Package: fect (via r-universe)

October 28, 2026

Type Package

Title Fixed Effects Counterfactual Estimators

Version 2.4.5

Date 2026-05-29

Author Yiqing Xu [aut, cre], Licheng Liu [aut], Ziyi Liu [aut], Ye Wang [aut], Tianzhu Qin [aut], Shiyun Hu [aut], Rivka Lipkovitz [aut]

Maintainer Yiqing Xu <yiqingxu@stanford.edu>

Description Provides tools for estimating causal effects in panel data using counterfactual methods, as well as other modern DID estimators. It is designed for causal panel analysis with binary treatments under the parallel trends assumption. The package supports scenarios where treatments can switch on and off and allows for limited carryover effects. It includes several imputation estimators, such as Gsynth (Xu 2017), linear factor models, and the matrix completion method. Detailed methodology is described in Liu, Wang, and Xu (2024) <doi:10.48550/arXiv.2107.00856> and Chiu et al. (2025) <doi:10.48550/arXiv.2309.15983>. Optionally integrates with the ``HonestDiDFEct'' package for sensitivity analyses compatible with imputation estimators. ``HonestDiDFEct'' is not on CRAN but can be obtained from <https://github.com/lzy318/HonestDiDFEct>.

URL https://yiqingxu.org/packages/fect/, https://github.com/xuyiqing/fect

BugReports https://github.com/xuyiqing/fect/issues

NeedsCompilation yes

License MIT + file LICENSE

Imports Rcpp (>= 0.12.3), ggplot2 (>= 2.1.0), GGally (>= 1.0.1), doParallel (>= 1.0.10), doFuture, foreach (>= 1.4.3), abind (>= 1.4-0), codetools, MASS, gridExtra, grid, fixest, doRNG, future, parallelly, mvtnorm, dplyr, future.apply, reshape2, rlang, scales, splines

Suggests panelView, testthat (>= 3.0.0), did, DIDmultiplegtDYN, ggrepel, HonestDiDFEct, withr

Depends R (>= 4.1.0)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.3

Encoding UTF-8

Config/testthat/edition 3

LazyData true

Repository https://cran.r-universe.dev

Date/Publication 2026-05-30 17:10:08 UTC

RemoteUrl https://github.com/cran/fect

RemoteRef HEAD

RemoteSha bbc71a2de6d26656af972114455e40224b7ec032

Contents

fect-package . . . 3
att.cumu . . . 4
did_wrapper . . . 5
effect . . . 7
esplot . . . 9
estimand . . . 15
fect . . . 17
fect-internal . . . 27
fect_iden . . . 28
fect_mspe . . . 28
fect_sens . . . 30
get.cohort . . . 32
gs2020 . . . 33
hh2019 . . . 33
imputed_outcomes . . . 33
interFE . . . 35
plot.fect . . . 37
print.fect . . . 50
print.interFE . . . 51
r.cv.rolling . . . 52
sim_base . . . 54
sim_gsynth . . . 55
sim_linear . . . 55
sim_region . . . 56
sim_trend . . . 56
simdata . . . 57
simgsynth . . . 58
turnout . . . 58

Index **59**

`fect-package` *Fixed Effects Counterfactual Estimators*

Description

The `fect` package implements counterfactual estimators for Time-Series Cross-Sectional (TSCS) data analysis, along with statistical tools to test their identification assumptions.

Details

The `fect` package provides tools for estimating treatment effects in TSCS datasets using a range of counterfactual estimators. These estimators first impute counterfactuals for treated observations by fitting an outcome model (fixed effects model, interactive fixed effects model, or matrix completion) to untreated observations. Then, the individual treatment effect for each treated observation is calculated as the difference between the observed and predicted counterfactual outcomes.

The package supports:

- Calculation of the Average Treatment Effect on the Treated (ATT) and period-specific ATTs.
- Placebo tests and equivalence tests to evaluate the validity of the identification assumptions.
- Robust estimation techniques for unbalanced panel datasets.

See `fect` for details.

Author(s)

Licheng Liu (<liulch@mit.edu>), Massachusetts Institute of Technology

Ye Wang (<yw1576@nyu.edu>), New York University

Yiqing Xu (<yiqingxu@stanford.edu>), Stanford University

Ziyi Liu (<zyliu2020@uchicago.edu>), University of Chicago

References

Athey, S., Bayati, M., Doudchenko, N., Imbens, G., and Khosravi, K. (2021). Matrix completion methods for causal panel data models. *Journal of the American Statistical Association, 116*(536), 1716-1730.

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica, 77*(4), 1229-1279.

Liu, L., Wang, Y., and Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science, 68*(1), 160-176.

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57-76.

att.cumu *Calculate Cumulative Treatment Effects*

Description

Calculate cumulative treatment effects based on the results of a fect object.

Usage

```
att.cumu(x, period = NULL, weighted = TRUE, alpha = 0.05, type = "on", plot = FALSE)
```

Arguments

x	A fect object.
period	A two-element numeric vector specifying the range of terms during which treatment effects are to be accumulated, e.g., period = c(-1, 1).
weighted	A logical flag specifying whether to calculate weighted cumulative treatment effects based on counts at each period. Default is TRUE.
alpha	A numerical value specifying the significance level. Default is 0.05.
type	A string that specifies the type of effect to calculate. Must be one of the following: "on" (switch-on treatment effect) or "off" (switch-off treatment effect). Default is "on".
plot	A logical flag indicating whether to plot cumulative effects. Default is FALSE.

Author(s)

Licheng Liu, Ye Wang, and Yiqing Xu

References

Athey, S., Bayati, M., Doudchenko, N., Imbens, G., and Khosravi, K. (2021). Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116(536), 1716-1730.

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica*, 77(4), 1229-1279.

Liu, L., Wang, Y., and Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science*, 68(1), 160-176.

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis*, 25(1), 57-76.

See Also

fect, plot.fect

did_wrapper *A Multi-Method Difference-in-Differences Estimator*

Description

Implements multiple difference-in-differences (DiD) estimators under a unified interface, supporting overall ATT and event-study estimates across staggered adoption settings.

Usage

```
did_wrapper(
  data,
  Y,
  D,
  X = NULL,
  index,
  method = c("twfe", "st", "iw", "cs_never", "cs_notyet", "didm"),
  se = c("default", "boot", "bootstrap", "jackknife"),
  nboots = 200,
  parallel = TRUE,
  core = NULL,
  time_to_treat_var = "Time_to_Treatment",
  treat_indicator = "treat",
  csdid.base_period = "universal",
  didm.effects = NA,
  didm.placebo = NA
)
```

Arguments

`data`	Input data frame.
`Y`	Outcome variable name (string).
`D`	Treatment indicator variable name (string).
`X`	Optional covariate vector for adjustment.
`index`	Character vector of unit and time variable names, e.g., `c("id", "time")`.
`method`	DiD method: `"twfe"`, `"st"`, `"iw"`, `"cs_never"`, `"cs_notyet"`, or `"didm"`.
`se`	Standard error method: `"default"`, `"boot"`, `"bootstrap"`, or `"jackknife"`.
`nboots`	Number of bootstrap replications (if applicable).
`parallel`	Logical; use parallel computation for bootstrapping.
`core`	Number of CPU cores to use if `parallel = TRUE`.
`time_to_treat_var`	Name of time-to-treatment variable; used internally.
`treat_indicator`	Name of treatment cohort indicator variable; used internally.

`csdid.base_period`
: Baseline period choice for Callaway–Sant'Anna estimators.

`didm.effects`
: Effects vector for `didm`; required if `method = "didm"`.

`didm.placebo`
: Placebo vector for `didm`; required if `method = "didm"`.

Details

This function:

1. Drops always-treated units.
2. Constructs event time and cohort variables.
3. Computes ATT using the specified DiD method.
4. Constructs event-study ATT curves.
5. Optionally estimates bootstrap or jackknife standard errors.

Supported methods include TWFE, stacked DiD, interaction-weighted DiD, Callaway–Sant'Anna estimators, placebo DiD, and DIDmultiplegt.

Value

A list of class `"did_wrapper"`:

`est.avg`
: Data frame with overall ATT, standard error, confidence interval, and p-value.

`est.att`
: Event-study ATT estimates by relative period, including standard errors and confidence intervals.

Author(s)

Rivka Lipkovitz

Examples

```
## Not run:
result_twfe <- did_wrapper(
  data   = df,
  Y      = "outcome",
  D      = "treat",
  index  = c("id", "time"),
  method = "twfe"
)
result_twfe$est.avg
result_twfe$est.att

## End(Not run)
```

`effect` *Calculate Cumulative or Sub-group Treatment Effects*

Description

Calculates cumulative or average treatment effects for specified units and time periods based on a fitted `fect` object. The function supports both cumulative effects over time and period-specific average treatment effects, with bootstrap-based uncertainty estimates.

Usage

```
effect(
  x,
  cumu = TRUE,
  id = NULL,
  period = NULL,
  plot = FALSE,
  count = TRUE,
  xlab = NULL,
  ylab = NULL,
  main = NULL
)
```

Arguments

`x`	A `fect` object containing treatment effect estimates and bootstrap results.
`cumu`	Logical. If `TRUE` (default), calculates cumulative treatment effects. If `FALSE`, calculates period-specific average treatment effects.
`id`	Character vector or NULL. Unit identifiers to include in the analysis. If `NULL` (default), all treated units are included.
`period`	Numeric vector of length 2 specifying the time window `c(start, end)` for effect calculation. If `NULL`, uses the maximum possible window based on the data.
`plot`	Logical. If `TRUE`, creates a visualization of the cumulative treatment effects with confidence intervals and a bar chart showing the number of treated units at each time point. Default is `FALSE`.
`count`	Logical. If `TRUE`, shows the count bars in the plot.
`xlab`	Character. X-axis label for the plot.
`ylab`	Character. Y-axis label for the plot.
`main`	Character. Main title for the plot.

Details

The function processes treatment effects in several steps:

1. Selects units based on the `id` parameter or includes all treated units if `id = NULL`.

2. Calculates relative time to treatment for each unit.

3. If `cumu = TRUE`, computes cumulative effects by summing average effects up to each period.

4. Performs bootstrap analysis to estimate uncertainty (standard errors, confidence intervals, and p-values).

The function supports different inference methods (bootstrap, jackknife, parametric) and adjusts calculations accordingly.

Note: The function requires bootstrap results in the input `fect` object (`keep.sims = TRUE` must be set when fitting the model).

Value

Returns a list containing:

`eff` Vector of point estimates for cumulative or average treatment effects.

`est.eff` Matrix containing the following columns:

- ATT: Point estimates
- S.E.: Standard errors
- CI.lower: Lower bound of confidence interval
- CI.upper: Upper bound of confidence interval
- p.value: Two-sided p-values

Warning

The function will stop with an error if:

- No bootstrap results are available in the input object
- The panel contains treatment reversals
- The specified ending period exceeds the maximum available period

Author(s)

Shiyun Hu, Licheng Liu, Ye Wang, and Yiqing Xu

References

Liu, L., Wang, Y., & Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science*, 68(1), 160-176.

See Also

`fect`, `plot.fect`

Examples

```
## Not run:
# Fit fect model with bootstrap
fit <- fect(Y ~ D + X, data = panel_data, keep.sims = TRUE)

# Calculate cumulative effects for all treated units
results <- effect(fit)

# Calculate period-specific effects for specific units
results_specific <- effect(fit,
                           cumu = FALSE,
                           id = c("unit1", "unit2"),
                           period = c(1, 4))

# View results
print(results$est.catt)

## End(Not run)
```

---

esplot *Event Study Visualization*

Description

Visualize dynamic treatment effects and create an event study plot. This function offers flexibility in displaying estimates, confidence intervals, and various annotations. It can handle data directly or from ‘did_wrapper‘ objects, calculate confidence intervals from standard errors if needed, and allows for connected (line/ribbon) or point-range style plots.

