NET WORK LISER

Discussion Paper Series

IZA DP No. 19053

October 2026

fevc: Leave-Out Bias-Corrected Variance Decompositions in Stata

Johannes Schmieder
Boston University
and IZA@LISER

The IZA Discussion Paper Series (ISSN: 2365-9793) ("Series") is the primary platform for disseminating research produced within the framework of the IZA@LISER Network, an unincorporated international network of labour economists coordinated by the Luxembourg Institute of Socio-Economic Research (LISER). The Series is operated by LISER, a Luxembourg public establishment (établissement public) registered with the Luxembourg Business Registers under number J57, with its registered office at 11, Porte des Sciences, 4366 Esch-sur-Alzette, Grand Duchy of Luxembourg.

Any opinions expressed in this Series are solely those of the author(s). LISER accepts no responsibility or liability for the content of the contributions published herein. LISER adheres to the European Code of Conduct for Research Integrity. Contributions published in this Series present preliminary work intended to foster academic debate. They may be revised, are not definitive, and should be cited accordingly. Copyright remains with the author(s) unless otherwise indicated.

ISSN: 2365-9793

fevc: Leave-Out Bias-Corrected Variance Decompositions in Stata*

Abstract

Variance decompositions from regressions with worker and firm fixed effects are widely used to measure heterogeneity and sorting in linked data. Variances computed directly from estimated effects overstate heterogeneity because they include estimation noise, and the bias can be large when few workers move between firms. This article introduces fevc, a Stata command that implements the leave-out bias correction of Kline, Saggio, and Sølvsten (2020, "Leave-Out Estimation of Variance Components," Econometrica 88: 1859–1898). fevc reports corrected variances of worker and firm effects, their covariance, and the variance of their sum. Researchers choose the unit left out, the treatment of controls, and the weights that define the population being described. The command also reports standard errors for projections of fixed effects on observed characteristics. It has two backends: a source-only Mata implementation that runs wherever Stata runs, and a parallel Rust plugin for large datasets. Calculations based on random projections report Monte Carlo standard errors that measure their numerical precision. In a public extract of Italian social-security records, the correction halves the estimated variance of firm effects and reverses the sign of the worker–firm covariance. In synthetic benchmarks with 6.4 million observations and 28 threads, the Rust backend is 1.3 to 3.5 times as fast as the MATLAB package of Kline, Saggio, and Sølvsten. Under match deletion, its measured peak memory is about 30 percent of the MATLAB package's.

JEL classification

C13, C23, C55, J31, J62

Keywords

fevc, two-way fixed effects, variance decomposition, leave-out estimation, limited-mobility bias, Mata, Rust, randomized linear algebra

Corresponding author

Johannes Schmieder
johannes@bu.edu

* I thank Raffaele Saggio for helpful comments. I used OpenAI's ChatGPT and Codex to assist with software development, testing, and the preparation and revision of this manuscript. I take full responsibility for the code, analysis, and text, including any remaining errors.

1 Introduction

How much of the variation in earnings reflects differences between workers, differences between firms, and the allocation of workers to firms? A regression with worker and firm fixed effects provides a natural framework for answering this question. Researchers use the variances of the two sets of effects and their covariance to describe wage heterogeneity and sorting (Abowd et al., 1999; Card et al., 2013). Similar questions arise in linked data on patients and physicians, brands and retailers, benefit take-up across employers, distributional firm wage premia, and authors and institutions (Kwok, 2024; Clark et al., 2025; Lachowska et al., 2025; Bérgolo et al., 2026; Dahis, 2026).

The variance of estimated effects, however, combines true heterogeneity with estimation noise. Even when the regression coefficients are unbiased, squaring them biases their second moments upward. Estimation errors in worker and firm effects are also correlated, which distorts their estimated covariance. These problems can be severe when few workers move between firms: a large number of observations does not ensure that the comparisons needed to separate worker and firm effects are precise (Andrews et al., 2008; Kline et al., 2020; Bonhomme et al., 2023).

Kline, Saggio, and Sølvsten (2020), hereafter KSS, develop leave-out bias corrections for quadratic forms in high-dimensional linear models. Their approach estimates the contribution of noise using a regression that excludes the observation whose error is being evaluated. Subtracting that contribution removes the plug-in bias under conditional mean zero and the appropriate independence and identification assumptions. Leaving out an entire worker–firm match extends the calculation to errors that are correlated within matches.

Stata users can already estimate two-way fixed-effect regressions on large linked datasets, for example with `felsdvreg` (Cornelissen, 2008), the procedure of Guimarães and Portugal (2010), or `reghdfe` (Correia, 2014). The implementations of the leave-out correction compared in this article, however, are written in Matlab, Julia, R, or Python (Saggio, 2024; High Dimensional Economics Lab, 2024; Moghani, 2026; Lamadon and Oppenheimer, 2022). Using them from Stata means exporting the data, rebuilding the estimation sample in another language, and importing the results. Sample construction is part of the estimator, because the correction requires a sample that remains connected after each deletion, and splitting it across two environments makes it harder to keep track of the population that the estimates describe.

This article presents `fevc`, a Stata command for variance decompositions in the KSS framework. A single command constructs the leave-out sample, estimates the model, and reports plug-in, bias, and corrected values for the variance of worker effects, the variance of firm effects, their covariance, and the variance of their sum. Stored results and postestimation commands document the retained sample and the computation. The command makes the choices that define the estimand explicit. Researchers decide whether single observations or entire worker–firm matches are left out, may declare their own dependence blocks within worker–firm pairs, choose whether controls are re-estimated in each leave-out fit, and keep the frequency weights that enter the regression separate from the target weights that define the population being described. Section S8 of the technical supplement compares these conventions with those of other implementations.

The command has two computational backends. A Rust plugin uses sparse iterative solvers, combinatorial multigrid preconditioning, and parallel computation to handle large datasets. A Mata backend, distributed entirely as source code, runs within Stata and serves computing environments whose policies permit ado and Mata code but restrict native binaries. By default, both backends use the random-projection approximation of KSS. Because its results depend on random draws,

`fevc` reports Monte Carlo standard errors that measure this numerical uncertainty. The command also reports leave-out standard errors for projections of fixed effects on observed characteristics, and it offers experimental inference for the variance components themselves, including decompositions that pool movers and eligible stayers.

A public extract of Italian social-security records illustrates the command. The correction attributes about half of the plug-in variance of firm effects to estimation noise and reverses the sign of the worker–firm covariance. In synthetic graphs with up to 6.4 million observations and in two confidential samples of Italian administrative data, the Rust backend is faster than the KSS Matlab package in every tested configuration. Its measured peak memory is lower in every synthetic match-deletion configuration and, at four or more threads, in both confidential samples.

Section 2 develops the econometric framework and the choices that define the estimand. Section 3 describes installation, syntax, options, computation, and stored results. Section 4 works through the public example, and section 5 examines bias, numerical precision, and computational performance. An appendix collects the essential formulas and conditions, and a technical supplement develops the proofs, numerical methods, experimental inference, and supporting evidence (Schmieder, 2026b).

2 Econometric framework

2.1 The model and the variance targets

Consider an observation $i = 1, \ldots, n$, such as a worker observed in a particular year. Let $w(i)$ identify its worker and $j(i)$ its firm. The worker–firm model is

$$y_i = \alpha_{w(i)} + \psi_{j(i)} + z_i'\gamma + \varepsilon_i, \tag{1}$$

where y_i is the outcome, α_w and ψ_j are worker and firm fixed effects, z_i contains observed controls with coefficients γ, and ε_i is an error with conditional mean zero. The worker and firm effects are identified only up to a normalization, such as a zero average firm effect, and only within a connected set of workers and firms (section 2.4). They therefore describe differences within the estimation sample.

Let $\rho_i > 0$ be the weight that observation i receives in describing the population of interest, with $\sum_i \rho_i = 1$; equal weights give $\rho_i = 1/n$. Write $\bar{\alpha} = \sum_i \rho_i \alpha_{w(i)}$ and $\bar{\psi} = \sum_i \rho_i \psi_{j(i)}$. The four targets are

$$V_\alpha = \sum_i \rho_i (\alpha_{w(i)} - \bar{\alpha})^2, \qquad V_\psi = \sum_i \rho_i (\psi_{j(i)} - \bar{\psi})^2, \tag{2}$$

$$C_{\alpha\psi} = \sum_i \rho_i (\alpha_{w(i)} - \bar{\alpha})(\psi_{j(i)} - \bar{\psi}), \qquad V_{\alpha+\psi} = V_\alpha + V_\psi + 2C_{\alpha\psi}. \tag{3}$$

These quantities describe the weighted dispersion of the effects across observations in the sample, not the sampling variance of an estimator. The covariance measures sorting: whether workers with high worker effects tend to be employed at firms with high firm effects. Because each target is centered, none depends on the arbitrary normalization of the effects.

The fourth target describes the variation in the combined worker and firm contribution to the outcome. With controls, it excludes their contribution and its covariance with the fixed effects, so it is the variance of $\alpha_{w(i)} + \psi_{j(i)}$ rather than of y_i. Because all four targets use the same sample and weights, their adding-up identity holds for any choice of population or weights.

2.2 Why a leave-out correction is needed

Replacing the effects by their least-squares estimates gives the *plug-in* decomposition. Because the estimates contain sampling error, their dispersion reflects both true heterogeneity and noise. Plug-in variances are therefore biased upward, and correlated estimation errors distort the worker–firm covariance. These biases can be severe when mobility is limited, because differences between firm effects are identified only through the few workers who move between firms (Andrews et al., 2008; Kline et al., 2020).

Throughout, expectations condition on the retained sample and the regression design, together with the weights, deletion units, and projection covariates introduced below; write $\mathcal{D}$ for this information. The error assumptions are imposed in the retained population. To see how the KSS correction works, suppose first that observations are independent and have unit regression weight. Write the model as $y = X\beta + \varepsilon$, where row i of X is x_i' and β collects the effects and controls under a normalization that identifies them. Let $S_{xx} = X'X$. Each target can be written as a quadratic form $\theta_t = \beta' A_t \beta$, where the symmetric matrix A_t selects the relevant effects, centers them, and applies the target weights. If σ_i^2 is the variance of ε_i, the bias of the plug-in estimate is

$$\mathbb{E}(\widehat{\beta}' A_t \widehat{\beta}) - \theta_t = \sum_i B_{t,ii}\sigma_i^2, \qquad B_{t,ii} = x_i' S_{xx}^{-1} A_t S_{xx}^{-1} x_i. \tag{4}$$

The weight $B_{t,ii}$ determines how much the error variance of observation i contributes to the bias in the plug-in estimate of target t. Correcting the bias thus requires an estimate of every σ_i^2. The squared full-sample residual is a poor candidate, because the fit absorbs part of each error, most of all for observations that carry much of the information about their own effects.

KSS instead pair the outcome with a residual from a fit that excludes the observation. Let $\widehat{\beta}_{-i}$ be the estimate computed without observation i. Because this fit uses only other observations, it is unbiased and independent of ε_i, and conditional mean zero implies

$$\widehat{\sigma}_i^2 = y_i(y_i - x_i'\widehat{\beta}_{-i}), \qquad \widehat{\theta}_t^{\mathrm{KSS}} = \widehat{\beta}' A_t \widehat{\beta} - \sum_i B_{t,ii}\widehat{\sigma}_i^2, \tag{5}$$

with $\mathbb{E}(\widehat{\sigma}_i^2 \mid \mathcal{D}) = \sigma_i^2$, provided the regression remains identified after the deletion. The corrected quadratic form is therefore unbiased for θ_t (Kline et al., 2020). The product deliberately combines the outcome itself with a leave-out residual; replacing it with a squared full-sample residual would reintroduce the fitting bias that the correction is designed to remove.

