Package 'fixes'

May 25, 2025

Type Package

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

Version 0.4.0

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, haven

VignetteBuilder knitr

URL https://github.com/yo5uke/fixes

BugReports https://github.com/yo5uke/fixes/issues

NeedsCompilation no

Author Yosuke Abe [aut, cre]

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

Repository CRAN

Date/Publication 2025-05-25 05:40:02 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

```
## Not run:
# 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")

## End(Not run)
```

---

run_es *Run Event Study with Fixed Effects*

Description

This function performs an event study using fixed effects regression based on a panel dataset. It generates dummy variables for each lead and lag period relative to the treatment timing, applies optional covariates and fixed effects, and estimates the model using ‘fixest::feols‘.

Usage

```
run_es(
  data,
  outcome,
  treatment,
  time,
  staggered = FALSE,
  timing,
  lead_range = NULL,
  lag_range = NULL,
  covariates = NULL,
  fe,
  cluster = NULL,
  weights = NULL,
  baseline = -1,
  interval = 1,
  time_transform = FALSE,
  unit = NULL
)
```

Arguments

data	A data frame containing the panel dataset.

outcome	The outcome variable, specified unquoted. You may use a raw variable name (e.g., 'y') or a transformation (e.g., 'log(y)').
treatment	The treatment indicator (unquoted). Can be binary numeric ('0/1') or logical ('TRUE/FALSE'). Typically equals 1 (or 'TRUE') in and after the treated period, 0 otherwise.
time	The time variable (unquoted). Used to calculate the relative timing.
staggered	Logical. If 'TRUE', allows treatment timing to vary across units. Requires 'timing' to be a column name. Default is 'FALSE'.
timing	The time period when the treatment occurs. If 'staggered = FALSE', must be a single numeric value (e.g., '2005'). If 'staggered = TRUE', must be an unquoted variable name representing the treatment timing for each unit. If 'time_transform = TRUE', specify 'timing' as an integer corresponding to the transformed time index within each unit (e.g., 5 for the fifth time point).
lead_range	Number of pre-treatment periods to include as leads (e.g., 5 = 'lead5', 'lead4', ..., 'lead1'). If 'NULL', the function will automatically determine the maximum possible lead across all units.
lag_range	Number of post-treatment periods to include as lags (e.g., 3 = 'lag0', 'lag1', 'lag2', 'lag3'). If 'NULL', the function will automatically determine the maximum possible lag across all units.
covariates	Optional covariates to include in the regression. Must be supplied as a one-sided formula (e.g., '~ x1 + x2').
fe	Fixed effects to control for unobserved heterogeneity. Must be a one-sided formula (e.g., '~ id + year').
cluster	Clustering specification for robust standard errors. Accepts either: - a character vector of column names (e.g., 'c("id", "year")'), or - a one-sided formula (e.g., '~ id + year' or '~ id^year'). Cluster variables are internally re-evaluated after filtering for the estimation window.
weights	Optional observation weights. Must be supplied as a one-sided formula (e.g., '~ popwt'). If 'NULL', unweighted regression is performed.
baseline	The relative time (e.g., '-1') to use as the reference period. The corresponding dummy variable will be excluded from the regression and added manually to the results with estimate 0. Must lie within the specified 'lead_range' and 'lag_range'. If not, an error will be thrown.
interval	The interval between time periods. Use '1' for annual data (default), '5' for 5-year intervals, etc.
time_transform	Logical. If TRUE, the time variable will be converted to a unit-level sequence (1, 2, 3, ...) based on its order within each unit. Useful for panel data with non-continuous time variables. Default is FALSE.
unit	The unit (individual) identifier for panel data. Required when 'time_transform = TRUE'. Must be an unquoted variable name (e.g., 'id').

Details

This function is intended for difference-in-differences or event study designs with panel data. It automatically: - Computes relative time: `(time - timing) / interval` - Generates dummy vari-

ables for specified leads and lags - Removes the baseline term from estimation and appends it back post-estimation - Uses `fixest::feols()` for fast and flexible estimation

Both fixed effects and clustering are fully supported. Observation weights can be specified using the ‘weights‘ argument.

If ‘time_transform = TRUE‘, the time variable is internally replaced with a unit-level sequence (e.g., 1, 2, 3, ...) based on its order within each unit (as specified by the ‘unit‘ argument). This is useful when the time variable is irregular (e.g., Date-type data or monthly data with gaps). Note that in this case, the ‘timing‘ argument must be specified based on the transformed index (e.g., 5 corresponds to the fifth time point in the sorted order within each unit).

Value

A tibble with the event study regression results, including: - ‘term‘: Name of the lead or lag dummy variable - ‘estimate‘: Coefficient estimate - ‘std.error‘: Standard error - ‘statistic‘: t-statistic - ‘p.value‘: p-value - ‘conf_high‘: Upper bound of 95 - ‘conf_low‘: Lower bound of 95 - ‘relative_time‘: Time scaled relative to the treatment - ‘is_baseline‘: Logical indicator for the baseline term (equals ‘TRUE‘ only for the excluded dummy)

Examples

```
## Not run:
# Assume df is a panel dataset with variables: id, year, y, treat, x1, x2, var1, var2, popwt

# Minimal example without covariates
run_es(
  data       = df,
  outcome    = y,
  treatment  = treat,
  time       = year,
  timing     = 2005,
  lead_range = 2,
  lag_range  = 2,
  fe         = ~ id + year,
  cluster    = ~ id,
  baseline   = -1
)

# Weighted regression
run_es(
  data       = df,
  outcome    = y,
  treatment  = treat,
  time       = year,
  timing     = 2005,
  lead_range = 2,
  lag_range  = 2,
  covariates = ~ x1 + x2,
  fe         = ~ id + year,
  cluster    = ~ id,
  weights    = ~ wt,
  baseline   = -1
```

```
)

# Example with staggered treatment timing
# Suppose `treat_time` indicates the year each unit was treated
run_es(
  data       = df,
  outcome    = y,
  treatment  = is_treated,
  time       = year,
  staggered  = TRUE,
  timing     = treat_time,  # a variable with treatment years per unit
  lead_range = 3,
  lag_range  = 4,
  fe         = ~ id + year,
  cluster    = ~ id,
  baseline   = -1
)

## End(Not run)
```

Index