# Sparse linear models in high-dimensions

**Marcelo Ortiz-Villavicencio**

EMORY UNIVERSITY

**February 23, 2024**

Econometrics Reading Group

# Roadmap

1. Sparse linear models in high-dimensions
   - Problem Formulation
   - Different Sparsity Models
2. Shrinkage Estimators and Regularizers
3. The Regularization Bias
4. Orthogonalization
   - Double Lasso
   - Simulation Study

# Sparse linear models in high-dimensions

# Introduction

- The goal of this chapter is to provide an overview of the most popular used *shrinkage estimators* in the machine learning literature and how are they useful in the context of *linear regression*q models from the perspective of econometrics analysis.
- Shrinkage estimators provide a feasible approach to potentially identify *relevant variables* from a *large pool* of covariates.
- So our fundamental problem is a *linear model* with a large parameter vector that *potentially* contains many zeros (i.e., sparsity).
- The main assumption is that, while the number of covariates is large, perhaps much larger than the number of observations, the number of *associated non-zero* coefficients is relatively small.
- We can extend this framework even for nonparametric models (e.g., kernel ridge regressions).
- Applications: IVs from a many potentially weak instruments.

# Problem Formulation

EMORY UNIVERSITY

- Let $\theta^* \in \mathbb{R}^d$ be an *unknown vector*, referred to as the regression vector.
- Suppose that we observe a vector $y \in \mathbb{R}^n$ and a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ that are linked via the standard linear model

$$y = \mathbf{X}\theta^* + w$$

  where $w \in \mathbb{R}^n$ is a vector of noise variables.
- This model can also be written in a scalarized form: for each index $i = 1, 2, \ldots, n$, we have $y_i = \langle x_i, \theta^* \rangle + w_i$, where $x_i^T \in \mathbb{R}^d$ is the $i$-th row of $\mathbf{X}$, and $y_i$ and $w_i$ are (respectively) the $i$-th entries of the vectors $y$ and $w$.
- The quantity $\langle x_i, \theta^* \rangle := \Sigma_{j=1}^d, x_{ij}\theta_j^*$ denotes the usual *Euclidean inner product* between the vector $x_i \in \mathbb{R}^d$ of predictors (or covariates), and the regression vector $\theta^* \in \mathbb{R}^d$.
- Thus, each response $y_i$ is a noisy version of a linear combination of $d$ covariates.

# Different Sparsity Models

- As we know when $d > n$ it's impossible to obtain any meaningful estimates of $\theta^*$ unless we impose a *low dimensional structure*.
- Key concept: Sparsity
- Let us define the *support set* of $\theta^*$ as

$$S(\theta^*) := \left\{ j \in \{1, 2, \ldots, d\} \mid \theta_j^* \neq 0 \right\},$$

- This set has cardinality $s := |S(\theta^*)|$ *substantially smaller* than $d$.
- Assuming that the model is exactly supported on $s$ coefficients may be overly restrictive, in which case it is also useful to consider various relaxations of hard sparsity, which leads to the notion of weak sparsity.
- Roughly speaking, a vector $\theta^*$ is *weakly sparse* if it can be *closely approximated* by a *sparse vector*.

# $\ell_q$-balls

- There are different ways in which to formalize such an idea, one way being via the $\ell_q$- "norms".
- For a parameter $q$ and radius $R_q > 0$, consider the set

$$\mathbb{B}_q\left(R_q\right) = \left\{\theta \in \mathbb{R}^d \mid \sum_{j=1}^{d} |\theta_j|^q \leq R_q\right\}.$$

- It is known as the $\ell_q$-ball of radius $R_q$.

# Shrinkage Estimators and Regularizers

# Shrinkage Estimators and Regularizers

EMORY UNIVERSITY

- The *size* of a parameter vector $\theta$ is the *number of elements* in the vector and the *length* of $\theta$ is the length of the vector as measured by an assigned *norm*.
- The $\ell_q$ norm of a vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)'$ denoted by the notation, $\|\boldsymbol{\theta}\|_q$, is defined as

$$\|\boldsymbol{\theta}\|_q \left( \sum_{j=1}^{d} |\theta_i|^q \right)^{1/q} \qquad q > 0,$$

  where $|\theta|$ denotes the absolute value of $\theta$.
- When $q = 2$, the $\ell_q$ norm is known as the *Euclidean Norm*, or *Euclidean Distance*

**Example**

Let $\boldsymbol{\theta} = (\theta_1, \theta_2)$, then the $\ell_2$ Euclidean norm of $\boldsymbol{\theta}$ is $\|\boldsymbol{\theta}\|_2 = \sqrt{|\theta_1|^2 + |\theta_2|^2}$.