The correction does not require n separate regressions. Let $P_{ii} = x_i' S_{xx}^{-1} x_i$ be the leverage of observation i, which measures how strongly its own outcome influences its fitted value, and let $\widehat{e}_i = y_i - x_i'\widehat{\beta}$. The usual deleted-residual identity is

$$y_i - x_i'\widehat{\beta}_{-i} = \frac{\widehat{e}_i}{1 - P_{ii}}. \tag{6}$$

The calculation therefore needs only the full-sample fit and, for each observation, the leverage P_{ii} and the loading $B_{t,ii}$. When $P_{ii} = 1$, the observation is essential for identification, and deleting it leaves the regression unidentified. Leverages close to one can produce large leave-out residuals; they mark observations that carry much of the information needed to separate the effects.

An unbiased variance estimator need not be positive in every sample. The product $\widehat{\sigma}_i^2$ can be negative, and so can a corrected variance. A negative estimate should be interpreted in light of its sampling and numerical precision, not as evidence of a negative population variance.

2.3 Centering the variance estimates

The published KSS estimator uses the uncentered product in (5), and the unbiasedness argument above applies to it. The KSS Matlab package instead subtracts the outcome mean from the first factor, using $(y_i - \bar{y})(y_i - x_i'\widehat{\beta}_{-i})$ (Saggio, 2024). Centering can substantially improve precision: the uncentered product scales the leave-out residual by the level of y_i, so when the outcome mean is far from zero, each product is noisy even though its expectation is correct. The cost is a possible bias: $\bar{y}$ depends on the same errors that enter the leave-out residual, so the centered product no longer has expectation exactly σ_i^2. With homoskedastic errors, the bias is zero whenever the regressors span a constant, as the fixed effects do. Under heteroskedasticity it generally is not, and a large sample alone does not establish that it is negligible.

`fevc` offers three conventions. The default, `centering(mean)`, subtracts the mean, over all retained observations, of the outcome that enters the correction, which under `nuisance(fixedoffset)` is net of the controls. It is used for point estimates, projections, and component inference. Its mean can differ from the Matlab package's when matches contain unequal numbers of observations. The option `centering(none)` uses the published uncentered product. For point estimates, `centering(corrected)` keeps the centered product but removes the bias it introduces. When the leave-out quantities are computed exactly and the controls are either re-estimated jointly or have a genuinely known contribution, this restores exact unbiasedness under the error assumptions above.[1] Removing the bias need not reduce mean squared error, because the correction can increase variance (Technical Supplement Section S6.4). Whether the default is adequate depends on the design and the target.

2.4 Deletion units and the leave-one-out set

The argument above assumes that errors are independent across observations, so that leaving observation i out makes $\widehat{\beta}_{-i}$ independent of its error. This is *observation deletion*. In panel data the assumption can be restrictive: the errors of a worker observed repeatedly at the same firm may be correlated, and deleting one observation would then leave correlated observations in the leave-out fit. KSS therefore also allow dependence within blocks of observations. *Match deletion* leaves out an entire worker–firm match when constructing the leave-out fit. It allows arbitrary error covariance within each match but requires independence across matches, including different matches of the same worker.

More generally, let g denote a declared block of observations, and let y_g, X_g, and $\widehat{\beta}_{-g}$ be its outcome vector, its design rows, and the coefficient estimate after deleting it. The cross-product $y_g(y_g - X_g\widehat{\beta}_{-g})'$ then estimates the error covariance within the block. Under independence across blocks and the same conditional-mean-zero and identification assumptions, the correction is unbiased. With $B_{t,g} = X_g S_{xx}^{-1} A_t S_{xx}^{-1} X_g'$, the corrected quadratic form is

$$\widehat{\theta}_t^{\mathrm{KSS}} = \widehat{\beta}' A_t \widehat{\beta} - \sum_g y_g' B_{t,g}(y_g - X_g\widehat{\beta}_{-g}). \tag{7}$$

For a single-observation block this reduces to (5). For a match with several observations, it includes

[1] The correction also requires a rank condition: for observation deletion, at least two residual degrees of freedom. Matches with constant regressors and general blocks have the conditions stated in the appendix (Schmieder, 2026b, Technical Supplement, Section S2). The result does not extend to the randomized calculation with a finite number of probes or to controls estimated once on the same sample and held fixed.

the within-match cross-products needed to remove the contribution of correlated errors. This is a correction to the point estimate itself; clustering a standard error afterwards would not accomplish it. Section S2 derives the block formula and its weighted representation.

Differences in worker and firm effects in (1) are identified only within a connected set of the worker–firm network, and researchers therefore commonly restrict the analysis to the largest connected set (LCS). The LCS, however, need not remain connected after an observation or match is deleted. If a match provides the only link between two groups of firms, for example, deleting it makes their relative firm effects unidentified. KSS therefore construct a *leave-one-out (LOO) set*: a connected sample on which the required effects remain identified after every deletion used in the correction (Kline et al., 2020). This stronger requirement can exclude observations that the LCS would retain.

The choice of deletion unit thus determines both the assumption about error dependence and the sample on which the correction can be computed. Controls can also make a leave-out regression unidentified even when the graph stays connected. Because the weights are renormalized after the sample is constructed, the estimates describe the retained population, not the input data.

In `fevc`, deletion blocks need not coincide with the worker–firm pairs used to estimate the fixed effects. The KSS Matlab code forms matches from its worker and firm identifiers, whereas `fevc` accepts a separate `deletionid()` nested within each worker–firm pair. A researcher who groups small firms to estimate a common effect for each group, for example, may retain observations that separate firm effects would lose from the LOO set. The firm identifier then denotes the group, while `deletionid()` can still identify the original worker × firm matches as deletion units. Several such blocks can then occupy one worker–group cell, each a separate deletion unit.[2] Independence must be defensible across these blocks, and a block may not span different estimation pairs.

2.5 Weights and controls

`fevc` distinguishes two kinds of weights. Frequency weights determine each stored row's contribution to estimating the regression coefficients. A positive integer frequency f_i indicates that row i represents f_i observations with identical outcomes and regressors, contributing $f_i(y_i - x_i'\beta)^2$ to the least-squares objective. Identical payroll records in the same deletion block can thus be stored once without changing the estimates or the correction; duplicating a record does not create independent information. Target weights determine each row's contribution to the population whose variances and covariances are reported. The option `targetweight()` specifies the row's total target weight a_i, giving it share $\rho_i = a_i / \sum_t a_t$ in the decomposition. Without `targetweight()`, $a_i = f_i$, so each underlying observation receives equal weight. In worker-year data with one row per observation, this default gives workers with longer histories more weight. Setting $a_i = 1/T_{w(i)}$, where T_w is the number of retained observations on worker w, gives each worker equal total weight while every worker-year still enters the regression equally. Target weights change both the plug-in decomposition and its bias correction, but not the regression fit. Section S2 gives the exact weighted compression identities.

Researchers typically add controls $z_i'\gamma$ to (1), such as experience profiles or year effects. In the general KSS framework, the controls are estimated jointly with the fixed effects, and their coefficients

[2]With match deletion and an explicit `deletionid()`, the distinction between movers and stayers in section 2.6 counts declared blocks rather than estimation firms. A worker with two blocks in one firm group can enter the mover sample, subject to the connectivity and rank checks, and each block is then deleted whole. A worker with a single block is treated as a stayer.

are re-estimated after each deletion. When the controls are estimated from information spread across many observations, deleting one observation or match may change $\widehat{\gamma}$ little. This motivates a simpler approximation, used by the main KSS Matlab routine: estimate the full model once and apply the leave-out correction to $y_i - z_i'\widehat{\gamma}$, holding the fitted control contribution fixed. `fevc` offers this approximation as `nuisance(fixedoffset)`. The default, `nuisance(joint)`, instead re-estimates the control coefficients in each leave-out regression and applies the KSS correction to the full model. For the uncentered estimator, joint re-estimation preserves the finite-sample unbiasedness argument without requiring the effect of deletion on the control coefficients to be negligible.

2.6 Movers and stayers

Workers observed at more than one firm are *movers*; those observed at only one firm are *stayers*. The distinction matters for the variance of worker effects under match deletion. When each worker–firm pair forms one deletion block, a stayer has a single block, so deleting it removes every observation on that worker. The worker's effect is then unidentified in the leave-out regression, and its contribution to the worker-effect variance cannot be corrected. Without further restrictions on dependence within matches, the match-deletion correction for V_α therefore applies only to movers (Kline et al., 2020). With an explicit `deletionid()`, the labels count declared deletion units instead of firms, so a worker at one firm can count as a mover (section 2.4).

With `stayers(movers)`, `fevc` estimates the model and all four targets on the retained mover sample. The default, `stayers(both)`, adds workers who are stayers in the input data, are employed at a firm in the retained mover sample, and have at least two underlying observations. The model is re-estimated on the combined sample, and the effects are centered with its pooled target weights. The decomposition therefore reflects differences in mean effects between movers and stayers as well as dispersion within each group. Workers who become stayers only because some of their matches are removed during sample construction are not added back.

Under match deletion with `stayers(both)`, the correction in (7) deletes mover matches whole but deletes stayer observations one at a time. Deleting one of a stayer's observations leaves the others to identify the worker's effect. The combined correction requires independence across all deletion units, that is, across mover matches and across the individual observations of each stayer. Errors within a mover match remain unrestricted, but serial correlation in a stayer's errors is ruled out. Including stayers thus changes both the population described and the assumptions needed for the correction. This article calls this partition the *match/stayer-hybrid* correction. It follows the KSS Matlab package, whose match-deletion routine also corrects stayers at the observation level (Saggio, 2024). Structured component inference uses the same partition, with the additional conditions described in section 3.9.

Observation deletion also defaults to `stayers(both)`, but it deletes the observations of movers and eligible stayers alike one at a time. With either deletion mode, `stayers(movers)` restricts the sample to movers.

2.7 Projecting fixed effects

Researchers often want to relate estimated effects to observed characteristics, for example by regressing firm wage effects on log firm size. Under the conditional-mean-zero assumption of the fixed-effect regression, the slope of such a regression is an unbiased estimate of the corresponding

projection slope. Its conventional standard error is unreliable, however, because the estimated effects are unequally precise and their errors are correlated. `fevc` reports the same slope as a separate regression on the retained sample with the same weights, together with leave-out standard errors that account for estimation error in the fixed effects, conditional on the design and the projection covariates. By default, these standard errors use the same mean-centered leave-out products as the point estimates. The exact leave-out covariance estimate is unbiased without centering and with jointly estimated controls or a genuinely known control offset; under mean centering it generally is not. Normal intervals further require that the slope be approximately normal and that its standard error be consistent, which includes negligible effects of centering and of an estimated control offset where they apply. The appendix states the formula, and Technical Supplement Section S4.1 gives the derivation.

3 Using `fevc`

3.1 Installation and requirements

`fevc` requires Stata 18 or later. The package, including its help file and the precompiled native plugins, can be installed from GitHub:

```
net install fevc, replace ///
    from("https://raw.githubusercontent.com/johannes-schmieder/fevc/main/")
```

The package contains the ado and Mata code and Rust plugins for macOS on Apple silicon and Intel, Linux on x86-64, and Windows on x86-64. The command loads the plugin for the current platform, so no compiler or separate download is needed. When no compatible plugin is available, the default `backend(auto)` uses the Mata implementation. Parallel computation does not require Stata/MP, because the Rust backend sets its own thread count through `nativethreads()`. The help file `help fevc` documents every option and includes runnable examples on simulated data.

