Multiple-Treatment Regression Weights

This note extends the Angrist-style regression-weighting calculation from one binary treatment to settings with several treatment arms. The important point is that there are two distinct extensions.

First, Goldsmith-Pinkham, Hull, and Kolesar study the regression that includes several treatment indicators on the right-hand side at the same time. This is not just the Angrist weighted average repeated once for each treatment. The coefficient on one treatment contains its own weighted average effect and, in general, zero-sum weighted averages of the effects of the other treatments.

Second, Lal studies the one-at-a-time regression-adjustment approach. This removes the simultaneous-regression contamination term, but it still targets a treatment-specific overlap-weighted average rather than the unweighted ATE. Consequently, the ordering of regression-adjusted treatment coefficients can differ from the ordering of ATEs.

The algebra below is written in the finite-cell language used in `applications/Angrist1998.lean`: cells x have mass m_x, treatment probabilities are cell-specific, and conditional mean effects can vary by cell.

The Lean file is `applications/MultipleTreatments.lean`. It deliberately keeps `applications/Angrist1998.lean` self-contained: the multiple-treatment file imports Angrist for the binary primitives, then builds the Goldsmith-Pinkham, Hull, and Kolesar simultaneous-regression result and the Lal one-at-a-time result on top of those primitives.

The formalization is finite-sample algebra. It does not prove a sampling limit, an asymptotic approximation, or a causal identification theorem from primitive random variables. Instead, it proves the exact algebra obtained after imposing saturated cells, cell masses, cell propensities, conditional mean potential outcomes, and an invertible residual treatment covariance matrix. Positivity and probability-normalization assumptions are not needed for the matrix identities themselves; they are needed only when interpreting the resulting weights as convex averages or probability weights.

1. Baseline: One Binary Treatment. Let $D \in \{0, 1\}$, let X be a finite cell, and write

$$p_x = \Pr(D = 1 \mid X = x), \qquad \tau_x = E[Y(1) - Y(0) \mid X = x].$$

After residualizing D and Y on cell indicators, the population regression coefficient is

$$\beta = \frac{\sum_x m_x p_x (1 - p_x) \tau_x}{\sum_x m_x p_x (1 - p_x)}.$$

Thus the coefficient is a convex average of conditional treatment effects, with weights proportional to within-cell treatment variance:

$$\omega_x = \frac{m_x p_x (1 - p_x)}{\sum_z m_z p_z (1 - p_z)}.$$

This is the Angrist overlap-weight result. It is convex because $m_x p_x (1 - p_x) \geq 0$, and it puts no weight on cells with no within-cell treatment variation.

2. Simultaneous Multiple-Treatment Regression. Goldsmith-Pinkham, Hull, and Kolesar consider regressions with multiple treatment indicators included jointly. Let treatment take values $D \in \{0, 1, \ldots, K\}$, where 0 is the control arm. Define treatment indicators

$$T_k = 1\{D = k\}, \qquad k = 1, \ldots, K.$$

The population regression residualizes the vector $T = (T_1, \ldots, T_K)$ and the outcome on controls or cell indicators. In the saturated finite-cell case, the within-cell covariance matrix of the treatment indicators is

$$\Sigma_x = \operatorname{diag}(p_x) - p_x p_x',$$

where $p_x = (p_{1x}, \ldots, p_{Kx})'$ and $p_{kx} = \Pr(D = k \mid X = x)$. The population residual covariance matrix is

$$\Gamma = \sum_x m_x \Sigma_x.$$

If $\tau_x = (\tau_{1x}, \ldots, \tau_{Kx})'$ collects treatment effects relative to control,

$$\tau_{kx} = E[Y(k) - Y(0) \mid X = x],$$

then the vector of simultaneous-regression coefficients is

$$\beta = \Gamma^{-1} \sum_x m_x \Sigma_x \tau_x.$$

This is the vector analogue of the Angrist numerator-denominator formula. The difference is that Σ_x is no longer a scalar variance. It has off-diagonal entries

$$(\Sigma_x)_{k\ell} = -p_{kx} p_{\ell x}, \qquad k \neq \ell,$$

for mutually exclusive arms. These off-diagonal terms are the source of contamination.