# Shrinkage Estimators and Regularizers

- The idea of a shrinkage estimator is to impose a *restriction on the length* of the estimated parameter vector $\hat{\theta}$.
- In other words, the idea is to *shrink* the parameter vector in order to identify the 0 elements in $\theta$.
- This can be framed as the following optimization problem:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\text{argmin}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{X})$$
$$\text{s.t. } pen(\boldsymbol{\theta}) \leq c,$$

  where $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{X})$ is the loss function, and $pen(\boldsymbol{\theta})$ is a *penalty* term or *regularizer* for any $c > 0$.
- Different definitions of $pen(\boldsymbol{\theta})$ lead to different shrinkage estimators.
- Let's write the previous optimization problem in its Lagrange form

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\text{argmin}} \mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{X}) + \lambda pen(\boldsymbol{\theta})$$

EMORY UNIVERSITY

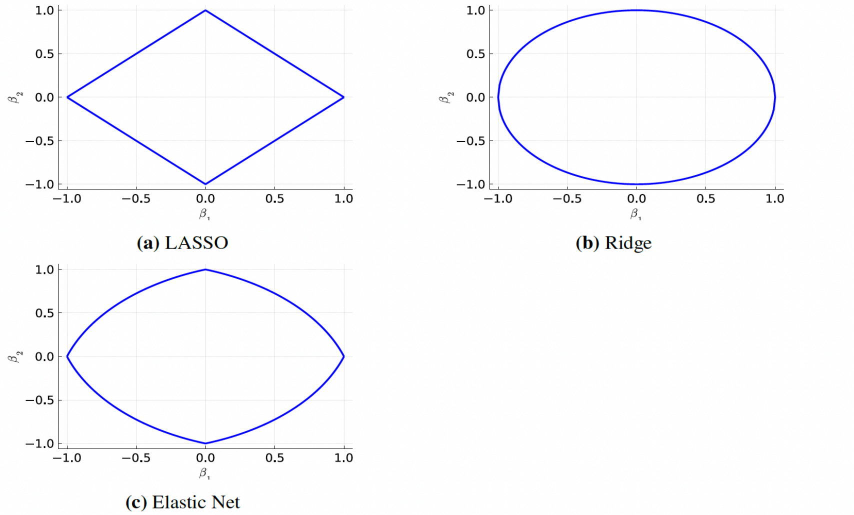
- A interesting class of regularizers is called the *Bridge estimator* as defined by Frank and Friedman (1993), which proposed the following regularizer in the equation

$$pen(\boldsymbol{\theta}; q) = \|\boldsymbol{\theta}\|_q^q = \sum_{j=1}^{d} |\theta_j|^q, \quad q \in \mathbb{R}^+.$$

  The Bridge estimator encompasses at least two shrinkage estimators as special cases.
- When $q = 1$, the Bridge estimator becomes the *Least Absolute Shrinkage and Selection Operator* (LASSO) as proposed by Tibshirani (1996)
- When $q = 2$, the Bridge estimator becomes the *Ridge estimator* as defined by Hoerl and Kennard (1970b, 1970a).
- We can define further a linear combination of LASSO and Ridge, which is called *Elastic Net*:

$$pen(\boldsymbol{\theta}; \alpha) = \sum_{j=1}^{d} \alpha |\theta_j| + (1 - \alpha) |\theta_j|^2$$

# Lasso, Ridge and Elastic Net




**(a)** LASSO

**(b)** Ridge

**(c)** Elastic Net

# Lasso, Ridge and Elastic Net

- An advantage of the $\ell_1$ norm i.e., LASSO, is that it can produce estimates with exactly zero values, i.e., elements in $\hat{\theta}$ can be exactly zero. This means we will have *corner solutions*.
- While the $\ell_2$ norm, i.e., Ridge, does not usually produce estimates with values that equal exactly 0. Ridge contour does not have the *sharp corners*.
- However, the Ridge does have a *computational advantage* over other variations of the Bridge estimator. $\implies$ Closed Form Solution