Usage

```
esplot(data, Period = NULL, Estimate = "ATT", SE = NULL,
       CI.lower = "CI.lower", CI.upper = "CI.upper", Count = NULL,
       proportion = 0.3, est.lwidth = NULL, est.pointsize = NULL,
       show.points = TRUE, fill.gap = TRUE, start0 = FALSE,
       only.pre = FALSE, only.post = FALSE, show.count = TRUE,
       stats = NULL, stats.labs = NULL, highlight.periods = NULL,
       highlight.colors = NULL, highlight.shapes = NULL,
       highlight.fill = FALSE,
       lcolor = NULL, lwidth = NULL,
       ltype = NULL, connected = FALSE, ci.outline = FALSE,
       main = NULL, xlim = NULL, ylim = NULL, xlab = NULL, ylab = NULL,
       gridOff = FALSE, stats.pos = NULL, theme.bw = TRUE,
       legacy.style = FALSE,
       cex.main = NULL, cex.axis = NULL, cex.lab = NULL,
       cex.text = NULL, axis.adjust = FALSE, color = "#000000",
       pre.color = NULL, post.color = NULL,
       count.color = "gray70", count.alpha = 0.4,
```

```
        count.outline.color = "grey69",
        xangle = NULL, yangle = NULL,
        xbreaks = NULL, ybreaks = NULL,
        legendOff = FALSE, cex.legend = NULL,
        pre.label = "Pre-treatment", post.label = "Post-treatment")
```

Arguments

data	The input data for the event study plot. Can be a data.frame or an object of class did_wrapper (in which case, the est.att component will be used).
Period	The name of the column in data representing the relative time period. If NULL (the default), the function will attempt to automatically identify the period column from common names (e.g., 'time', 'Time', 'period', 'Period', 'event.time', 'event_time', 'rel_time') or use numeric rownames if available.
Estimate	The name of the column in data containing the point estimates (e.g., Average Treatment Effect on the Treated). Default is "ATT".
SE	The name of the column in data containing the standard errors. If columns for CI.lower and CI.upper (as specified by their respective arguments, defaulting to "CI.lower" and "CI.upper") are not found in data, and an SE column is provided and found, 95% confidence intervals will be calculated using Estimate +/- 1.96 * SE. Default is NULL.
CI.lower	The name of the column in data for the lower bound of the confidence interval. Default is "CI.lower". If this column is not found, it may be calculated from Estimate and SE (if SE is provided and found in data).
CI.upper	The name of the column in data for the upper bound of the confidence interval. Default is "CI.upper". If this column is not found, it may be calculated from Estimate and SE (if SE is provided and found in data).
Count	Optional. The name of the column in data indicating a count measure (e.g., number of observations) for each time period. Used if show.count = TRUE or for xlim determination based on proportion. Default is NULL.
proportion	Numeric, between 0 and 1. If Count is specified and xlim is not, this proportion is used to determine the default x-axis limits. Periods where the count is below proportion * max(Count) might be excluded from the default view. Default is 0.3.
est.lwidth	Numeric. The line width for the estimate line (if connected = TRUE) or the main vertical line of the point-range (if connected = FALSE). Default is NULL, which resolves to 0.6 if connected = FALSE (the default for connected), 1.2 if connected = TRUE and show.points = FALSE (the default for show.points), and 0.7 if connected = TRUE and show.points = TRUE.
est.pointsize	Numeric. The size of the points. If connected = TRUE and show.points = TRUE, this is the size of points at integer time periods. If connected = FALSE, this controls the size of the central point in the point-range (via fatten aesthetic). Default is NULL, which resolves to 2 if connected = FALSE (the default for connected), 3 if connected = TRUE and show.points = FALSE (the default for show.points), and 1.2 if connected = TRUE and show.points = TRUE.

show.points Logical. If connected = TRUE, whether to display points at integer time periods on top of the line and ribbon. Default is TRUE.

fill.gap Logical. If connected = FALSE, whether to fill gaps in the sequence of time periods with an estimate and confidence interval of 0. This is useful when some integer time periods are missing from the input data. Default is TRUE.

start0 Logical. If TRUE, the vertical line separating pre- and post-treatment periods is drawn at x = -0.5, implying period 0 is the first post-treatment period. If FALSE (default), the line is at x = 0.5, implying period 0 is the last pre-treatment period.

only.pre Logical. If TRUE, the plot will only display pre-treatment periods. The vertical separator line will be omitted. Default is FALSE.

only.post Logical. If TRUE, the plot will only display post-treatment periods. The vertical separator line will be omitted. Default is FALSE.

show.count Logical. Whether to display a bar plot of the values from the Count column at the bottom of the main plot. Default is TRUE.

stats Optional. A numeric vector of statistics (e.g., p-values) to be printed on the plot.

stats.labs Optional. A character vector of labels corresponding to the stats values. Must be the same length as stats.

highlight.periods
Optional. A numeric vector of time periods to highlight with different colors. For connected = TRUE, these define intervals from period - 0.5 to period + 0.5. For connected = FALSE, individual points at these periods are highlighted.

highlight.colors
Optional. A character vector of colors corresponding to highlight.periods. If NULL and highlight.periods is provided, default rainbow colors are used. Must be the same length as highlight.periods.

highlight.shapes
Optional. An integer vector of point shapes corresponding to highlight.periods (e.g., 17 = triangle, 18 = diamond). Must be the same length as highlight.periods. Default is NULL.

highlight.fill Logical. If TRUE, draws a lightened-tone background rectangle behind each highlighted period (auto-derived from highlight.colors). If FALSE (the default), only the colored glyph at each highlight period is drawn. The colored glyph alone is usually sufficient signal that a period belongs to a test window, and the glyph-only look survives grayscale printing without surprise. Set TRUE for slide / talk figures where the rectangle helps a remote audience locate the window.

lcolor Optional. Color(s) for the reference lines. Can be a single color (applied to both horizontal y=0 line and vertical pre/post separator line) or a vector of two colors (first for horizontal, second for vertical). If NULL, defaults to "#aaaaaa" if theme.bw = TRUE, otherwise "white". Default is NULL.

lwidth Optional. Line width(s) for the reference lines. Can be a single width or a vector of two widths (similar to lcolor). If NULL, defaults to 1.5 if theme.bw = TRUE, otherwise 2. Default is NULL.

ltype Optional. Linetype(s) for the reference lines. Can be a single linetype (applied to both horizontal y=0 line and vertical pre/post separator line) or a vector of

two linetypes (first for horizontal, second for vertical). Default is NULL, which resolves to c("solid", "dashed") under the modern recipe (theme.bw = TRUE with legacy.style = FALSE) and to c("solid", "solid") otherwise.

connected — Logical. If TRUE, estimates and confidence intervals are plotted as a connected line with a ribbon. This involves interpolating values between observed time points (at 0.5 steps by default) to create a smoother appearance. If FALSE (default), geom_pointrange is used, showing individual estimates and their CIs as points with ranges for each observed time period.

ci.outline — Logical. If connected = TRUE, whether to draw an outline around the confidence interval ribbon. The outline color is a slightly darker version of the fill color. Default is FALSE.

main — Optional. The main title for the plot. If NULL (default), a default title "Estimated Dynamic Treatment Effects" is used. If an empty string "" is provided, no title is displayed.

xlim — Optional. A numeric vector of length 2 specifying the x-axis limits (c(min, max)). If NULL (default), limits are determined automatically based on the data range, potentially filtered by proportion if Count is used.

ylim — Optional. A numeric vector of length 2 specifying the y-axis limits (c(min, max)). If NULL (default), limits are determined automatically to encompass all estimates and confidence intervals, with potential expansion if show.count = TRUE.

xlab — Optional. The label for the x-axis. If NULL (default), "Time Relative to Treatment" is used. If an empty string "" is provided, no label is displayed.

ylab — Optional. The label for the y-axis. If NULL (default), "Effect on Y" is used. If an empty string "" is provided, no label is displayed.

gridOff — Logical. Whether to turn off major and minor grid lines. Default is TRUE.

stats.pos — Optional. A numeric vector of length 2 (c(x, y)) specifying the coordinates for the top-left position of the stats text block. If NULL (default), the position is automatically determined.

theme.bw — Logical. Whether to use ggplot2::theme_bw(). Default is TRUE.

legacy.style — Logical. If TRUE, reproduces the pre-2.3.1 visual defaults (bold centered title, larger axis text, solid vline, blue placebo triangles, no peach highlight rectangle) regardless of the value of theme.bw. If FALSE (the default) and theme.bw = TRUE, the modern v2.3.1 visual recipe is used. Pass legacy.style = TRUE to byte-identical-reproduce figures rendered under earlier fect versions.

cex.main — Optional. Numeric scaling factor for the plot title font size. The base size used by ggplot is 16. Default is NULL (uses base size 16).

cex.axis — Optional. Numeric scaling factor for the axis tick mark labels font size. The base size used by ggplot is 15. Default is NULL (uses base size 15).

cex.lab — Optional. Numeric scaling factor for the axis title (x and y labels) font size. The base size used by ggplot is 15. Default is NULL (uses base size 15).

cex.text — Optional. Numeric scaling factor for annotated text elements (e.g., stats text, count label). The base size used by ggplot for annotation is 5. Default is NULL (uses base size 5).

axis.adjust	Logical. If TRUE, x-axis tick labels are rotated 45 degrees for better readability with many labels. Default is FALSE.
color	Character. The primary color for plotting estimates, points, lines, and confidence interval fills/lines (unless overridden by highlight.colors for specific periods). Default is "#000000" (black).
pre.color	Optional. Character. Color for pre-treatment period estimates. If NULL (default), uses the value of color.
post.color	Optional. Character. Color for post-treatment period estimates. If NULL (default), uses the value of color.
count.color	Character. The fill color for the bars if show.count = TRUE. Default is "gray70".
count.alpha	Numeric. Alpha transparency for the count bars if show.count = TRUE. Default is 0.4.
count.outline.color	Character. The color for the outline of count bars if show.count = TRUE. Default is "grey69".
xangle	Optional. Numeric. Rotation angle (in degrees) for x-axis tick labels. Default is NULL (no rotation).
yangle	Optional. Numeric. Rotation angle (in degrees) for y-axis tick labels. Default is NULL (no rotation).
xbreaks	Optional. A numeric vector of break points for the x-axis. If NULL (default), breaks are determined automatically.
ybreaks	Optional. A numeric vector of break points for the y-axis. If NULL (default), breaks are determined automatically.
legendOff	Logical. If TRUE, the legend is suppressed. Default is FALSE.
cex.legend	Optional. Numeric scaling factor for the legend text size. Default is NULL (uses ggplot2 defaults).
pre.label	Character. Label for the pre-treatment period in the legend. Default is "Pre-treatment".
post.label	Character. Label for the post-treatment period in the legend. Default is "Post-treatment".

Value

p	A ggplot object representing the event study plot.

Author(s)

Licheng Liu, Yiqing Xu, Ziyi Liu, Zhongyu Yin, Rivka Lipkovitz

Examples

```
# Basic example with simulated data
set.seed(123)
event_data <- data.frame(
  time = -5:5,
  ATT = cumsum(rnorm(11, 0, 0.2)) + c(rep(0,5), 0, 0.5, 1, 1.2, 1.5, 1.3),
  SE = runif(11, 0.1, 0.3)
)
```

```
event_data$CI.lower <- event_data$ATT - 1.96 * event_data$SE
event_data$CI.upper <- event_data$ATT + 1.96 * event_data$SE
event_data$count <- sample(50:150, 11, replace = TRUE)
event_data$count[event_data$time == -5 | event_data$time == 5] <- 20 # for proportion demo

# Default plot (point-range)
esplot(event_data, Period = "time", Estimate = "ATT",
       CI.lower = "CI.lower", CI.upper = "CI.upper")

# # Connected plot with ribbon
# esplot(event_data, Period = "time", Estimate = "ATT",
#        CI.lower = "CI.lower", CI.upper = "CI.upper",
#        connected = TRUE, show.points = TRUE)

# # Connected plot using SE for CI calculation
# event_data_no_ci <- event_data[, c("time", "ATT", "SE", "count")]
# esplot(event_data_no_ci, Period = "time", Estimate = "ATT", SE = "SE",
#        connected = TRUE, ci.outline = TRUE, color = "blue")

# # Show count bars and stats
# esplot(event_data, Period = "time", Estimate = "ATT",
#        CI.lower = "CI.lower", CI.upper = "CI.upper", Count = "count",
#      show.count = TRUE, stats = c(0.03, 0.12), stats.labs = c("P-val Pre", "P-val Post"),
#        main = "Event Study with Counts and Stats", proportion = 0.2)

# # Highlight specific periods (connected)
# esplot(event_data, Period = "time", Estimate = "ATT", SE = "SE",
#        connected = TRUE, highlight.periods = c(-1, 2),
#        highlight.colors = c("orange", "green"),
#        main = "Highlighted Periods (Connected)")

# # Highlight specific periods (point-range)
# esplot(event_data, Period = "time", Estimate = "ATT", SE = "SE",
#        connected = FALSE, highlight.periods = c(-1, 2),
#        highlight.colors = c("orange", "green"),
#        main = "Highlighted Periods (Point-Range)")

# # Only post-treatment period, custom labels
# esplot(event_data, Period = "time", Estimate = "ATT", SE = "SE",
#        only.post = TRUE, xlab = "Years Post-Intervention", ylab = "Impact Metric",
#        start0 = TRUE, color = "darkred", est.lwidth = 1.5)

# Using did_wrapper object (conceptual example, requires `did` package and setup)
# if (requireNamespace("did", quietly = TRUE)) {
#   # Assume `did_out` is an output from `did::att_gt` or similar
#   # and `did_wrapper_obj` is created, e.g.,
#   # did_wrapper_obj <- list(est.att = event_data) # Simplified for example
#   # class(did_wrapper_obj) <- "did_wrapper"
#   # esplot(did_wrapper_obj) # Would use defaults: Period="time", Estimate="ATT"
# }
```

estimand *Post-hoc Estimand Dispatcher*

Description

Computes a post-hoc estimand from a `fect` fit, with bootstrap or jackknife uncertainty. The `type` argument selects from a closed enum of mathematically-defined estimands; the `by` argument selects the grouping axis. The accessor `imputed_outcomes` is the underlying long-form data source for any estimand the dispatcher does not ship natively.

Usage

```
estimand(
  fit,
  type   = c("att", "att.cumu", "aptt", "log.att"),
  by     = c("event.time", "cohort", "calendar.time", "overall"),
  test        = c("none", "placebo", "carryover"),
  cells       = NULL,
  weights     = NULL,
  window      = NULL,
  direction   = c("on", "off"),
  vartype     = c("bootstrap", "jackknife", "parametric", "none"),
  conf.level  = 0.95,
  ci.method   = NULL
)
```

Arguments

`fit`	A `fect` object.
`type`	Estimand type. `"att"` returns per-cell mean treatment effect aggregated per group; `"att.cumu"` returns cumulative ATT through each event time (replaces `effect`); `"aptt"` returns the average proportional treatment effect on the treated (Chen and Roth 2024 QJE); `"log.att"` returns the mean log-scale treatment effect. `"aptt"` and `"log.att"` require `keep.sims = TRUE` at fit time for SE/CI.
`by`	Grouping axis. One of `"event.time"` (default), `"cohort"`, `"calendar.time"`, `"overall"`, or any column name resolvable in the fit's panel data.
`test`	Selects which subset of cells to aggregate over. `"none"` (default) uses the standard treated post-treatment cells. `"placebo"` restricts aggregation to the pre-treatment cells in `fit$placebo.period` that were masked-and-imputed during the placebo fit; e.g.\ `estimand(fit, "aptt", "event.time", test = "placebo")` returns a per-event-time placebo APTT series. Requires `placeboTest = TRUE` at fit time and forces `direction = "on"`. `"carryover"` is the analogous extension for early post-reversal cells in `fit$carryover.period`; requires `carryoverTest = TRUE` at fit time, a panel with reversals, and forces `direction = "off"`. Both are incompatible with `type = "att.cumu"`.