For the k-th coefficient,

$$\beta_k = \sum_{\ell=1}^{K} \sum_x \lambda_{k\ell}(x) \tau_{\ell x},$$

where

$$\lambda_{k\ell}(x) = m_x e_k' \Gamma^{-1} \Sigma_x e_\ell.$$

Summing the weights over cells gives

$$\sum_x \lambda_{k\ell}(x) = e_k' \Gamma^{-1} \left(\sum_x m_x \Sigma_x \right) e_\ell = e_k' \Gamma^{-1} \Gamma e_\ell = 1\{k = \ell\}.$$

So the own-treatment weights sum to one, while the weights on every other treatment's effects sum to zero:

$$\sum_x \lambda_{kk}(x) = 1, \qquad \sum_x \lambda_{k\ell}(x) = 0 \quad \text{for } \ell \neq k.$$

This is the clean algebraic content of the Goldsmith-Pinkham, Hull, and Kolesar decomposition. The k-th coefficient equals an own-treatment weighted average plus contamination terms from the other treatment effects:

$$\beta_k = \underbrace{\sum_x \lambda_{kk}(x) \tau_{kx}}_{\text{own-treatment component}} + \underbrace{\sum_{\ell \neq k} \sum_x \lambda_{k\ell}(x) \tau_{\ell x}}_{\text{contamination components}}.$$

The contamination weights integrate to zero, not one. Therefore they cannot be interpreted as convex averaging weights. They disappear under special cases, for example if the other treatment's conditional effects are constant across cells, because a zero-sum weighting of a constant is zero. In general they do not disappear.

This explains why the single-treatment Angrist result does not mechanically extend to simultaneous multiple-treatment regressions. With one treatment, Σ_x is the scalar $p_x(1 - p_x)$ and there are no off-diagonal covariance terms. With multiple mutually exclusive treatments, the residualized treatment indicators are mechanically related within cells, and the inverse covariance matrix propagates those relationships across all coefficients.

Formalized Finite-Cell Statement. The finite-cell theorem to formalize is a matrix identity. Assume:

- a finite cell type C with masses m_c;
- a finite treatment type with K active arms and one omitted control arm;
- cell propensities $p_c : \{1, \ldots, K\} \to \mathbb{R}$;
- conditional effects $\tau_c : \{1, \ldots, K\} \to \mathbb{R}$;
- $\Gamma = \sum_c m_c(\mathrm{diag}(p_c) - p_c p_c')$ is invertible.

Define

$$\beta = \Gamma^{-1} \sum_c m_c(\mathrm{diag}(p_c) - p_c p_c') \tau_c.$$

Then for each active arm k,

$$\beta_k = \sum_{\ell=1}^{K} \sum_c m_c e_k' \Gamma^{-1} (\mathrm{diag}(p_c) - p_c p_c') e_\ell \tau_{\ell c},$$

with the normalization identity

$$\sum_c m_c e_k' \Gamma^{-1}(\mathrm{diag}(p_c) - p_c p_c') e_\ell = 1\{k = \ell\}.$$

This theorem is mostly linear algebra: expand the kth coordinate of the vector equation and then use $\Gamma^{-1}\Gamma = I$. The econometric interpretation is attached afterwards: the $\ell = k$ block is the own weighted component, while the $\ell \neq k$ blocks are contamination components.

The Lean implementation is in `applications/MultipleTreatments.lean`:

- `cellTreatmentCovariance` defines Σ_c.
- `residualTreatmentCovariance` defines Γ.
- `simultaneousCoefficient` defines β.
- `simultaneousWeight` defines the cell-arm weights $\lambda_{k\ell}(c)$.
- `simultaneousCoefficient_eq_sum_weights` proves the coefficient expansion.
- `simultaneousWeight_sum_cell`, `simultaneousWeight_sum_cell_self`, and `simultaneousWeight_sum_cell_ne` prove the one-or-zero normalization.

Here is the Lean shape of the Goldsmith-Pinkham, Hull, and Kolesar block. The finite cell type is `c`, the active treatment-arm type is `k`, and the omitted control arm is implicit. The type `k` indexes only active treatments $1, \ldots, K$.

```
variable {c k : Type*}
variable [Fintype c] [Fintype k] [DecidableEq k]
```

The assumptions `[Fintype c]` and `[Fintype k]` tell Lean that all cell and arm sums are finite. The assumption `[DecidableEq k]` lets Lean decide whether two active arms are equal, which is needed to write diagonal entries like `if i = j then ... else ...`.

