Machine Learning for Microeconometrics
Part 3: Causal Inference with Lasso

A. Colin Cameron
Univ.of California - Davis

.

May 2022

Course Outline

- **1:** Variable selection and cross validation
- **2.** Shrinkage methods
  - ridge, lasso, elastic net
- **Part 3: ML for causal inference using lasso**
  - **OLS with many controls, IV with many instruments**
- **4**. Other methods for prediction
  - nonparametric regression, principal components, splines
  - neural networks
  - regression trees, random forests, bagging, boosting
- **5.** More ML for causal inference
  - ATE with heterogeneous effects and many controls.
- **6.** Classification and unsupervised learning
  - classification (categorical y) and unsupervised learning (no y).

Introduction

- Consider the leading first work on inference with machine learning.
- This focuses on OLS and IV estimation of the partial linear model
  - using LASSO to select among potential controls and/or instruments
  - make assumption of sparsity.
- Work from 2010 on by Belloni, Chernozhukov and Hansen and their coauthors.
- This has been implemented in Stata version 16.

Overview

1.1 Partial Linear Model

- A **partial linear** control function model specifies

$$y = \mathbf{d}'\boldsymbol{\alpha} + g(\mathbf{x}_c) + u \text{ where } g(\cdot) \text{ is unknown.}$$

- Interest lies in estimating $\boldsymbol{\alpha}$
  - $\mathbf{d}$ are policy or treatment variables of interest
    - for simplicity we will later focus on the scalar case
  - $\mathbf{x}_c$ are "nuisance" control variables
  - $g(\cdot)$ is an unknown function
- Selection on observables assumption is made
  - consistent OLS estimation of $\boldsymbol{\alpha}$ requires $E[u|\mathbf{d}, \mathbf{x}_c] = 0$
  - this is more plausible the better is $g(\mathbf{x}_c)$.

1.2 Robinson (1988) Semiparametric Estimator

- Robinson (1988) proposed **semiparametric estimation**

$$y = \mathbf{d}'\boldsymbol{\alpha} + g(\mathbf{x}_c) + u, \quad E[u|\mathbf{x}_c] = 0$$

where $g(\cdot)$ is unknown.

- Then

$$\begin{aligned} E[y|\mathbf{x}_c] &= E[\mathbf{d}|\mathbf{x}_c]'\boldsymbol{\alpha} + g(\mathbf{x}_c) + 0 \\ y - E[y|\mathbf{x}_c] &= (d - E[\mathbf{d}|\mathbf{x}_c])'\boldsymbol{\alpha} + u \end{aligned}$$

- Estimate by OLS regression of kernel residuals on kernel residuals

$$u_{y|\mathbf{x}_c} = u'_{\mathbf{d}|\mathbf{x}_c}\boldsymbol{\alpha} + v$$

  - Kernel regression of y on $\mathbf{x}_c$ gives residual $u_{y|\mathbf{x}_c}$
  - Kernel regression of $\mathbf{d}$ on $\mathbf{x}_c$ gives residuals $u_{\mathbf{d}|\mathbf{x}_c}$
- OLS of $u_{y|\mathbf{x}_c}$ on $u_{\mathbf{d}|\mathbf{x}_c}$ gives root-N consistent asymptotically normal $\widehat{\boldsymbol{\alpha}}$.

Curse of Dimensionality

- The Robinson method entails kernel regression on a vector $\mathbf{x}_c$.
- So only works if $\mathbf{x}_c$ is of low dimension
  - e.g. $y =$ energy consumption; $\mathbf{d} =$ usual demand determinants; $\mathbf{x}_c$ is time of day (scalar).
- Instead we are interested in a high-dimensional set of controls $\mathbf{x}_c$
  - kernel regression fails due to the **curse of dimensionality**
    - ★ the sample size required for adequate local regression grows exponentially with the dimension of $\mathbf{x}_c$.
- Solution: use a machine learner rather than kernel regression
  - here use the LASSO instead of kernel regression
    - ★ requires a sparsity assumption
    - ★ and use of clever methods.

2.1 Partialling-out estimator

- Allow for complexity by assuming

$$g(\mathbf{x}_c) \simeq \mathbf{x}'\gamma + r$$

  where $\mathbf{x}$ consists of flexible transformations of $\mathbf{x}_c$ such as polynomials, interactions, splines, ... and r is an approximation error that disappears at appropriate rate.

- Then

$$y = \mathbf{d}'\boldsymbol{\alpha} + \mathbf{x}'\gamma + r + u.$$

- Belloni, Chernozhukov and coauthors have suggested several LASSO-based methods that yield root-N consistent and asymptotically normal estimates of $\boldsymbol{\alpha}$
  - we start with the partialling-out estimator
  - consider scalar d for simplicity.

2.1 Partialling-out Estimator

- Recall $y = \alpha \times d + \mathbf{x}'\gamma + r + u$.
- The method is similar to Robinson except use the LASSO not kernel regression
  - 1. Perform LASSO of d on $\mathbf{x}$ and obtain residual $\widehat{u}_d$ from OLS regression of d on the selected variables.
  - 2. Perform LASSO of y on $\mathbf{x}$ and obtain residual $\widehat{u}_y$ from OLS regression of y on the selected variables.
  - 3. Obtain $\widehat{\alpha}$ from OLS regression of $\widehat{u}_y$ on $\widehat{u}_d$.
- A key assumption is the sparsity assumption that the true model is small relative to the sample size N and grows at rate no more than $\sqrt{N}$.
  - $s/(\sqrt{N}/\ln p)$ should be small
  - $p = \dim(\mathbf{x})$ is the number of potential regressors
  - s is the number of variables in the true model.