`cells` Optional filter on which treated cells to include. Accepts `NULL` (default), a logical vector, or a one-sided formula. See `imputed_outcomes`.

`weights` Aggregation-weight handling. `NULL` (default) uses `fit$W.agg` if the fit was built with `W` or `W.agg`; otherwise uniform.

`window` Optional event-time window `c(L, R)`; convenience sugar for `cells = ~ event.time >= L & event.time <= R`.

`direction` Either `"on"` (default) or `"off"`; see `imputed_outcomes`.

`vartype` `"bootstrap"` (default), `"jackknife"`, `"parametric"`, or `"none"`. Selects which variance method to source replicates from. The output `vartype` column reports the method actually used at fit time (read from `fit$vartype`); this argument is informational and does not re-aggregate replicates. Wild-bootstrap fits (`vartype = "wild"` at fit time) are consumed transparently and reported as `"wild"` in the output column.

`conf.level` Two-sided confidence level. Defaults to 0.95.

`ci.method` One of `"basic"`, `"percentile"`, `"bc"` (bias-corrected percentile), `"bca"` (bias-corrected accelerated; Efron 1987 in full), or `"normal"` (Wald: $\hat{\theta} \pm z \cdot SE$). Default is `NULL`, which triggers a per-type default: `"att"` -> `"normal"` (matches what `fit$est.att` already uses internally), `"att.cumu"` -> `"percentile"` (matches what `att.cumu()` does internally), `"aptt"` -> `"bca"` and `"log.att"` -> `"bca"` (ratio / log estimators benefit from joint bias and acceleration corrections when the bootstrap distribution is skewed; `"bc"` can degenerate at the boundary when the point estimate falls outside the bootstrap support). Pass an explicit value to override. `"bca"` requires bootstrap- or wild-bootstrap-sourced replicates.

Value

A data frame with columns `<by_key>`, `estimate`, `se`, `ci.lo`, `ci.hi`, `n_cells`, and `vartype`. Always tidy regardless of `type` or `by`.

Numerical equality with existing slots

`estimand(fit, "att", "event.time")` returns `estimate`, `se`, `ci.lo`, `ci.hi` byte-identical to columns `ATT`, `S.E.`, `CI.lower`, `CI.upper` of `fit$est.att`, when default arguments are used. This invariant is asserted by package tests.

References

Chen, Jiafeng, and Jonathan Roth. 2024. "Logs with Zeros? Some Problems and Solutions." *Quarterly Journal of Economics* 139 (2): 891–936.

See Also

`imputed_outcomes` for the underlying long-form accessor; `fect` for the fitting interface.

Examples

```
## Not run:
library(fect)
fit <- fect(Y ~ D, data = simdata, index = c("id", "time"),
            method = "fe", force = "two-way",
            se = TRUE, nboots = 200, parallel = FALSE)

## Default: per-event-time ATT (matches fit$est.att numerically).
est <- estimand(fit, "att", "event.time")
head(est)

## End(Not run)
```

`fect` *Fixed Effects Counterfactual Estimators*

Description

Implements counterfactual estimators in TSCS data analysis and statistical tools to test their identification assumptions.

Usage

```
fect(formula = NULL, data, Y, D, X = NULL,
             W = NULL, W.est = NULL, W.agg = NULL,
             group = NULL,
             na.rm = FALSE,
             index, force = "two-way",
             time.component.from = "notyettreated", em = TRUE,
             r = 0, lambda = NULL, nlambda = 10,
             CV = NULL, k = 20, cv.prop = 0.1, cv.method = "rolling",
             cv.nobs = 3, cv.donut = 1, cv.buffer = 1, criterion = "mspe",
             binary = FALSE, QR = FALSE,
             method = "fe",  se = FALSE, vartype = "bootstrap",
             para.error = "auto", cl = NULL,
             ci.method = "normal", quantile.CI = NULL,
             nboots = 200, alpha = 0.05,
             parallel = TRUE, cores = NULL, tol = 1e-5,
             max.iteration = 5000, seed = NULL,
             min.T0 = NULL, max.missing = NULL,
             proportion = 0.3, pre.periods = NULL,
             f.threshold = 0.5, tost.threshold = NULL,
             knots = NULL, degree = 2,
             group.fe = NULL,
             cfe = NULL,
             Z = NULL, gamma = NULL, Q = NULL, kappa = NULL,
             Q.type = NULL,
```

```
 Q.bspline.degree = NULL,
 Z.param = NULL, Q.param = NULL,
 balance.period = NULL, fill.missing = FALSE,
 placeboTest = FALSE, placebo.period = NULL,
carryoverTest = FALSE, carryover.period = NULL, carryover.rm = NULL,
 loo = FALSE, permute = FALSE, m = 2,
 normalize = FALSE, keep.sims = FALSE,
 cm = FALSE,
 loading.bound = "none", gamma.loading = NULL,
 gamma.loading.grid = NULL,
 cv.rule = "1se")
```

Arguments

formula	an object of class "formula": a symbolic description of the model to be fitted, e.g, Y~D+X1+X2
data	a data frame, can be a balanced or unbalanced panel data.
Y	the outcome indicator.
D	the treatment indicator. The treatment should be binary (0 and 1).
X	time-varying covariates. Covariates that have perfect collinearity with specified fixed effects are dropped automatically.
W	a string giving the column name of a weight variable. Convenience default that populates both W.est and W.agg when those are left NULL. Suitable for survey or sample weights, where the same column applies to both the outcome-model fit and the across-treated-obs aggregation.
W.est	a string giving the column name of a weight variable that enters the outcome-model fit (the weighted least squares applied inside the IFE / MC / CFE solver). When NULL, falls back to W. Use this (with W.agg = NULL) when the weight reflects fit-side considerations and the estimand is the unweighted average ATT across treated cells.
W.agg	a string giving the column name of a weight variable that enters the across-treated-obs aggregation (att.on, est.avg, est.att). When NULL, falls back to W. Use this (with W.est = NULL) when the user's estimand differs from "ATT for the treated units in the analysis sample" — common cases include calibration weights to a target population or post-stratification weights that should adjust the summary but not the model fit. A clean in-package solution for inverse-probability weights for confounding adjustment is under development for fect 3.0 (a cross-fit doubly-robust path); W.agg is not a substitute for that work and does not deliver the doubly-robust properties an IPW user expects. In v2.3.1, W.est and W.agg (when both supplied) must point to the same column; truly distinct columns for fit vs. aggregation (e.g. combined survey x IPW designs) are scheduled for v2.4.0.
group	the group indicator. If specified, the group-wise ATT will be estimated.
na.rm	a logical flag indicating whether to list-wise delete missing observations. Default to FALSE. If na.rm = FALSE, it allows the situation when Y is missing but

D is not missing for some observations. If na.rm = TRUE, it will list-wise delete observations whose Y, D, or X is missing.

index — a character vector specifying the unit (first element) and time (second element) indicators. For most methods, must be of length 2. For method = "cfe", additional elements (third, fourth, etc.) specify extra fixed-effect grouping variables. Every observation should be uniquely defined by the pair of the unit and time indicator.

force — a string indicating whether unit or time or both fixed effects will be imposed. Must be one of the following, "none", "unit", "time", or "two-way". The default is "two-way".

time.component.from — Controls which units provide the time-varying model components (time fixed effects, factor structure, temporal dynamics). Options are "notyettreated" (default) — all units contribute during their pre-treatment periods, or "nevertreated" — only never-treated units estimate the time components, which are then projected onto treated units.

em — a logical flag indicating whether to use the EM algorithm for missing data in the estimation sample. Default is TRUE. Setting em = FALSE requires a complete estimation sample and is only compatible with time.component.from = "nevertreated".

r — an integer specifying the number of factors. If CV = TRUE, the cross validation procedure will select the optimal number of factors from r to 5.

lambda — a single or sequence of positive numbers specifying the hyper-parameter sequence for matrix completion method. If lambda is a sequence and CV = 1, cross-validation will be performed.

nlambda — an integer specifying the length of hyper-parameter sequence for matrix completion method. Default is nlambda = 10.

CV — a logical flag indicating whether cross-validation will be performed to select the optimal number of factors or hyper-parameter in matrix completion algorithm. If r is not specified, the procedure will search through r = 0 to 5.

k — an integer specifying number of cross-validation rounds. Default is k = 20.

cv.prop — a numerical value specifying the proportion of testing set compared to sample size during the cross-validation procedure.

cv.method — a string specifying the cross-validation masking strategy. One of "rolling" (default; standard time-series rolling-window CV), "block" (random scattered anchors with contiguous-block masking), or "loo" (leave-one-out, available for fect_nevertreated). The legacy aliases "all_units" (= "block") and "treated_units" (block masking restricted to treated pre-treatment cells) are still accepted but emit a deprecation message; both will be replaced by the unified (cv.method, cv.units) API in v2.4.0.

cv.nobs — an integer specifying the length of continuous observations within a unit in the testing set. Default is cv.nobs = 3.

cv.donut — an integer specifying the length of removed observations at the head and tail of the continuous observations specified by cv.nobs. Used by block CV (cv.method = "block", or the legacy aliases "all_units"/"treated_units"). Default is 1 (matches cv.buffer for rolling CV).

`cv.buffer` an integer specifying the length of past-side buffer cells masked from training (but not scored) immediately before each rolling-window holdout. Used only by `cv.method = "rolling"`; the future side is dropped from training by construction. Analogous to `cv.donut` for block CV but applied only on the past side. Default is 1.

`criterion` criterion used for model selection. Default is "mspe". `"mspe"` for the mean squared prediction error, `"gmspe"` for the geometric-mean squared prediction errors, `"moment"` for period-weighted residuals in test sets, `"pc"` for an information criterion method.

`binary` This version doesn't support this option.

`QR` This version doesn't support this option.

`method` a string specifying which imputation algorithm will be used. `"fe"`, `"ife"`, `"mc"`, `"gsynth"`, or `"cfe"`. Default is `"fe"`.

`se` a logical flag indicating whether uncertainty estimates will be produced.

`vartype` a string specifying the type of variance estimator, e.g. `"bootstrap"`. Three values are supported: `"bootstrap"` (nonparametric cluster-bootstrap; the safe default), `"jackknife"` (leave-one-unit-out), and `"parametric"` (two-stage pseudo-treated parametric bootstrap). The `"parametric"` option is restricted to the gsynth-style regime: it requires `time.component.from = "nevertreated"`, no treatment reversal, and `method` not in `c("mc", "both")`. These three conditions correspond to Gates A, B, and C in the three-gate defense system (see `ARCHITECTURE.md`). For all other settings, `vartype = "bootstrap"` is recommended.

`para.error` a string specifying the residual-error model used by the parametric bootstrap path; sub-option of `vartype = "parametric"` (silently ignored otherwise). One of `"auto"`, `"ar"`, `"empirical"`, or `"wild"`. Default `"auto"` resolves at fit time to `"empirical"` on a fully-observed panel and `"ar"` on a panel with missing cells; the resolved label is stored on `fit$para.error`. `"ar"` estimates an AR(1) error process from control residuals (works on any panel shape); `"empirical"` resamples residuals i.i.d. from the main-fit pool (requires fully- observed panel); `"wild"` applies unit-level Rademacher sign-flips over the empirical residual pool (requires fully-observed panel).

`cl` a string specifying the cluster column for cluster bootstrapping. When `group.fe` is set to a single column and `cl` is unset, `cl` auto-defaults to `group.fe[1]` — the natural choice when treatment varies at the group level (Bertrand, Duflo & Mullainathan 2004). To override — for example, to cluster at the unit level when group-level FE is absorbed — pass `cl = index[1]` (e.g., `cl = "id"`). Note that `cl = NULL` does NOT disable clustering: the case bootstrap always resamples units (which is unit-level clustering); the auto-default only changes the resample unit to a coarser level. `cl = FALSE` is rejected with a guiding error.

`ci.method` a string controlling how the bootstrap distribution becomes the CIs reported in fect's `est.*` slots. Two values: `"normal"` (default; Wald CI: $\hat{\theta} \pm z \cdot SE$) or `"basic"` (reflected pivot: $[2\hat{\theta} - q_{1-\alpha/2},\ 2\hat{\theta} - q_{\alpha/2}]$, the literature-standard "percentile" CI per Davison & Hinkley (1997, §5.2.1) and what `boot::boot.ci(type = "basic")` returns). For `"percentile"`, `"bc"`, or `"bca"` CIs (typically wanted on alternative estimands like `aptt` and `log.att`), call `estimand` after fitting.

quantile.CI — deprecated as of v2.4.2. Use ci.method instead. quantile.CI = FALSE maps to ci.method = "normal"; quantile.CI = TRUE maps to ci.method = "basic". Both mappings still work but emit a one-time deprecation warning.

nboots — an integer specifying the number of bootstrap runs. Ignored if se=FALSE. Default 200, sufficient for the standard error and the normal-CI structure that fect ships in est.att/att.avg. For tail-quantile-based CI methods accessed via estimand() (ci.method "basic", "percentile", "bc", "bca"), bump to nboots = 1000 or higher (Efron 1987 Section 3; DiCiccio & Efron 1996 Section 4); estimand() emits a warning when called on under-replicated fits.

alpha — the significance level for hypothesis tests and confidence intervals. Default 0.05.

parallel — controls which operations run in parallel. Accepted values:

TRUE Enable parallel computing for both CV and bootstrap (default in fect()).