**Proposition (Closed Form Solution of Ridge)**

*When $q = 2$ and $\mathcal{L}(\theta; \mathbf{y}, \mathbf{X}) = (\mathbf{y} - \mathbf{X}\theta)^\top(\mathbf{y} - \mathbf{X}\theta)$ i.e., mean-square loss function, there is a closed form solution, namely $\hat{\theta}_{Ridge} = \left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}$*

- We can generalize this even for functions using *Kernel Ridge Regressions* and *RKHS* learning theory in nonparametric estimation! (more on this in 2 chapters)

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \left( \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \right)$$

# The Regularization Bias

# Lasso and Post-Lasso

- Let's focus in Lasso in this section.
- Denote by $\mathcal{L}(\theta) = n^{-1} \sum_{i=1}^{n} \left(Y_i - X_i^\top \theta\right)^2$ the mean-square loss function.
- The Lasso estimator is defined as:

$$\widehat{\theta} \in \underset{\theta \in \mathbb{R}^d}{\text{argmin}} \mathcal{L}(\theta) + \lambda \|\theta\|_1 .$$

- The Lasso minimizes the sum of the empirical mean-square loss and a penalty or regularization term $\lambda \|\theta\|_1$.
- Notice that the solution to previous program is not necessarily unique.
- $\lambda$ sets the trade-off between *fit* and *sparsity*
- Caution: In presence of a high-dimensional $\theta_0$ (true parameter) for which the *sparsity assumption* is not assumed to hold, using the Lasso estimator is not a good idea.

# Lasso and Post-Lasso

- Lasso has a cousin called *Post-Lasso*.
- This algorithm has been studied at Belloni and Chernozhukov (2011) and Belloni and Chernozhukov (2013)
- It is a *two-step estimator* in which a second step is added to the Lasso procedure in order to remove the bias that comes from shrinkage.
- That second step consists in running an OLS regression using only the covariates associated with a non-zero coefficient in the Lasso step.
    1. Run the Lasso regression and denote $\hat{s}(\theta)$ the estimator of the *support set* of $\theta$, i.e., the non-zero Lasso coefficients.
    2. Run an OLS regression including only the covariates corresponding to the *non-zero coefficients* in $\hat{s}(\theta)$ from above.

# The Regularization Bias

- A natural appeal of the Post-Lasso estimator is that it is a powerful tool for variable selection.
- However, we have to discuss the *regularization bias* which is nothing more than an *omitted variable bias* arising from the same mechanism described previously.

**Remark**

*Model selection and estimation cannot be achieved optimally at the same time.*

- Yang (2005) shows that for any model selection procedure to be consistent, it must behave *sub-optimally* for estimating the regression function and vice-versa.

# The bias of simple plug-in estimators in linear models

**Example (Linear model with high-dimensional controls)**

Consider the iid sequence of random variables $(Y_i, D_i, X_i)_{i=1}^n$ such that:

$$Y_i = D_i\theta_0 + X_i^\top \beta_0 + \varepsilon_i,$$

with $\varepsilon_i$ such that $\mathbb{E}[\varepsilon] = 0, \mathbb{E}[\varepsilon]^2 = \sigma^2 < \infty$ and $\varepsilon_i \perp (D_i, X_i)$. $\theta_0$ capture the treatment effect of a binary treatment $D_i \in \{0, 1\}$. $X_i$ is of dimension $d > 1$. $d$ is allowed to be much larger than $n$ and to grow with $n$. Denote by $\mu_d := \mathbb{E}(X \mid D = d)$ for $d \in \{0, 1\}$ and $\pi_0 := \mathbb{E}[D]$.

- In this example we are interested in estimate *treatment effect* $\theta_0$
- So $\beta_0$ is just a *nuisance parameter*.

