Package ‘fetwfe’

February 21, 2025

Title Fused Extended Two-Way Fixed Effects

Version 0.4.4

Description Calculates the fused extended two-way fixed effects (FETWFE) estimator for unbiased and efficient estimation of difference-in-differences in panel data with staggered treatment adoption. This estimator eliminates bias inherent in conventional two-way fixed effects estimators, while also employing a novel bridge regression regularization approach to improve efficiency and yield valid standard errors. Provides flexible tuning parameters (including user-specified or data-driven choices for penalty parameters), detailed output including overall and cohort-specific treatment effects with confidence intervals, and extensive diagnostic tools. See details in Faletto (2024) (<doi:10.48550/arXiv.2312.05985>).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports expm, glmnet, grpreg

Suggests bacondecomp, knitr, rmarkdown, dplyr

VignetteBuilder knitr

NeedsCompilation no

Author Gregory Faletto [aut, cre]

Maintainer Gregory Faletto <gfaletto@gmail.com>

Repository CRAN

Date/Publication 2025-02-21 11:40:24 UTC

Contents

fetwfe *Fused extended two-way fixed effects*

Description

Implementation of fused extended two-way fixed effects. Estimates overall ATT as well as CATT (cohort average treatment effects on the treated units).

Usage

```
fetwfe(
  pdata,
  time_var,
  unit_var,
  treatment,
  covs,
  response,
  indep_counts = NA,
  sig_eps_sq = NA,
  sig_eps_c_sq = NA,
  lambda.max = NA,
  lambda.min = NA,
  nlambda = 100,
  q = 0.5,
  verbose = FALSE,
  alpha = 0.05
)
```

Arguments

`pdata` Dataframe; the panel data set. Each row should represent an observation of a unit at a time. Should contain columns as described below.

`time_var` Character; the name of a single column containing a variable for the time period. This column is expected to contain integer values (for example, years). Recommended encodings for dates include format YYYY, YYYYMM, or YYYYMMDD, whichever is appropriate for your data.

`unit_var` Character; the name of a single column containing a variable for each unit. This column is expected to contain character values (i.e. the "name" of each unit).

`treatment` Character; the name of a single column containing a variable for the treatment dummy indicator. This column is expected to contain integer values, and in particular, should equal 0 if the unit was untreated at that time and 1 otherwise. Treatment should be an absorbing state; that is, if unit `i` is treated at time `t`, then it must also be treated at all times `t` + 1, ..., `T`. Any units treated in the first time period will be removed automatically. Please make sure yourself that at least some units remain untreated at the final time period ("never-treated units").

`covs` Character; a vector containing the names of the columns for covariates. All of these columns are expected to contain integer or numeric values (so if you use categorical values, encode them using e.g. binary indicators before passing the data to this function). At least one covariate must be provided.

`response` Character; the name of a single column containing the response for each unit at each time. The response must be an integer or numeric value.

`indep_counts` (Optional.) Integer; a vector. If you have a sufficiently large number of units, you can optionally randomly split your data set in half (with `N` units in each data set). The data for half of the units should go in the `pdata` argument provided above. For the other `N` units, simply provide the counts for how many units appear in the untreated cohort plus each of the other `R` cohorts in this argument `indep_counts`. The benefit of doing this is that the standard error for the average treatment effect will be (asymptotically) exact instead of conservative. The length of `indep_counts` must equal 1 plus the number of treated cohorts in `pdata`. All entries of `indep_counts` must be strictly positive (if you are concerned that this might not work out, maybe your data set is on the small side and it's best to just leave your full data set in `pdata`). The sum of all the counts in `indep_counts` must match the total number of units in `pdata`. Default is NA (in which case conservative standard errors will be calculated if `q < 1`.)

`sig_eps_sq` (Optional.) Numeric; the variance of the row-level IID noise assumed to apply to each observation. See Section 2 of Faletto (2024) for details. It is best to provide this variance if it is known (for example, if you are using simulated data). If this variance is unknown, this argument can be omitted, and the variance will be estimated using the estimator from Pesaran (2015, Section 26.5.1) with ridge regression. Default is NA.

`sig_eps_c_sq` (Optional.) Numeric; the variance of the unit-level IID noise (random effects) assumed to apply to each observation. See Section 2 of Faletto (2024) for details. It is best to provide this variance if it is known (for example, if you are using simulated data). If this variance is unknown, this argument can be omitted, and the variance will be estimated using the estimator from Pesaran (2015, Section 26.5.1) with ridge regression. Default is NA.

`lambda.max` (Optional.) Numeric. A penalty parameter `lambda` will be selected over a grid search by BIC in order to select a single model. The largest `lambda` in the grid will be `lambda.max`. If no `lambda.max` is provided, one will be selected automatically. For `lambda <= 1`, the model will be sparse, and ideally all of the following are true at once: the smallest model (the one corresponding to `lambda.max`) selects close to 0 features, the largest model (the one corresponding to `lambda.min`) selects close to `p` features, `nlambda` is large enough so that models are considered at every feasible model size, and `nlambda` is small enough so that the computation doesn't become infeasible. You may want to manually tweak `lambda.max`, `lambda.min`, and `nlambda` to try to achieve these goals, particularly if the selected model size is very close to the model corresponding to `lambda.max` or `lambda.min`, which could indicate that the range of `lambda` values was too narrow. You can use the function outputs `lambda.max_model_size`, `lambda.min_model_size`, and `lambda_star_model_size` to try to assess this. Default is NA.

lambda.min (Optional.) Numeric. The smallest lambda penalty parameter that will be considered. See the description of lambda.max for details. Default is NA.

nlambda (Optional.) Integer. The total number of lambda penalty parameters that will be considered. See the description of lambda.max for details. Default is 100.

q (Optional.) Numeric; determines what L_q penalty is used for the fusion regularization. q = 1 is the lasso, and for 0 < q < 1, it is possible to get standard errors and confidence intervals. q = 2 is ridge regression. See Faletto (2024) for details. Default is 0.5.

verbose Logical; if TRUE, more details on the progress of the function will be printed as the function executes. Default is FALSE.

alpha Numeric; function will calculate (1 - alpha) confidence intervals for the cohort average treatment effects that will be returned in catt_df.