2.2 Stata poregress command

- `poregress` depvar *varsofinterest*, options
  - *varsofinterest* is **d**
  - option `controls`([*alwaysvars*)] *othervars* splits **x** into controls to always include and controls to be selected by Stata
  - default option `plugin` determines the penalty λ by plug-in formula rather than by CV or adaptive CV.
- For independent heteroskedastic errors use the following.
- The plug-in penalty is $\lambda = c\sqrt{N}\Phi(1 - \frac{\gamma}{2p})$ where $c = 1.1$ and $\gamma = 0.1/\ln\{\max(p, N)\}$.
- LASSO has individual loadings for each regressor
  - $\kappa_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij}\widehat{\varepsilon}_i)^2}$ for normalized $\mathbf{x}_{ij}$ and $\widehat{\varepsilon}_i$ is a residual from a sequence of first-stage LASSOs.

2.3. Application of Partialling-out Estimator

- Data is 2003 data from the U.S. Medical Expenditure Panel Survey
  - people aged 65-90 years.
- Dependent variable `ltotexp` is log total medical expenditures.
- Regressor of interest is `suppins`
  - indicator variable for supplemental insurance beyond Medicare.
- Add many control variables to hopefully control for endogeneity of `suppins`
  - use LASSO to reduce number of control variables

Data and key variables

```
. * Data for inference on suppins example: 5 continuous and 13 binary variables
. use mus203mepsmedexp.dta, clear
(A.C.Cameron & P.K.Trivedi (2021): Microeconometrics using Stata, 2e)

. keep if ltotexp != .
(109 observations deleted)

. describe ltotexp suppins

              storage   display    value
variable name   type    format     label      variable label
---------------------------------------------------------------------------------
ltotexp         float   %9.0g                 ln(totexp) if totexp > 0
suppins         float   %9.0g                 =1 if has supp priv insurance

. summarize ltotexp suppins

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     ltotexp |      2,955    8.059866    1.367592   1.098612   11.74094
     suppins |      2,955    .5915398    .4916322          0          1
```

Continuous regressors

```
. * Continuous variables
. global xlist2 income educyr age famsze totchr

. describe $xlist2

              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------------------------------
income          double  %12.0g                annual household income/1000
educyr          double  %12.0g                Years of education
age             double  %12.0g                Age
famsze          double  %12.0g                Size of the family
totchr          double  %12.0g                # of chronic problems

. summarize $xlist2

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      income |      2,955    22.68353    22.60988         -1     312.46
      educyr |      2,955    11.82809    3.405095          0         17
         age |      2,955    74.24535    6.375975         65         90
      famsze |      2,955    1.890694    .9644483          1         13
      totchr |      2,955    1.808799    1.294613          0          7
```

Binary regressors

```
. * Discrete binary variables
. global dlist2 female white hisp marry northe mwest south ///
>     msa phylim actlim injury priolist hvgg

. describe $dlist2

              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------------------------------
female          double  %12.0g                =1 if female
white           double  %12.0g                =1 if white
hisp            double  %12.0g                =1 if Hispanic
marry           double  %12.0g                =1 if married
northe          double  %12.0g                =1 if northeast area
mwest           double  %12.0g                =1 if Midwest area
south           double  %12.0g                =1 if south area (West is excluded)
msa             double  %12.0g                =1 if metropolitan statistical area
phylim          double  %12.0g                =1 if has functional limitation
actlim          double  %12.0g                =1 if has activity limitation
injury          double  %12.0g                =1 if condition is caused by an accident/injury
priolist        double  %12.0g                =1 if has medical conditions that are on the priority list
hvgg            float   %9.0g                 =1 if health status is excellent, good or very good
```

OLS without & with products & cross products of controls

- Little change when add all the interactions

```
. * OLS on small model and full model
. global rlist2 c.($xlist2)##c.($xlist2) i.($dlist2) c.($xlist2)#i.($dlist2)

. qui regress ltotexp suppins $xlist2 $dlist2, vce(robust)

. estimates store OLSSMALL

. qui regress ltotexp suppins $rlist2, vce(robust)

. estimates store OLSFULL

. estimates table OLSSMALL OLSFULL, keep(suppins) b(%9.4f) se stats(N df_m r2)
```

Variable	OLSSMALL	OLSFULL
suppins	0.1706	0.1868
	0.0469	0.0478
N	2955	2955
df_m	19.0000	99.0000
r2	0.2682	0.3028

Partialling-out Lasso with plug-in lambda

- Estimate between preceding OLS estimates with similar standard error

```
. * Partialing-out partial linear model using default plugin lambda
. poregress ltotexp suppins, controls($rlist2)

Estimating lasso for ltotexp using plugin
Estimating lasso for suppins using plugin

Partialing-out linear model         Number of obs               =      2,955
                                    Number of controls          =        176
                                    Number of selected controls =         21
                                    Wald chi2(1)                =      15.43
                                    Prob > chi2                 =     0.0001

------------------------------------------------------------------------------
             |               Robust
     ltotexp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     suppins |   .1839193   .0468223     3.93   0.000     .0921493    .2756892
------------------------------------------------------------------------------
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

Lassoinfo

- 21 overall, 12 for y and 9 for d
  - so distinct variables chosen for y and d

```
. * Lasso information
. lassoinfo

    Estimate:  active
     Command:  poregress
```

Variable	Model	Selection method	lambda	No. of selected variables
ltotexp	linear	plugin	.080387	12
suppins	linear	plugin	.080387	9

lassoknots gives the variables chosen

- For y (`ltotexp`) `totchr`, `actlim`, `phylim` especially important.
- For d (`suppins`) `income` especially important.

```
. lassoknots, for(ltotexp)

---------------------------------------------------------------------------------------------------
     |                No. of                |
     |               nonzero In-sample      |                    Variables (A)dded, (R)emoved,
  ID |     lambda      coef. R-squared      |                          or left (U)nchanged
-----+--------------------------------------+------------------------------------------------------
 * 1 |    .080387         12    0.2390      | A totchr            0.actlim            c.age#c.totchr
     |                                      |   0.hisp#c.totchr   0.hvgg#c.totchr     1.white#c.totchr
     |                                      |   c.educyr#c.totchr 1.phylim#c.educyr   0.phylim#c.famsze
     |                                      |   0.actlim#c.famsze 0.female#c.totchr   1.priolist#c.educyr
---------------------------------------------------------------------------------------------------
* lambda selected by plugin assuming heteroskedastic errors.

