Package 'LeaveOutKSS'

May 7, 2026

Type Package

Title Leave-Out Variance Component Estimation for Two-Way Fixed Effects Models

Version 0.1.0

Author Vahid Moghani [aut, cre]

Maintainer Vahid Moghani <contact@vahid-moghani.com>

Description Implements leave-out estimation of variance components in two-way fixed effects models as an 'R' translation of the original 'MATLAB' package of Kline, Saggio, and Solvsten (2020) <doi:10.3982/ECTA16410>. The package includes graph-based connected-set pruning, leave-out bias correction, leverage computation by exact and randomized algorithms, fixed effect estimation helpers, and companion model-fit summaries for matched worker-firm panels in the spirit of Abowd, Kramarz, and Margolis (1999) <doi:10.1111/1468-0262.00020>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports data.table, Matrix, igraph, sanic, parallel, utils, doParallel, foreach

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-04-21 19:02:34 UTC

Contents

`build_adj` *Build a Firm-to-Firm Mobility Adjacency Matrix*

Description

Constructs a symmetric sparse adjacency matrix of firm mobility links using worker transitions. Only movers contribute edges.

Usage

```
build_adj(id, firmid)
```

Arguments

`id`	Worker identifier vector.
`firmid`	Firm identifier vector.

Value

A sparse square adjacency matrix whose nonzero entries count observed worker moves between firms.

See Also

`connected_set()`, `pruning_unbal_v3()`

Examples

```
build_adj(
  id = c(1, 1, 2, 2, 3, 3),
  firmid = c(1, 2, 2, 3, 3, 3)
)
```

`connected_set` *Restrict a Panel to Its Largest Connected Set of Firms*

Description

Builds a mobility graph from worker moves across firms and keeps only the largest connected component of firms. This is the first graph-based trimming step used by the leave-out routines before leave-one-worker-out pruning.

Usage

```
connected_set(
  y,
  id,
  firmid,
  lagfirmid,
  controls,
  prov_indicator = rep(1, length(y)),
  progress = FALSE
)
```

Arguments

`y`	Numeric outcome vector.
`id`	Worker identifier vector.
`firmid`	Firm identifier vector.
`lagfirmid`	Lagged firm identifier vector, typically constructed within worker.
`controls`	Matrix of controls aligned with the observations.
`prov_indicator`	Optional provider indicator carried along for interface compatibility.
`progress`	Logical scalar indicating whether stage messages should be emitted.

Details

The graph is built from observed worker transitions between lagged and current firms. Firms not connected to the largest component are removed. The function relabels worker and firm identifiers internally but preserves the originals in the returned table.

Value

A list with two elements: `DT`, a `data.table` containing the restricted sample and original identifiers, and `DT_controls`, the correspondingly restricted controls.

See Also

`pruning_unbal_v3()`, `strongc_set()`, `build_adj()`

`fast_fe_est` *Fit a One-Way or Two-Way Fixed Effects Model*

Description

Solves a fixed effects model using conjugate gradients and returns fitted values and adjusted outcomes as an object. When `firmid` is omitted, the routine estimates a one-way worker fixed effects model. When `firmid` is supplied, it estimates a two-way worker-firm fixed effects model.

Usage

```
fast_fe_est(
  y,
  id,
  firmid = NULL,
  controls = NULL,
  csv_file = NULL,
  progress = FALSE
)
```

Arguments

`y`	Numeric outcome vector.
`id`	Worker identifier vector.
`firmid`	Optional firm identifier vector. If `NULL`, a one-way model is fitted.
`controls`	Optional matrix or vector of controls.
`csv_file`	Optional path for exporting the fitted values table as a `.csv` file.
`progress`	Logical scalar indicating whether stage progress messages should be emitted.

Details

This helper is useful when the goal is to recover fitted values and residualized outcomes rather than the leave-out variance decomposition. The returned fitted-values table includes `y_hat`, `y_adj`, and the original identifiers. When `csv_file` is supplied, that table is also written to disk.