FALSE Disable all parallel computing.

"cv" Enable parallel CV only; bootstrap runs serially.

"boot" Enable parallel bootstrap only; CV runs serially.

c("cv","boot") Explicit form of TRUE: parallel for both.

When parallel = TRUE, auto-enable thresholds apply: CV parallelism engages only when Nco * TT exceeds the per-method threshold (ife = 20000, mc = 20000, cfe = 60000). Explicit "cv" overrides the threshold. Nested parallelism (calling fect() from within a future_lapply or foreach %dopar% block) should use parallel = FALSE to avoid deadlock. When using parallel with method = "mc", parallel CV computes all candidate lambda values without early stopping (the serial path uses a break_check short-circuit to skip lambdas with diminishing MSPE returns). This guarantees numerical identity between serial and parallel results but may compute a few extra lambda values compared to the serial path. Use parallel = FALSE to preserve the short-circuit behavior.

cores — an integer indicating the number of cores for parallel computing.

tol — a positive number indicating the relative tolerance for the EM update check (||fit_new - fit_old||_F / ||fit_old||_F < tol). Default tightened from 1e-3 to 1e-5 in v2.4.3 because the older default produced under-converged IFE/CFE point estimates that shifted up to 40% relative to the converged value (inference was preserved, but the reported numbers were stopping-point-dependent).

max.iteration — the maximal number of EM iterations. Default raised from 1000 to 5000 in v2.4.3 to accommodate the tighter tol; canonical IFE/CFE fits converge in 700-2000 iters at tol = 1e-5. A warning() is emitted when the EM hits this cap without satisfying the tol gate (under-convergence diagnostic).

seed — an integer seed for random number generation.

min.T0 — an integer specifying the minimum number of pre-treatment periods for each treated unit.

max.missing — an integer specifying the maximum number of missing observations allowed per unit.

proportion — a numeric value specifying which pre-treatment periods are used for goodness-of-fit tests.

pre.periods a vector specifying the range of pre-treatment periods used for the goodness-of-fit test.

f.threshold a numeric threshold for an F-test in equivalence testing. Default 0.5.

tost.threshold a numeric threshold for two-one-sided t-tests.

knots a numeric vector specifying knots (currently unused; reserved for future use).

degree an integer specifying the degree (currently unused; reserved for future use).

group.fe a character vector of column names naming additional simple additive fixed-effect groupings to absorb (e.g., group.fe = "state" when rows are counties and treatment varies at state level). Each entry must be a column in data. Each entry must be *nested in* index[1] — i.e., constant within each level of the unit identifier — otherwise an error is raised. When group.fe is set, method = "fe" is silently routed to method = "cfe" (FE is a subset of CFE, identical result); method = "ife", "mc", "both", and "gsynth" hard-error (use method = "cfe" with r > 0 for free latent factors with group-level FE). When group.fe has length 1 and cl is unset, cl auto-defaults to group.fe[1]. To cluster at the unit level instead, pass cl = index[1] (e.g., cl = "id"); cl = NULL does NOT change behavior (the case bootstrap always resamples units regardless of cl), and cl = FALSE is rejected with a guiding error.

cfe a vector of lists specifying interactive fixed effects for method="cfe".

Z a vector specifying the time-invariant covariates for the Z matrix.

gamma a vector specifying the time-varying covariates for the gamma matrix.

Q a vector specifying the time-varying covariates for the Q matrix.

kappa a vector specifying the time-invariant covariates for the kappa matrix.

Q.type a vector specifying the type of Q matrix.

Q.bspline.degree an integer specifying the degree used when Q.type includes "bspline" in method="cfe". If NULL, a default degree is chosen based on the number of distinct time values.

Z.param a list specifying the parameters for the Z matrix.

Q.param a list specifying the parameters for the Q matrix.

balance.period a length-2 vector specifying a time range for a balanced sample.

fill.missing a logical flag indicating whether to allow missing observations in a balanced sample.

placeboTest a logical flag indicating whether to perform a placebo test.

placebo.period an integer or 2-element numeric vector specifying pseudo-treatment periods.

carryoverTest a logical flag for carryover tests.

carryover.period an integer or 2-element numeric vector specifying pseudo-carryover periods.

carryover.rm an integer specifying the range of post-treatment periods to treat as carryover.

loo a logical flag for leave-one-period-out goodness-of-fit tests.

permute a logical flag indicating whether to run a permutation test.

m an integer specifying the block length for permutation tests. Default 2.

normalize — a logical flag indicating whether to scale outcome and covariates.

keep.sims — a logical flag indicating whether to save unit-time level bootstrap effects. Default keep.sims = FALSE. If se = FALSE, this argument is ignored.

cm — a logical flag indicating whether to enable causal moderation analysis. When TRUE, the estimator decomposes the treatment effect into effect modification and causal moderation components. Currently available for method = "fe" and method = "ife". Default is FALSE.

loading.bound — a string controlling whether treated-unit factor loadings are bounded inside the convex hull of control loadings. "none" (default) reproduces standard GSC behavior. "simplex" constrains each treated unit's loading to be a non-negative convex combination of control loadings via an entropy-regularized simplex projection, ensuring the imputed counterfactual lies pointwise in the convex hull of factor-implied control outcomes. Currently applies only to method = "ife" or method = "gsynth" (equivalent forms), and requires time.component.from = "nevertreated".

gamma.loading — scalar regularization strength for the "simplex" projection. NULL (default) triggers 5-fold cross-validation over gamma.loading.grid. A numeric value is used directly. Ignored when loading.bound = "none".

gamma.loading.grid — a numeric vector of candidate gamma.loading values for cross-validation. NULL (default) uses 10^seq(-2, 2, length.out = 9). Ignored when loading.bound = "none" or when gamma.loading is supplied.

cv.rule — a string selecting the cross-validation rule for choosing the number of factors r (or matrix-completion penalty lambda). One of:

"1se" **(default)** The 1-SE rule (Breiman, Friedman, Olshen and Stone 1984; Hastie, Tibshirani and Friedman 2009, Section 7.10): pick the smallest r whose mean CV criterion is within one fold-SE of the minimum-CV-error r. Biases toward parsimony in a fold-aware way — when CV is precise, it allows larger r; when CV is noisy, it gravitates to simpler models.

"min" Pick the r that minimizes the mean CV criterion (no tolerance).

"1pct" Legacy pre-2.3.0 heuristic: pick the smallest r within 1% relative tolerance of the best mean CV criterion. Use this for byte-identical reproducibility of pre-2.3.0 fits.

Ignored when CV = FALSE.

Details

fect implements counterfactual estimators for TSCS data. It first imputes counterfactuals by fitting an outcome model using untreated observations, then estimates the individual treatment effect as the difference between observed and predicted outcomes. Finally, it computes average treatment effects on the treated (ATT) and period-specific ATTs. Placebo and equivalence tests help evaluate identification assumptions.

Value

Y.dat — T-by-N matrix of the outcome variable.

D.dat	T-by-N matrix of the treatment variable.
I.dat	T-by-N matrix of observation indicators (observed/missing).
Y	name of the outcome variable.
D	name of the treatment variable.
X	name of any time-varying covariates.
W	name of the weight variable.
index	name of the unit and time indicators.
force	specified fixed effects option.
T	number of time periods.
N	number of units.
p	number of time-varying observables.
r.cv	number of factors (selected by cross-validation if needed).
lambda.cv	optimal hyper-parameter for matrix completion, if applicable.
beta	coefficients for any covariates in an interactive fixed effects model.
sigma2	mean squared error.
IC	information criterion.
est	results of the fitted model.
MSPE	mean squared prediction error from cross-validation.
CV.out	results of the cross-validation procedure.
niter	number of iterations.
factor	estimated time-varying factors.
lambda	estimated loadings.
lambda.tr	estimated loadings for treated units.
lambda.co	estimated loadings for control units.
mu	estimated grand mean.
xi	estimated time fixed effects.
alpha	estimated unit fixed effects.
alpha.tr	estimated unit fixed effects for treated units.
alpha.co	estimated unit fixed effects for control units.
validX	logical indicating if valid covariates exist.
validF	logical indicating if factors exist.
id	vector of unit IDs.
rawtime	vector of time periods.
obs.missing	matrix indicating missingness patterns.
Y.ct	T-by-N matrix of predicted outcomes under no treatment.
eff	T-by-N matrix of estimated individual treatment effects.
res	residuals for observed values.

eff.pre	effects for treated units in pre-treatment periods.
eff.pre.equiv	pre-treatment effects under baseline (two-way FE) model.
pre.sd	by-period residual standard deviations for pre-treatment ATT.
att.avg	overall average treatment effect on the treated.
att.avg.W	weighted ATT.
att.avg.unit	by-unit average treatment effect on the treated.
time	time index for switch-on treatment effect.
count	count of observations for each switch-on effect time.
att	switch-on treatment effect.
att.on.W	weighted switch-on effect.
time.off	time index for switch-off treatment effect.
att.off	switch-off treatment effect.
att.off.W	weighted switch-off effect.
count.off	count for each switch-off period.
att.placebo	ATT for placebo periods.
att.carryover	ATT for carryover periods.
eff.calendar	ATT by calendar time.
eff.calendar.fit	loess-fitted ATT by calendar time.
N.calandar	number of treated observations each calendar period.
balance.avg.att	ATT for balanced sample.
balance.att	switch-on ATT for balanced sample.
balance.time	time index for balanced sample.
balance.count	count for each time in balanced sample.
balance.att.placebo	ATT for placebo period in balanced sample.
group.att	ATT for different groups.
group.output	list of switch-on treatment effects by group.
est.att.avg	inference for att.avg.
est.att.avg.unit	inference for att.avg.unit.
est.att	inference for att.
est.att.W	inference for weighted att.
est.att.off	inference for switch-off.
est.att.off.W	inference for weighted switch-off.
est.placebo	inference for placebo ATT.
est.carryover	inference for carryover ATT.

`est.eff.calendar`
: inference for `eff.calendar`.

`est.eff.calendar.fit`
: inference for `eff.calendar.fit`.

`est.balance.att`
: inference for balanced sample switch-on.

`est.balance.avg`
: inference for balanced sample average ATT.

`est.balance.placebo`
: inference for balanced sample placebo.

`est.avg.W`
: inference for `att.avg.W`.

`est.beta`
: inference for `beta`.

`est.group.att`
: inference for group-specific ATT.

`est.group.output`
: inference for group output.

`att.avg.boot`
: bootstrap draws for `att.avg`.

`att.avg.unit.boot`
: bootstrap draws for `att.avg.unit`.

`att.count.boot`
: bootstrap draws for `count`.

`att.off.boot`
: bootstrap draws for `att.off`.

`att.off.count.boot`
: bootstrap draws for `count.off`.

`att.placebo.boot`
: bootstrap draws for `att.placebo`.

`att.carryover.boot`
: bootstrap draws for `att.carryover`.

`balance.att.boot`
: bootstrap draws for `balance.att`.

`att.bound`
: equivalence confidence interval for pre-trend.

`att.off.bound`
: equivalence confidence interval for switch-off.

`beta.boot`
: bootstrap draws for `beta`.

`test.out`
: F-test and equivalence test results for pre-treatment fit.

`loo.test.out`
: leave-one-period-out test results.

`permute`
: permutation test results.

Author(s)

Licheng Liu; Ye Wang; Yiqing Xu; Ziyi Liu

References

Athey, S., Bayati, M., Doudchenko, N., Imbens, G., and Khosravi, K. (2021). Matrix completion methods for causal panel data models. *Journal of the American Statistical Association, 116*(536), 1716-1730.

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica, 77*(4), 1229-1279.

Liu, L., Wang, Y., and Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science, 68*(1), 160-176.

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57-76.

See Also

`plot.fect`, `print.fect`

Examples

```
library(fect)
data(simdata)
out <- fect(Y ~ D + X1 + X2, data = simdata,
            index = c("id","time"), force = "two-way",
            CV = TRUE, r = c(0, 5), se = 0, parallel = FALSE)
```

`fect-internal` *Internal FEct Functions*

Description

Internal fect functions

Details

These are not to be called by the user (or in some cases are just waiting for proper documentation to be written :).

fect_iden *Over-identification test for causal moderation (cm)*

Description

Implements the over-identification test described in the slide: run two auxiliary regressions of residualized outcomes on the moderator and covariates (including quadratic and interaction terms) with unit and time fixed effects, then test using $n \times R^2 \sim \chi^2(df)$, where df is the number of *nonlinear* (quadratic + interaction) terms that remain in the fitted model.

Usage

```
fect_iden(
  x,
  moderator,
  covariates = NULL,
  quadratic = TRUE,
  interaction = TRUE
)
```

Arguments

x	A ‘fect‘ object. Must be estimated with ‘cm=TRUE‘ so that ‘x$est.cm‘ exists.
moderator	Character scalar. Name of the moderator M_{it} (must be in ‘x$X‘).
covariates	Optional character vector. Names of other covariates X_{it}. Default is all covariates in ‘x$X‘ except ‘moderator‘.
quadratic	Logical. Include quadratic terms M_{it}^2 and X_{it}^2. Default TRUE.
interaction	Logical. Include interaction terms $M_{it} \times X_{it}$. Default TRUE.

Value

A list with elements ‘e1‘ and ‘e0‘, each containing: - ‘n‘: sample size used in the auxiliary regression - ‘r2‘: R-squared of the auxiliary regression - ‘stat‘: $n \times R^2$ - ‘df‘: degrees of freedom (# nonlinear terms kept) - ‘p‘: p-value from chi-square test - ‘model‘: the fitted ‘fixest‘ model

fect_mspe *Mean Squared Prediction Error Evaluation for fect Objects*

Description

Evaluates the prediction accuracy of one or more `fect` model fits by hiding a subset of control observations and comparing counterfactual predictions against actual values.