. lassoknots, for(suppins)

---------------------------------------------------------------------------------------------------
     |                No. of                |
     |               nonzero In-sample      |                    Variables (A)dded, (R)emoved,
  ID |     lambda      coef. R-squared      |                          or left (U)nchanged
-----+--------------------------------------+------------------------------------------------------
 * 1 |    .080387          9    0.0809      | A age               income              1.hvgg#c.income
     |                                      |   0.hisp#c.educyr   1.marry#c.income    1.white#c.educyr
     |                                      |   0.marry#c.famsze  c.income#c.totchr   0.northe#c.income
---------------------------------------------------------------------------------------------------
* lambda selected by plugin assuming heteroskedastic errors.
```

Partialling out done manually

- The following gives same results as earlier `poregress`

```
. * Partialing out done manually
. qui lasso linear suppins $rlist2, selection(plugin)

. qui predict suppins_lasso, postselection

. qui generate u_suppins = suppins - suppins_lasso

. qui lasso linear ltotexp $rlist2, selection(plugin)

. qui predict ltotexp_lasso, postselection

. qui generate u_ltotexp = ltotexp - ltotexp_lasso

. regress u_ltotexp u_suppins, vce(robust) noconstant noheader
```

u_ltotexp	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
u_suppins	.1839193	.0468223	3.93	0.000	.0921117	.2757268

Cross validation instead of plugin lambda

- Cross validation selects 73 controls (40 for y and 50 for d).

```
. * Cross validation instead
. poregress ltotexp suppins, controls($rlist2) selection(cv) rseed(10101)

Estimating lasso for ltotexp using cv
Estimating lasso for suppins using cv

Partialing-out linear model          Number of obs                =      2,955
                                     Number of controls           =        176
                                     Number of selected controls  =         73
                                     Wald chi2(1)                 =      15.58
                                     Prob > chi2                  =     0.0001

------------------------------------------------------------------------------
             |               Robust
     ltotexp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     suppins |   .1852675   .0469368     3.95   0.000     .0932731    .2772619
------------------------------------------------------------------------------
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

2.4 Clustered Data

- Data are grouped with correlated observations within group and uncorrelated across groups
  - y_{ig} is outcome for individual i in cluster g, $i = 1, ..., N_g$, $g = 1, ..., G$.
- Two methods for the LASSO have objective function

$$\text{Method 1} \quad : \quad Q_\lambda(\boldsymbol{\beta}) = \frac{1}{G}\sum_{g=1}^{G}\sum_{i=1}^{N_g}(y_{ig} - \mathbf{x}'_{ig}\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

$$\text{Method 2} \quad : \quad Q_\lambda(\boldsymbol{\beta}) = \frac{1}{G}\sum_{g=1}^{G}\frac{1}{N_g}\sum_{i=1}^{N_g}(y_{ig} - \mathbf{x}'_{ig}\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

- Stata uses method 2.

Clustered Data (continued)

- Belloni, Chernozhukov, Hansen and Kozbur (2016), "Inference in High-Dimensional Panel Models with an Application to Gun Control", JBES, 590-606.
- Consider balanced panel model with fixed effects
  - mean difference data (y and $\mathbf{x}$ and possibly $\mathbf{z}$) to get rid of fixed effects
  - so now clustered data with fixed effects now eliminated.
- Then consider two uses of machine learning in the partial linear model
  - section 4.1: select subset of many potential instruments
  - section 4.2: select subset of many controls.
- They use as method 1. giving equal weight to all mean-differenced observations.

3.1. Orthogonalization defined

- Define $\boldsymbol{\alpha}$ as parameters of interest and $\boldsymbol{\eta}$ as nuisance parameters.
- Estimate $\widehat{\boldsymbol{\alpha}}$ is obtained following first step estimate $\widehat{\boldsymbol{\eta}}$ of $\boldsymbol{\eta}$
  - First stage: $\widehat{\boldsymbol{\eta}}$ solves $\sum_{i=1}^{n} \omega(\mathbf{w}_i, \boldsymbol{\eta}) = \mathbf{0}$
  - Second stage: $\widehat{\boldsymbol{\alpha}}$ solves $\sum_{i=1}^{n} \psi(\mathbf{w}_i, \boldsymbol{\alpha}, \widehat{\boldsymbol{\eta}}) = \mathbf{0}$.
- The distribution of $\widehat{\boldsymbol{\alpha}}$ is usually affected by the noise due to estimating $\boldsymbol{\eta}$
  - e.g. Heckman's two-step estimator in selection models.
- But this is not always the case
  - e.g. the asymptotic distribution of feasible GLS is not affected by first-stage estimation of variance model parameters to get $\widehat{\Omega}$.
- Result: The distribution of $\widehat{\boldsymbol{\alpha}}$ is unaffected by first-step estimation of $\boldsymbol{\eta}$ if the function $g(\cdot)$ satisfies
  - $E[\partial\psi(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\eta})/\partial\boldsymbol{\eta}] = \mathbf{0}$; see next slide.
- So choose functions $\psi(\cdot)$ that satisfy the orthogonalization condition

$$E[\partial\psi(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\eta})/\partial\boldsymbol{\eta}] = \mathbf{0}.$$

Orthogonalization (continued)