3.2 Syntax

`fevc` *depvar* [*controls*] [*if*] [*in*] [*weight*]`, worker(`*varname*`) firm(`*varname*`)` [*options*]

The outcome *depvar* is followed by optional numeric controls, which may use factor-variable notation such as `i.year`. Only frequency weights, `[fw=`*varname*`]`, are allowed. The `if` and `in` qualifiers restrict the input before the command constructs the leave-out sample. The principal options are:

Option	Description (default)
Required	
`worker(`*varname*`)`	identifier of the first effect dimension
`firm(`*varname*`)`	identifier of the second effect dimension
Model and estimand	
`deletion(match|observation)`	unit left out in the correction (`match`)
`deletionid(`*varname*`)`	dependence blocks nested within worker–firm pairs
`stayers(both|movers)`	include eligible stayers or movers only (`both`)
`nuisance(joint|fixedoffset)`	re-estimate controls in deleted fits or hold them fixed (`joint`)
`centering(mean|none|corrected)`	outcome centering in the leave-out products (`mean`)
`targetweight(`*varname*`)`	row masses defining the decomposed population (frequency)
Calculation	

`algorithm(jla|exact|auto)`	randomized or deterministic calculation (`jla`)
`probes(#)`	random directions per stage (200)
`seed(#)`	estimation seed (8675309)
`mcse(all|off)`	numerical Monte Carlo standard errors (`all`)
`backend(auto|mata|rust)`	implementation (`auto`: Rust when available)
Numerical settings	
`rng(auto|stata|counter_v1)`	random-number generator (`auto`)
`engine(auto|generic|compressed)`	internal representation (`auto`)
`preconditioner(auto|diagonal|cmg)`	iterative-solver preconditioner (`auto`)
`exact_limit(#)`	exact calculation dimension limit (500)
`batch(auto|#)`	linear systems solved together (`auto`)
`nativethreads(#)`	cap on Rust threads (`c(processors)`)
`tolerance(#)`, `maxiter(#)`	solver tolerance and iteration limit (automatic; 10,000)
`memory_gib(#)`, `memorycheck(warn|error|off)`	optional memory budget and its policy (none; `warn`)
Projection	
`project(`*varlist*`)`	covariates for a fixed-effect projection
`projecteffect(worker|firm)`	effect to project (required with `project()`)
`projectweight(frequency|target)`	projection weights (`frequency`)
Experimental component inference	
`inference(none|highrank|q1)`	experimental component intervals (`none`); see Supplement S5
`inferencemodel(`*mode*`)`	explicitly select `structured_common` or `structured_leverage` for structured inference
Reporting	
`nolog`, `verbose`, `nodisplay`	suppress progress, add detail, or suppress output

As a minimal example, the command

```
fevc log_wage i.year, worker(worker_id) firm(firm_id)
```

estimates the model with year controls and reports the corrected decomposition. It uses the defaults listed above: match deletion with eligible stayers, joint re-estimation of the controls, mean centering, and the randomized calculation with 200 probes. The option names `worker()` and `firm()` follow the earnings application; they can identify any two linked sets of units. Section 4 shows and interprets the output.

3.3 Options

The options for the model and estimand determine what the decomposition measures, and section 2 explains each of them. Two details matter in practice: `centering(corrected)` applies to point estimates only, and an explicit `targetweight()` replaces the default frequency mass rather than multiplying it.

Among the calculation options, `algorithm(auto)` chooses the exact calculation when the number of identified parameters is at most `exact_limit()`. `seed()` makes the result reproducible, and the command restores the caller's random-number state afterwards. Turning off the Monte Carlo standard errors with `mcse(off)` leaves the point estimates unchanged, and an explicitly requested backend is never replaced silently.

For point estimates, most users can leave the numerical settings at their defaults and check the convergence diagnostics and the Monte Carlo standard errors; projections and component inference need the additional settings described below. Rust and Mata use different random-number generators by default, so the same seed gives different random directions in the two backends; requesting a generator explicitly also fixes the backend. The compressed engine is specialized for

match deletion without controls, and the generic engine handles controls, both deletion modes, and stayers. Larger batches trade memory for speed. The solver tolerance and iteration limit govern convergence, not identification or the dependence assumptions. An explicit memory budget guides batching and produces a warning, or an error with `memorycheck(error)`, when the command's forecast allocations exceed it. Without a budget, no memory-based planning takes place, and the forecast is never a limit on Stata's total memory. The reporting options change neither the estimates nor the stored results.

3.4 Sample construction and target weights

For match deletion, the command first constructs a connected sample of movers whose firm comparisons survive the required deletions. Its selection rule also removes workers whose own removal would disconnect the network. When workers are observed at several firms, this rule can exclude observations even if deleting each of their matches separately would preserve identification, so the retained sample need not be the largest one that supports leave-match-out estimation (section S1.4). With the default `stayers(both)`, eligible stayers at retained firms are then added. The command also checks that the leave-out regressions remain identified with the controls included, and it reports a failure rather than substituting another estimator. With controls, the randomized calculation relies on a conservative sufficient check, so a reported `UNVERIFIED_DELETION_RANK` does not prove that a leave-out regression is unidentified; the exact calculation checks the leave-out matrices directly, within its size limit (section S3.2).

The estimates describe the retained population. `estat sample` reports each step of the construction, and `e(sample)` marks the retained rows. Because target weights refer to the retained sample, giving each worker equal weight requires a first run to identify that sample. The following code counts each worker's retained observations and assigns each a mass of $1/T_w$:

```
fevc log_wage i.year, worker(worker_id) firm(firm_id)
generate byte retained = e(sample)
bysort worker_id: egen long Ti = total(retained)
generate double worker_weight = cond(retained, 1/Ti, 1)
fevc log_wage i.year, worker(worker_id) firm(firm_id) ///
    targetweight(worker_weight)
assert e(sample) == retained
```

Excluded rows keep positive placeholder weights, so that the classification of movers and stayers, and hence the retained population, is unchanged. With frequency weights f_s, use the worker's retained frequency sum for T_w and the mass f_s/T_w.

3.5 Exact and randomized calculations

The option `algorithm(exact)` computes the leverages and target-specific bias loadings directly from the full-sample normal equations and obtains the leave-out residuals from (6) or its block counterpart. It is useful for small datasets and for checking the approximation on a tractable example.

The default `algorithm(jla)` uses the *Johnson–Lindenstrauss approximation* (JLA) of Kline et al. (2020, section 3), which builds on random projections (Johnson and Lindenstrauss, 1984) and randomized trace estimation (Hutchinson, 1989). Each leverage and bias loading is a squared length or inner product of high-dimensional vectors. JLA projects these vectors onto a small set of random

directions and approximates their lengths and inner products by averages over the directions, first for the leverages and then for the target-specific loadings. This replaces many exact calculations with a few hundred sparse linear solves. More probes improve precision at the cost of more solves, and the precision required depends on the distribution of weights and leverages, not only on the number of observations (section 5).

Because the result depends on random draws, the command reports by default a numerical Monte Carlo standard error (MCSE) for each corrected component. Holding the data fixed, the MCSE approximates how much the estimate would vary if both the leverage and the target directions were drawn again (section S3.5). It measures numerical uncertainty only, excluding the bias that remains with a finite number of probes, solver and floating-point error, and sampling uncertainty. With `centering(corrected)`, it also omits the variance of the centering correction and its covariance with the rest of the estimate. The exact calculation reports an MCSE of zero; a diagnostic that cannot be computed is reported as missing. Researchers should compare the MCSE with the differences they intend to interpret, especially for a covariance close to zero, and increase `probes()` when it matters. Reruns with several prespecified seeds and a tighter `tolerance()` provide further checks.

3.6 Computation and the two backends

Most of the computation consists of solving many linear systems with the same regression design. For JLA, `fevc` uses sparse iterative solves, without forming the inverse of the normal equations, and it reuses the design and the solver setup for the fitted effects and for all random directions.

To see how random projections reduce the work, consider unit weights and the regression projection matrix $P = XS_{xx}^{-1}X'$. For a vector q with independent Rademacher entries, each equal to -1 or 1 with equal probability, $\mathbb{E}(qq') = I$ and $P^2 = P$. Hence

$$\mathbb{E}\{(Pq)_i^2\} = P_{ii}. \tag{8}$$

One solve, $S_{xx}u = X'q$, produces $Pq = Xu$ and therefore a random measurement of every observation's leverage. Averaging over probes estimates all the leverages together, and related projections supply the target-specific loadings. These identities underlie the JLA method of KSS (Kline et al., 2020); `fevc` uses the normalized leverage estimates and the finite-probe adjustment of Kline et al. (2021). The number of probes, rather than the number of observations, determines the number of large solves. Computing the numerical MCSE reuses the same directions and the prepared solver, so it adds solves but no new random draws.

The worker–firm structure is especially useful here. Without controls, eliminating worker effects reduces the normal equations to a system in firm effects with matrix

$$S_{\mathrm{f}} = X_{\mathrm{f}}'X_{\mathrm{f}} - X_{\mathrm{f}}'X_{\mathrm{w}}(X_{\mathrm{w}}'X_{\mathrm{w}})^{-1}X_{\mathrm{w}}'X_{\mathrm{f}}, \tag{9}$$

where X_{w} and X_{f} are the worker and firm indicator matrices. This is a graph Laplacian whose links summarize the worker mobility that connects firms. Its sparsity makes matrix–vector products inexpensive, while weak links between large groups of firms can make iterative solution slow. Combinatorial multigrid (CMG) preconditioning groups strongly connected vertices into progressively smaller graphs and uses corrections on these graphs to remove errors that decay slowly in the original system (Koutis et al., 2011). The hierarchy is prepared once and reused across right-hand sides. Sparse solves, random projections, and CMG also underpin the KSS Matlab package.

I implemented a simplified CMG solver in Mata, with graph aggregation, local error corrections, and a direct solve on the smallest graph, and ported the full Matlab/C solver to Rust as a standalone package (Schmieder, 2026a). On eligible two-way graphs, the Rust solver eliminates workers with up to four distinct firm matches, which reduces the system's dimension, and retains higher-degree workers to avoid creating too many firm-to-firm connections. The benefit therefore depends on the mobility graph. Rust threads share the graph, hierarchy, and working space across right-hand sides, whereas the benchmarked KSS Matlab configuration uses separate worker processes with potentially duplicated data and solver state.

Calculation	Mata backend	Rust backend
Exact and randomized point estimation	Supported	Supported
Observation and match deletion, stayers	Supported	Supported
Joint or fixed-offset controls	Supported	Supported; at most 32 control columns in JLA
Numerical MCSE	Supported	Supported
Preconditioners	Diagonal; simplified CMG	Diagonal; CMG with worker elimination
Fixed-effect projections	Exact calculation	Randomized calculation with the explicit options in section 3.8
Experimental component inference	Exact observation smoother; structured match inference	Structured observation and match inference by JLA

The limit of 32 control columns counts each factor-variable indicator separately.

3.7 Postestimation and stored results

The main output reports, for each component, the plug-in estimate, the estimated bias, the corrected estimate, and its share of the target-weighted outcome variance. The corrected value equals the plug-in value minus the estimated bias. The sorting row is twice the worker–firm covariance, so the worker variance, firm variance, and sorting rows add up to the variance of the sum. Four postestimation commands display further detail: `estat decomposition` repeats the table, and its `full` option adds the raw covariance and model fit; `estat sample` reports the sample construction; `estat computation` reports the backend, method, and numerical settings; and `estat diagnostics` reports numerical and resource diagnostics, including the status of the numerical MCSE.