# The bias of simple plug-in estimators in linear models

Two-step estimator:

1. Run a *Lasso regression* of $Y$ on $D$ and $\mathbf{X}$, forcing $D$ to remain in the model by excluding $\theta_0$ from the penalty part in the Lasso. Exclude all the elements in $\mathbf{X}$ that correspond to a zero coefficients $\hat{\beta}^{\text{lasso}}$
2. Run an *OLS regression* of $Y$ on $D$ and the *set of selected* $\mathbf{X}$ to obtain the post-selection estimator $\hat{\theta}^{\text{post}}$

- Denote $\widehat{\beta}$ the corresponding estimator for $\beta_0$ obtained in step 2. Notice that for $j \in \{1, \ldots, d\}$, if $\widehat{\beta}_j^{\text{lasso}} = 0$ then $\widehat{\beta}_j = 0$.
- Also denote by $\widehat{\pi} := n^{-1} \sum_{i=1}^{n} D_i$. Therefore,

$$\widehat{\theta} := \frac{\frac{1}{n} \sum_{i=1}^{n} D_i \left( Y_i - X_i^\top \widehat{\beta} \right)}{\widehat{\pi}} = \frac{1}{n_1} \sum_{D_i=1} \left( Y_i - X_i^\top \widehat{\beta} \right),$$

where $n_d := \sum_{i=1}^{n} \mathbf{1}\{D_i = d\}, d \in \{0, 1\}$.

# Regularization Bias of $\hat{\theta}$

EMORY UNIVERSITY

**Lemma**

*Under the previous linear model, if* $\mu_1 \neq 0$*, then* $\sqrt{n}\left(\widehat{\theta} - \theta_0\right) \to \infty$

**Sketch of the proof:** Substitute the linear model in the expression of $\hat{\theta}$ to get

$$\sqrt{n}\left(\widehat{\theta} - \theta_0\right) = \widehat{\pi}^{-1}\left[\frac{1}{n}\sum_{i=1}^{n} D_i X_i\right]^\top \sqrt{n}\left(\beta_0 - \widehat{\beta}\right) + \widehat{\pi}^{-1}\sqrt{n}\left[\frac{1}{n}\sum_{i=1}^{n} D_i \varepsilon_i\right]$$

By CLT, CMT, LLN and Slutsky

$$\left[\frac{1}{n}\sum_{i=1}^{n} D_i X_i\right] \xrightarrow{p} \pi_0 \mu_1 .$$

$$\widehat{\pi}^{-1}\left[\frac{1}{\sqrt{n}}\sum_{i=1}^{n} D_i \varepsilon_i\right] \xrightarrow{d} \mathcal{N}\left(0, \frac{\sigma^2}{\pi_0}\right)$$

$$\left|\left[\frac{1}{n}\sum_{i=1}^{n} D_i X_i\right]' \sqrt{n}\left(\beta_0 - \widehat{\beta}\right)\right| \approx s\sqrt{\log d} \to \infty$$

# Orthogonalization

# Orthogonalization

- This is the main idea in Chernozhukov et al. (2017); Belloni et al. (2017); Chernozhukov et al. (2018).
- To build the intuition, assume that the parameter of interest, $\theta_0$ solves the equation $\mathbb{E}m\left(Z_i, \theta_0, \beta_0\right) = 0$ for some known *score function* $m(\cdot)$, a vector of observables $Z_i$ and nuisance parameter $\beta_0$.
- In the simplest case, think about the *score function* as the first derivative of the log-likelihood functions in the parametric case.
- From our example: $Z_i = (Y_i, D_i, X_i)$, and $m\left(Z_i, \theta, \beta\right) := \left(Y_i - D_i\theta - X_i^\top \beta\right) D_i$.
- The derivative of the estimating moment with respect to nuisance parameter is not zero:

$$\mathbb{E}\partial_\beta m\left(Z_i, \theta_0, \beta_0\right) = -\pi_0\mu_1 \neq 0.$$

Idea: Can we replace $m(\cdot)$ by another score function $\psi(\cdot)$ and use a different nuisance parameter $\eta_0$ such that

$$\mathbb{E}\partial_\eta \psi\left(Z_i, \theta_0, \eta_0\right) = 0$$

# Neyman Orthogonality

- We say that any function $\psi$ that satisfies previous condition is an *orthogonal score* or *Neyman-Orthogonal*
- Intuition: The moment condition for estimating $\theta_0$ is not affected by small perturbations around the true value of the nuisance parameter $\eta_0$.
- Changing the estimating moment can neutralize the effect of the first step estimation and suppress the *regularization bias*.