- Why does this work? By Taylor series expansion

$$
\begin{aligned}
& \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi(\mathbf{w}_i, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\eta}}) \\
= \; & \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \psi(\mathbf{w}_i, \boldsymbol{\alpha}_0, \boldsymbol{\eta}_0) + \frac{1}{n} \sum_{i=1}^{n} \left. \frac{\partial \psi(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\eta})}{\partial \boldsymbol{\alpha}'} \right|_{\boldsymbol{\alpha}_0, \boldsymbol{\eta}_0} \times \sqrt{n}(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}_0) \\
& + \frac{1}{n} \sum_{i=1}^{n} \left. \frac{\partial \psi(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\eta})}{\partial \boldsymbol{\eta}'} \right|_{\boldsymbol{\alpha}_0, \boldsymbol{\eta}_0} \times \sqrt{n}(\widehat{\boldsymbol{\eta}} - \boldsymbol{\eta}_0)
\end{aligned}
$$

- By a law of large numbers $\frac{1}{n} \sum_{i=1}^{n} \left. \frac{\partial \psi(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right|_{\boldsymbol{\alpha}_0, \boldsymbol{\eta}_0}$ converges to its expected value which is zero if $E[\partial \psi(\mathbf{w}_i, \boldsymbol{\alpha}, \boldsymbol{\eta}) / \partial \boldsymbol{\eta}] = \mathbf{0}$.
- So the term involving $\widehat{\boldsymbol{\eta}}$ drops out.
- For more detail see Cameron and Trivedi (2005, p.201).

3.2 Orthogonalization for partialling-out estimator

- Consider the partially linear model with scalar d and manipulate

$$\begin{array}{rrcll} & y & = & \alpha d + g(\mathbf{x}) + u & \text{where } E[u|d,\mathbf{x}] = 0 \\ \Rightarrow & E[y|\mathbf{x}] & = & \alpha E[d|\mathbf{x}] + g(\mathbf{x}) & \text{as } E[u|\mathbf{x}_2] = 0 \\ & y - E[y|\mathbf{x}] & = & \alpha(d - E[d|\mathbf{x}]) + u & \text{subtracting} \end{array}$$

- Robinson (1988) differencing estimator
  - use kernel methods to get $\widehat{E}[y|\mathbf{x}]$ and $\widehat{E}[d|\mathbf{x}]$
  - $\widehat{\alpha}$ from OLS regress $(y - \widehat{E}[y|\mathbf{x}])$ on $(d - \widehat{E}[d|\mathbf{x}])$
- Instead here use machine learning methods for $\widehat{E}[y|\mathbf{x}]$ and $\widehat{E}[d|\mathbf{x}]$.
- Recall that OLS of y on $\mathbf{x}$ has f.o.c. $\sum_i \mathbf{x}_i u_i = \mathbf{0}$
  - so is sample analog of population moment condition $E[\mathbf{x}u] = \mathbf{0}$.
- So partialling-out estimator therefore solves population moment condition
  - $E[(d - E[d|\mathbf{x}])\{y - E[y|\mathbf{x}] - (d - E[d|\mathbf{x}])\alpha\}] = 0.$

Orthogonalization for partialling-out estimator (continued)

- Partialling-out solves population condition $E[\psi(\cdot)] = 0$ where

$$\psi(\cdot) = (d - E[d|\mathbf{x}])\{y - E[y|\mathbf{x}] - (d - E[d|\mathbf{x}])\alpha\}.$$

- Define $\eta_1 = E[d|\mathbf{x}]$ and $\eta_2 = E[y|\mathbf{x}]$, so

$$\begin{aligned} \psi(w, \alpha, \boldsymbol{\eta}) &= (d - \eta_1)\{y - \eta_2 - (d - \eta_1)\alpha\} \\ &= (d - \eta_1)(y - \eta_2) - \alpha(d - \eta_1)^2\}. \end{aligned}$$

- Then differentiating

$$\begin{aligned} \partial\psi(\mathbf{w}, \alpha, \boldsymbol{\eta})/\partial\eta_1 &= -(y - \eta_2) + 2\alpha(d - \eta_1) \\ \partial\psi(\mathbf{w}, \alpha, \boldsymbol{\eta})/\partial\eta_2 &= -(d - \eta_1) \end{aligned}$$

- The orthogonalization condition $E[\partial\psi(\mathbf{w}, \alpha, \boldsymbol{\eta})/\partial\boldsymbol{\eta}] = 0$ hold as

$$\begin{aligned} E[-(y - \eta_2) + 2(d - \eta_1)\alpha|\mathbf{x}] &= -(E[y|\mathbf{x}] - \eta_2) + 2(E[d|\mathbf{x}] - \eta_1) \\ &= -(\eta_2 - \eta_2) + 2(\eta_1 - \eta_1) = 0 \\ \text{and } E[-(d - \eta_1)|\mathbf{x}] &= -(E[d|\mathbf{x}] - \eta_1) = \mathbf{0}. \end{aligned}$$

Orthogonalization for partialling-out estimator (continued)

- More formally $\eta_{1i} = E[d_{1i}|\mathbf{x}_{1i}]$ and $\eta_{2i} = E[d_{1i}|\mathbf{x}_{1i}]$ vary with i.
- A formal treatment deals with functionals $\eta_{1i} = \eta_1(\mathbf{x}_i), \eta_{2i} = \eta_2(\mathbf{x}_{2i})$
  - this allows a range of machine learners for d_i and y_i - not just lasso.
- For simplicity consider the linear case where

$$\eta_{1i} = E[d_i|\mathbf{x}] = \mathbf{x}_i'\boldsymbol{\pi}_1 \text{ and } \eta_{2i} = E[y_i|\mathbf{x}] = \mathbf{x}_i'\boldsymbol{\pi}_2$$

- Then

$$\begin{aligned} \psi(w_i, \alpha, \boldsymbol{\pi}) &= (d_i - \mathbf{x}_i'\boldsymbol{\pi}_1)\{y_i - \mathbf{x}_i'\boldsymbol{\pi}_2 - (d_i - \mathbf{x}_i'\boldsymbol{\pi}_1)\alpha\} \\ \partial\psi(w_i, \alpha, \boldsymbol{\pi})/\partial\boldsymbol{\pi}_2 &= -(d_i - \mathbf{x}_i'\boldsymbol{\pi}_1)\mathbf{x}_i \\ E[\partial\psi(w_i, \alpha, \boldsymbol{\pi})/\partial\boldsymbol{\pi}_2|\mathbf{x}_i] &= E[-(d_i - \mathbf{x}_i'\boldsymbol{\pi}_1)\mathbf{x}_i|\mathbf{x}_i] \\ &= -(\mathbf{x}_i'\boldsymbol{\pi}_1 - \mathbf{x}_i'\boldsymbol{\pi}_1)\mathbf{x}_i = \mathbf{0} \end{aligned}$$

- Similarly $E[\partial\psi(w_i, \alpha, \boldsymbol{\pi})/\partial\boldsymbol{\pi}_1] = 0$.

