Package ‘fixes’

March 18, 2025

Type Package

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

Version 0.1.0

Description

Provides functions for creating, analyzing, and visualizing event study models using fixed-effects regression.

Depends R (>= 4.1.0)

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

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Yosuke Abe [aut, cre]

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

Repository CRAN

Date/Publication 2025-03-18 15:10:06 UTC

Contents

`plot_es` *Plot Event Study Results*

Description

This function creates a plot for event study results using ‘ggplot2‘. Users can choose between ribbon-style confidence intervals or error bars to visualize the estimates and their uncertainty.

Usage

```
plot_es(
  data,
  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"
)
```

Arguments

`data`	A dataframe containing the results from the ‘run_es‘ function. The dataframe must include the following columns: - ‘relative_time‘: The scaled time relative to the treatment. - ‘estimate‘: The estimated coefficients. - ‘conf_low‘: The lower bound of the 95 - ‘conf_high‘: The upper bound of the 95 - ‘std.error‘: The standard errors (required if ‘type = "errorbar"‘).
`type`	The type of confidence interval visualization: "ribbon" (default) or "errorbar". - "ribbon": Shaded area representing the confidence intervals. - "errorbar": Vertical error bars for each estimate.
`vline_val`	The x-intercept for the vertical reference line (default: 0). Typically represents the time of treatment.
`vline_color`	The color of the vertical reference line (default: "#000").
`hline_val`	The y-intercept for the horizontal reference line (default: 0). Usually represents the null effect line.
`hline_color`	The color of the horizontal reference line (default: "#000").
`linewidth`	The width of the lines in the plot (default: 1).
`pointsize`	The size of the points for the estimates (default: 2).
`alpha`	The transparency level for the ribbon (default: 0.2).

`barwidth` The width of the error bars (default: 0.2).

`color` The color of the lines and points (default: "#B25D91FF").

`fill` The fill color for the ribbon (default: "#B25D91FF").

Details

This function provides a flexible visualization tool for event study results. Users can customize the appearance of the plot by adjusting the parameters for line styles, point sizes, colors, and confidence interval types.

Column Requirements: The input dataframe (‘data‘) must include: - ‘relative_time‘: A numeric column for the time relative to the treatment. - ‘estimate‘: The estimated coefficients for each relative time. - ‘conf_low‘ and ‘conf_high‘: The bounds of the confidence intervals. - ‘std.error‘: The standard errors (only required if ‘type = "errorbar"‘).

Type Options: - ‘"ribbon"‘: A shaded area to represent the confidence intervals. - ‘"errorbar"‘: Error bars for each point. Standard errors (‘std.error‘) are required.

Value

A ggplot object displaying the event study results. The plot includes: - A line connecting the estimates over relative time. - Points for the estimated coefficients. - Either ribbon-style confidence intervals or error bars, depending on ‘type‘. - Vertical and horizontal reference lines for better interpretability.

Note

If ‘type = "errorbar"‘, ensure that the ‘std.error‘ column is present in the input dataframe. Missing values in the ‘std.error‘ column for any term will result in incomplete confidence intervals.

Examples

```
# Simulate panel data
df <- tibble::tibble(
  firm_id = rep(1:50, each = 10),  # 50 firms over 10 years
  state_id = rep(sample(1:10, size = 50, replace = TRUE), each = 10),
  year = rep(2000:2009, times = 50),
 is_treated = rep(sample(c(1, 0), size = 50, replace = TRUE, prob = c(0.5, 0.5)), each = 10),
  y = rnorm(500, mean = 0, sd = 1)  # Simulated outcome variable
)

# Run event study
event_study <- run_es(
  data       = df,
  outcome    = y,
  treatment  = is_treated,
  time       = year,
  timing     = 2005,
  lead_range = 5,              # Corresponds to years 2000-2004 (relative time: -5 to -1)
  lag_range  = 4,               # Corresponds to years 2006-2009 (relative time: 1 to 4)
  fe         = firm_id + year,
  cluster    = "state_id",
  baseline   = -1,
  interval   = 1
)

# Basic plot
plot_es(event_study)

# Use error bars instead of ribbon confidence intervals
plot_es(event_study, type = "errorbar")

# Adjust vertical reference line
plot_es(event_study, type = "errorbar", vline_val = -0.5)

# Customize axis breaks and title
library(ggplot2)
plot_es(event_study, type = "errorbar") +
  ggplot2::scale_x_continuous(breaks = seq(-5, 4, by = 1)) +
  ggplot2::ggtitle("Result of Event Study")
```

