Ribbon transparency (default 0.2).
<code>linewidth</code>	Line width (default 2).
<code>markersize</code>	Marker size (default 8).
<code>show_ribbon</code>	Logical; if TRUE, shows confidence interval as a ribbon band (default TRUE).
<code>height</code>	Plot height in pixels (default NULL for auto).
<code>width</code>	Plot width in pixels (default NULL for auto).

**Details**

The hover tooltip displays:

- Relative time to treatment
- Point estimate (coefficient)
- Confidence interval bounds
- Standard error
- P-value

**Value**

A plotly object that can be displayed interactively.

**Examples**

```
## Not run:  
# Assuming res <- run_es(...)  
plot_es_interactive(res)  
plot_es_interactive(res, ci_level = 0.99, show_ribbon = FALSE)  
  
## End(Not run)
```

---

run\_es

*Event Study Estimation for Panel Data*

---

**Description**

Runs an event study regression on panel data, supporting both classic (universal timing) and staggered (unit-varying timing via sunab). The function builds the design (lead/lag factor or sunab), estimates with **fixest**, and returns a tidy table with metadata.

**Usage**

```
run_es(  
  data,  
  outcome,  
  treatment,  
  time,  
  timing,  
  fe = NULL,  
  lead_range = NULL,  
  lag_range = NULL,  
  covariates = NULL,  
  cluster = NULL,  
  weights = NULL,  
  baseline = -1L,  
  interval = 1,  
  time_transform = FALSE,  
  unit = NULL,  
  staggered = FALSE,  
  method = c("classic", "sunab"),  
  conf.level = 0.95,  
  vcov = "HC1",  
  vcov_args = list()  
)
```

**Arguments**

<code>data</code>	A <code>data.frame</code> containing panel data.
<code>outcome</code>	Unquoted outcome (name or expression, e.g., <code>log(y)</code> ).
<code>treatment</code>	Unquoted treatment indicator (0/1 or logical). Used only when <code>method = "classic"</code> .
<code>time</code>	Unquoted time variable (numeric or Date).
<code>timing</code>	For <code>classic</code> : a numeric/Date (universal) or a variable (unquoted) if <code>staggered = TRUE</code> . For <code>sunab</code> : an unquoted variable with adoption time.
<code>fe</code>	One-sided fixed-effects formula, e.g., <code>~ id + year</code> . Can be <code>NULL</code> for no fixed effects.
<code>lead_range, lag_range</code>	Integers for pre/post windows. If <code>NULL</code> , determined automatically.
<code>covariates</code>	One-sided formula of additional controls, e.g., <code>~ x1 + log(x2)</code> .
<code>cluster</code>	Cluster specification (one-sided formula like <code>~ id + year</code> , a single character column name, or a vector of length <code>nrow(data)</code> ).
<code>weights</code>	Observation weights (a name/one-sided formula or a numeric vector of length <code>nrow(data)</code> ).
<code>baseline</code>	Integer baseline period (default <code>-1</code> ); reference period excluded from results for both <code>"classic"</code> and <code>"sunab"</code> methods.
<code>interval</code>	Numeric spacing of the time variable (default <code>1</code> ; ignored internally for Dates).
<code>time_transform</code>	Logical; if <code>TRUE</code> , creates consecutive integer time within unit.
<code>unit</code>	Unit identifier variable (required when <code>time_transform = TRUE</code> ); also used for metadata when supplied.
<code>staggered</code>	Logical; if <code>TRUE</code> , <code>timing</code> is a variable ( <code>classic</code> ) or is used by <code>sunab</code> .
<code>method</code>	Either <code>"classic"</code> or <code>"sunab"</code> (default: <code>"classic"</code> ).
<code>conf.level</code>	Numeric vector of confidence levels (default <code>0.95</code> ).
<code>vcov</code>	VCOV type passed to <code>fixest::vcov()</code> or used via <code>broom::tidy(vcov = ...)</code> . Default <code>"HC1"</code> .
<code>vcov_args</code>	List of additional arguments forwarded to <code>fixest::vcov()</code> .

**Value**

A `data.frame` of class `"es_result"` with columns:

- `term`, `estimate`, `std.error`, `statistic`, `p.value`
- `conf_low_XX`, `conf_high_XX` (for each requested `conf.level`)
- `relative_time` (integer; `0 = event`), `is_baseline` (logical; marks the reference period)

Attributes include: `lead_range`, `lag_range`, `baseline`, `interval`, `call`, `model_formula`, `conf.level`, `N`, `N_units`, `N_treated`, `N_nevertreated`, `fe`, `vcov_type`, `cluster_vars`, `staggered`, `sunab_used`.

**Key Features**

- One-step event study: specify outcome, treatment, time, timing, fixed effects, and (optionally) covariates.
- Switch between Classic (factor expansion) and Staggered-SAFE (`method = "sunab"`).
- Flexible clustering, weights, and VCOV choices (e.g., `vcov = "HC1" | "HC3" | "CR2" | "iid" ...`).
- Automatic lead/lag window detection and customizable baseline period.
- Returns an `"es_result"` object compatible with `print()` and `autoplot()`.

# Index

`autoplot.es_result`, [2](#)

`plot_es`, [2](#), [2](#)

`plot_es_interactive`, [3](#)

`run_es`, [5](#)