4.1 Cross-Fit Partialling-Out Estimator

- The preceding partialling out used the same data at the first stage as at the second stage.
- A better procedure uses different data in the first stage lassos to that used for the second stage estimation of α.
- Superficially this leads to a loss of precision in estimating α due to a smaller sample size
  - this is avoided by the following method.
- Split the sample into K folds and for fold $k = 1, ..., K$
  - use most data for LASSO estimation of nuisance part
    - ★ yields model for prediction $\widehat{d} = \mathbf{x}'\widehat{\boldsymbol{\pi}}_d^{(k)}$ and $\widehat{y} = \mathbf{x}'\widehat{\boldsymbol{\pi}}_y^{(k)}$
  - use remaining smaller data to get predicted residuals in fold k
    - ★ compute residuals $\widetilde{u}_d^{(k)} = d^{(k)} - \mathbf{x}^{(k)\prime}\widehat{\boldsymbol{\pi}}_d^{(k)}$ and $\widetilde{u}_y^{(k)} = y^{(k)} - \mathbf{x}^{(k)\prime}\widehat{\boldsymbol{\pi}}_y^{(k)}$.

Cross-Fit Partialling-Out Estimator (continued)

- Given vectors of residuals $\widetilde{u}_d^{(k)}$ and $\widetilde{u}_y^{(k)}$ in each of the K folds , $k = 1, .., K$ there are two ways to estimate α.
- 1. Combine all residuals into N residuals $\widetilde{u}_y$ and $\widetilde{u}_d$, regress and get $\widehat{\alpha}$
  - Stata default
- 2. For each $k = 1, ..., K$ obtain $\widehat{\alpha}^{(k)}$ from OLS of $\widetilde{u}_y^{(k)}$ on $\widetilde{u}_d^{(k)}$
  - then form the average $\widehat{\alpha} = \frac{1}{K} \sum_{k=1}^{K} \widehat{\alpha}^{(k)}$
  - there is little loss in efficiency as we average over K independent samples
- Cross-fit partialling out under either method 1. or 2. reduces the complications of data mining
  - it allows s to grow at rate N and not $\sqrt{N}$.

4.3 Stata xporegress command

- `xporegress` depvar *varsofinterest*, options
  - *varsofinterest* is **d**
- Option `controls`([*alwaysvars)] othervars* splits **x** into controls to always include and controls to be selected by Stata.
- Default option `plugin` determines the penalty λ by plug-in formula rather than by CV or adaptive CV.
  - default forms N residuals.
- Option `technique(dml1)` computes K estimates $\widehat{\alpha}^{(k)\prime}$ and averages.
- Option `resample(#)` of `xporegress` uses more than one K-fold split so results not dependent on the random split
  - should use in final results.

4.4 Cross-fitting partialling-out Application

- Leads to similar results.

```
. * Crossfit partialing out (double/debiased) using default plugin
. xporegress ltotexp suppins, controls($rlist2) rseed(10101) nolog

Cross-fit partialing-out              Number of obs                =      2,955
linear model                          Number of controls           =        176
                                      Number of selected controls  =         31
                                      Number of folds in cross-fit =         10
                                      Number of resamples          =          1
                                      Wald chi2(1)                 =      15.66
                                      Prob > chi2                  =     0.0001

------------------------------------------------------------------------------
             |               Robust
     ltotexp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     suppins |   .1856171   .0469096     3.96   0.000      .093676    .2775582
------------------------------------------------------------------------------
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

Selected variables across the folds

- Number of selected variables across the ten folds

```
. * Summarize the number of selected variables across the ten folds
. lassoinfo

    Estimate:  active
     Command:  xporegress
```

Variable	Model	Selection method	No. of selected variables min	median	max
ltotexp	linear	plugin	11	13	14
suppins	linear	plugin	7	9	11

4.5 Multiple Sample Splits

- The sample-splitting adds noise.
- To control for this can do the following
  - S times repeat the sample splitting method (e.g. $S = 500$)
  - each time get a $\widehat{\alpha}_s$ (from averaging the K $\widehat{\alpha}'_{ks}$) and $\widehat{\sigma}^2_s = Var[\widehat{\alpha}_s]$
- Then $\overline{\widehat{\alpha}} = \frac{1}{S}\sum_{s=1}^{S}\widehat{\alpha}_s$
- And $Var[\widehat{\alpha}] = \frac{1}{S}\sum_{s=1}^{S}\widehat{\sigma}^2_s + \frac{1}{S}\sum_{s=1}^{S}(\widehat{\alpha}_s - \overline{\widehat{\alpha}})^2$.
- This is option `resample(#)` of `xporegress`
  - should use in final results.

Multiple Splits Application

- This took along time and standard error is larger.