Usage

```
fect_mspe(
  out.fect,
  seed = NULL,
  cv.method = "rolling",
  cv.nobs = 3,
  cv.donut = 1,
  cv.buffer = 1,
  cv.prop = 0.1,
  min.T0 = 5,
  k = 20,
  criterion = "mspe",
  W = NULL,
  norm.para = NULL,
  proportion = 0
)
```

Arguments

out.fect	A fitted fect object or a named list of fitted fect objects to compare.
seed	Optional integer; random seed for reproducibility.
cv.method	Character; cross-validation masking strategy. One of "rolling" (default), "block" (random scattered anchors with contiguous-block masking), or "loo". Legacy aliases "all_units" (= "block") and "treated_units" are still accepted but emit a deprecation message.
cv.nobs	Integer; number of observations to mask per unit per fold. Default is 3.
cv.donut	Integer; number of periods around treatment onset to exclude from masking. Used by block CV. Default is 1.
cv.buffer	Integer; number of past-side cells masked from training (but not scored) before each rolling-window holdout. Used only by cv.method = "rolling". Default is 1.
cv.prop	Numeric; for block CV, proportion of observations to mask per round; for rolling CV, fraction of eligible units sampled per fold. Default is 0.2.
min.T0	Integer; minimum number of pre-treatment periods required. Default is 5.
k	Integer; number of cross-validation folds. Default is 5.
criterion	Character; scoring criterion. One of "mspe", "wmspe", "gmspe", "wgmspe", "mad", "moment", "gmoment". Default is "mspe".
W	Optional TT x N observation weight matrix. Default is NULL.
norm.para	Optional normalization vector. Default is NULL.
proportion	Numeric; proportion cutoff for count.T.cv (same as fect_cv). Default is 0.

Value

A list containing:

summary	A data frame with mean scores across replications for each model, including MSPE, WMSPE, GMSPE, WGMSPE, MAD, Moment, GMoment, RMSE, and Bias.
records	A data frame with per-replication results including Rep, Model, Hidden_N, and all score columns.
fits	The refitted fect objects from the last replication.
criterion	The scoring criterion used.
scores	The scores from the last replication.

Examples

```
## Not run:
out <- fect(Y ~ D, data = df, index = c("unit", "time"), method = "ife", r = 2)
mspe <- fect_mspe(out.fect = out, hide_n = 10, seed = 42, n_rep = 5)
mspe$summary

## End(Not run)
```

---

fect_sens *Sensitivity Analysis for fect Objects under Relative Magnitude and Smoothness Restrictions*

Description

Conducts sensitivity analyses on fect model objects under relative magnitude (RM) and smoothness (C-LF) assumptions, producing robust confidence intervals and parameter sets.

Usage

```
fect_sens(
  fect.out,
  post.periods = NA,
  l_vec = NA,
  Mbarvec = seq(0, 1, by = 0.1),
  Mvec = seq(0, 0.25, 0.05),
  periodMbarvec = c(0, 0.5),
  periodMvec = c(0, 0.1),
  parallel = FALSE,
  cores = NULL
)
```

Arguments

fect.out	A fitted fect object.
post.periods	Vector of post-treatment periods for sensitivity analysis. Default uses all available post-treatment periods.

l_vec	Optional weighting vector for averaging ATT across post-treatment periods. Default weights by treated-unit counts.
Mbarvec	Values of Mbar for overall RM-based sensitivity analysis.
Mvec	Values of M for overall smoothness-based sensitivity analysis.
periodMbarvec	Values of Mbar for period-specific RM sensitivity analysis.
periodMvec	Values of M for period-specific smoothness sensitivity analysis.
parallel	Logical; if TRUE, uses parallel computation where supported. Default is FALSE.
cores	Optional integer; number of workers when parallel = TRUE.

Details

This function:

1. Extracts ATT estimates and variance-covariance matrices from fect.out.
2. Constructs weighted averages across post-treatment periods.
3. Computes robust confidence sets under RM and C-LF assumptions.
4. Optionally computes robust bounds for each post-treatment period separately.

Robust sets are computed using functions from the **HonestDiDFEct** package.

Value

An updated fect object including:

sensitivity.rm	Relative Magnitude (RM) sensitivity results for average and period-by-period ATT.
sensitivity.smooth	Smoothness (C-LF) sensitivity results for average and period-by-period ATT.

Author(s)

Rivka Lipkovitz

Examples

```
## Not run:
out <- fect(Y ~ D, data = df, index = c("unit", "time"), method = "fe", se = TRUE)
out_sens <- fect_sens(
  fect.out = out,
  post.periods = c(1, 2, 3, 4),
  Mbarvec = seq(0, 1, by = 0.2),
  Mvec = c(0, 0.05, 0.1),
  periodMbarvec = c(0, 0.5),
  periodMvec = c(0, 0.1)
)
names(out_sens$sensitivity.rm)
names(out_sens$sensitivity.smooth)

## End(Not run)
```

`get.cohort` *Obatin the Cohort Index*

Description

Obtain the cohort index based on the provided panel data.

Usage

```
get.cohort(data, D, index,
                varname = NULL,
                start0 = FALSE,
                entry.time = NULL,
                drop.always.treat = FALSE)
```

Arguments

`data`	a data frame, can be a balanced or unbalanced panel data.
`D`	the treatment indicator. The treatment should be binary (0 and 1).
`index`	a two-element string vector specifying the unit and time indicators. Must be of length 2. Every observation should be uniquely defined by the pair of the unit and time indicator.
`varname`	a vector of strings that specifies the variable names for the cohort index and relative periods to be generated.
`start0`	a logical flag that indicates whether period 0 is the first post-treatment period. By default, it is set to FALSE, meaning that period 1 is considered the initial post-treatment period.
`entry.time`	a list of intervals representing the initial calendar time of treatment for each cohort. Units that received treatment within each specified interval are categorized as belonging to a single cohort.
`drop.always.treat`	a logical flag that indicates whether to drop all always-treated units.

Details

`get.cohort` function preprocesses the data and generates the cohort index for different groups. If not specified in `varname`, the function automatically creates four new variables: 'FirstTreat' stores the initial calendar time when a unit is first treated, 'Cohort' stores the cohort time, 'Time_to_Treatment' tracks the period relative to the treatment, and 'Time_to_Exit' tracks the period relative to the treatment exit. Users have the option to specify custom names for these newly generated variables using the `varname` parameter. By default, the function sets `start0=FALSE`, where the relative period 1 corresponds to the first post-treatment/post-exit-treatment period. If `start0=TRUE`, period 0 becomes the first post-treatment/post-exit-treatment period. Additionally, users can categorize units into specific cohorts using the `entry.time` parameter. For instance, if `entry.time=list(c(1,5),c(6,10))`, units that receive their first treatment within the calendar intervals `c(1,5)` and `c(6,10)` are grouped into two distinct cohorts. Units that have never been treated are always categorized as the "control cohort".

Value

`data` a new data frame containing the cohort index.

Author(s)

Licheng Liu; Yiqing Xu, Ziyi Liu; Zhongyu Yin

---

gs2020 *Simulated Gsynth-like Panel Data (No Reversal)*

Description

A simulated dataset for demonstration and testing.

Format

data.frame

---

hh2019 *Simulated Panel Data Example*

Description

A simulated dataset used in package vignettes and examples.

Format

data.frame

---

imputed_outcomes *Imputed Potential Outcomes Accessor*

Description

Returns the cell-level imputed potential-outcome surface from a fitted `fect` object as a long-form data frame. This is the rock-bottom accessor for any post-hoc estimand: read off the columns and aggregate however you like. Used internally by `estimand` for the shipped estimand types; exposed publicly so users can compose custom estimands the package does not ship.

Usage

```
imputed_outcomes(
  fit,
  cells = NULL,
  replicates = FALSE,
  direction = c("on", "off")
)
```

Arguments

fit	A fect object (output of fect).
cells	Optional filter on which treated cells to return. Accepts NULL (default; all treated cells), a logical vector aligned with the unfiltered row order, or a one-sided formula evaluated against the long-form data (e.g., ~ event.time %in% 1:5 & !id %in% bad_ids).
replicates	Logical. FALSE (default) returns one row per treated cell with the point-estimate Y0_hat. TRUE expands by bootstrap/jackknife replicate; requires the fit to have been built with keep.sims = TRUE.
direction	Either "on" (default; event time relative to treatment onset) or "off" (relative to treatment exit; only meaningful on reversal panels).

Value

A data frame with one row per treated cell or per (treated cell, replicate). Columns:

id Unit identifier from fit$id.

time Calendar time from fit$rawtime.

event.time Event time at this cell (relative to onset or exit per direction).

cohort First-treatment calendar time for the unit (direction = "on") or first-exit calendar time (direction = "off").

treated Logical; always TRUE in returned rows.

Y_obs Observed outcome at this cell.

Y0_hat Imputed counterfactual outcome (Y_obs - eff).

eff Cell-level score: contribution to the ATT estimator. For imputation estimators this equals Y_obs - Y0_hat; for doubly-robust estimators it equals (Y_obs - Y0_hat) + eff_debias.

eff_debias Debias correction; 0 for plain imputation estimators, populated for DR estimators.

W.agg Aggregation weight at this cell; 1 if the fit was built without W or W.agg.

replicate (only when replicates = TRUE) 1..nboots for bootstrap, or the dropped-unit index for jackknife.

Memory cost

With replicates = TRUE the returned data frame has n_treated_cells * nboots rows. For typical panels this is manageable; for large panels ($TT \times N \geq 50{,}000$ and nboots ≥ 500) consider filtering via cells before expansion.

Slot contract

This function reads from `fit$Y.dat`, `fit$D.dat`, `fit$T.on` (or `T.off`), `fit$eff`, `fit$eff.boot` (when `replicates = TRUE`), `fit$eff_debias` (when populated), `fit$W.agg`, `fit$id`, `fit$rawtime`. These slots have a documented stable shape; see the package design notes.

See Also

`estimand` for the typed dispatcher built on top of this accessor; `fect` for the fitting interface.

Examples

```
## Not run:
library(fect)
fit <- fect(Y ~ D, data = simdata, index = c("id", "time"),
            method = "fe", force = "two-way",
            se = TRUE, nboots = 200, parallel = FALSE,
            keep.sims = TRUE)

## Point-estimate long form: one row per treated cell.
po <- imputed_outcomes(fit)
head(po)

## Filter to first 5 event times.
po5 <- imputed_outcomes(fit, cells = ~ event.time %in% 1:5)

## Bootstrap replicate expansion (requires keep.sims = TRUE).
po_b <- imputed_outcomes(fit, replicates = TRUE)
nrow(po_b) == nrow(po) * 200    # one row per (cell, replicate)

## End(Not run)
```

---

interFE *Interactive Fixed Effects Models*

Description

Estimating interactive fixed effect models.

Usage

```
interFE(formula = NULL, data, Y, X = NULL, W = NULL,
         index, r = 0, force = "two-way",
         se = FALSE, nboots = 500, seed = NULL,
         tol = 1e-3, max_iteration = 500,
         binary = FALSE, QR = FALSE, normalize = FALSE)
```

Arguments

formula	an object of class "formula": a symbolic description of the model to be fitted.
data	a data frame.
Y	outcome variable.
X	time-varying covariates.
W	weights.
index	a two-element string vector specifying the unit (group) and time indicators.
r	an integer specifying the number of factors.
force	a string indicating whether unit/time/both fixed effects will be imposed: "none", "unit", "time", or "two-way".
se	logical; if TRUE, computes bootstrap-based standard errors.
nboots	number of bootstrap runs (ignored if se=FALSE).
seed	random seed.
tol	tolerance for EM algorithm.
max_iteration	max number of EM iterations.
binary	logical flag for a probit link (not fully supported here).
QR	logical flag for QR-based factor analysis in probit model (not fully supported).
normalize	logical; if TRUE, scales outcome/covariates.

Details

interFE estimates interactive fixed effect models as in Bai (2009).

Value

beta	estimated coefficients.
mu	estimated grand mean.
factor	estimated time-varying factors.
lambda	estimated factor loadings.
VNT	diagonal matrix of r eigenvalues.
niter	number of iterations before convergence.
alpha	estimated unit fixed effects (if imposed).
xi	estimated time fixed effects (if imposed).
residuals	model residuals.
sigma2	residual mean squared error.
IC	information criterion.
ValidX	logical for whether valid covariates exist.
dat.Y	matrix of outcome data.
dat.X	array of independent variables.

Y	name of the outcome variable.
X	name of time-varying control variables.
index	name of unit/time indicators.
est.table	table of final estimates.
est.boot	matrix of bootstrap results.

Author(s)

Licheng Liu; Ye Wang; Yiqing Xu

References

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica, 77*(4), 1229-1279.

See Also

print.interFE, fect

Examples

```
library(fect)
data(simdata)
d <- simdata[-(1:150),] # remove the treated units
out <- interFE(Y ~ X1 + X2, data = d, index=c("id","time"),
               r = 2, force = "two-way", nboots = 50)
```

---

plot.fect *Plot Method for 'fect' Objects*

Description

Visualizes results from a fect estimation.