Point estimation posts no sampling covariance matrix `e(V)`, so `lincom` and `test` do not apply to the decomposition. The main stored results are:

Scalars	
`e(N)`, `e(worker_levels)`, `e(firm_levels)`	retained observations, workers, and firms
`e(deletion_units)`, `e(probes)`, `e(seed)`	deletion units, probes per stage, and seed
`e(mcse_available)`	1 if the numerical MCSE is usable
Macros	
`e(deletion)`, `e(stayers)`, `e(nuisance)`, `e(centering)`	estimand choices
`e(algorithm)`, `e(backend_selected)`, `e(rng_selected)`	method, backend, and generator used
`e(mcse_mode)`, `e(mcse_method)`, `e(mcse_status)`	numerical-MCSE mode, method, and status
`e(status)`, `e(withholding_status)`	result status; reason a result was withheld
Matrices	
`e(b)`	corrected worker variance, firm variance, covariance, and variance of the sum
`e(results)`	rows `plugin`, `bias_correction`, `corrected`, and `mcse`
`e(mcse)`, `e(mcse_cov)`, `e(mcse_cov_raw)`	numerical MCSE; usable and raw 4×4 numerical covariances
`e(decomposition)`	decomposition with the sorting row and shares
`e(projection_b)`, `e(projection_V)`, `e(projection_results)`	projection coefficients, covariance, and table
`e(V)`	component covariance, posted only by supported inference requests
Functions	
`e(sample)`	marks the retained sample

The numerical covariance matrices are never posted as `e(V)`. If a validation check that applies to the whole command fails, the command exits with an error and records the reason in `e(withholding_status)`. Other failures affect only the result concerned. A failure of the numerical MCSE alone can leave the point estimates in place, and component inference in the Rust backend can withhold one target's interval or the joint covariance while reporting the remaining results. A successful return therefore does not guarantee that every requested diagnostic is available; `ereturn list` shows what was computed, and the supplement describes when each interval is available.

3.8 Fixed-effect projections

To relate firm effects to an observed characteristic, add `project()` and `projecteffect()` to the decomposition command. The command adds a constant and reports the projection coefficients with leave-out standard errors that account for the estimation error in the fixed effects and for the declared dependence blocks. Either worker or firm effects can be projected. The projection covariates, including the constant, must have full rank under the projection weights, which use frequency mass by default and target mass with `projectweight(target)`.

By default, projections use the exact Mata calculation, which forms a dense inverse and is therefore limited to small designs by `exact_limit()`. Larger designs need the randomized Rust calculation, requested explicitly with `backend(rust)`, `rng(counter_v1)`, `algorithm(jla)`, and an explicit `preconditioner(cmg)` or `preconditioner(diagonal)`. It uses the generic engine, which may be stated with `engine(generic)` or left at `engine(auto)`. Both calculations support observation and match deletion, jointly estimated controls, and stayers. Centering changes the standard errors but not the coefficients: mean centering is the default, `centering(none)` is available, and `centering(corrected)` is not. Under mean centering the covariance estimate need not be exactly unbiased, and normal intervals require the conditions in section 2.7. The numerical MCSE does not cover the projection coefficients or their covariance. Section 4 gives an example.

3.9 Experimental component inference

Standard errors for the variance components themselves are harder to obtain. Kline, Saggio, and Sølvsten (2020) derive the sampling variance of leave-out estimators and estimate the required products of error variances with independent split-sample predictions. These predictions impose stronger requirements on the worker–firm graph than point estimation and add substantial computation; KSS provide conservative adjustments for cases in which they are unavailable. The KSS Matlab package contains additional routines, not exposed through its main interface, that instead fit a model for the error variances and use the fitted values in the covariance calculation (Saggio, 2024). `fevc` builds on this fitted-variance approach and adapts it to mean-centered estimates, whose sampling covariance differs from that of the uncentered estimator.

The opt-in requests `inference(highrank)` and `inference(q1)` report experimental intervals for the four components. They use different reference distributions. Highrank uses a normal reference distribution. It requires that no single direction in the data dominate the target, a condition KSS call a diffuse spectrum, and that individual units have limited influence. q1 allows for one dominant direction, such as a single imprecisely estimated contrast that carries much of the target, together with a diffuse remainder, and uses the corresponding nonstandard reference distribution of KSS. Both depend on an adequate model for the error variances and on assumptions beyond those needed for point estimation. Under mean centering, the calculation treats the observed mean as fixed and omits the uncertainty from estimating it. The procedures can also be numerically fragile. Their intervals should therefore be read as suggestive evidence about the uncertainty of the estimated components; a successful computation does not establish nominal coverage.

Structured match inference covers movers and eligible stayers together, or movers only with `stayers(movers)`. Movers have several original deletion units and stayers have one. Errors may be correlated within a mover unit, but mover units and individual stayer observations, including frequency copies of a stayer row, must be mutually independent. The variance model must be chosen explicitly: `inferencemodel(structured_common)` makes the error variance a quadratic function of ranks of leverage, target loadings and mover regression mass, and `structured_leverage` a function of leverage alone. Both fit separate mover and stayer coefficients jointly. The controls must be held fixed with `nuisance(fixedoffset)`, and the uncertainty from estimating them is omitted; having few controls does not by itself make this omission harmless. Component inference is not available with `centering(corrected)`, nor for match deletion with jointly estimated controls.

Exact Mata serves small designs and Rust's randomized calculation larger ones. Both accept frequency and target weights and use simulation to compute the traces in the covariance formula. Highrank posts `e(V)` only when its joint covariance and all individual checks pass; q1 reports intervals without a Gaussian `e(V)`. A withheld interval is not a zero standard error. Structured inference and `project()` require separate calls. By default, Rust uses 2,048 random probes to fit the variance model (`inferencegramprobes()`), and both backends use 1,000 covariance simulations (`inferencesimulations()`); these budgets are separate from the 200 point-estimation probes. Technical Supplement Table S5.1 and Section S5.12 list the supported requests and options.

For observation deletion, exact Mata also offers a different method that smooths observation-level variance estimates. It is used for supported observation-deletion requests that do not name a structured model. In two tested designs with mean centering and highrank intervals, it produced intervals in only 11.4% and 18.8% of attempts, and smaller q1 designs also had low availability (Technical Supplement Table S6.9). These failures do not describe structured match inference.

Because a failed inference call can also withhold the point results, it is prudent to save a point-only estimate before experimenting with inference. The MCSE of the point estimates does not cover the inference calculations.

4 Illustration with the public Veneto examples

KSS illustrate leave-out estimation with Italian social-security records from the Veneto Worker History file (Kline et al., 2020, section 9). The KSS Matlab package includes a public teaching extract from these records, which I use here to show how to estimate and interpret a decomposition in Stata. The extract is a small public dataset, not the confidential data used in the published application.

The extract has 128,632 observations on workers in two regions. The decomposition and the component inference use the 71,614 observations on workers in one region who are observed in both 1999 and 2001; the fixed-effect projection uses all observations to compare the two regions.[3] Both analyses use the log wage in the file without further controls and include movers and eligible stayers. Table 1 reports the samples that `fevc` retains; the decomposition keeps 56,044 observations on 28,022 workers and 1,684 firms. Comparisons across specifications should account for differences in their retained populations.

Table 1: Samples in the public Veneto examples

Example	Input rows	Retained rows	Mover rows	Stayer rows	Workers	Firms
Default decomposition	71,614	56,044	12,828	43,216	28,022	1,684
Firm-effect projection	128,632	89,666	19,944	69,722	44,833	2,974

Notes: Both examples use the same public dataset. The decomposition restricts the input to workers with regional code 1 who are observed in both years; the projection uses both regions. Retained rows are divided into movers and eligible stayers from the input data, and the worker and firm counts cover both groups. Target weights give each retained observation equal mass. The correction assumes independent errors across a stayer's observations but allows arbitrary dependence within a mover's match.

4.1 Variance decomposition

The following code imports the data, selects the decomposition sample, and estimates its variance components:

```
import delimited using kss_veneto.csv, ///
    varnames(nonames) asdouble clear
rename (v1 v2 v3 v4 v5) (worker firm year region log_wage)
keep if region == 1
bysort worker (year): keep if _N == 2
fevc log_wage, worker(worker) firm(firm)
```

All settings are defaults: match deletion with eligible stayers, mean centering, and the randomized calculation with 200 probes and seed 8675309. After its progress messages, the command displays the following output:

[3]The KSS Matlab package distributes the file as `lincom.csv`; the replication package supplies it as `kss_veneto.csv`. The code listings assume that this file is in the working directory.

```
KSS leave-out variance decomposition
Sample:       56,044 of       71,614 requested rows retained ( 78.3%)
Dimensions:     28,022 workers;      1,684 firms;     56,044 deletion units
Target: retained movers plus eligible attached stayers
Deletion: match; nuisance=joint
Method: jla; backend=rust
Numerics: engine=generic; preconditioner=cmg
JLA work: probes=      200; seed=   8,675,309
Note: sample construction removed       15,570 requested rows; type estat sample
>  for the stage accounting.
Note: movers use match blocks; stayers use observation deletion.
      Stayer correction is not match-robust.

Additive worker-firm decomposition
------------------------------------------------------------------------------
Component                     Plug-in Estimated bias KSS corrected   % of Var(Y)
------------------------------------------------------------------------------
Worker variance              .1035376      .01840106      .0851365         68.38
Firm variance               .01982105     .009582543     .01023851          8.22
Sorting (2 x cov.)         -.00781809     -.01717639    .009358298          7.52
Total worker-firm var.       .1155405      .01080721      .1047333         84.12
------------------------------------------------------------------------------
KSS corrected = plug-in - estimated bias; sorting is 2 x covariance.
Target-weighted Var(Y):       .1245015
Approximate numerical MCSE (all main point probes)
  MCSE method: crossfit_if_v1; status: ok_local
  worker= .00010214; firm=  .0000929; covariance=  .0000896; total= .00002767
  Main point estimates only; excludes projection/inference calculations.
  Excludes finite-probe bias, solver/floating-point error and sampling uncertain
> ty.

Inference: not requested; e(V) not posted; MCSE is numerical, not sampling.
```

The header describes the retained sample, the target population, the deletion convention, and the method. The notes warn that the stayer part of the correction assumes independence across a stayer's observations. The table and the MCSE block follow the layout described in section 3.7.

The same decomposition can be computed with the Mata backend and followed by postestimation commands:

```
fevc log_wage, worker(worker) firm(firm) backend(mata)
estat computation
estat diagnostics
generate byte decomposition_sample = e(sample)
```

The generated indicator keeps the estimation sample for later analysis. Because the Mata backend uses a different random-number generator, the same seed produces different random directions. Its corrected firm variance of 0.01031 lies within the numerical uncertainty of the two calls.

Table 2 collects the results. The correction reduces both variance estimates. The worker variance falls from 0.10354 to 0.08514, and the firm variance from 0.01982 to 0.01024. The firm component is particularly affected: almost half of its plug-in value is attributed to estimation noise. Treating the estimated effects as if they were the true effects would therefore substantially overstate firm heterogeneity in this sample.

The covariance changes sign, from -0.00391 to 0.00468. The plug-in estimates suggest that

Table 2: Variance decomposition in the public Veneto example

Target	Plug-in	Bias correction	KSS	Numerical MCSE
Worker variance	0.10354	0.01840	0.08514	0.000102
Firm variance	0.01982	0.00958	0.01024	0.000093
Worker–firm covariance	-0.00391	-0.00859	0.00468	0.000090
Variance of sum	0.11554	0.01081	0.10473	0.000028

Notes: Rust calculation with the default mean centering. The bias correction is subtracted from the plug-in estimate. The numerical MCSE approximates the variation from both the leverage and the target projections on the fixed data, with 200 probes per stage. It excludes finite-probe bias, solver error, and sampling uncertainty.

workers with higher worker effects tend to be employed by firms with lower firm effects; after the correction, the estimated association is positive. This change is consistent with limited mobility: when few workers move, an overestimated worker effect tends to be paired with an underestimated firm effect, so the estimation errors of the two effects are negatively correlated and pull the plug-in covariance down.

The variance of the sum falls from 0.11554 to 0.10473. This change is smaller than the combined reductions in the two variances because the total includes twice the covariance, whose correction offsets part of those reductions.