```
. xporegress ltotexp suppins, controls($rlist2) rseed(10101) nolog resample(10)

Cross-fit partialing-out              Number of obs               =      2,955
linear model                          Number of controls          =        176
                                      Number of selected controls =         40
                                      Number of folds in cross-fit =        10
                                      Number of resamples         =         10
                                      Wald chi2(1)                =      14.90
                                      Prob > chi2                 =     0.0001
```

ltotexp	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
suppins	.1814719	.0470151	3.86	0.000	.0893239	.2736199

```
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

5. Double Selection Estimator

- A third method for estimating the partial linear model
  - used in the Belloni et al 2014 JEP article.
- Recall $y = \alpha \times d + \mathbf{x}'\gamma + r + u$.
- The method is
  - 1. Perform LASSO of y on $\mathbf{x}$ and denote selected regressor $\mathbf{x}_y$
  - 2. Perform LASSO of d on $\mathbf{x}$ and denote selected regressor $\mathbf{x}_d$.
  - 3. Obtain $\widehat{\alpha}$ from OLS regression of y on d and the union of $\mathbf{x}_y$ and $\mathbf{x}_d$.
- Use Stata command `dsregress`.

Double Selection Estimator Application

- Double selection yields similar results to before.

```
. * Double selection partial linear model using default plugin lambda
. dsregress ltotexp suppins, controls($rlist2)

Estimating lasso for ltotexp using plugin
Estimating lasso for suppins using plugin

Double-selection linear model         Number of obs               =      2,955
                                      Number of controls          =        176
                                      Number of selected controls =         21
                                      Wald chi2(1)                =      15.30
                                      Prob > chi2                 =     0.0001

------------------------------------------------------------------------------
             |               Robust
     ltotexp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     suppins |   .1836224   .0469429     3.91   0.000      .091616    .2756289
------------------------------------------------------------------------------
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

6.1 Generalized Linear Models

- In economics we extend from OLS using the GMM framework
  - this handles both nonlinearity and endogeneity.
- In statistics the main extension is to commonly-used nonlinear models
  - **generalized linear models** (GLM) for independent data
  - generalized estimating equations (GEE) for clustered and panel data.
- GLM's specify $E[y|\mathbf{x}] = G(\mathbf{x}'\boldsymbol{\beta})$ for specified $G(\cdot)$
  - $G(a) = a$ for linear model
  - $G(a) = \exp(a)$ for Poisson for count y
  - $G(a) = \Lambda(a) = \frac{e^a}{1+e^a}$ for logit for binary y.
- For OLS, logit and Poisson the resulting estimating equations are

$$\begin{aligned} \textstyle\sum_{i=1}^n \{y_i - G(\mathbf{x}_i'\boldsymbol{\beta})\}\mathbf{x}_i &= \mathbf{0} \\ \textstyle\sum_{i=1}^n \text{residual}_i \times \text{regressors}_i &= \mathbf{0}. \end{aligned}$$

- And an **IV estimator** given $\mathbf{z}_i$ satisfying $E[\{y_i - G(\mathbf{x}_i'\boldsymbol{\beta})\}|\mathbf{z}_i] = 0$ is

$$\textstyle\sum_{i=1}^n \{y_i - G(\mathbf{x}_i'\boldsymbol{\beta})\}\mathbf{z}_i = \mathbf{0}.$$

Partialling-out for Generalized Linear Models

- Now consider a **partial linear GLM**

$$\begin{aligned} E[y|\mathbf{x}] &= G(\alpha d + g(\mathbf{x}_c)) \text{ for specified } G(\cdot) \\ &\simeq G(\alpha d + \mathbf{x}'\boldsymbol{\beta}) \text{ for specified } G(\cdot) \end{aligned}$$

- If $\boldsymbol{\beta}$ were known then we would estimate the scalar α as solving the single equation

$$\textstyle\sum_{i=1}^{n} \{y_i - G(\alpha d_i + \mathbf{x}_i'\boldsymbol{\beta})\} z_i = 0$$

  for some scalar "instrument" z_i.
- We instead estimate $\widetilde{\boldsymbol{\beta}}$ so the estimating equation for α is

$$\textstyle\sum_{i=1}^{n} \psi(\mathbf{w}_i, \alpha, \widetilde{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \{y_i - G(\alpha d_i + \mathbf{x}_i'\widetilde{\boldsymbol{\beta}})\} z_i = 0.$$

- The key is

Partialling-out for Generalized Linear Models

- The estimating equation for α is

$$\sum_{i=1}^{n} \psi(\mathbf{w}_i, \alpha, \widetilde{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \{y_i - G(\alpha d_i + \mathbf{x}_i' \widetilde{\boldsymbol{\beta}})\} z_i = 0.$$

- The partialling-out GLM estimator does the following
  - 1. Post-lasso logit of y_i on d_i and $\mathbf{x}_i$ gives first-stage $\widetilde{\alpha}$ and $\widetilde{\boldsymbol{\beta}}$.
  - 2. Construct an "instrument" z_i for d_i
    - ⋆ this is the tricky bit and is based on $\widetilde{\alpha}$ and $\widetilde{\boldsymbol{\beta}}$
  - 3. Estimator α solves the preceding sample moment condition.

Partialling-out for Generalized Linear Models (continued)

- The population moment condition is

$$E[\psi(\mathbf{w}, \alpha, \boldsymbol{\beta})] = E[\{y - G(\alpha d + \mathbf{x}'\boldsymbol{\beta})\} \times z] = 0.$$

- The hard part is constructing the "instrument" z from the $\mathbf{x}'s$ and $\widetilde{\alpha}$ and $\widetilde{\boldsymbol{\beta}}$
  - the instrument is relevant

$$E\left[\frac{\psi(\mathbf{w}, \alpha, \boldsymbol{\beta})}{\partial \alpha}\right] = E[\{y - G(\alpha d + \mathbf{x}_i'\boldsymbol{\beta})\} \times G'(\alpha d + \mathbf{x}_i'\boldsymbol{\beta}) \times d \times z] \neq 0$$

  - the instrument is such that the orthogonalization condition holds

$$E\left[\frac{\psi(\mathbf{w}, \alpha, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right] = E[\{y - G(\alpha d + \mathbf{x}_i'\boldsymbol{\beta})\} \times G'(\alpha d + \mathbf{x}_i'\boldsymbol{\beta}) \times \mathbf{x} \times z] = \mathbf{0}.$$

  - an "efficient" instrument is chosen.
- For details see Stata documentation and Belloni, Chernozhukov and Wei (2016), JBES, 606-609.

Logit Model

- Logit commands are `pologit`, `xpologit` and `dslogit`.
- Marginal effects are not identified as they depend on $\boldsymbol{\beta}$ and here we have only estimated α

$$\begin{aligned} \Pr[y=1|d,\mathbf{x}] &= \Lambda(\alpha \times d + \mathbf{x}'\boldsymbol{\beta}) \\ \frac{\partial}{\partial d}\Lambda(\alpha \times d + \mathbf{x}'\boldsymbol{\beta}) &= \alpha \times \Lambda'(\alpha \times d + \mathbf{x}'\boldsymbol{\beta}). \end{aligned}$$

- But logit coefficients have an odds ratio interpretation

$$\frac{p}{1-p} = \frac{\Pr[y=1|d,\mathbf{x}]}{\Pr[y=0|d,\mathbf{x}]} = \exp(\alpha \times d + \mathbf{x}'\boldsymbol{\beta})$$

- Example: $\alpha = 0.2$ then a one unit change in d increases the odds-ratio by a multiple $e^{0.2} = 1.22$.

Logit Model Application

- Define a binary outcome dy for whether or not totexp > 4000
  - then dy=1 for 42% of sample and dy=0 for 58%
  - so log-odds at $\bar{y}$ is $\bar{y}/(1-\bar{y}) = 0.72$.
  - here do both partialling-out and double selection