Usage

```
## S3 method for class 'fect'
plot(
    x,
    type = NULL,
    restrict = "rm",
    loo = FALSE,
    highlight = NULL,
    highlight.fill = FALSE,
    plot.ci = NULL,
    show.points = TRUE,
    loess.fit = TRUE,
    show.group = NULL,
```

```
bound = NULL,
show.count = TRUE,
proportion = 0.3,
pre.periods = NULL,
f.threshold = NULL,
tost.threshold = NULL,
effect.bound.ratio = FALSE,
stats = NULL,
stats.labs = NULL,
raw = "none",
vis = NULL,
main = NULL,
xlim = NULL,
ylim = NULL,
xlab = NULL,
ylab = NULL,
xangle = NULL,
yangle = NULL,
xbreaks = NULL,
ybreaks = NULL,
xticklabels = NULL,
yticklabels = NULL,
gridOff = NULL,
legendOff = FALSE,
legend.pos = NULL,
legend.nrow = NULL,
legend.labs = NULL,
stats.pos = NULL,
show.stats = TRUE,
theme.bw = TRUE,
nfactors = NULL,
include.FE = TRUE,
id = NULL,
cex.main = NULL,
cex.main.sub = NULL,
cex.axis = NULL,
cex.lab = NULL,
cex.legend = NULL,
cex.text = NULL,
axis.adjust = FALSE,
axis.lab = "both",
axis.lab.gap = c(0, 0),
shade.post = FALSE,
start0 = FALSE,
return.test = FALSE,
return.data = FALSE,
balance = NULL,
weight = NULL,
```

```
lcolor = NULL,
lwidth = NULL,
ltype = NULL,
line.color = NULL,
line.width = NULL,
count = NULL,
preset = NULL,
connected = NULL,
ci.outline = FALSE,
color =  NULL,
pre.color = NULL,
post.color = NULL,
est.lwidth = NULL,
est.pointsize = NULL,
count.color =  NULL,
count.alpha = NULL,
count.outline.color = NULL,
placebo.color =  NULL,
carryover.color =  NULL,
carryover.rm.color = NULL,
sens.original.color = NULL,
sens.colors =  NULL,
counterfactual.color =  NULL,
counterfactual.raw.controls.color =  NULL,
counterfactual.raw.treated.color =  NULL,
counterfactual.linetype =  NULL,
box.control =  NULL,
box.treat =  NULL,
calendar.color = NULL,
calendar.lcolor = NULL,
calendar.cicolor = NULL,
heterogeneous.color = NULL,
heterogeneous.lcolor = NULL,
heterogeneous.cicolor = NULL,
equiv.color =  NULL,
status.treat.color =  NULL,
status.control.color = NULL,
status.missing.color =  NULL,
status.removed.color =  NULL,
status.placebo.color = NULL,
status.carryover.color =  NULL,
status.carryover.rm.color =  NULL,
status.balanced.post.color = NULL,
status.balanced.pre.color = NULL,
status.background.color = NULL,
covariate = NULL,
covariate.labels = NULL,
covariate.value = NULL,
```

```
    covariate.value.range = FALSE,
    relative.time = FALSE,
    pretreatment = FALSE,
    num.pretreatment = 3,
    cm = FALSE,
    legacy.style = FALSE,
    ...
)
```

Arguments

x	A fitted fect object.
type	Plot type. Options include: "gap", "equiv" (equivalence test), "status" (treatment status), "exit" (exiting treatment effects), "factors", "loadings", "calendar" (ATT by calendar time), "box" (individual effects distribution), "counterfactual", "sens" (sensitivity analysis plot, e.g., for Rambachan & Roth bounds), "sens_es" (event-study style sensitivity plot), or "cumul" (cumulative effect plot). Default is context-dependent (e.g., "gap", or "equiv" if loo=TRUE).
restrict	For sensitivity plots (type = "sens" or type = "sens_es"). Specifies the type of restriction: "rm" (restriction on M, for relative magnitude bounds) or "sm" (smoothness restriction, for bounds based on second differences). Default is "rm".
loo	Logical; if TRUE (default is FALSE), use leave-one-out estimates for pre-treatment periods in equivalence tests or gap plots.
highlight	Controls which test-period types receive the colored-glyph treatment in placebo / carryover plots. Accepts: • NULL (default): auto — highlight every test type that ran at fit time (placebo, carryover, carryover.rm). • TRUE: same as NULL but explicit. • FALSE: no highlighting; all periods render as plain pre/post circles. • A character subset of c("placebo", "carryover", "carryover.rm"): highlight only the named test types; other test periods render as plain circles. For example, highlight = "placebo" on a fit with both placeboTest = TRUE and carryoverTest = TRUE draws orange triangles at placebo periods but renders carryover periods as ordinary post-treatment circles.
highlight.fill	Logical. If TRUE, draws a lightened-tone background rectangle behind each highlighted period (in the same hue as the period's accent glyph). If FALSE (the default), only the colored glyph is drawn. The glyph alone is usually sufficient signal that a period belongs to a test window, and the glyph-only look survives grayscale printing without surprise. Set TRUE for slide / talk figures where the rectangle helps locate the window for a remote audience.
plot.ci	Specifies the confidence interval to be plotted. Options include: "0.9", "0.95", or "none". Default is NULL, which is context-dependent (e.g., "0.95" for gap plots, "0.9" for equivalence plots, "none" if no CIs are available in the fect object).

show.points	Logical; if TRUE (default), shows point estimates on event study style plots (e.g., gap, equiv, exit plots).
show.group	Optional character string or NULL (default); if specified, plots results for a particular subgroup defined in the fect call.
bound	Which bounds to display in bounding/equivalence tests: "none" (default for most plots), "min" (minimal effect bound), "equiv" (equivalence range), or "both" (default for type="equiv").
show.count	Logical; if TRUE (default), shows a histogram of observation counts (number of treated units) in relevant plots like gap, calendar, or box plots.
proportion	Numeric (0 to 1); for event study plots, restricts plotted time points to those where the number of treated units is at least this fraction of the maximum number of treated units observed across all periods. Default is 0.3.
pre.periods	Optional numeric vector; specifies pre-treatment periods to be used for bounding and equivalence tests. Default is NULL, which uses the pre-treatment periods defined in the fect object.
f.threshold	Numeric or NULL (default); F-test threshold for equivalence checks. If NULL, uses the value from the fect object.
tost.threshold	Numeric or NULL (default); threshold for TOST-based equivalence checks (e.g., for pre-trend, placebo, or carryover tests). If NULL, uses the value from the fect object.
effect.bound.ratio	Logical; if TRUE (default is FALSE), shows the ratio of ATT to the minimal bound in equivalence plots.
stats	Character vector or NULL (default); specifies which test statistics to display on the plot (e.g., "F.p", "equiv.p", "placebo.p"). Default is context-dependent based on plot type and tests performed.
stats.labs	Character vector or NULL (default); custom labels for the displayed statistics. Must match the length of stats.
raw	For type = "counterfactual" plot: "none" (default; shows average treated and counterfactual lines), "band" (shows 5-95% quantile band for raw control/treated data), or "all" (shows all raw control/treated unit trajectories).
vis	(Deprecated) Formerly controlled line display for placebo/carryover tests. This is now handled by highlight and internal logic.
main	Character string or NULL (default); main plot title. If NULL, a default title is used based on the plot type.
xlim	Numeric vector of length 2 or NULL (default); specifies x-axis limits.
ylim	Numeric vector of length 2 or NULL (default); specifies y-axis limits.
xlab	Character string or NULL (default); x-axis label. If NULL, a default label is used based on the plot type.
ylab	Character string or NULL (default); y-axis label. If NULL, a default label is used based on the plot type.
xangle	Numeric or NULL (default); angle (in degrees) for x-axis tick labels.
yangle	Numeric or NULL (default); angle (in degrees) for y-axis tick labels.

xbreaks	Numeric vector or NULL (default); positions of major ticks on the x-axis.
ybreaks	Numeric vector or NULL (default); positions of major ticks on the y-axis.
xticklabels	Character vector or NULL (default); custom labels for the x-axis ticks. Primarily used in status plots.
yticklabels	Character vector or NULL (default); custom labels for the y-axis ticks. Primarily used in status plots.
gridOff	Logical or NULL (default). If TRUE, removes major and minor grid lines. If NULL (which defaults to FALSE internally for this argument), the grid is off for most plots but on for type = "status" (where it forms tile outlines).
legendOff	Logical; if TRUE (default is FALSE), hides the legend.
legend.pos	Character string or NULL (default); position of the legend (e.g., "bottom", "top", "right", "left", "none", or c(x,y) coordinates). Default is typically "bottom".
legend.nrow	Integer or NULL (default); number of rows in the legend.
legend.labs	Character vector or NULL (default); custom legend labels.
stats.pos	Numeric vector of length 2 or NULL (default); (x,y) coordinates for displaying the text of test statistics on the plot.
show.stats	Logical; if TRUE (default), displays test statistics on the plot if stats is not "none".
theme.bw	Logical; if TRUE (default), uses a black-and-white ggplot2 theme (theme_bw()). If FALSE and no preset is specified, uses theme_grey().
nfactors	Integer or NULL (default); number of factors to plot when type is "factors" or "loadings". Defaults to min(r.cv, 4), where r.cv is the number of estimated factors.
include.FE	Logical; if TRUE (default), includes fixed effects as additional "factors" in factor/loading plots if applicable (i.e., if unit or time fixed effects were estimated as part of the factor model).
id	Character vector or NULL (default); when specified, plots only the chosen unit(s). For counterfactual plots (type="counterfactual"), only one ID is typically supported for individual unit display; otherwise, averages are shown.
cex.main	Numeric or NULL (default); font size multiplier for the main plot title. Base size is 16pt.
cex.main.sub	Numeric or NULL (default); font size multiplier for a possible subtitle. Base size is 16pt.
cex.axis	Numeric or NULL (default); font size multiplier for axis tick labels. Base size is 15pt.
cex.lab	Numeric or NULL (default); font size multiplier for axis labels. Base size is 15pt.
cex.legend	Numeric or NULL (default); font size multiplier for legend text. Base size is 15pt.
cex.text	Numeric or NULL (default); font size multiplier for text annotations on the plot (e.g., statistics values). Base size is 5pt.
axis.adjust	Logical; if TRUE (default is FALSE), attempts to adjust axis labels for plots with many time points (e.g., by rotating x-axis labels).

axis.lab — Controls axis labeling in type="status" plots. Options: "both" (default), "time", "unit", or "off".

axis.lab.gap — Numeric vector of length 1 or 2; gap spacing for axis labels in type="status" plots (sets frequency of labels). Default is c(0, 0) (label every tick).

shade.post — Logical; if FALSE (default). If TRUE, shades post-treatment periods in certain plots. For counterfactual plots, if NULL (the default for the argument is FALSE), it effectively becomes TRUE internally unless explicitly set to FALSE.

start0 — Logical; if TRUE (default is FALSE), re-indexes the first post-treatment period as 0 in event-time plots (gap, equiv, exit). The period before treatment becomes -1.

return.test — Logical; if TRUE (default is FALSE), returns a list containing the plot object and relevant test statistics instead of just the plot.

return.data — Logical; if TRUE (default is FALSE), returns a data frame with the data used to construct the plot instead of returning a plot object. If return.test=TRUE as well, returns a list with data and test.out.

balance — Logical or NULL (default); if TRUE, uses a "balance sample" (if available from the fect object, e.g., from setting balanced.sample = TRUE in fect) for plotting.

weight — Logical or NULL (default); if TRUE, applies weighting (if available from the fect object, e.g., from providing weights in fect) to the estimates being plotted.

lcolor — Character vector of length 1 or 2, or NULL (default). If NULL, uses a default color (e.g., "#AAAAAA70" if theme.bw=TRUE, else "white" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

lwidth — Numeric vector of length 1 or 2, or NULL (default). If NULL, uses a default width (e.g., 1.5 if theme.bw=TRUE, else 2 for the base style). The preset argument may define a different width. Specifying this argument directly overrides any preset or base style setting.

ltype — Character vector of length 1 or 2, or NULL (default). If NULL, uses a default linetype (e.g., c("solid", "solid") for the base style). The preset argument may define a different linetype (e.g., preset="vibrant" uses c("solid", "dashed")). Specifying this argument directly overrides any preset or base style setting.

line.color — (Deprecated) Use color for main estimate lines or lcolor for reference lines instead.

line.width — (Deprecated) Use est.lwidth for main estimate lines or lwidth for reference lines instead.

count — (Deprecated) Use show.count to control visibility and count.color, count.alpha, count.outline.color for appearance.

preset — A character string specifying a color and style preset: NULL (default, uses a base style, often with theme_bw if theme.bw=TRUE), "vibrant", or "grayscale". This sets a group of defaults for many plot elements. Individual appearance arguments (like color, lcolor, etc.) can override settings from a preset or the base style.

connected — Logical or NULL (default). For event study type plots (e.g., type="gap"), if TRUE, connects points with lines; if FALSE, shows point-ranges. If NULL, defaults to FALSE for the base style, and TRUE if preset = "vibrant".

`ci.outline` Logical; if `FALSE` (default). For event study plots where CIs are shown as ribbons, if `TRUE`, adds an outline to the CI ribbon.

`color` Character string or `NULL` (default). If `NULL`, uses a default color for main plot lines/points (e.g., `"black"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`pre.color` Character string or `NULL` (default). Color for pre-treatment period estimates. If `NULL`, uses the value of `color`.

`post.color` Character string or `NULL` (default). Color for post-treatment period estimates. If `NULL`, uses the value of `color`.

`est.lwidth` Numeric or `NULL`; line width for the main estimate line in event study plots. If `NULL` (default), the width is determined based on `connected` and `show.points` (e.g., '0.6' for point-ranges with the base style, '0.7' for connected points with markers with the base style). The `preset` argument may influence this default. Specifying a numeric value directly overrides any other setting.

`est.pointsize` Numeric or `NULL`; size of points for the main estimate in event study plots. If `NULL` (default), the size is determined based on `connected` and `show.points` (e.g., '2' for point-ranges with the base style, '1.2' for connected points with markers with the base style). The `preset` argument may influence this default. Specifying a numeric value directly overrides any other setting.