Value

An object of class `"fast_fe_est_result"` containing the fitted values table, model metadata, and elapsed time.

See Also

leave_out_KSS(), rsquared_comp()

Examples

```
path <- system.file("extdata", "test.csv", package = "LeaveOutKSS")
dt <- data.table::fread(path, header = FALSE)

res <- fast_fe_est(
  y = dt[[4]],
  id = dt[[1]],
  firmid = dt[[2]],
  controls = cbind(year = dt[[3]])
)

print(res)
```

---

kss_quadratic_form *Evaluate Plug-In and Kline, Saggio, and Solvsten (KSS)-Corrected Quadratic Forms*

Description

Computes a covariance-like quadratic form from transformed coefficient vectors and subtracts the Kline, Saggio, and Solvsten (KSS) bias adjustment based on observation-specific variances and Bii weights.

Usage

```
kss_quadratic_form(sigma_i, A_1, A_2, beta, Bii)
```

Arguments

sigma_i	Vector of leave-out variance estimates.
A_1	Matrix used to transform the coefficient vector on the left side of the quadratic form.
A_2	Matrix used to transform the coefficient vector on the right side of the quadratic form.
beta	Estimated coefficient vector.
Bii	Vector of observation-specific bias terms for the target variance component.

Value

A named list with theta, the plug-in estimate, and theta_KSS, the bias-corrected estimate.

See Also

leave_out_KSS(), lincom_KSS()

Examples

```
A <- diag(2)
kss_quadratic_form(
  sigma_i = c(1, 2),
  A_1 = A,
  A_2 = A,
  beta = c(0.5, 1),
  Bii = c(0.1, 0.2)
)
```

---

leave_out_KSS *Leave-Out Bias-Corrected Variance Decomposition in a Two-Way Fixed Effects Model*

Description

Estimates plug-in and leave-out bias-corrected variance components for a two-way fixed effects model as part of the R translation of the original 'MATLAB' package of Kline, Saggio, and Solvsten (2020). The function starts from worker identifiers, firm identifiers, and an outcome, constructs the leave-one-worker-out connected set, optionally partials out controls, computes statistical leverages either exactly or via the Johnson-Lindenstrauss approximation (JLA), and returns decomposition summaries together with estimated worker and firm effects.

Usage

```
leave_out_KSS(
  y,
  id,
  firmid,
  controls = NULL,
  leave_out_level = "matches",
  type_algorithm = "JLA",
  simulations_JLA = 200,
  lincom_do = 0,
  Z_lincom = NULL,
  labels_lincom = NULL,
  csv_file = NULL,
  txt_file = NULL,
  paral = TRUE,
  Cd = 12345,
  progress = FALSE
)
```

Arguments

y	Numeric outcome vector.
id	Worker identifier vector.
firmid	Firm identifier vector.
controls	Optional matrix or vector of controls. When supplied, the function prepends an intercept internally and residualizes the outcome with respect to worker, firm, and control regressors before computing variance components.
leave_out_level	Character scalar. Use "matches" to leave out entire worker-firm matches or "obs" to leave out person-year observations.
type_algorithm	Character scalar. Use the randomized Johnson-Lindenstrauss approximation ("JLA") to the leverages or "exact" for the exact algorithm.
simulations_JLA	Integer number of random projections when type_algorithm = "JLA".
lincom_do	Integer flag equal to 0 or 1. When 1, the function also calls lincom_KSS() to regress estimated firm effects on user-supplied observables.
Z_lincom	Optional matrix of observables used by lincom_KSS() when lincom_do = 1. The main decomposition may collapse the estimation sample to match means, but the optional lincom step is still run on the post-pruning observation-level sample so observation weights are preserved.
labels_lincom	Optional labels for the columns of Z_lincom.
csv_file	Optional path for exporting the estimated effects table as a .csv file.
txt_file	Optional path for exporting a text summary of the decomposition.
paral	Logical scalar indicating whether leverage computation should use the parallel routine leverages_parallel().
Cd	Integer random seed passed to base::set.seed().
progress	Logical scalar indicating whether stage progress messages should be emitted.