run_es *Run Event Study with Fixed Effects*

Description

This function performs an event study using fixed effects regression. It first generates lead and lag dummy variables relative to the treatment timing, scales the time intervals if specified, and then estimates the regression model.

Usage

```
run_es(
  data,
  outcome,
  treatment,
  time,
  timing,
  lead_range,
  lag_range,
  fe,
  cluster = NULL,
  baseline = -1,
  interval = 1
)
```

Arguments

`data` A dataframe containing the dataset.

`outcome` The name of the outcome variable (e.g., "y"). Should be unquoted.

`treatment` The name of the treatment variable (e.g., "treated"). Should be unquoted.

`time` The name of the time variable (e.g., "year"). Should be unquoted.

`timing` The time period when the treatment occurred. For example, if the treatment was implemented in 1995, set ‘timing = 1995‘.

`lead_range` Number of time periods to include before the treatment (negative leads). For example, ‘lead_range = 3‘ includes 3 periods before the treatment.

`lag_range` Number of time periods to include after the treatment (positive lags). For example, ‘lag_range = 2‘ includes 2 periods after the treatment.

`fe` A vector of fixed effects variables or an additive expression (e.g., firm_id + year). These variables account for unobserved heterogeneity.

`cluster` An optional variable for clustering standard errors. For example, ‘cluster = "state"‘.

`baseline` The relative time period to use as the baseline (default: -1). The corresponding dummy variable is excluded from the regression and treated as the reference group. For example, if ‘baseline = 0‘, the treatment year is the baseline.

`interval` The time interval between observations (default: 1). For example, use ‘interval = 5‘ for datasets where time steps are in 5-year intervals.

Details

This function is designed for panel data and supports time intervals other than 1 (e.g., 5-year intervals). It automatically scales the relative time variable using the ‘interval‘ parameter.

Steps: 1. Compute the relative time for each observation as ‘(time - timing) / interval‘. 2. Generate lead and lag dummy variables within the specified ranges (‘lead_range‘, ‘lag_range‘). 3. Construct and estimate the fixed effects regression model using ‘fixest::feols‘. 4. Format the regression results into a tidy dataframe.

If ‘interval > 1‘, ensure that the specified ‘lead_range‘ and ‘lag_range‘ correspond to the number of time intervals, not the absolute number of years.

Value

A tidy dataframe with regression results. This includes: - ‘term‘: The lead or lag variable names. - ‘estimate‘: Estimated coefficients. - ‘std.error‘: Standard errors. - ‘conf.high‘: Upper bound of the 95 - ‘conf.low‘: Lower bound of the 95 - ‘relative_time‘: Scaled relative time based on the specified ‘interval‘.

Examples

```
# Simulate panel data
df <- tibble::tibble(
  firm_id = rep(1:50, each = 10),  # 50 firms over 10 years
  state_id = rep(sample(1:10, size = 50, replace = TRUE), each = 10),
  year = rep(2000:2009, times = 50),
 is_treated = rep(sample(c(1, 0), size = 50, replace = TRUE, prob = c(0.5, 0.5)), each = 10),
  y = rnorm(500, mean = 0, sd = 1)  # Simulated outcome variable
)

# Run event study
event_study <- run_es(
  data       = df,
  outcome    = y,
  treatment  = is_treated,
  time       = year,
  timing     = 2005,
  lead_range = 5,             # Corresponds to years 2000-2004 (relative time: -5 to -1)
  lag_range  = 4,              # Corresponds to years 2006-2009 (relative time: 1 to 4)
  fe         = firm_id + year,
  cluster    = ”state_id”,
  baseline   = -1,
  interval   = 1
)
```

Index