The firm-variance correction of 0.00958 is about one hundred times its numerical MCSE of 0.000093, so numerical variation is negligible relative to the correction. With `centering(none)`, the same call gives a corrected firm variance of 0.01076 with a numerical MCSE of 0.00037; mean centering thus reduces the numerical MCSE by a factor of about four here. Neither number speaks to sampling uncertainty. Technical Supplement Section S6.2 reports checks across complete reruns with different seeds at 200 and 1,000 probes.

4.2 Component inference for movers and stayers

To attach experimental sampling intervals to this decomposition of movers and stayers, request a structured variance model and the randomized match-inference calculation:

```
fevc log_wage, worker(worker) firm(firm) ///
    deletion(match) stayers(both) nuisance(fixedoffset) ///
    backend(rust) rng(counter_v1) algorithm(jla) ///
    engine(generic) preconditioner(cmg) ///
    tolerance(1e-10) nativethreads(4) ///
    inference(highrank) inferencemodel(structured_common)
```

With the complete settings in Technical Supplement Section S5.14, a point-only call and the highrank and q1 calls retain the same 56,044 observations and give the same point estimates. Both inference calls return intervals for all four components, and highrank also posts their joint covariance. The standard error of the firm variance is 0.00103, giving a 95% interval of $[0.00822, 0.01226]$. With no controls, no control-estimation uncertainty is omitted, but the interval still rests on the independence of stayer observations, the variance model, the conditions behind the reference distribution, and a negligible effect of treating the mean as fixed. The diagnostics also show that the leading direction of each variance and of the covariance is concentrated: a single unit accounts for about 27 percent of its squared weight (Technical Supplement Table S5.4). The similarity of the highrank and q1 intervals therefore does not verify the one-mode conditions, and this illustration

does not establish coverage.

4.3 Projection of firm effects on a regional indicator

A researcher may also want to describe how firm effects differ across regions. The projection reloads the full dataset with both regions and projects firm effects on an indicator `region2` for regional code 1, with code −1 as the reference category:

```
import delimited using kss_veneto.csv, ///
    varnames(nonames) clear
rename (v1 v2 v3 v4 v5) (worker firm year region log_wage)
generate byte region2 = region == 1
fevc log_wage, worker(worker) firm(firm) ///
    project(region2) projecteffect(firm) ///
    backend(rust) rng(counter_v1) algorithm(jla) ///
    engine(generic) preconditioner(cmg)
```

The retained design has too many parameters for the default exact projection, so the call requests the randomized calculation with the five options listed in section 3.8. The projection describes the distribution of firm effects across retained observations, using frequency weights. Its slope equals the slope from regressing the estimated firm effects on the indicator in the same sample with the same weights; the leave-out calculation changes only how the uncertainty in the estimated effects is measured.

Table 3: Projection of firm effects on a regional indicator

Term	Estimate	KSS SE	Full-sample residual SE	95% KSS interval
Constant[a]	-0.27874	0.13403	0.07031	[-0.54142, -0.01605]
Region 2	0.25982	0.09518	0.05015	[0.07328, 0.44636]

Notes: The KSS covariance uses mean-centered, symmetrized leave-out cross-products within the declared blocks. The full-sample residual covariance replaces them with outer products of full-sample residuals, using the same blocks and loadings. [a]The constant depends on the normalization, which sets the last retained firm effect to zero.

The estimated regional difference in firm effects is 0.25982 (table 3). Its KSS standard error is 0.09518, compared with 0.05015 when full-sample residuals are used in the same formula; both calculations use the same estimation loadings and dependence blocks. Replacing full-sample residual products with mean-centered leave-out products thus nearly doubles the standard error here. The resulting 95 percent normal interval is $[0.07328, 0.44636]$. Without centering, the KSS standard error is 0.10253. Unlike the constant, the slope does not depend on the normalization of the firm effects. It is a descriptive projection in this retained sample; reading it as the effect of a firm's location would require a causal argument.

5 Statistical and computational performance

The experiments address three practical questions: how much bias the leave-out correction removes, how accurately the randomized calculation approximates exact KSS, and how much time and memory large problems require. The statistical experiments draw repeated outcomes on a fixed design. The approximation experiment holds the data fixed and redraws the random projections.

The performance comparison runs paired calls of `fevc` and the KSS Matlab package on the same data and host. The `fevc` experiments use `centering(none)`; exact calculation is finite-sample unbiased under the assumptions in section 2. Mean centering adds one mean calculation and no linear solves, so it does not change the computational burden.

5.1 Bias correction in a fixed design

The bias experiment has 24 observations, six workers, four firms, and four periods, with 17 declared matches, and remains identified after the requested deletions. Outcomes combine worker and firm effects, two controls, and an error. One specification has independent heteroskedastic observation errors and uses observation deletion. The other adds a shock common to all observations in a match and uses match deletion. In both, the controls are re-estimated jointly after each deletion. Each specification uses 5,000 outcome replications; section S6 gives the complete coefficient and error distributions.

Table 4: Plug-in and exact KSS estimates in the fixed-design experiment

Target	Plug-in bias	MCSE	KSS bias	MCSE	Plug-in RMSE	KSS RMSE
Independent observation errors						
Worker variance	0.0708	(0.0022)	0.0024	(0.0023)	0.1710	0.1594
Firm variance	0.0452	(0.0014)	-0.0005	(0.0014)	0.1070	0.1001
Covariance	-0.0090	(0.0005)	-0.0004	(0.0005)	0.0376	0.0373
Variance of sum	0.0979	(0.0025)	0.0012	(0.0025)	0.1995	0.1790
Independent match blocks						
Worker variance	0.1390	(0.0035)	-0.0042	(0.0037)	0.2847	0.2589
Firm variance	0.0913	(0.0021)	-0.0005	(0.0022)	0.1721	0.1549
Covariance	-0.0149	(0.0008)	0.0002	(0.0008)	0.0561	0.0550
Variance of sum	0.2006	(0.0037)	-0.0044	(0.0039)	0.3269	0.2785

Notes: 5,000 outcome replications per error design. Truth is the fixed-design quadratic target. MCSE is the Monte Carlo standard error of estimated bias; RMSE is root mean squared error.

The leave-out correction removes most of the plug-in bias (table 4). Under independent observation errors, plug-in bias is 0.0708 for worker variance, 0.0452 for firm variance, and 0.0979 for the variance of their sum. The corresponding exact KSS biases are 0.0024, −0.0005, and 0.0012. With within-match error correlation, plug-in bias in the total rises to 0.2006, while match-deletion KSS bias is −0.0044. Across the eight component-by-design comparisons, the largest absolute KSS bias is 1.15 Monte Carlo standard errors.

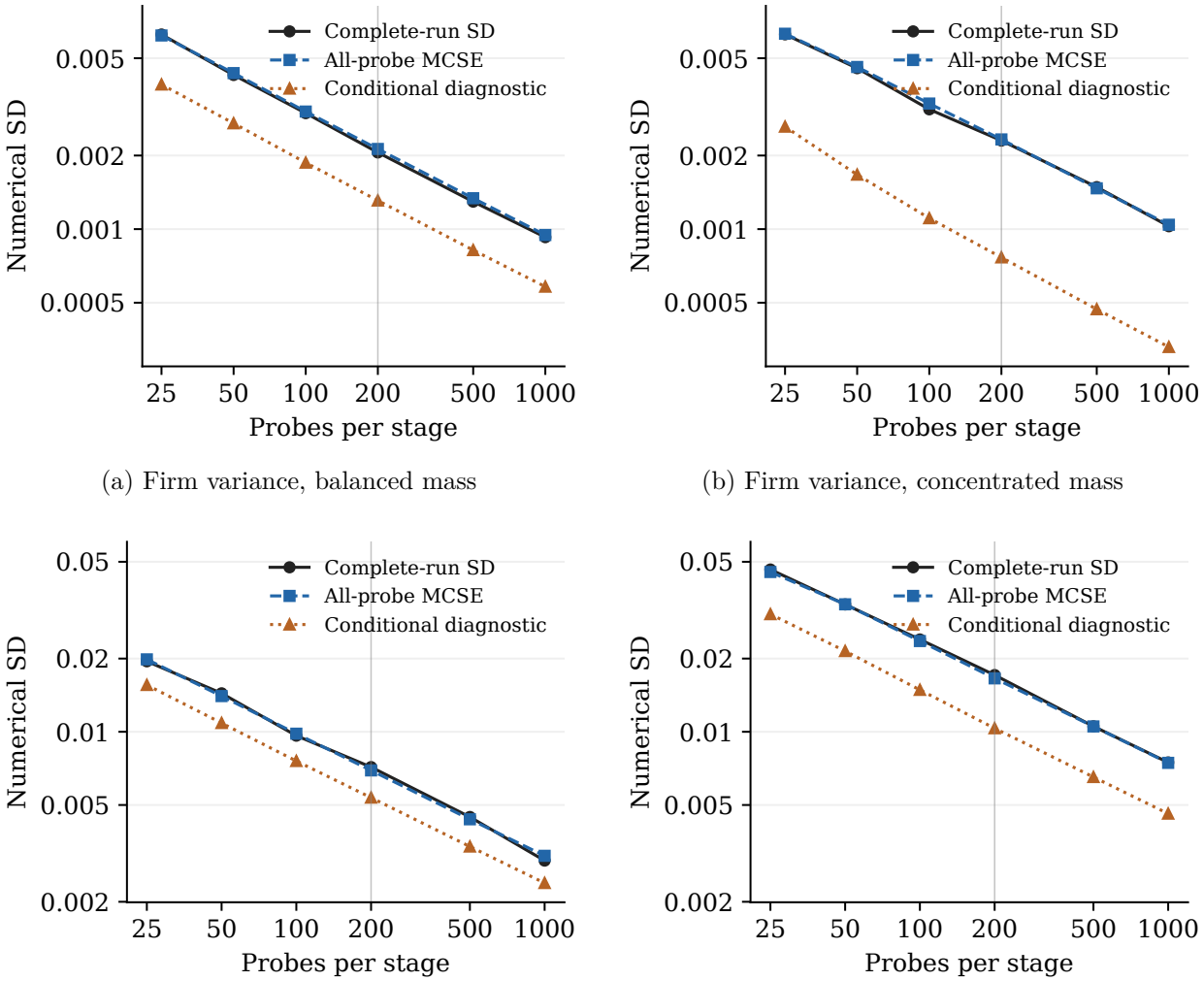
The correction also lowers the root mean squared error of the worker variance, the firm variance, and the total in both designs. Its effect on the covariance's root mean squared error is small, because that component's plug-in bias is small relative to its sampling variation. These results illustrate the finite-design unbiasedness argument: when the deletion unit matches the error dependence, the correction works even when errors within a match are correlated.

5.2 Accuracy of the randomized approximation

The approximation experiment uses the same 24-row graph and fixes the outcome. Exact match-deletion KSS supplies the comparison value. The randomized calculation uses 25, 50, 100, 200, 500,

or 1,000 probes per stage, with 1,000 complete-call seeds at each probe count. A balanced design gives every stored row frequency one. A concentrated design gives two rows in one match frequency 12, increasing both that match's regression information and its default target mass. Repeated runs measure numerical variation on the same data, as distinct from the outcome variation in the bias experiment.

Figure 1: Numerical precision and probe count



(a) Firm variance, balanced mass

(b) Firm variance, concentrated mass

(c) Variance of sum, balanced mass

(d) Variance of sum, concentrated mass

Notes: Mata JLA on fixed data, with 1,000 complete-call seeds at each probe count. Complete-run SD is the standard deviation of the point estimates. All-probe MCSE, the reported numerical MCSE, is the square root of the mean raw estimated numerical variance. The conditional curve is the root mean square of a developer diagnostic that holds the leverage directions fixed. Finite raw covariances are retained even when the reported diagnostic is withheld. The concentrated design changes regression frequencies and default target weights. Both axes are logarithmic; the vertical line marks 200 probes.