The within-cell treatment covariance matrix is defined entrywise:

```
noncomputable def cellTreatmentCovariance (p : c → k → ℝ) (x : c) : Matrix k k ℝ :=
  fun i j => (if i = j then p x i else 0) - p x i * p x j
```

This is the Lean version of

$$\Sigma_x = \mathrm{diag}(p_x) - p_x p_x'.$$

The entrywise definition is intentionally simple for Lean. We still expose the usual mathematical shape through simp and helper theorems:

```
@[simp] theorem cellTreatmentCovariance_apply
theorem cellTreatmentCovariance_apply_self
theorem cellTreatmentCovariance_apply_ne
```

These say, respectively,

$$\Sigma_x(j, \ell) = 1\{j = \ell\} p_{jx} - p_{jx} p_{\ell x},$$

$$\Sigma_x(j, j) = p_{jx}(1 - p_{jx}),$$

and, when $j \neq \ell$,

$$\Sigma_x(j, \ell) = -p_{jx} p_{\ell x}.$$

The aggregate covariance matrix is the cell-mass weighted sum:

```
noncomputable def residualTreatmentCovariance
    (cellMass : c → ℝ) (p : c → k → ℝ) : Matrix k k ℝ :=
  ∑ x : c, cellMass x • cellTreatmentCovariance p x
```

Mathematically this is

$$\Gamma = \sum_x m_x \Sigma_x.$$

The coefficient vector is defined as

```
noncomputable def simultaneousCoefficient
    (cellMass : c → ℝ) (p τ : c → k → ℝ)
    [Invertible (residualTreatmentCovariance cellMass p)] : k → ℝ :=
  invOf (residualTreatmentCovariance cellMass p) *ᵥ
    ∑ x : c, cellMass x • (cellTreatmentCovariance p x *ᵥ τ x)
```

Lean's Unicode notation invOf means `invOf`, and `*ᵥ` means matrix-vector multiplication. The typeclass assumption `[Invertible (residualTreatmentCovariance cellMass p)]` is the finite-sample full-rank condition. It is the Lean-side version of assuming Γ^{-1} exists.

The coefficient weights are then

```
noncomputable def simultaneousWeight
    (cellMass : c → ℝ) (p : c → k → ℝ)
    [Invertible (residualTreatmentCovariance cellMass p)] (j l : k) (x : c) :
    ℝ :=
  cellMass x *
    ((invOf (residualTreatmentCovariance cellMass p) *
      cellTreatmentCovariance p x) j l)
```

This is exactly

$$\lambda_{j\ell}(x) = m_x(\Gamma^{-1}\Sigma_x)_{j\ell}.$$

Lean Proof Of The Coefficient Expansion. The theorem

```
theorem simultaneousCoefficient_eq_sum_weights
```

proves

$$\beta_j = \sum_{\ell}\sum_{x} \lambda_{j\ell}(x)\tau_{\ell x}.$$

The proof has three conceptual steps.

First, a private helper lemma pushes matrix-vector multiplication through the finite cell sum:

```
private theorem mulVec_sum_smul_mulVec
```

It proves

$$A\left(\sum_x m_x B_x v_x\right) = \sum_x m_x (AB_x)v_x.$$

In Lean this is just `Matrix.mulVec_sum`, `Matrix.mulVec_smul`, and `Matrix.mulVec_mulVec`.

Second, the theorem unfolds `simultaneousCoefficient` and applies that helper with $A = \Gamma^{-1}$, $B_x = \Sigma_x$, and $v_x = \tau_x$.

Third, Lean expands the jth coordinate and swaps the finite sums over cells and arms using `Finset.sum_comm`. That is the point where the vector equation becomes the displayed double sum over ℓ and x.

The proof is short in the final file, but that concision hides the actual Lean work: most of the effort is getting the finite sums, scalar multiplication, and matrix-vector coordinates into exactly the same normal form.