Details

Relative to the original 'MATLAB' package, this implementation follows the same broad sequence: connected-set construction, leave-out pruning, optional residualization of controls, leverage computation, and bias correction of the variance of firm effects, the covariance of worker and firm effects, and the variance of worker effects.

The decomposition is based on an Abowd, Kramarz, and Margolis (1999; AKM)-style model with worker effects, firm effects, and optional controls. By default, the function leaves out matches, which corresponds to allowing unrestricted heteroskedasticity and arbitrary serial correlation within worker-firm matches, in line with the discussion in the original vignette. When leave_out_level = "obs", the correction is based on leaving out one person-year observation at a time.

When controls are supplied, the function first estimates their coefficients in the leave-out connected set and then works with the residualized outcome. When lincom_do = 1, the function additionally reports linear projections of firm effects on observables using lincom_KSS().

The input vectors must be sorted by worker identifier and, within worker, from earlier to later time periods before calling the function. When `controls` or `Z_lincom` are supplied, they must follow that same sorted row order.

The returned object is the primary estimation record. It stores the decomposition summaries, estimated worker and firm effects, and optional lincom output. When `csv_file` or `txt_file` are supplied, those summaries are also written to disk.

Value

An object of class `"leave_out_kss_result"` containing biased and bias-corrected estimates, estimated worker and firm effects, optional lincom results, sample summaries, and elapsed time.

References

Kline, P., Saggio, R., and Solvsten, M. (2020). Leave-out estimation of variance components. *Econometrica*, 88(5), 1859-1898.

Abowd, J. M., Kramarz, F., and Margolis, D. N. (1999). High wage workers and high wage firms. *Econometrica*, 67(2), 251-333.

See Also

`leave_out_KSS_fe()`, `rsquared_comp()`, `lincom_KSS()`, `leverages()`, `leverages_parallel()`

Examples

```
path <- system.file("extdata", "test.csv", package = "LeaveOutKSS")
dt <- data.table::fread(path, header = FALSE)
data.table::setorder(dt, V1, V3)

res <- leave_out_KSS(
  y = dt[[4]],
  id = dt[[1]],
  firmid = dt[[2]],
  simulations_JLA = 5,
  paral = FALSE,
  progress = FALSE
)

print(res)
```

`leave_out_KSS_fe` *Leave-Out Bias-Corrected Decomposition with Internally Expanded Fixed-Effect Controls*

Description

Variant of `leave_out_KSS()` that allows selected control columns to be treated as categorical regressors and expanded into dummy variables inside the routine. This mirrors the use case discussed in the original 'MATLAB' vignette where time effects or other discrete controls are partialled out before the leave-out variance decomposition is computed.

Usage

```
leave_out_KSS_fe(
  y,
  id,
  firmid,
  controls = NULL,
  absorb_col = NULL,
  leave_out_level = "matches",
  type_algorithm = "JLA",
  simulations_JLA = 200,
  lincom_do = 0,
  Z_lincom = NULL,
  labels_lincom = NULL,
  csv_file = NULL,
  txt_file = NULL,
  paral = TRUE,
  Cd = 12345,
  progress = FALSE
)
```

Arguments

`y`	Numeric outcome vector.
`id`	Worker identifier vector.
`firmid`	Firm identifier vector.
`controls`	Optional matrix or vector of controls. When supplied, the function prepends an intercept internally and residualizes the outcome with respect to worker, firm, and control regressors before computing variance components.
`absorb_col`	Optional integer vector identifying columns of `controls` that should be treated as categorical variables and expanded into dummies after the internal intercept column is added.
`leave_out_level`	Character scalar. Use `"matches"` to leave out entire worker-firm matches or `"obs"` to leave out person-year observations.
`type_algorithm`	Character scalar. Use the randomized Johnson-Lindenstrauss approximation (`"JLA"`) to the leverages or `"exact"` for the exact algorithm.
`simulations_JLA`	Integer number of random projections when `type_algorithm = "JLA"`.
`lincom_do`	Integer flag equal to `0` or `1`. When `1`, the function also calls `lincom_KSS()` to regress estimated firm effects on user-supplied observables.