Figure 1 compares the reported numerical uncertainty with the actual dispersion of complete randomized runs. The numerical MCSE is designed to capture variation from both stages of random directions, whereas the conditional diagnostic omits variation in the leverage directions. The plotted

MCSE averages estimated variances rather than standard deviations, and the MCSE of the total includes the numerical covariances between components. In every plotted cell, the numerical MCSE is within six percent of the complete-run standard deviation.

At 200 probes, the numerical MCSE of the firm variance is 0.00213 in the balanced design and 0.00233 in the concentrated design, compared with complete-run standard deviations of 0.00206 and 0.00230. The conditional diagnostic gives 0.00131 and 0.00077. Randomness in the leverage directions can thus account for a substantial part of the numerical uncertainty, especially for an individual variance component.

Concentrating observations and target mass in one match makes the approximation more demanding, even though the stored graph is small. A given number of probes therefore does not deliver the same precision in every application. Technical Supplement Section S6.2 and Table S6.2 report bias against exact KSS, dispersion, and availability for all four components, with Rust checks at 200 and 1,000 probes.

5.3 Projection standard errors

An independent dense Python calculation checks the projection covariance formula by refitting every leave-out regression exactly. Evaluating the formula outside the command separates its statistical behavior from the implementation, which is checked separately against exact Mata and Rust results on fixed inputs. The design has 400 independent matches with two observations each, heteroskedastic errors, within-match correlation 0.45, and the uncentered estimator. Across 3,000 replications, the mean estimated slope variance is within 0.92 percent of the true value of 0.00032228, and the other two covariance entries are within 1.34 percent. A nominal 95 percent interval for the slope covers in 94.0 percent of replications, with Monte Carlo standard error 0.0043. An accurate average covariance estimate thus coexists with modest undercoverage. Technical Supplement Section S6.3 specifies the design.

5.4 Time and memory across graph classes

The performance comparison uses six synthetic graph classes: near-degree-three workers in a well-mixed market, degree-five workers in well-mixed or segmented markets, mixed worker degrees from two to eight in either market structure, and degree-four workers with a bottleneck between two markets. A worker's degree is the number of firms at which the worker is observed. Every worker has distinct firm matches with one observation per match, and every firm is matched with 100 workers; to keep round sample sizes, one or two workers in the near-degree-three class have degree four. One percent of workers connect the two markets of the segmented and bottleneck designs. The low-degree classes include workers that the Rust solver eliminates in its degree-four reduction as well as workers that remain in its graph. Section S7.2 gives the construction.

The size grid doubles from 100,000 to 6.4 million observations, covering seven sizes at 28 native threads. The degree-four, degree-five, and mixed-degree classes cover both observation and match deletion; the near-degree-three class covers match deletion only. A separate grid uses one, four, seven, 14, and 28 threads at 400,000 observations, with both deletion modes for the two mixed-degree classes and match deletion for the near-degree-three class. Together the grids contain 97 distinct configurations. Each experiment runs one warm-up call per package, followed by three measured pairs in rotated order. Figures 2 and 3 show match deletion only. Their upper panels vary

observations for all six classes at 28 threads; their lower panels vary threads for the near-degree-three and two mixed-degree classes at 400,000 observations. Table 5 reports both deletion modes at the largest sample size.

Each pair runs sequentially on identical retained data on the same host. Specifications use no controls, unit frequencies, uniform target weights, `stayers(both)`, and 200 probes. `fevc` uses the Rust backend, automatic solver and batch selection, and at most 10,000 iterations, with no explicit tolerance or memory-budget option. Stata uses $\min(T, 4)$ licensed processors, while the native solver independently uses T threads. The KSS Matlab package uses T single-threaded workers and its own mean-centered correction and numerical settings (Technical Supplement Section S8). The synthetic graphs contain no stayers; the Veneto comparison below exercises repeated matches and stayers.

Setup-inclusive time begins after the common data import and validation, before any package-specific setup, and ends when the estimate returns. It includes the creation of a fresh Matlab worker pool, but not Matlab process startup or data import. Estimator-only time excludes package setup and pool creation. Time panels show medians of three measured calls, and ratio panels show medians of the paired ratios.

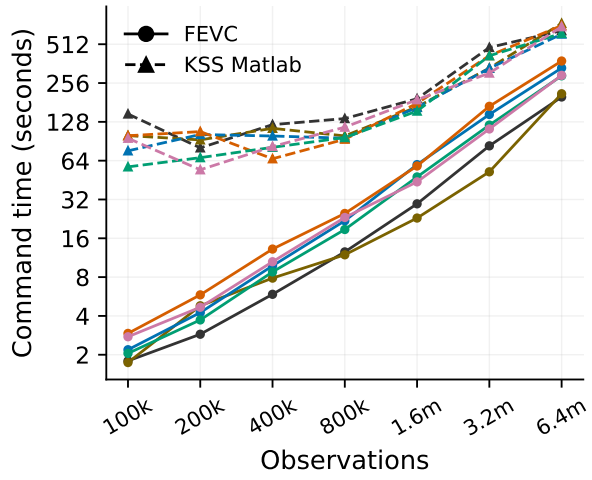
Memory is the peak process-tree proportional set size (PSS) over the setup-inclusive interval, with one measured pair per configuration. PSS counts private resident pages fully and apportions shared pages across processes, and it includes loaded data and application overhead. These sampled footprints are not exact minimum RAM requirements: sampling can miss brief peaks, sweeps are not atomic, the observer adds overhead, and a single pair per configuration cannot show run-to-run variation. Technical Supplement Section S7.1 gives the measurement definitions, sampling limits, and numerical checks. Its Veneto tables also report summed resident set size (RSS), which counts shared pages in full.

`fevc` has a median paired setup-inclusive time ratio below one in all 102 evaluated grid cells (97 distinct configurations). At 6.4 million observations and 28 threads, speedups are 1.35–2.94 for observation deletion and 1.85–3.49 for match deletion. Median setup-inclusive times are 198.6–367.2 seconds versus 494.0–583.0 seconds for observation deletion, and 199.6–379.3 seconds versus 613.5–742.5 seconds for match deletion, for `fevc` and KSS Matlab, respectively. Excluding package setup leaves speedups of 1.27–2.94 and 1.90–5.00. Pool startup thus contributes to the setup-inclusive advantage but does not account for all of it. At this size, the degree-four bottleneck design gives the largest setup-inclusive speedup under both deletion modes. Under match deletion, the near-degree-three class gives a 3.30-fold setup-inclusive speedup and a 5.00-fold estimator-only speedup. Neither low-degree class is representative of graphs with higher worker degrees.

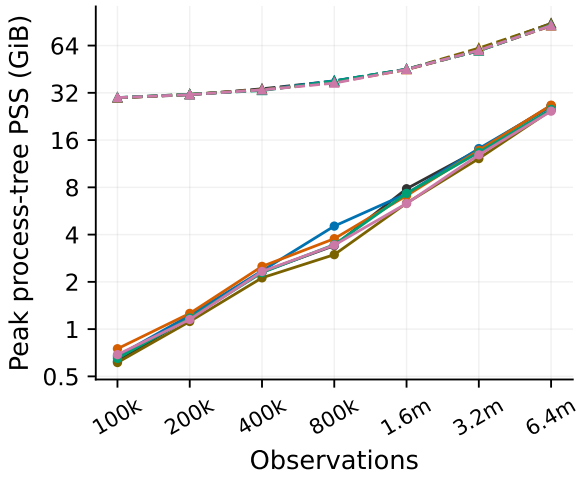
Under match deletion at 6.4 million observations, measured PSS peaks are 24.5–26.7 GiB for `fevc` and 85.4–88.6 GiB for KSS Matlab; paired ratios range from 0.281 to 0.312. Because hardware can differ between size cells, the curves across sizes do not form a scaling experiment on fixed hardware.

Table 5 reports the largest synthetic size for both deletion modes and the two Veneto benchmarks, all at 28 threads. One single-thread mixed-degree, well-mixed observation-deletion cell in the thread grid favors Matlab during estimation alone, so the setup-inclusive advantage does not imply faster estimation in every setting. Because each synthetic match contains one observation, the two deletion modes exercise distinct software paths on identical deletion units; they do not establish performance for repeated-observation matches.

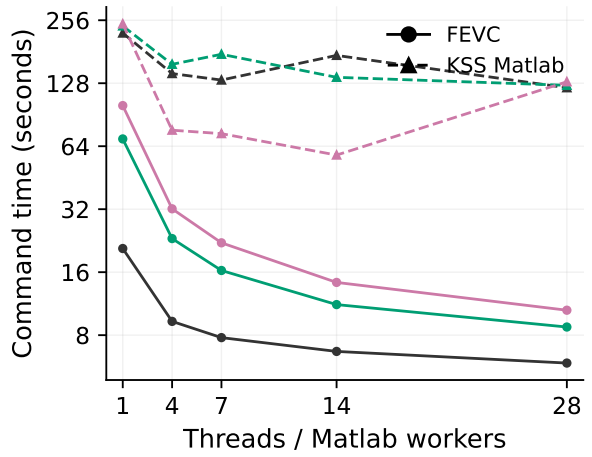
Figure 2: Match deletion: setup-inclusive time and memory



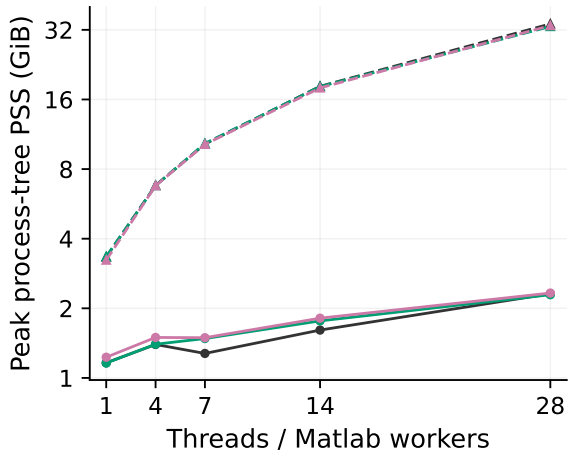
(a) Time by observations, 28 threads



(b) Memory by observations, 28 threads



(c) Time by threads, 400,000 observations

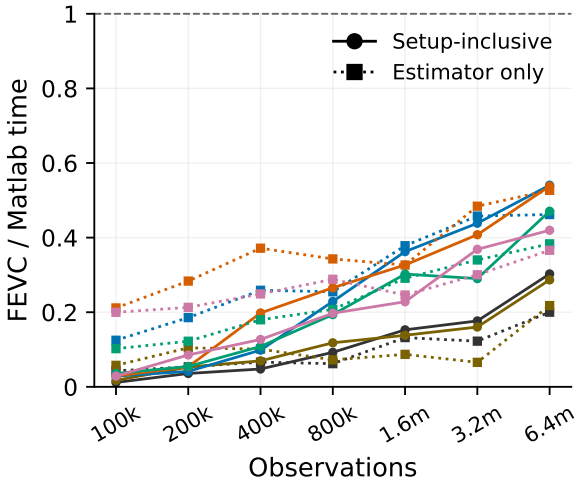


(d) Memory by threads, 400,000 observations

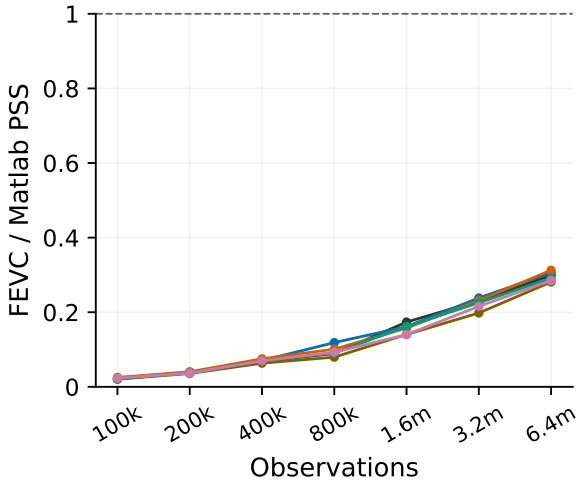
Degree 3: well-mixed
Degree 5: well-mixed
Degrees 2-8: well-mixed
Degree 4: bottleneck
Degree 5: segmented
Degrees 2-8: segmented

Notes: Match deletion throughout. The upper panels show all six classes; the lower panels show degree three and the two mixed-degree classes. Degree three includes one or two degree-four workers to preserve round N. Solid circles denote `fevc` and dashed triangles KSS Matlab. Time points are medians of three measured setup-inclusive times. Memory is sampled peak process-tree PSS over the same interval, one pair per point. Vertical axes are logarithmic with powers-of-two ticks. Stata uses $\min(T, 4)$ processors; native `fevc` uses T threads and Matlab T single-threaded workers. Hosts are fixed within pairs and thread sweeps, but can differ between size cells and sweeps.