```
. * Logit variant of partial linear model and partialing-out estimator
. generate dy = totexp > 4000

. qui logit dy suppins $rlist2, or vce(robust)

. estimates store FULL

. qui pologit dy suppins, controls($rlist2) selection(plugin) coef

. estimates store PARTIALOUT

. qui dslogit dy suppins, controls($rlist2) coef

. estimates store DOUBSEL

. estimates table FULL PARTIALOUT DOUBSEL, keep(suppins) b(%9.4f) se ///
>     stats(N df_m k_controls_sel)
```

Variable	FULL	PARTIAL~T	DOUBSEL
suppins	0.2792	0.2632	0.2680
	0.0936	0.0892	0.0892
N	2955	2955	2955
df_m	99.0000		
k_controls~l		19.0000	19.0000

legend: b/se

Exponential Conditional Mean Model (Poisson)

- Note that Poisson regression is applicable to any model with exponential conditional mean
  - it is not restricted to counts or Poisson
  - but do be sure to use robust standard errors.
- Poisson commands are `popoisson`, `xpopoisson` and `dspoisson`.
- Marginal effects are not identified as they depend on $\boldsymbol{\beta}$ and here we have only estimated α

$$\begin{aligned} E[y|d, \mathbf{x}] &= \exp(\alpha \times d + \mathbf{x}'\boldsymbol{\beta}) \\ \frac{\partial}{\partial d} \exp(\alpha \times d + \mathbf{x}'\boldsymbol{\beta}) &= \alpha \times \exp(\alpha \times d + \mathbf{x}'\boldsymbol{\beta}). \end{aligned}$$

- But exponential coefficients have a semi-elasticity or multiplicative interpretation.
- Example: $\alpha = 0.2$ then a one unit change in d increases the conditional mean by a multiple $e^{0.2} = 1.22$.

6.2 Linear Instrumental Variables

- Consider a partial linear model where the treatment of interest is scalar (for simplicity).
- Add complication that the treatment is endogenous and just-identified (for simplicity).
- Problem is that if we add too many controls then we are more likely to have a weak instrument as the instrument has less incremental contribution after controlling for the exogenous variables.
- Solution is to extend earlier partialling-out to restrict number of controls.
- More generally `poivregress` command applies to multiple endogenous regressors ($\mathbf{d}$), regressors to always include ($\mathbf{w}$) and controls to reduce ($\mathbf{x}$). There are instruments $\mathbf{z}$ with $\dim[\mathbf{z}] \geq \dim[\mathbf{x}]$.

$$y = \mathbf{d}'\alpha + \mathbf{w}'\delta + \mathbf{x}'\gamma + u.$$

Partialling-out for Linear Instrumental Variables

- For simplicity consider scalar d and no variables to definitely include:

$$y = \alpha \times d + \mathbf{x}'\gamma + u.$$

- The partialling-out method is
  - 1. Perform LASSO of y on $\mathbf{x}$ and obtain residual $\widehat{u}_y$ from OLS regression of y on the selected variables.
  - 2. Do the following
    - ★ perform LASSO of d on $\mathbf{x}$ and obtain prediction $\widehat{d}$ from OLS of d on the selected variables
    - ★ perform LASSO of $\widehat{d}$ on $\mathbf{x}$ and obtain prediction $\widetilde{d}$ and residual $\widetilde{u}$ from OLS of $\widehat{d}$ on the selected variables
  - 3. Calculate $\widehat{u}_d = d - \widetilde{d}$ which has purged out the role of $\mathbf{x}$.
  - 4. Obtain $\widehat{\alpha}$ from IV regression of $\widehat{u}_y$ on $\widetilde{u}$ with instrument $\widehat{u}_d$.

Partialling-out IV Application

- Example from Acemoglu, Johnson and Robinson (2001), AER, 1369-1401
- Consider country GDP and role of secure institutions
  - y : `loggdp` (log PPP GDP per capita in 1995, World Bank)
  - d : `avexpr` (average protection against expropriation risk)
  - z : `logem4` (log settler mortality - a long time ago)
  - $\mathbf{x}$: measures of country latitude, temperature, humidity, soil types and natural resources.
- Problem: 24 potential controls and $n = 64$.

Data summary

- From output not given Cor(d, z) = Cor[avexpr,logem4)= -0.52.