`count.color` Character string or `NULL` (default). If `NULL`, uses a default color for count bars (e.g., `"grey70"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`count.alpha` Numeric (0-1) or `NULL` (default). If `NULL`, uses a default alpha for count bars (e.g., `0.4` for the base style). The `preset` argument may define a different alpha. Specifying this argument directly overrides any `preset` or base style setting.

`count.outline.color`
Character string or `NULL` (default). If `NULL`, uses a default outline color for count bars (e.g., `"grey69"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`placebo.color` Character string or `NULL` (default). If `NULL`, uses a default color for placebo elements (e.g., `"blue"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`carryover.color`
Character string or `NULL` (default). If `NULL`, uses a default color for carryover elements (e.g., `"red"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`carryover.rm.color`
Character string or `NULL` (default). If `NULL`, uses a default color for removed carryover elements (e.g., `"blue"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

sens.original.color
: Character string or NULL (default). If NULL, uses a default color for original estimates in sensitivity plots (e.g., "darkblue" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

sens.colors
: Vector of colors or NULL (default). If NULL, uses a default palette for sensitivity bounds (e.g., c("#218C23","#FF34B4","#FF521B","#2B59C3") for the base style). The preset argument may define a different palette. Specifying this argument directly overrides any preset or base style setting.

counterfactual.color
: Character string or NULL (default). If NULL, uses a default color for counterfactual lines (e.g., "steelblue" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

counterfactual.raw.controls.color
: Character string or NULL (default). If NULL, uses a default color for raw control lines (e.g., "#4682B420" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

counterfactual.raw.treated.color
: Character string or NULL (default). If NULL, uses a default color for raw treated lines (e.g., "#77777750" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

counterfactual.linetype
: Character string or NULL (default). If NULL, uses a default linetype for counterfactual lines (e.g., "longdash" for the base style). The preset argument may define a different linetype. Specifying this argument directly overrides any preset or base style setting.

box.control
: Character string or NULL (default). If NULL, uses a default fill color for control boxplots (e.g., "skyblue" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

box.treat
: Character string or NULL (default). If NULL, uses a default fill color for treated boxplots (e.g., "pink" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

calendar.color
: Character string or NULL (default). If NULL, uses a default color for fitted lines in calendar plots (e.g., "skyblue" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

calendar.lcolor
: Character string or NULL (default). If NULL, uses a default color for average ATT lines in calendar plots (e.g., "red" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

`calendar.cicolor`
: Character string or `NULL` (default). If `NULL`, uses a default color for CI in calendar plots (e.g., `"red"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`heterogeneous.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for fitted lines in heterogeneous plots (e.g., `"red"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`heterogeneous.lcolor`
: Character string or `NULL` (default). If `NULL`, uses a default color for average ATT lines in heterogeneous plots (e.g., `"red"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`heterogeneous.cicolor`
: Character string or `NULL` (default). If `NULL`, uses a default color for CI in heterogeneous plots (e.g., `"red"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`equiv.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for equivalence bounds (e.g., `"red"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`status.treat.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for treated status (e.g., `"#06266F"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`status.control.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for control status (e.g., `"#B0C4DE"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`status.missing.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for missing status (e.g., `"#FFFFFF"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`status.removed.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for removed status (e.g., `"#A9A9A9"` for the base style). The `preset` argument may define a different color. Specifying this argument directly overrides any `preset` or base style setting.

`status.placebo.color`
: Character string or `NULL` (default). If `NULL`, uses a default color for placebo status (e.g., `"#66C2A5"` for the base style). The `preset` argument may define a

different color. Specifying this argument directly overrides any preset or base style setting.

status.carryover.color
: Character string or NULL (default). If NULL, uses a default color for carryover status (e.g., "#E78AC3" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

status.carryover.rm.color
: Character string or NULL (default). If NULL, uses a default color for removed carryover status (e.g., "#ffc425" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

status.balanced.post.color
: Character string or NULL (default). If NULL, uses a default color for balanced post-treatment status (e.g., "#00852B" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

status.balanced.pre.color
: Character string or NULL (default). If NULL, uses a default color for balanced pre-treatment status (e.g., "#A5CA18" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

status.background.color
: Character string or NULL (default). If NULL, uses a default background color for status plots (e.g., "gray90" for the base style). The preset argument may define a different color. Specifying this argument directly overrides any preset or base style setting.

covariate
: Character string or NULL (default). If NULL, uses a default covariate for heterogeneous plots (e.g., "X" for the base style). The preset argument may define a different covariate. Specifying this argument directly overrides any preset or base style setting.

covariate.labels
: Character vector or NULL (default). If NULL, uses a default covariate labels for heterogeneous plots (e.g., "X1" for the base style). The preset argument may define a different covariate labels. Specifying this argument directly overrides any preset or base style setting.

covariate.value
: Numeric or NULL (default); a specific covariate value at which to evaluate the HTE plot. If NULL, the full range is used.

covariate.value.range
: Logical; if TRUE, plots HTE over the range of covariate values. Default is FALSE.

relative.time
: Logical; if TRUE, uses relative time (time since treatment) on the x-axis for HTE plots. Default is FALSE.

pretreatment
: Logical; if TRUE, restricts HTE analysis to pre-treatment periods only. Default is FALSE.

num.pretreatment
: Integer; number of pre-treatment periods to include when pretreatment = TRUE. Default is 3.

cm
: Logical; if TRUE, plots causal moderation estimates instead of effect modification estimates for HTE plots. Requires the fect object to have been estimated with cm = TRUE. Default is FALSE.

legacy.style
: Logical; if TRUE, reproduces the pre-2.3.1 visual defaults (bold centered title, larger axis text, solid treatment-onset line, blue placebo triangles, no peach highlight rectangle) regardless of the value of theme.bw. If FALSE (the default) and theme.bw = TRUE, the modern v2.3.1 visual recipe is used. Pass legacy.style = TRUE for byte-identical reproduction of figures rendered under earlier fect versions.

loess.fit
: Logical; if TRUE (default), overlays a LOESS smoothing curve on HTE scatter plots.

...
: Additional graphical parameters passed to internal plotting routines, primarily those accepted by esplot for event study style plots (gap, equiv, exit, sens_es, cumul).

Details

plot.fect generates various visualizations for objects estimated via fect. Depending on the selected type, it can show treatment-effect dynamics ("gap", "exit", "cumul"), equivalence test outcomes ("equiv"), factor/loadings for factor models ("factors", "loadings"), raw vs. counterfactual trajectories ("counterfactual"), treatment status ("status"), effects by calendar time ("calendar"), distribution of individual effects ("box"), or sensitivity analysis results ("sens", "sens_es"). The preset argument allows for quick customization of plot aesthetics using predefined color schemes. Many individual color and style parameters can be set to further customize the appearance, and these will override any settings from a preset or the base style. The function heavily relies on ggplot2 for plotting and esplot for event-study style visualizations.

Value

By default, a ggplot2 object representing the plot.

If return.data=TRUE, returns a list containing:

p
: The ggplot2 object.

data
: A named list of data frames containing the data used to construct each layer of the plot (no extra all-NA columns are added).

If return.test=TRUE as well, the list also contains:

test.out
: A list or data frame containing relevant test statistics (such as F-tests, equivalence p-values, placebo test p-values, etc.), if applicable to the plot type and options chosen.

If return.test=TRUE and return.data=FALSE, returns a list containing p and test.out.

Author(s)

Licheng Liu, Ye Wang, Yiqing Xu, Ziyi Liu, Rivka Lipkovitz

References

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57–76.

Liu, L., Wang, Y., & Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science, 66*(1), 220-237. (Provides context for `esplot` which is used internally for many plot types)

Rambachan, A., & Roth, J. (2023). A More Credible Approach to Parallel Trends. *Review of Economic Studies, 90*(5), 2555-2591. (Provides context for sensitivity analysis plots like `type="sens"` and `type="sens_es"`)

Examples

```
library(fect)
# For CRAN checks, use a small number of bootstraps
# In practice, use a larger number (e.g., nboots = 200 or more)
if(requireNamespace("ggplot2") && requireNamespace("ggrepel")) {
  data(simdata)
  # Estimate with fixed effects method
  out.fect <- fect(
    Y ~ D + X1 + X2,
    data = simdata,
    index = c("id","time"),
    method = "fe",
    force = "two-way",
    se = TRUE,
    parallel = FALSE,
    nboots = 5 # nboots low for example
  )

  # Default gap plot
  plot(out.fect, main = "Estimated ATT (FEct)", ylab = "Effect of D on Y")

  # Gap plot with vibrant preset and custom line color
  # plot(out.fect, preset = "vibrant", color = "darkgreen",
  #      main = "Estimated ATT (Vibrant Preset, Custom Line)")

  # Counterfactual plot for the first treated unit
  # Need to know the ID of a treated unit. Let's find one.
  treated_ids <- unique(simdata$id[simdata$D == 1])
  if (length(treated_ids) > 0) {
    plot(out.fect, type = "counterfactual", id = treated_ids[1],
         main = paste("Counterfactual for Unit", treated_ids[1]))
  }

  # Status plot
  plot(out.fect, type = "status")

  # Cumulative effect plot (if est.eff is available from fect call)
  # This example might not have it by default, but showing how to call
  # out.fect.cumul <- fect(Y ~ D, data = simdata, index = c("id","time"), method = "fe",
  #                        cumulative = TRUE, se = TRUE, parallel = FALSE, nboots = 5)
```

```
  # if (exists("out.fect.cumul")) {
  #   plot(out.fect.cumul, type = "cumul", main = "Cumulative ATT")
  # }

  # Example for sensitivity plot (requires IFE/GSYNTH method and sensitivity analysis)
  # \donttest{
  #   out.ife <- fect(Y ~ D, data = simdata, index = c("id","time"),
  #                   method = "ife", se = TRUE, r = 2,
  #                 sensitivity.analysis = TRUE, sensitivity.plot = FALSE, # run analysis
  #                   parallel = FALSE, nboots = 5) # nboots low for example
  #   if (!is.null(out.ife$sensitivity.rm)) {
  #     plot(out.ife, type = "sens", restrict = "rm",
  #          main = "Sensitivity Analysis (Relative Magnitude)")
  #     plot(out.ife, type = "sens_es", restrict = "rm",
  #          main = "Event-Study Sensitivity (Relative Magnitude)")
  #   }
  # }
}
```

`print.fect` *Print Results*

Description

Print results of the matrix completion method.

Usage

```
## S3 method for class 'fect'
print(x, switch.on = TRUE,
       switch.off = FALSE,time.on.lim = NULL, time.off.lim = NULL, ...)
```

Arguments

`x`	a `fect` object.
`switch.on`	logical; if `TRUE`, print switch-on effect.
`switch.off`	logical; if `TRUE`, print switch-off effect.
`time.on.lim`	two-element numeric vector specifying the switch-on effect range.
`time.off.lim`	two-element numeric vector specifying the switch-off effect range.
`...`	other arguments.

Value

No return value.

Author(s)

Licheng Liu; Ye Wang; Yiqing Xu; Ziyi Liu

References

Athey, S., Bayati, M., Doudchenko, N., Imbens, G., and Khosravi, K. (2021). Matrix completion methods for causal panel data models. *Journal of the American Statistical Association, 116*(536), 1716-1730.

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica, 77*(4), 1229-1279.

Liu, L., Wang, Y., and Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science, 68*(1), 160-176.

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57-76.

See Also

`fect`, `plot.fect`

Examples

```
library(fect)
data(simdata)
out <- fect(Y ~ D + X1 + X2, data = simdata,
            index = c("id","time"), force = "two-way",
            CV = TRUE, r = c(0, 5), se = 0, parallel = FALSE)
print(out)
```

---

print.interFE *Print Results*

Description

Print results of interactive fixed effects estimation.

Usage

```
## S3 method for class 'interFE'
print(x, ...)
```

Arguments

x	an `interFE` object.
...	other arguments.

Value

No return value.

Author(s)

Licheng Liu; Ye Wang; Yiqing Xu

References

Bai, J. (2009). Panel data models with interactive fixed effects. *Econometrica, 77*(4), 1229-1279.

See Also

interFE, fect

Examples

```
library(fect)
data(simdata)
d <- simdata[-(1:150),] # remove the treated units
out <- interFE(Y ~ X1 + X2, data = d, index=c("id","time"),
               r = 2, force = "two-way", nboots = 50)
print(out)
```

---

r.cv.rolling *Rolling-window cross-validation for rank selection*

Description

Picks the number of factors ‘r‘ for an interactive-fixed-effects model via standard rolling-window cross-validation. For each of ‘k‘ folds, a fraction ‘cv.prop‘ of eligible units (controls plus treated pre-treatment) is sampled; only sampled units carry a mask. For each sampled unit, a random anchor time ‘t*‘ is drawn and the fold’s training set excludes ‘t* - cv.buffer, ..., t* - 1‘ (buffer), ‘t*, ..., t* + cv.nobs - 1‘ (the held-out, scored block), and ‘t* + cv.nobs, ..., end_of_eligible(t)‘ (rolling-window future drop; for treated units, ‘end_of_eligible‘ is the cell strictly before treatment onset). MSPE is scored at the held-out block only and averaged across folds.