Lean Proof Of Weight Normalization. The theorem

```
theorem simultaneousWeight_sum_cell
```

proves

$$\sum_x \lambda_{j\ell}(x) = 1\{j = \ell\}.$$

The proof again uses one private helper:

```
private theorem matrix_mul_sum_smul_apply
```

It proves the entrywise identity

$$\left(A\sum_x m_x B_x\right)_{j\ell} = \sum_x m_x (AB_x)_{j\ell}.$$

Applying this with $A = \Gamma^{-1}$ and $B_x = \Sigma_x$ rewrites the sum of weights as

$$\left(\Gamma^{-1}\sum_x m_x \Sigma_x\right)_{j\ell} = (\Gamma^{-1}\Gamma)_{j\ell}.$$

The remaining Lean step is the full-rank condition:

`invOf_mul_self (residualTreatmentCovariance cellMass p)`

which proves $\Gamma^{-1}\Gamma = I$. Reading off the (j,ℓ) entry of the identity matrix gives `if j = l then 1 else 0`. The wrappers `simultaneousWeight_sum_cell_self` and `simultaneousWeight_sum_cell_ne` state the two econometric cases directly: own weights sum to one, and contamination weights sum to zero.

3. One-at-a-Time Regression Adjustment. Lal studies the alternative approach of estimating treatment-arm effects one at a time. Instead of putting all treatment indicators into one regression, fix an arm k and run a binary-treatment adjustment for that arm. In its cleanest control-comparison version, restrict attention to units with $D \in \{0, k\}$ and define

$$Q_k = 1\{D = k\}.$$

Within that pairwise comparison sample, let

$$q_{kx} = \Pr(Q_k = 1 \mid X = x, D \in \{0, k\}).$$

The same FWL argument as in the Angrist calculation gives

$$\beta_k^{\text{one}} = \frac{\sum_x m_x^{(k)} q_{kx}(1-q_{kx})\tau_{kx}}{\sum_x m_x^{(k)} q_{kx}(1-q_{kx})},$$

where $m_x^{(k)} = \Pr(X = x, D \in \{0, k\})$. Equivalently, in terms of the original cell probabilities p_{0x} and p_{kx},

$$m_x^{(k)} q_{kx}(1-q_{kx}) = m_x \frac{p_{0x}p_{kx}}{p_{0x}+p_{kx}},$$

whenever $p_{0x} + p_{kx} > 0$.

Thus the one-at-a-time coefficient is again an overlap-weighted average of k's conditional effect relative to control:

$$\beta_k^{\text{one}} = \sum_x \omega_{kx}^{\text{one}} \tau_{kx},$$

with

$$\omega_{kx}^{\text{one}} = \frac{m_x^{(k)} q_{kx}(1-q_{kx})}{\sum_z m_z^{(k)} q_{kz}(1-q_{kz})}.$$

These weights are nonnegative and sum to one. This is why one-at-a-time regression adjustment avoids the Goldsmith-Pinkham, Hull, and Kolesar contamination term: after fixing k, the treatment is binary again.

The Lean implementation reuses the Angrist binary primitives:

- `Lal.pairwiseCellMass` defines $m_x^{(k)}$.
- `Lal.pairwisePropensity` defines q_{kx}.
- `Lal.oneAtATimeCoefficient` is the pairwise binary regression coefficient.
- `Lal.oneAtATimeOverlapEffect` is the corresponding overlap-weighted estimand.
- `Lal.oneAtATimeCoefficient_eq_overlapEffect` proves the one-at-a-time coefficient is the Angrist overlap estimand.
- `Lal.pairwise_overlapWeight_eq_original` rewrites the pairwise overlap mass as $m_x p_{0x}p_{kx}/(p_{0x}+p_{kx})$ when the pairwise denominator is nonzero.

The key design choice is that the Lal block does not reprove Angrist. It specializes the Angrist finite-cell theorem to the pairwise comparison sample. The namespace is nested under `MultipleTreatments`:

```
namespace Lal
```

The pairwise sample mass is

```
def pairwiseCellMass (cellMass p0 pk : g → ℝ) (x : g) : ℝ :=
  cellMass x * (p0 x + pk x)
```

This is $m_x^{(k)} = m_x(p_{0x} + p_{kx})$.

The pairwise treatment probability is

```
noncomputable def pairwisePropensity (p0 pk : g → ℝ) (x : g) : ℝ :=
  pk x / (p0 x + pk x)
```

This is

$$q_{kx} = \frac{p_{kx}}{p_{0x} + p_{kx}}.$$

The one-at-a-time coefficient is then not a new regression object. It is exactly the Angrist binary coefficient applied to the pairwise mass and pairwise propensity:

```
noncomputable def oneAtATimeCoefficient
    (cellMass p0 pk y0 yk : g → ℝ) : ℝ :=
  Angrist1998.cellRegressionCoefficient
    (pairwiseCellMass cellMass p0 pk) (pairwisePropensity p0 pk) y0 yk
```