`Z_lincom`	Optional matrix of observables used by `lincom_KSS()` when `lincom_do = 1`. The main decomposition may collapse the estimation sample to match means, but the optional lincom step is still run on the post-pruning observation-level sample so observation weights are preserved.
`labels_lincom`	Optional labels for the columns of `Z_lincom`.
`csv_file`	Optional path for exporting the estimated effects table as a `.csv` file.
`txt_file`	Optional path for exporting a text summary of the decomposition.
`paral`	Logical scalar indicating whether leverage computation should use the parallel routine `leverages_parallel()`.
`Cd`	Integer random seed passed to `base::set.seed()`.
`progress`	Logical scalar indicating whether stage progress messages should be emitted.

Details

The function follows the same workflow as `leave_out_KSS()` but modifies the control-adjustment step. When `absorb_col` is supplied, the corresponding columns are treated as categorical effects and expanded into dummy variables inside the leave-out connected set before residualization. This is convenient for year effects or other high-level discrete controls that are easier to supply in coded form than as a pre-built model matrix.

As with `leave_out_KSS()`, the input vectors must be sorted by worker identifier and, within worker, from earlier to later time periods before calling the function. Any supplied control columns must follow that same row order.

The rest of the decomposition logic is unchanged: the function constructs a leave-one-worker-out connected set, computes leverages, and returns plug-in and bias-corrected variance components together with estimated worker and firm effects. When `csv_file` or `txt_file` are supplied, those summaries are also written to disk.

Value

An object of class `"leave_out_kss_result"` containing biased and bias-corrected estimates, estimated worker and firm effects, optional lincom results, sample summaries, and elapsed time.

References

Kline, P., Saggio, R., and Solvsten, M. (2020). Leave-out estimation of variance components. *Econometrica*, 88(5), 1859-1898.

See Also

`leave_out_KSS()`, `rsquared_comp()`

Examples

```
path <- system.file("extdata", "test.csv", package = "LeaveOutKSS")
dt <- data.table::fread(path, header = FALSE)
data.table::setorder(dt, V1, V3)
```

```
res <- leave_out_KSS_fe(
  y = dt[[4]],
  id = dt[[1]],
  firmid = dt[[2]],
  controls = cbind(year = dt[[3]]),
  absorb_col = 1,
  simulations_JLA = 5,
  paral = FALSE,
  progress = FALSE
)

print(res)
```

`leverages` *Compute Statistical Leverages and Bias Terms*

Description

Computes the observation-level leverage quantities used in the Kline, Saggio, and Solvsten (KSS) bias correction, either exactly or with a Johnson-Lindenstrauss approximation (JLA).

Usage

```
leverages(X_fe, X_pe, X, xx, type_algorithm, scale, progress = FALSE)
```

Arguments

`X_fe`	Matrix used for the firm-effect variance component.
`X_pe`	Matrix used for the person-effect variance component.
`X`	Main design matrix.
`xx`	Crossproduct matrix `t(X) %*% X`.
`type_algorithm`	Character scalar, either `"exact"` or `"JLA"`.
`scale`	Number of random projections when `type_algorithm = "JLA"`.
`progress`	Logical scalar indicating whether leverage progress should be displayed.

Details

The exact branch solves one linear system per observation. The Johnson-Lindenstrauss approximation (JLA) branch follows the randomized projection logic described in the original vignette to approximate the same quantities at lower computational cost on large panels.

Value

A list with elements `Pii`, `Mii`, `correction_JLA`, `Bii_fe`, `Bii_cov`, and `Bii_pe`.