Usage

```
r.cv.rolling(
  formula,
  data,
  index,
  method = c("ife", "gsynth", "cfe"),
  r.max = 5L,
  cv.nobs = 3L,
  cv.buffer = 1L,
  k = 20L,
  cv.prop = 0.1,
  cv.rule = c("1se", "min", "1pct"),
  min.T0 = 5L,
```

```
  force = "unit",
  seed = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

formula	A model formula, e.g. 'Y ~ D + X1 + X2'.
data	A long-format data frame.
index	Character vector identifying the panel structure. For 'method = "ife"' and 'method = "gsynth"', length 2: 'c("unit", "time")'. For 'method = "cfe"', length >= 2: the first two entries are 'c("unit", "time")' and any additional entries are extra grouping fixed-effect columns forwarded to the inner 'fect()' call's 'index =' argument.
method	Estimator. One of '"ife"' (IFE-EM, internally 'time.component.from = "notyet-treated"'), '"gsynth"' (GSC, internally 'time.component.from = "nevertreated"'), or '"cfe"' (Complex Fixed Effects, internally 'time.component.from = "notyet-treated"'). All three paths populate 'Y.ct.full' at masked positions, so MSPE scoring works uniformly. For CFE, rolling CV picks 'r' only; CFE-specific arguments ('Z', 'gamma', 'Q', 'Q.type', 'kappa', extra index columns) are forwarded via '...' and held fixed at their user-supplied values.
r.max	Largest candidate rank to evaluate. CV is run over '0:r.max'.
cv.nobs	Length of the held-out (scored) block per unit per fold. Default 3.
cv.buffer	Number of observations immediately BEFORE the held-out block to drop from training (the past-side buffer that attenuates AR-leakage). Default 1. Analogous to 'cv.donut' in the existing 'cv.method = "all_units"' / '"treated_units"' strategies, but applied only on the past side: the future side is dropped by construction.
k	Number of folds. Each fold draws a fresh sample of units and a fresh set of per-unit anchors; the per-r MSPE is averaged across folds and the SE used by the '"1se"' rule reflects fold-to-fold variability. Default 20 (matches the default for the existing CV strategies).
cv.prop	Fraction of eligible units sampled per fold. Only sampled units receive a mask in that fold; the rest stay fully observed and contribute training data at every period. Default 0.1 (paired with 'k = 20' for ~2x coverage of every eligible unit across folds). Across 'k' folds, every eligible unit lands in the holdout roughly 'k * cv.prop' times in expectation. Must satisfy '0 < cv.prop <= 1'. On small panels (n_eligible < 30) consider raising further, since per-fold MSPE precision scales with 'cv.prop * n_eligible * cv.nobs'.
cv.rule	Rule for picking 'r' from the MSPE curve: '"1se"' (default), '"min"', or '"1pct"'.
min.T0	Minimum observations required strictly before the anchor. Sets the lower bound on valid anchor positions. Default 5.
force	One of '"none"', '"unit"', '"time"', '"two-way"'. Default '"unit"'.
seed	Optional integer base seed; per-fold seeds derive from 'seed + fold_id' for reproducibility. Default 'NULL' (use the ambient RNG).

verbose	If TRUE (default), print per-fold per-r MSPE.
...	Additional arguments forwarded to ‘fect()‘. For ‘method = "cfe"‘, the user holds CFE structural arguments (‘Z‘, ‘gamma‘, ‘Q‘, ‘Q.type‘, ‘Q.bspline.degree‘, ‘kappa‘, etc.) fixed at their spec via ‘...‘; rolling CV varies only ‘r‘.

Details

Per-fold unit sampling is required: masking every eligible unit at the same time leaves no donor data at the masked time points and breaks factor identification. Sampling ‘cv.prop‘ of units per fold keeps unsampled units fully observed at all periods.

This is the standard time-series CV design (cf. ‘forecast::tsCV‘, ‘tidymodels::sliding_window‘, ‘caret::createTimeSlices‘) adapted to panel data: each sampled unit gets its own anchor per fold, drawn uniformly from valid positions.

Value

List with components: - ‘r.cv‘: chosen rank. - ‘cv.rule‘: rule applied. - ‘mspe‘: data.frame of per-r MSPE (averaged across folds), SE across folds, and held-out cell counts. - ‘mspe.per.fold‘: r-by-k matrix of per-fold MSPE. - ‘k‘, ‘cv.nobs‘, ‘cv.buffer‘, ‘cv.prop‘: parameters used. - ‘n.units.masked‘: distinct units that contributed to at least one fold’s holdout.

Examples

```
## Not run:
  library(fect)
  data(simdata)
  res <- r.cv.rolling(Y ~ D, data = simdata, index = c("id", "time"),
                      method = "ife", r.max = 5,
                      cv.nobs = 3, cv.buffer = 1, k = 20)
  res$r.cv
  ## then use the chosen r in a CV-disabled fit:
  fit <- fect(Y ~ D, data = simdata, index = c("id", "time"),
              method = "ife", time.component.from = "notyettreated",
              CV = FALSE, r = res$r.cv, se = TRUE)

## End(Not run)
```

---

sim_base *Simulated data (no interactive fixed effects)*

Description

A simulated dataset with continuous outcomes and no interactive fixed effects (factor loadings set to zero). Same DGP as `simdata` but with the factor contribution removed from the outcome.

Format

dataframe

References

Liu, L., Wang, Y., and Xu, Y. (2022). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science, 68*(1), 160-176.

`sim_gsynth` *Simulated data for Gsynth (no treatment reversal)*

Description

A simulated dataset with no treatment reversal, used to demonstrate the generalized synthetic control method (never-treated estimation regime).

Format

dataframe

References

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57-76.

`sim_linear` *Simulated panel data with unit-specific linear time trends (block DID)*

Description

A simulated balanced panel with 200 units and 50 time periods. All 80 treated units receive treatment at period 41 (block DID design). The DGP includes unit-specific slopes on a linear time trend, where treated units have systematically steeper slopes. Used to demonstrate unit-specific time trends with `Q.type = "linear"` in the CFE estimator.

Format

A data frame with 10000 rows and 4 columns:

time Time period (1 to 50)

id Unit identifier (1 to 200)

Y Outcome variable

D Treatment indicator (0/1)

sim_region — *Simulated unbalanced panel with region-specific time effects*

Description

A simulated unbalanced panel with 500 units belonging to 5 regions and 20 time periods. The DGP includes region-specific linear time trends as the confounding source. Treatment probability and timing depend on region, and units in higher-numbered regions enter the panel later, creating unbalancedness. Used to demonstrate additional fixed effects in the CFE estimator.

Format

A data frame with columns:

time Time period (1 to 20)

id Unit identifier (1 to 500)

region Region indicator (1 to 5)

Y Outcome variable

D Treatment indicator (0/1)

region_time Region-by-time interaction factor

sim_trend — *Simulated panel data with unit-specific sinusoidal time trends (block DID)*

Description

A simulated balanced panel with 200 units and 50 time periods. All 80 treated units receive treatment at period 41 (block DID design). The DGP includes unit-specific amplitudes on a half-cycle sinusoidal time trend, where treated units have systematically larger amplitudes.

Format

A data frame with 10000 rows and 4 columns:

id Unit identifier (1 to 200)

time Time period (1 to 50)

Y Outcome variable

D Treatment indicator (0/1)

`simdata` *Simulated panel data with two latent factors*

Description

A simulated panel dataset with continuous outcomes used throughout the package vignettes to demonstrate factor-augmented counterfactual estimators. The data-generating process follows Liu, Wang, and Xu (2024) with one modification (see Format).

The panel has $N = 200$ units and $T = 35$ time periods. Treatment switches on and off over time (99 of 150 treated units experience at least one reversal), reflecting a general treatment pattern rather than simple staggered adoption. The outcome includes two latent factors ($r = 2$), so the parallel-trends assumption is violated and the standard fixed-effects estimator is biased. Treatment assignment loads on the same factors and fixed effects that enter the outcome—units with larger λ_i and α_i are more likely to be treated—so the confounding is structural and cannot be removed by two-way fixed effects alone.

Format

A data frame with the following columns:

id unit identifier (1–200)

time time period (1–35)

Y observed outcome

error idiosyncratic error $\varepsilon_{it} \sim N(0, 2)$

eff realized treatment effect τ_{it}

tr_cum, tr_prob treatment-probability constructions

D treatment indicator

X1, X2 observed time-varying covariates $\sim N(0, 1)$ with coefficients 1 and 3

alpha unit fixed effect $\alpha_i \sim N(0, 1)$

xi time fixed effect ξ_t (AR(1) with drift)

F1, F2 latent time factors $f_t \in \mathbb{R}^2$ (one trending, one white noise)

L1, L2 unit-specific factor loadings $\lambda_i \sim N(0.5, 1)$

FL1, FL2 per-cell factor-loading products $\lambda_{i,k} \cdot f_{t,k}$ $(k = 1, 2)$

The DGP is

$$Y_{it} = \tau_{it} D_{it} + X_{1,it} + 3X_{2,it} + \mu + 3\alpha_i + \xi_t + 2\,\lambda_i' f_t + \varepsilon_{it},$$

with grand mean $\mu = 5$ and treatment effect $\tau_{it} \sim N(0.4 \cdot \mathrm{tr_cum}_{it}/T,\ 0.2)$.

The $2\,\lambda_i' f_t$ term doubles the latent factor contribution relative to the original Liu, Wang, and Xu (2024) DGP. The doubling strengthens the factor signal-to-noise ratio (variance of the factor contribution to variance of the residual) from approximately 2.7 to 10.9, which makes the factor structure clearly recoverable by cross-validated rank-selection procedures on this dataset. The unmodified DGP is preserved in earlier package versions; see `git log data/simdata.rda` for the prior file.

References

Liu, L., Wang, Y., and Xu, Y. (2024). A Practical Guide to Counterfactual Estimators for Causal Inference with Time-Series Cross-Sectional Data. *American Journal of Political Science*, 68(1), 160–176.

`simgsynth` *Simulated data for Gsynth*

Description

A simulated dataset with no treatment reversal.

Format

dataframe

References

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57-76.

`turnout` *EDR and Voter Turnout in the US*

Description

State-level voter turnout data.

Format

dataframe

References

Springer, M. J. (2014). How the States Shaped the Nation: American Electoral Institutions and Voter Turnout, 1920-2000. University of Chicago Press.

Xu, Y. (2017). Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models. *Political Analysis, 25*(1), 57-76.

Index

∗ **datasets**
gs2020, 33
hh2019, 33
sim_base, 54
sim_gsynth, 55
sim_linear, 55
sim_region, 56
sim_trend, 56
simdata, 57
simgsynth, 58
turnout, 58
∗ **ts**
fect-internal, 27
_gsynth_XXinv *(*fect-internal*)*, 27
_gsynth_Y_demean *(*fect-internal*)*, 27
_gsynth_beta_iter *(*fect-internal*)*, 27
_gsynth_beta_iter_ub *(*fect-internal*)*, 27
_gsynth_data_ub_adj *(*fect-internal*)*, 27
_gsynth_fe_ad_covar_iter *(*fect-internal*)*, 27
_gsynth_fe_ad_inter_covar_iter *(*fect-internal*)*, 27
_gsynth_fe_ad_inter_iter *(*fect-internal*)*, 27
_gsynth_fe_ad_iter *(*fect-internal*)*, 27
_gsynth_fe_add *(*fect-internal*)*, 27
_gsynth_fe_add2 *(*fect-internal*)*, 27
_gsynth_inter_fe *(*fect-internal*)*, 27
_gsynth_inter_fe_mc *(*fect-internal*)*, 27
_gsynth_inter_fe_ub *(*fect-internal*)*, 27
_gsynth_panel_beta *(*fect-internal*)*, 27
_gsynth_panel_est *(*fect-internal*)*, 27
_gsynth_panel_factor *(*fect-internal*)*, 27
_gsynth_panel_factor_ub *(*fect-internal*)*, 27
_gsynth_panel_fe *(*fect-internal*)*, 27
_gsynth_panel_fe_ub *(*fect-internal*)*, 27

att.cumu, 4

beta_iter *(*fect-internal*)*, 27
beta_iter_ub *(*fect-internal*)*, 27

cv.sample *(*fect-internal*)*, 27

data_ub_adj *(*fect-internal*)*, 27
did_wrapper, 5

effect, 7, *15*
equiv_test *(*fect-internal*)*, 27
esplot, 9
estimand, 15, *20*, *33*, *35*

fe_ad_covar_iter *(*fect-internal*)*, 27
fe_ad_inter_covar_iter *(*fect-internal*)*, 27
fe_ad_inter_iter *(*fect-internal*)*, 27
fe_ad_iter *(*fect-internal*)*, 27
fe_add *(*fect-internal*)*, 27
fe_add2 *(*fect-internal*)*, 27
fect, *3, 4*, *8*, *16*, 17, *34, 35*, *37*, *40*, *48*, *50–52*
fect-internal, 27
fect-package, 3
fect.default *(*fect-internal*)*, 27
fect.formula *(*fect-internal*)*, 27
fect_fe *(*fect-internal*)*, 27
fect_iden, 28
fect_mc *(*fect-internal*)*, 27
fect_mspe, 28
fect_sens, 30

get.cohort, 32
gs2020, 33

hh2019, 33

imputed_outcomes, *15, 16*, 33
inter_fe *(*fect-internal*)*, 27
inter_fe_mc *(*fect-internal*)*, 27
inter_fe_ub *(*fect-internal*)*, 27
interFE, 35, *51, 52*

interFE.default *(*fect-internal*)*, 27
interFE.formula *(*fect-internal*)*, 27

panel_beta *(*fect-internal*)*, 27
panel_est *(*fect-internal*)*, 27
panel_factor *(*fect-internal*)*, 27
panel_factor_ub *(*fect-internal*)*, 27
panel_fe *(*fect-internal*)*, 27
panel_fe_ub *(*fect-internal*)*, 27
plot.fect, *4*, *8*, *27*, 37, *51*
print.fect, *27*, 50
print.interFE, *37*, 51

r.cv.rolling, 52

sim_base, 54
sim_gsynth, 55
sim_linear, 55
sim_region, 56
sim_trend, 56
simdata, 57
simgsynth, 58

turnout, 58

XXinv *(*fect-internal*)*, 27

Y_demean *(*fect-internal*)*, 27