Likewise, the target estimand is exactly the Angrist overlap estimand applied to the same pairwise objects:

```
noncomputable def oneAtATimeOverlapEffect
    (cellMass p0 pk y0 yk : g → ℝ) : ℝ :=
  Angrist1998.overlapWeightedTreatmentEffect
    (pairwiseCellMass cellMass p0 pk) (pairwisePropensity p0 pk) y0 yk
```

The theorem

```
theorem oneAtATimeCoefficient_eq_overlapEffect
```

then proves the Lal one-at-a-time target with one unfold and one call to the Angrist theorem:

```
exact Angrist1998.cellRegressionCoefficient_eq_overlapWeightedTreatmentEffect
```

This is the formal payoff from keeping Angrist self-contained. The pairwise Lal result becomes a reuse theorem, not a parallel proof.

The theorem

```
theorem pairwise_overlapWeight_eq_original
```

connects the pairwise notation back to the original multi-arm cell probabilities:

$$m_x^{(k)} q_{kx}(1 - q_{kx}) = m_x \frac{p_{0x} p_{kx}}{p_{0x} + p_{kx}}.$$

The Lean proof is exactly the algebra one would do on paper: unfold the definitions, clear the nonzero denominator with `field_simp`, and finish the polynomial identity with `ring`.

But avoiding contamination does not mean recovering the ATE. The target is

$$\mathrm{WATE}_k = \sum_x \omega_{kx}^{\mathrm{one}} \tau_{kx},$$

not

$$\mathrm{ATE}_k = \sum_x m_x \tau_{kx}.$$

The weights depend on the overlap pattern for treatment k versus control. Different treatments can therefore be averaged over different parts of the covariate distribution.

4. Lal's Ranking-Reversal Point. The ranking issue is easiest to state by normalizing the one-at-a-time weights. Let $\gamma_k(X)$ be the normalized overlap weight for treatment k, so that $E[\gamma_k(X)] = 1$. Then

$$\mathrm{WATE}_k = E[\gamma_k(X)\tau_k(X)].$$

Since $E[\gamma_k(X)] = 1$,

$$\mathrm{WATE}_k = E[\tau_k(X)] + \mathrm{Cov}(\gamma_k(X), \tau_k(X)).$$

Therefore

$$\mathrm{WATE}_j - \mathrm{WATE}_k = (\mathrm{ATE}_j - \mathrm{ATE}_k) + [\mathrm{Cov}(\gamma_j(X), \tau_j(X)) - \mathrm{Cov}(\gamma_k(X), \tau_k(X))]\,.$$

If $\mathrm{ATE}_j > \mathrm{ATE}_k$ but $\mathrm{WATE}_j < \mathrm{WATE}_k$, the regression-adjusted ranking reverses the ATE ranking. Algebraically, the reversal condition is

$$\mathrm{Cov}(\gamma_j(X), \tau_j(X)) - \mathrm{Cov}(\gamma_k(X), \tau_k(X)) < -(\mathrm{ATE}_j - \mathrm{ATE}_k).$$

This is the sense in which Lal's problem is different from the simultaneous multiple-treatment problem. The one-at-a-time regression does not mix in the effects of other arms through zero-sum contamination weights. Instead, each arm gets its own overlap-weighted estimand. Rankings can still fail because the estimands being ranked are not the ATEs and are not averaged with common weights.

The Lean implementation records this bookkeeping as:

- `Lal.weightedAverage_eq_ate_add_covariance`, the finite-cell identity $\mathrm{WATE}_k = \mathrm{ATE}_k$ plus the weight-effect deviation term.
- `Lal.weightedRankingGap_eq_ateGap_add_covarianceGap`, the corresponding two-treatment ranking-gap identity.

The ranking theorems are intentionally stated as algebraic identities over finite cells. The theorem name uses "covariance" because that is the econometric reading, but the Lean statement is slightly more primitive. It proves

$$\sum_x m_x \gamma_x \tau_x = \sum_x m_x \tau_x + \sum_x m_x(\gamma_x - 1)\tau_x.$$

When the cell masses sum to one and the normalized weights satisfy $\sum_x m_x \gamma_x = 1$, the last term is the usual covariance-style weight-effect deviation:

$$\sum_x m_x(\gamma_x - 1)\tau_x = \mathrm{Cov}_m(\gamma_x, \tau_x).$$

Lean does not need those normalization assumptions for the algebraic identity itself. This is a useful theorem shape: later we can add probability-weighted corollaries without changing the core finite-sum proof.