Value

A named list with the following elements:

att_hat The estimated overall average treatment effect for a randomly selected treated unit.

att_se If q < 1, a standard error for the ATT. If indep_counts was provided, this standard error is asymptotically exact; if not, it is asymptotically conservative. If q >= 1, this will be NA.

catt_hats A named vector containing the estimated average treatment effects for each cohort.

catt_ses If q < 1, a named vector containing the (asymptotically exact, non-conservative) standard errors for the estimated average treatment effects within each cohort.

cohort_probs A vector of the estimated probabilities of being in each cohort conditional on being treated, which was used in calculating att_hat. If indep_counts was provided, cohort_probs was calculated from that; otherwise, it was calculated from the counts of units in each treated cohort in pdata.

catt_df A dataframe displaying the cohort names, average treatment effects, standard errors, and 1 - alpha confidence interval bounds.

beta_hat The full vector of estimated coefficients.

treat_inds The indices of beta_hat corresponding to the treatment effects for each cohort at each time.

treat_int_inds The indices of beta_hat corresponding to the interactions between the treatment effects for each cohort at each time and the covariates.

sig_eps_sq Either the provided sig_eps_sq or the estimated one, if a value wasn't provided.

sig_eps_c_sq Either the provided sig_eps_c_sq or the estimated one, if a value wasn't provided.

lambda.max Either the provided lambda.max or the one that was used, if a value wasn't provided. (This is returned to help with getting a reasonable range of lambda values for grid search.)

`lambda.max_model_size`
: The size of the selected model corresponding `lambda.max` (for `q` <= `1`, this will be the smallest model size). As mentioned above, for `q` <= `1` ideally this value is close to 0.

`lambda.min`
: Either the provided `lambda.min` or the one that was used, if a value wasn't provided.

`lambda.min_model_size`
: The size of the selected model corresponding `lambda.min` (for `q` <= `1`, this will be the largest model size). As mentioned above, for `q` <= `1` ideally this value is close to `p`.

`lambda_star`
: The value of `lambda` chosen by BIC. If this value is close to `lambda.min` or `lambda.max`, that could suggest that the range of `lambda` values should be expanded.

`lambda_star_model_size`
: The size of the model that was selected. If this value is close to `lambda.max_model_size` or `lambda.min_model_size`, That could suggest that the range of `lambda` values should be expanded.

`X_ints`
: The design matrix created containing all interactions, time and cohort dummies, etc.

`y`
: The vector of responses, containing `nrow(X_ints)` entries.

`X_final`
: The design matrix after applying the change in coordinates to fit the model and also multiplying on the left by the square root inverse of the estimated covariance matrix for each unit.

`y_final`
: The final response after multiplying on the left by the square root inverse of the estimated covariance matrix for each unit.

`N`
: The final number of units that were in the data set used for estimation (after any units may have been removed because they were treated in the first time period).

`T`
: The number of time periods in the final data set.

`R`
: The final number of treated cohorts that appear in the final data set.

`d`
: The final number of covariates that appear in the final data set (after any covariates may have been removed because they contained missing values or all contained the same value for every unit).

`p`
: The final number of columns in the full set of covariates used to estimate the model.

Author(s)

Gregory Faletto

References

Faletto, G (2024). Fused Extended Two-Way Fixed Effects for Difference-in-Differences with Staggered Adoptions. *arXiv preprint arXiv:2312.05985*. https://arxiv.org/abs/2312.05985.
Pesaran, M. H. . Time Series and Panel Data Econometrics. Number 9780198759980 in OUP Catalogue. Oxford University Press, 2015. URL https://ideas.repec.org/b/oxp/obooks/9780198759980.html.

Examples

```
set.seed(23451)

library(bacondecomp)

data(divorce)

# sig_eps_sq and sig_eps_c_sq, calculated in a separate run of `fetwfe(),
# are provided to speed up the computation of the example
res <- fetwfe(
    pdata = divorce[divorce$sex == 2, ],
    time_var = "year",
    unit_var = "st",
    treatment = "changed",
    covs = c("murderrate", "lnpersinc", "afdcrolls"),
    response = "suiciderate_elast_jag",
    sig_eps_sq = 0.1025361,
    sig_eps_c_sq = 4.227651e-35,
    verbose = TRUE)

# Average treatment effect on the treated units (in percentage point
# units)
100 * res$att_hat

# Conservative 95% confidence interval for ATT (in percentage point units)

low_att <- 100 * (res$att_hat - qnorm(1 - 0.05 / 2) * res$att_se)
high_att <- 100 * (res$att_hat + qnorm(1 - 0.05 / 2) * res$att_se)

c(low_att, high_att)

# Cohort average treatment effects and confidence intervals (in percentage
# point units)

catt_df_pct <- res$catt_df
catt_df_pct[["Estimated TE"]] <- 100 * catt_df_pct[["Estimated TE"]]
catt_df_pct[["SE"]] <- 100 * catt_df_pct[["SE"]]
catt_df_pct[["ConfIntLow"]] <- 100 * catt_df_pct[["ConfIntLow"]]
catt_df_pct[["ConfIntHigh"]] <- 100 * catt_df_pct[["ConfIntHigh"]]

catt_df_pct
```

Index