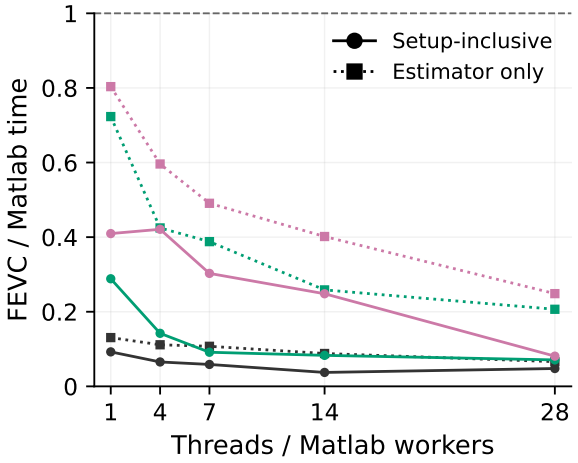
Figure 3: Match deletion: paired `fevc`-to-KSS Matlab time and memory ratios



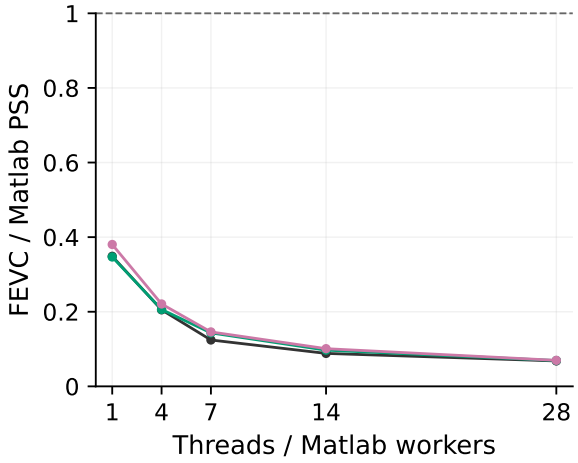
(a) Time ratio by observations, 28 threads



(b) Memory ratio by observations, 28 threads



(c) Time ratio by threads, 400,000 observations



(d) Memory ratio by threads, 400,000 observations

Degree 3: well-mixed
Degree 5: well-mixed
Degrees 2-8: well-mixed
Degree 4: bottleneck
Degree 5: segmented
Degrees 2-8: segmented

Notes: Match deletion throughout, with the same grids as figure 2. Time points are medians of three same-host paired ratios; solid curves show setup-inclusive time and dotted curves estimator-only time. Memory points are ratios of sampled peak process-tree PSS for the same package pairs. Vertical axes are linear from zero to one; dashed horizontal lines mark parity. A ratio of 0.25 means one-quarter of Matlab's time or memory. Each thread sweep uses one host; hardware may differ between size cells and sweeps.

Table 5: Synthetic and Veneto performance at 28 threads

Graph or sample	FEVC (s)	Matlab (s)	Time ratio	Est. ratio	PSS ratio
Observation deletion, 6.4 million observations					
Degree 4: bottleneck	198.6	583.0	0.340	0.340	—
Degree 5: well-mixed	318.9	551.2	0.579	0.689	—
Degree 5: segmented	367.2	494.0	0.743	0.786	—
Degrees 2-8: well-mixed	277.1	510.4	0.547	0.558	—
Degrees 2-8: segmented	279.2	524.2	0.541	0.569	—
Match deletion, 6.4 million observations					
Degree 3: well-mixed	199.6	652.4	0.303	0.200	0.299
Degree 4: bottleneck	211.4	742.5	0.287	0.218	0.281
Degree 5: well-mixed	334.9	613.5	0.541	0.462	0.306
Degree 5: segmented	379.3	714.5	0.538	0.527	0.312
Degrees 2-8: well-mixed	291.4	621.7	0.471	0.383	0.291
Degrees 2-8: segmented	293.8	700.1	0.420	0.366	0.285
Veneto data					
1999/2001: observation	25.2	232.4	0.110	0.178	0.224
1996–2001: match	104.5	278.8	0.375	0.264	0.328

Notes: 28 native threads; timing summaries use three measured pairs. Synthetic graphs have 6.4 million observations. The Veneto samples have 1,319,994 observations in 1999/2001 and 5,163,428 in 1996–2001. Times are separate package medians of setup-inclusive time. Time ratios are medians of paired `fevc`-to-Matlab values, not ratios of separate medians; "Est." excludes package setup and pool creation. PSS ratios compare sampled peak process-tree PSS, with one package pair per measured configuration. A dash denotes unmeasured PSS, not zero. Degree three is match-only. The 1996–2001 sample uses the match/stayer-hybrid correction.

The two packages draw different random directions and use different finite-probe corrections. For comparison, the Matlab package's targets, which divide by $N-1$, are multiplied by $(N-1)/N$ to match `fevc`'s normalization. Table S7.1 reports the differences across all 303 measured pairs; these do not show that the two approximations are equally accurate.

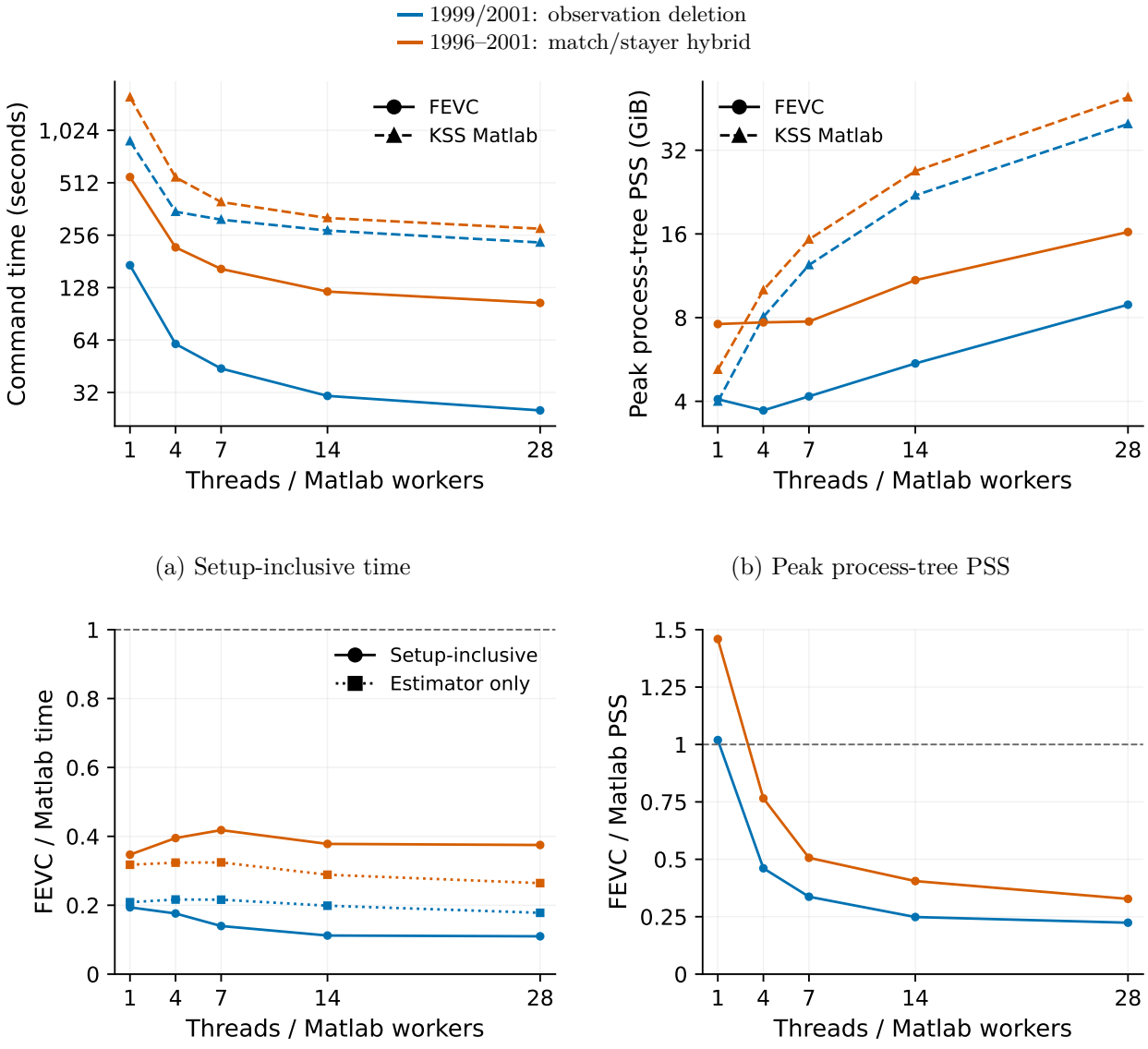
5.5 Performance on real-world data: Veneto

To test `fevc`'s performance on real-world data, I turn to the confidential Veneto data. Figure 4 compares two pooled samples from these records, rather than the public teaching extract of section 4: the 1999/2001 sample with 1,319,994 observations and observation deletion, and the full 1996–2001 panel with the match/stayer-hybrid correction. The samples differ in deletion method, preparation, and execution host, so the curves do not isolate the effect of adding years.

The longer window follows Kline et al. (2020, appendix A, table A.I). The reconstructed sample contains 5,163,428 observations, 1,106,419 workers, 86,647 firms, and 1,700,466 worker–firm matches, including 666,090 stayers. It has 18 fewer observations than the published pooled count of 5,163,446, so it is not an exact replication of the published sample. After singletons are removed, the network remains connected when any single worker is deleted. Year effects and quadratic and cubic age terms are estimated jointly with the worker and firm effects and subtracted once to form a common adjusted outcome; this preparation is not timed.

For the six-year panel, both packages estimate the four targets with equal weight on every observation and 200 probes, using the match/stayer-hybrid correction. This differs from the firm-variance specification of KSS's Table A.I, and it does not allow for arbitrary error dependence

Figure 4: Veneto performance: two-period and six-year samples



(a) Setup-inclusive time

(b) Peak process-tree PSS

(c) Paired time ratios

(d) Paired PSS ratios

Notes: Both samples retain movers and stayers and use 200 probes: 1,319,994 observations in 1999/2001 with observation deletion; 5,163,428 in 1996–2001 with the match/stayer-hybrid correction. Colors identify samples in every panel. Time panels show separate package medians and median paired ratios from three calls. Setup-inclusive time includes package setup and fresh pool creation; PSS covers the same interval and apportions shared pages. Non-atomic sampling can miss brief peaks; these are not exact minimum RAM requirements. Dotted timing ratios exclude package setup. Upper vertical axes use powers-of-two log ticks; lower axes are linear, with parity at one. The memory-ratio axis extends to 1.5 to retain the one-thread result. Stata uses $\min(T, 4)$ processors, native `fevc` T threads, and Matlab T single-threaded workers. Each package pair uses the same host and input; hosts can differ between the sample/deletion sweeps, while hardware is fixed within each thread sweep.

within stayers' matches. The comparator is the KSS Matlab package, not the original Econometrica replication code. The complete sweep runs on one host at one, four, seven, 14, and 28 threads, with one warm-up and three measured calls per package and thread count. All 40 reported calls pass the output checks.