See Also

`leverages_parallel()`, `leave_out_KSS()`

leverages_parallel *Parallel Computation of Statistical Leverages and Bias Terms*

Description

Parallel version of `leverages()` using `foreach` and `doParallel`.

Usage

```
leverages_parallel(X_fe, X_pe, X, xx, type_algorithm, scale, progress = FALSE)
```

Arguments

`X_fe`	Matrix used for the firm-effect variance component.
`X_pe`	Matrix used for the person-effect variance component.
`X`	Main design matrix.
`xx`	Crossproduct matrix `t(X) %*% X`.
`type_algorithm`	Character scalar, either `"exact"` or `"JLA"`.
`scale`	Number of random projections when `type_algorithm = "JLA"`.
`progress`	Logical scalar indicating whether leverage progress should be displayed.

Details

The exact and Johnson-Lindenstrauss approximation (JLA) branches mirror `leverages()`, but the repeated linear solves are distributed across worker processes. This routine is intended for larger problems where the leverage stage dominates runtime.

Value

A list with the same elements returned by `leverages()`.

See Also

`leverages()`, `leave_out_KSS()`

lincom_KSS — *Linear Projections of Estimated Firm Effects with Kline, Saggio, and Solvsten (KSS) Standard Errors*

Description

Regresses transformed fixed effects on observables and reports both naive and Kline, Saggio, and Solvsten (KSS)-corrected standard errors. This corresponds to the "lincom" discussion in the original vignette on regressing firm effects on observables.

Usage

```
lincom_KSS(y, X, Z, Transform, sigma_i, labels = NULL)
```

Arguments

`y`	Outcome vector used to estimate the original model.
`X`	Design matrix used to estimate the fixed effects model.
`Z`	Matrix of observables used in the linear projection.
`Transform`	Matrix that maps model coefficients into the fixed effect of interest, typically firm effects.
`sigma_i`	Observation-specific leave-out variance estimates.
`labels`	Optional labels for the columns of `Z`.

Value

An object of class `"lincom_kss_result"` containing a results table with coefficient estimates, naive standard errors, KSS-corrected standard errors, and t statistics.

See Also

`leave_out_KSS()`, `kss_quadratic_form()`

print.fast_fe_est_result — *Print a Fixed Effects Fit Result*

Description

Print a Fixed Effects Fit Result

Usage

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

Arguments

x	A result returned by fast_fe_est().
...	Unused.

Value

x, invisibly.

---

print.leave_out_kss_result *Print a LeaveOutKSS Decomposition Result*

Description

Print a LeaveOutKSS Decomposition Result

Usage

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

Arguments

x	A result returned by leave_out_KSS() or leave_out_KSS_fe().
...	Unused.

Value

x, invisibly.

---

print.lincom_kss_result *Print a Lincom Result*

Description

Print a Lincom Result

Usage

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

Arguments

x	A result returned by lincom_KSS().
...	Unused.

Value

x, invisibly.

---

print.rsquared_comp_result *Print an R-Squared Comparison Result*

Description

Print an R-Squared Comparison Result

Usage

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

Arguments

x	A result returned by rsquared_comp().
...	Unused.

Value

x, invisibly.

---

pruning_unbal_v3 *Prune to a Leave-One-Worker-Out Connected Set*

Description

Iteratively removes articulation workers from the worker-firm mobility graph until the remaining sample stays connected after dropping any single worker. This implements the leave-one-worker-out connectivity requirement used by the main Kline, Saggio, and Solvsten (KSS) routines.

Usage

```
pruning_unbal_v3(
  y,
  firmid,
  id,
  id_old,
  firmid_old,
  controls,
  prov_indicator = rep(1, length(y)),
  progress = FALSE
)
```

Arguments

y	Numeric outcome vector.
firmid	Firm identifier vector.
id	Worker identifier vector.
id_old	Original worker identifiers.
firmid_old	Original firm identifiers.
controls	Matrix of controls aligned with the observations.
prov_indicator	Optional provider indicator carried along with the sample.
progress	Logical scalar indicating whether iterative pruning progress should be emitted.