The ranking theorem applies this identity twice:

```
rw [weightedAverage_eq_ate_add_covariance cellMass weightJ τJ,
  weightedAverage_eq_ate_add_covariance cellMass weightK τK]
ring
```

After rewriting both WATEs into ATE plus deviation, the final theorem is just the ring identity

$$(a + c) - (b + d) = (a - b) + (c - d).$$

This is the formal version of the Lal ranking-reversal decomposition. The substantive content is not that Lean discovers a reversal; it proves that any ranking gap in the one-at-a-time regression estimands decomposes exactly into an ATE gap plus the difference in the two treatment-specific overlap deviations.

5. What Is And Is Not Proved. The formalized results are exact finite-cell identities:

- Goldsmith-Pinkham, Hull, and Kolesar simultaneous regression: the coefficient vector equals $\Gamma^{-1}\sum_x m_x\Sigma_x\tau_x$; each coordinate expands into own and contamination blocks; the own block weights sum to one and the contamination block weights sum to zero.
- Lal one-at-a-time regression: each pairwise coefficient is the Angrist overlap-weighted estimand for the corresponding binary comparison.
- Lal ranking accounting: the difference between two one-at-a-time WATEs equals the ATE difference plus the difference in treatment-specific overlap deviations.

The file does not yet prove:

- nonnegativity of the Goldsmith-Pinkham, Hull, and Kolesar own weights under additional restrictions;
- any asymptotic convergence statement for sample OLS;
- primitive causal identification from random variables, sigma-algebras, or conditional independence assumptions;
- the final rank-reversal inequality as a named theorem.

Those would be natural corollaries or later layers. The core algebra is now in place, and it is deliberately stated at a level where both papers can reuse it.

6. Formalization Status. The natural sequence is now:

1. Finish the existing Angrist finite-cell theorem for one binary treatment. This is in `applications/Angrist1998.lean` as `cellRegressionCoefficient_eq_overlapWeightedTreatmentEffect`.
2. Prove the Goldsmith-Pinkham, Hull, and Kolesar matrix decomposition. This is in `applications/MultipleTreatments.lean` as `simultaneousCoefficient_eq_sum_weights`:

$$\beta_k = \sum_{\ell=1}^{K}\sum_c m_c e_k'\Gamma^{-1}\Sigma_c e_\ell \tau_{\ell c}.$$

   The key normalization is

$$\sum_c m_c e_k'\Gamma^{-1}\Sigma_c e_\ell = 1\{k=\ell\}.$$

   The normalization theorem `simultaneousWeight_sum_cell` separates own-treatment weights from contamination weights.

3. Prove the one-at-a-time pairwise overlap result for each arm k. This is in `applications/MultipleTreatments.lean` as `Lal.oneAtATimeCoefficient_eq_overlapEffect`:

$$\beta_k^{\text{one}} = \frac{\sum_x m_x^{(k)} q_{kx}(1-q_{kx})\tau_{kx}}{\sum_x m_x^{(k)} q_{kx}(1-q_{kx})}.$$

   This is a thin reuse of the Angrist theorem after instantiating the binary treatment as $Q_k = 1\{D=k\}$ inside the k-versus-control comparison sample.

4. Prove the ranking identity. This is in `applications/MultipleTreatments.lean` as `Lal.weightedRankingGap_eq_ateGap_add_`

$$\text{WATE}_j - \text{WATE}_k = \text{ATE}_j - \text{ATE}_k + \text{Cov}(\gamma_j,\tau_j) - \text{Cov}(\gamma_k,\tau_k).$$

   The rank-reversal condition is still a direct inequality rearrangement from this identity.

The conceptual progression is therefore:

Angrist → simultaneous multiple treatments with contamination → one-at-a-time overlap WATEs → ranking reversals from treatment-

References.

- Angrist (1998), "Estimating the Labor Market Impact of Voluntary Military Service Using Social Security Data on Military Applicants."
- Goldsmith-Pinkham, Hull, and Kolesar, "Contamination Bias in Linear Regressions." Local reference: `refs/2106.05024v5.pdf`.
- Lal, "Does Regression Produce Representative Causal Rankings?" Local reference: `refs/regrank.pdf`.