Six-year median setup-inclusive times fall from 551.9 and 1,585.5 seconds for `fevc` and Matlab at one thread to 104.5 and 278.8 seconds at 28 threads. Paired speedups are 2.39–2.88, and estimator-only comparisons also favor `fevc` throughout. Peak PSS is 7.59 versus 5.20 GiB at one thread (ratio 1.46) and 16.26 versus 49.63 GiB at 28 threads (ratio 0.33). In the same memory measurement, the summed resident set size of Matlab's 28 workers is 61.99 GiB; apportioning shared pages reduces its measured footprint by about 20%, while the private memory of its workers remains. For the two-period sample with observation deletion, `fevc` uses 4.07 versus 3.99 GiB at one thread but less PSS at four through 28 threads, reaching 8.90 versus 39.76 GiB at 28. Times fall from 172.0 and 887.2 seconds at one thread to 25.2 and 232.4 seconds at 28, and paired speedups span 5.16–9.11. Technical Supplement Tables S7.3 and S7.2 report both samples, including both deletion modes for the two-period input.

In the two-period sweep, which runs on a single host, the memory gap widens with the number of threads, consistent with Rust threads sharing solver storage that Matlab's separate worker processes may duplicate (section 3.6); each point rests on one memory pair. Because the data are confidential, the replication materials contain only aggregate diagnostics. Across 15 measured six-year pairs, the maximum absolute difference between the two packages' corrected targets after normalization is 0.005622. Because the random streams and finite-probe formulas differ, this is a performance comparison, not a demonstration of equal numerical precision.

6 Conclusions

`fevc` brings the KSS leave-out correction into Stata and makes explicit the choices that give the variance components their meaning: the population described, the weights, the treatment of controls, and the blocks within which errors may be correlated. The public Veneto example shows why the correction matters. It halves the estimated variance of firm effects and changes the sign of the worker–firm covariance.

For the default mean-centered calculation, numerical Monte Carlo standard errors approximate the variation from both stages of random directions and help decide whether more probes are needed. In the public example, the default mean centering reduces the firm-variance MCSE to about one-quarter of its uncentered value. Sampling uncertainty and finite-probe bias require separate assessment, and inference for the components remains experimental.

The Rust backend makes these calculations practical on large linked datasets. Across the six synthetic graph classes, it is faster than the KSS Matlab package in setup-inclusive time in every tested configuration and uses less measured memory under match deletion. Pool startup contributes materially to this advantage, but at 6.4 million observations it remains when setup is excluded; Matlab is faster during estimation alone in one single-thread configuration. In the 1996–2001 Veneto panel with stayers, `fevc` is faster at every tested thread count, including when setup is excluded; its measured memory is higher at one thread but lower at four through 28 threads. Where site policies permit ado and Mata code but restrict native plugins, the source-only Mata backend provides an alternative within Stata, subject to the supported requests described in section 3.

References

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

Andrews, M. J., L. Gill, T. Schank, and R. Upward (2008). High wage workers and low wage firms: Negative assortative matching or limited mobility bias? *Journal of the Royal Statistical Society: Series A 171*(3), 673–697.

Bérgolo, M., R. Ceni, M. Fondo, and D. Vergara (2026). Minimum wages and the distribution of firm wage premia. Working paper, July 2026 version; first circulated February 2025.

Bonhomme, S., K. Holzheu, T. Lamadon, E. Manresa, M. Mogstad, and B. Setzler (2023). How much should we trust estimates of firm effects and worker sorting? *Journal of Labor Economics 41*(2), 291–322.

Card, D., J. Heining, and P. Kline (2013). Workplace heterogeneity and the rise of West German wage inequality. *Quarterly Journal of Economics 128*(3), 967–1015.

Clark, R., J.-F. Houde, and X. Zhu (2025). Long-term vertical contracts, geography, and the persistence of brand shares. Working paper, May 5, 2025 version; previously circulated as "Share Dispersion of National Brands: The Role of Retailers".

Cornelissen, T. (2008). The Stata command `felsdvreg` to fit a linear model with two high-dimensional fixed effects. *Stata Journal 8*, 170–189.

Correia, S. (2014). REGHDFE: Stata module to perform linear or instrumental-variable regression absorbing any number of high-dimensional fixed effects. Statistical Software Components S457874, Boston College Department of Economics.

Dahis, R. (2026). Growing downward: Persistent inequality in publishing in economics. Working paper, June 9, 2026 version.

Guimarães, P. and P. Portugal (2010). A simple feasible procedure to fit models with high-dimensional fixed effects. *Stata Journal 10*, 628–649.

High Dimensional Economics Lab (2024). VarianceComponentsHDFE.jl. Software repository.

Hutchinson, M. F. (1989). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics—Simulation and Computation 18*, 1059–1076.

Johnson, W. B. and J. Lindenstrauss (1984). Extensions of lipschitz mappings into a hilbert space. In *Conference in Modern Analysis and Probability*, Volume 26 of *Contemporary Mathematics*, pp. 189–206. Providence, RI: American Mathematical Society.

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

Kline, P., R. Saggio, and M. Sølvsten (2021). Improved stochastic approximation of regression leverages for bias correction of variance components. Technical note, January 21; revised October 2026.

Koutis, I., G. L. Miller, and D. Tolliver (2011). Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *Computer Vision and Image Understanding 115*(12), 1638–1646.

Kwok, J. H. (2024). How do primary care physicians influence healthcare? evidence on practice styles and switching costs from Medicare. Working paper, May 18, 2024 version.

Lachowska, M., I. Sorkin, and S. A. Woodbury (2025). Employers and unemployment insurance take-up. *American Economic Review 115*(8), 2529–2573.

Lamadon, T. and A. A. Oppenheimer (2022). PyTwoWay: Two-way models in Python. Software repository.

Moghani, V. (2026). *LeaveOutKSS: Leave-Out Variance Component Estimation for Two-Way Fixed Effects Models*. Comprehensive R Archive Network.

Saggio, R. (2024). LeaveOutTwoWay: Bias-corrected estimates of variance components in two fixed-effects models. Software repository.

Schmieder, J. (2026a). CMG: Combinatorial multigrid in Rust. Software repository, https://github.com/johannes-schmieder/CMG.

Schmieder, J. (2026b). Technical supplement to FEVC: Leave-out bias-corrected variance decompositions in Stata. Technical supplement, Boston University. October. Accompanies the FEVC article.

A Essential formulas and conditions

This appendix uses the notation of section 2: $S_{xx} = X'X$, A_t for a target matrix, $P = XS_{xx}^{-1}X'$, and $M = I - P$. All design and target objects refer to the retained population, and frequencies and target mass are distinct. None, Mean, and Corrected denote the three `centering()` options of section 2.3. The complete definitions and proofs are in Schmieder (2026b, Technical Supplement, Sections S1–S3).

A.1 Block correction

For a deletion block g, let $M_{gg} = I - X_g S_{xx}^{-1} X_g'$ and $B_{t,gg} = X_g S_{xx}^{-1} A_t S_{xx}^{-1} X_g'$. In an identified coefficient representation,

$$r_g = M_{gg}^{-1}(y_g - X_g\widehat{\beta}), \qquad \widehat{\theta}_t^{\text{None}} = \widehat{\beta}' A_t \widehat{\beta} - \sum_g y_g' B_{t,gg} r_g. \tag{A.1}$$

Existence requires every leave-out design to have full column rank, equivalently $M_{gg} \succ 0$. Conditional mean zero, finite second moments, and uncorrelated errors across blocks give exact unbiasedness; the covariance within a block is unrestricted. Replacing y_g by $y_g - \bar{y}\mathbf{1}_g$ in the correction defines Mean and generally changes its expectation.

A.2 Corrected centering

For observation deletion let $d_i = M_{ii} > 0$, $\Delta_0 = \text{diag}(d_i)$ and $s_i = (y_i - \bar{y})r_i$. The exact Corrected calculation solves

$$(\Delta_0 - M/n)\widetilde{\sigma}^2 = \Delta_0 s, \qquad \widehat{\theta}_t^{\text{Corrected}} = \widehat{\beta}' A_t \widehat{\beta} - \sum_i B_{t,ii}\widetilde{\sigma}_i^2. \tag{A.2}$$

The system is positive definite when $n - k \geq 2$. The convenient positive-weight auxiliary regression additionally requires $nd_i > 1$. For constant-regressor matches the existence condition is instead $G - k \geq 2$, with physical match sizes retained in both the mean and correction. With general blocks, let J be the block-indicator matrix, m_g the block size, and $a_g = \mathbf{1}_g' M_{gg}\mathbf{1}_g/m_g$. The full correction requires

$$\mathcal{K} = n\,\text{diag}(a_g) - J'MJ \succ 0 \quad \Longleftrightarrow \quad \text{rank}(MJ) \geq 2. \tag{A.3}$$

Section S2 gives the two product orientations, right-hand side, covariance recovery and positive-weight reduction. These conditions concern exact inverse actions. They do not assert finite-probe unbiasedness or interval validity, and a nearly singular correction can amplify numerical and sampling error. Shift invariance requires $Xb_0 = \mathbf{1}$ and $A_t b_0 = 0$.

A.3 Implemented finite-projection adjustment

Following Kline et al. (2021), first consider the case without separately estimated controls. For R Rademacher directions define $U_b = (Pq_b)_i^2$, $V_b = (Mq_b)_i^2$, their means $\bar{p}, \bar{m}$, normalized shares $\widehat{h} = \bar{p}/(\bar{p}+\bar{m})$, $\widehat{m} = 1-\widehat{h}$, and raw moments $a = \overline{U^2}$, $b_2 = \overline{V^2}$, $c_2 = \overline{UV}$. The leading bias, variance and adjusted inverse are

$$\begin{aligned} \widehat{b}_i &= \{\widehat{m}a - \widehat{h}b_2 + (\widehat{m}-\widehat{h})c_2\}/R, \\ \widehat{v}_i &= \{\widehat{m}^2 a + \widehat{h}^2 b_2 - 2\widehat{h}\widehat{m}c_2\}/R, \\ F_i &= \widehat{m}^{-1} + \widehat{b}_i\widehat{m}^{-2} - \widehat{v}_i\widehat{m}^{-3}. \end{aligned} \tag{A.4}$$

Section S3.2 supplies the exact-control and block versions and the reciprocal-denominator conditions. The adjustment removes a leading expansion term, not all finite-probe bias. With Corrected, the randomized calculation adds the split-pool centering increment in equation (S3.19). Its reported MCSE is that of Mean on the same probes and excludes the variance of the increment and its covariance with the Mean estimate; see Section S3.5.

A.4 Projection covariance

Let $L' = (R_p'W_pR_p)^{-1}R_p'W_pE$, where R_p contains full-rank projection covariates, W_p their weights, and $E\beta$ the selected effects. With $S = XS_{xx}^{-1}L$ and deleted residual r_g from (A.1),

$$\widehat{\pi} = L'\widehat{\beta}, \qquad \widehat{V}_\pi = \sum_g S_g' \,\mathrm{sym}(z_g r_g')S_g. \tag{A.5}$$

Here $z_g = y_g$ for None or $y_g - \bar{y}\mathbf{1}_g$ for Mean. Under the maintained block assumptions and appropriate nuisance treatment, None has the exact covariance expectation; Mean generally does not. Centering leaves the slope unchanged. Normal intervals require both a normal limit and consistent studentization; see Technical Supplement Section S4.1.