# Asymptotic Normality of Orthogonal Estimator

EMORY UNIVERSITY

## Assumption (Orthogonal Moment Condition)

*The (scalar) parameter of interest, $\theta_0$ is given by:*

$$\mathbb{E}\psi\left(Z_i, \theta_0, \eta_0\right) = 0$$

*for some known real-valued function $\psi(\cdot)$ satisfying the orthogonality condition, a vector of observables $Z_i$ and a high-dimensional sparse nuisance parameter $\eta_0$ such that $\|\eta_0\|_0 \leq s$.*

## Assumption (High-Quality Nuisance Estimation)

*Let first-step estimator $\widehat{\eta}$ such that with high-probability:*

$$\begin{aligned} \|\widehat{\eta}\|_0 &\lesssim s \\ \|\widehat{\eta} - \eta_0\|_1 &\lesssim \sqrt{s^2 \log d / n} \\ \|\widehat{\eta} - \eta_0\|_2 &\lesssim \sqrt{s \log d / n} \end{aligned}$$

# Asymptotic Normality of Orthogonal Estimator

EMORY UNIVERSITY

## Assumption (Affine-Quadratic Model)

*The function* $\psi(\cdot)$ *is such that:*

$$\psi\left(Z_i, \theta, \eta\right) = \Gamma_1\left(Z_i, \eta\right)\theta - \Gamma_2\left(Z_i, \eta\right)$$

*where* $\Gamma_j, j = 1, 2$*, are functions with all their second order derivatives with respect to* $\eta$ *constant over the convex parameter space of* $\eta$*.*

The estimator we are going to consider is $\check{\theta}$ such that:

$$\frac{1}{n}\sum_{i=1}^{n}\psi\left(Z_i, \check{\theta}, \widehat{\eta}\right) = 0.$$

## Theorem (Asymptotic Normality)

*The estimator* $\check{\theta}$ *in the affine-quadratic model and under previous assumptions:*
$\sqrt{n}\left(\check{\theta} - \theta_0\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma_\Gamma^2\right)$*, with* $\sigma_\Gamma^2 := \mathbb{E}\left[\psi\left(Z_i, \tau_0, \eta_0\right)^2\right] / \mathbb{E}\left[\Gamma_1\left(Z_i, \eta_0\right)\right]^2$*.*

# Intuition comes from FWL Theorem

- The Orthogonalization framework can be generalized for other ML learner algorithms.
- This is the main idea of *Double Machine Learning* (DML) (More on this in next chapters)
- DML builds on the FWL theorem to isolate the effect of interest, introducing a key idea: the use of ML models in the orthogonalization process.
  1. $\hat{D} = f(X) + v$
     $\Rightarrow \tilde{D} = D - \hat{X}$
  2. $\hat{Y} = g(X) + u$
     $\Rightarrow \tilde{Y} = Y - \hat{Y}$
  3. $\tilde{Y} = \theta_0 + \theta_1 \tilde{X} + \varepsilon$

EMORY UNIVERSITY

- We can define a Lasso procedure where *Neyman-Orthogonality* holds

The Double Lasso procedure:

1. We run Lasso regressions of $Y_i$ on $X_i$ and $D_i$ on $X_i$

$$\hat{\gamma}_{YX} = \arg\min_{\gamma \in \mathbb{R}^p} \quad \sum_i \left(Y_i - \gamma^\top X_i\right)^2 + \lambda_1 \sum_j \hat{\psi}_j^Y \left|\gamma_j\right|,$$
$$\hat{\gamma}_{DX} = \arg\min_{\gamma \in \mathbb{R}^p} \quad \sum_i \left(D_i - \gamma^\top X_i\right)^2 + \lambda_2 \sum_j \hat{\psi}_j^D \left|\gamma_j\right|,$$

   where $\hat{\psi}_j$ are penalty loadings normally equal to 1. Then, we obtain the resulting residuals:

$$\check{Y}_i = Y_i - \hat{\gamma}_{YX}^\top X_i,$$
$$\check{D}_i = D_i - \hat{\gamma}_{DX}^\top X_i.$$

   In place of Lasso, we can use Post-Lasso or other Lasso relatives.