Details

The routine constructs a bipartite worker-firm graph for movers, identifies articulation workers, removes them, and recomputes the largest connected component until no articulation worker remains.

Value

A list containing the pruned outcome, identifiers, controls, and provider indicator.

See Also

connected_set(), build_adj(), leave_out_KSS()

---

rsquared_comp *Compare Two-Way Fixed Effects and Saturated-Model R-Squared Values*

Description

Computes goodness-of-fit summaries for a two-way fixed effects model and for a saturated worker-firm interaction model on the same sample. The function is intended as a diagnostic companion to the leave-out decomposition routines and follows the same basic data-preparation conventions.

Usage

```
rsquared_comp(
  y,
  id,
  firmid,
  controls = NULL,
  txt_file = NULL,
  progress = FALSE
)
```

Arguments

y	Numeric outcome vector.
id	Worker identifier vector.
firmid	Firm identifier vector.
controls	Optional matrix or vector of additional controls.
txt_file	Optional path for exporting a text summary of the comparison.
progress	Logical scalar indicating whether stage progress messages should be emitted.

Details

The two-way fixed effects model includes worker effects, firm effects, and optional controls. The saturated model replaces separate worker and firm effects with worker-firm interaction indicators. Comparing the two summaries can be useful when evaluating how much additional fit is obtained by moving from the standard Abowd, Kramarz, and Margolis (1999; AKM) specification to a fully saturated match design.

Value

An object of class "rsquared_comp_result" containing a summary table for the two fitted models and the elapsed time.

See Also

leave_out_KSS(), leave_out_KSS_fe(), fast_fe_est()

Examples

```
path <- system.file("extdata", "test.csv", package = "LeaveOutKSS")
dt <- data.table::fread(path, header = FALSE)

res <- rsquared_comp(
  y = dt[[4]],
  id = dt[[1]],
  firmid = dt[[2]],
  progress = FALSE
)
```

```
print(res)
```

`sigma_for_stayers` *Approximate Leave-Out Variance Terms for Stayers*

Description

Computes the stayer-specific adjustment used when the main decomposition is performed at the match level. In that case, the current implementation uses a leave-one-observation-out style adjustment for stayers, following the approximation discussed in the original vignette.

Usage

```
sigma_for_stayers(y, id, firmid, peso, b)
```

Arguments

`y`	Outcome vector in person-year space.
`id`	Worker identifier vector in collapsed match space.
`firmid`	Firm identifier vector in collapsed match space.
`peso`	Match weights used to expand back to person-year space.
`b`	Estimated coefficient vector from the worker-firm fixed effects regression.

Value

A vector of averaged stayer variance adjustments at the match level.

See Also

`leave_out_KSS()`, `leave_out_KSS_fe()`

`strongc_set` *Restrict a Panel to Firms Above a Minimum Graph Degree Threshold*

Description

Graph-based trimming helper that keeps firms whose degree in the mobility graph is at least `min_degree`. This is a stronger restriction than the basic connected-set filter and can be useful when the analyst wants a denser firm network.

Usage

```
strongc_set(y, id, firmid, controls, min_degree = 1, progress = FALSE)
```

Arguments

y	Numeric outcome vector.
id	Worker identifier vector.
firmid	Firm identifier vector.
controls	Matrix of controls aligned with the observations.
min_degree	Minimum graph degree required for a firm to remain in the sample.
progress	Logical scalar indicating whether graph summary messages should be emitted.

Value

A list with DT and DT_controls, analogous to connected_set().

See Also

connected_set()

---

summary.leave_out_kss_result

Summarize a LeaveOutKSS Decomposition Result

Description

Summarize a LeaveOutKSS Decomposition Result

Usage

```
## S3 method for class 'leave_out_kss_result'
summary(object, ...)
```

Arguments

object	A result returned by leave_out_KSS() or leave_out_KSS_fe().
...	Unused.

Value

object, invisibly.

Index