2. We run the least squares regression of $\check{Y}_i$ on $\check{D}_i$ to the estimator $\check{\theta}$.

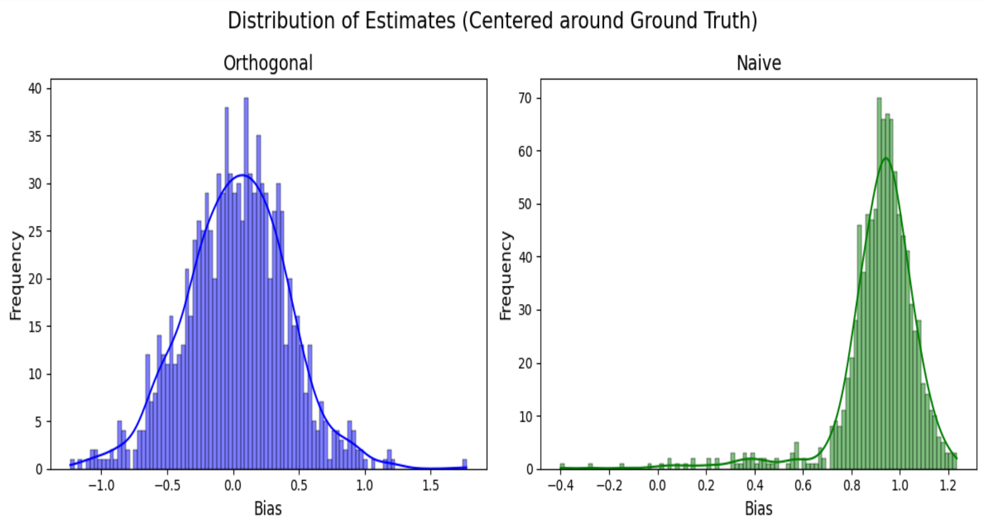
# Simulation Study

We compare the performance of the *naive* (e.g., Post-Lasso) and *orthogonal* methods (e.g., Double Lasso) in a computational experiment where $d = n = 100$, $\beta_j = 1/j^2, \gamma_j = 1/j^2$, and

$$Y = 1 \cdot D + \beta^\top X + \varepsilon_Y, \quad X \sim N(0, I), \varepsilon_Y \sim N(0, 1)$$
$$D = \gamma^\top X + \tilde{D}, \quad \tilde{D} \sim N(0, 1)/4$$

Here the true parameter is 1.

# Simulation Study



# Code

```
# Initialize constants
B = 1000  # Number of iterations
n = 100  # Sample size
d = 100  # Number of features

# Sim Parameters
mean = 0
sd = 1

# Initialize arrays to store results
naive = np.zeros(B)
orthogonal = np.zeros(B)
```

# Code

```
# Iterate through B simulations
for i in tqdm(range(B)):
    # Generate parameters:
    gamma = (1 / (np.arange(1, d + 1) ** 2)).reshape(d, 1)
    beta = (1 / (np.arange(1, d + 1) ** 2)).reshape(d, 1)

    # Generate covariates / random data
    X = np.random.normal(mean, sd, n * d).reshape(n, d)
    D = (X @ gamma) + np.random.normal(mean, sd, n).reshape(n, 1) / 4

    # Generate Y using DGP
    Y = D + (X @ beta) + np.random.normal(mean, sd, n).reshape(n, 1)

    # Single selection method using rlasso
    r_lasso_estimation = hdmpy.rlasso(np.concatenate((D, X), axis=1), Y, post=True)
    coef_array = r_lasso_estimation.est['coefficients'].iloc[2:, :].to_numpy()
    SX_IDs = np.where(coef_array != 0)[0]

    # Check if any X coefficients are selected
    if sum(SX_IDs) == 0:
        # If no X coefficients are selected, regress Y on D only
        naive[i] = sm.OLS(Y, sm.add_constant(D)).fit().params[1]
    else:
        # If X coefficients are selected, regress Y on selected X and D
        X_D = np.concatenate((D, X[:, SX_IDs]), axis=1)
        naive[i] = sm.OLS(Y, sm.add_constant(X_D)).fit().params[1]

    # Double Lasso Partialling Out
    resY = hdmpy.rlasso(X, Y, post=False).est['residuals']
    resD = hdmpy.rlasso(X, D, post=False).est['residuals']
    orthogonal[i] = sm.OLS(resY, sm.add_constant(resD)).fit().params[1]
```

# Thanks!

marcelo.ortiz@emory.edu
marcelortiz.com
@marcelortizv

# References

- Wainwright, M. J. (2019). *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press.
- Gaillac, C., & L'Hour, J. (2021). *Machine Learning for Econometrics. ENSAE Paris – IP Paris*.
- Ma, T. (2022). *Lecture Notes for Machine Learning Theory (CS229M/STATS214)*.