```
. * Read in Acemoglu-Johnson-Robinson data and define globals
. qui use mus228ajr.dta, clear

. global xlist lat_abst edes1975 avelf temp* humid* steplow deslow ///
>     stepmid desmid drystep  drywint goldm iron silv zinc oilres landlock

. describe logpgp95 avexpr logem4

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------------------------
logpgp95        float   %9.0g                 log PPP GDP pc in 1995, World Bank
avexpr          float   %9.0g                 average protection against expropriation risk
logem4          float   %9.0g                 log settler mortality

. summarize logpgp95 avexpr logem4, sep(0)

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    logpgp95 |         64    8.062237    1.043359   6.109248   10.21574
      avexpr |         64    6.515625    1.468647        3.5         10
      logem4 |         64    4.657031    1.257984   2.145931   7.986165
```

poivregress results

- Across the various Lassos **five control variables are selected**.

```
. * Partialling-out IV using plugin for lambda
. poivregress logpgp95 (avexpr=logem4), controls($xlist) selection(plugin, hom)

Estimating lasso for logpgp95 using plugin
Estimating lasso for avexpr using plugin
Estimating lasso for pred(avexpr) using plugin

Partialing-out IV linear model      Number of obs                  =         64
                                    Number of controls             =         24
                                    Number of instruments          =          1
                                    Number of selected controls    =          5
                                    Number of selected instruments =          1
                                    Wald chi2(1)                   =       8.74
                                    Prob > chi2                    =     0.0031

------------------------------------------------------------------------------
             |               Robust
    logpgp95 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      avexpr |   .8798503   .2976286     2.96   0.003      .296509    1.463192
------------------------------------------------------------------------------
Endogenous:   avexpr
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
```

6.3 Belloni, Chernozhukov and Hansen (JEP, 2014)

- Belloni, Chernozhukov and Hansen (2014), "High-dimensional methods and inference on structural and treatment effects," *Journal of Economic Perspectives*, Spring, 29-50
- Accessible paper. Three applications using LASSO.
- 1. IV with excess of instruments and use LASSO to select subset.
  - Application to house prices (y) affected by takings law (d) with 147 potential instruments and $n = 184$. Lasso picked just one instrument.
- 2. OLS with excess of controls and use double selection method.
  - Application to crime rate (y) affected by abortion rate (d) with 284 controls and $n = 550$. Around 10 controls are selected.
- 3. Just-identified IV with single y, d and z. Three LASSOs of y, x and z on $\mathbf{x}$ and then use the union of the chosen $\mathbf{x}$'s as controls in IV of y on d with instrument d.
  - so like double selection rather than partialling-out IV of `poivregress`.
  - Application same as the Acemoglu et al. example in these slides.

7.1 Double or Debiased Machine Learning

- Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey and Robins (2018), "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*.
- Interest lies in estimation of key parameter(s) controlling for high-dimensional nuisance parameters.
- Two components to double ML or debiased ML and subsequent inference
  - Work with orthogonalized moment conditions to allow consistent estimation of parameter(s) of interest.
  - Use sample splitting (cross fitting) to remove bias induced by overfitting.

Double or Debiased Machine Learning (continued)

- Then get asymptotic normal confidence intervals for parameters of interest
  - where a variety of ML methods can be used
    - random forests, lasso, ridge, deep neural nets, boosted trees, ensembles
  - that don't necessarily need sparsity
  - and theory does not require Donsker properties.
- Can apply to
  - partial linear model (with exogenous or endogenous regressor)
    - done in these slides using LASSO
  - ATE and ATET under unconfoundedness
    - will be covered in part 5
  - LATE in an IV setting.

7.2 Caution

- The LASSO methods are easy to estimate using Stata 16
  - they'll be (blindly) used a lot.
- However in any application
  - is the underlying assumption of sparsity reasonable?
  - has the asymptotic theory kicked in?
  - are the default values of c and γ reasonable?
  - are model assumptions such as instrument validity reasonable?
- Wüthrich and Zhu (2022) find that the lasso methods can fail to pick up all relevant control variables leading to considerable omitted variables bias
  - an alternative is to include all potential regressors directly and use recently developed methods for inference with many controls.

8. References

- Chapter 28.8 “Machine Learning for prediction and inference” in A. Colin Cameron and Pravin K. Trivedi (2022), Microeconometrics using Stata, Second edition, forthcoming.
- Belloni, Chernozhukov and Hansen and coauthors have many papers
  - focus on the following papers.
- Belloni, Chernozhukov and Hansen (2014), “High-dimensional methods and inference on structural and treatment effects,” Journal of Economic Perspectives, Spring, 29-50
  - accessible paper with three applications.
- Ahrens, Hansen and Schaffer (2020), “lassopack: Model selection and prediction with regularized regression in Stata,” Stata Journal, 176-235 (also ArXiv:1901.05397).
  - more detail on LASSO methods as well as on Stata add-on commands
  - generally supplanted by Stata version 16 commands but does some things not in Stata 16.

References (continued)

- Belloni, Chernozhukov and Hansen (2011), "Inference Methods for High-Dimensional Sparse Econometric Models," Advances in Economics and Econometrics, ES World Congress 2010, ArXiv 2011
  - even more detail and summarizes several of their subsequently published papers.
- Alex Belloni, D. Chen, Victor Chernozhukov and Ying Wei (2016), "Post-Selection Inference for Generalized Linear Models With Many Controls," JBES, 34(4), 606-619.
- Alex Belloni, D. Chen, Victor Chernozhukov and Christian Hansen (2012), "Sparse Models and Methods for Optimal Instruments with an Application to Eminent Domain", Econometrica, Vol. 80, 2369-2429.
  - IV application.
- Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey and James Robins (2018), "Double/debiased machine learning for treatment and structural parameters," The Econometrics Journal, 21, C1-C68.

References (continued)

- Kaspar Wüthrich and Ying Zhu (2022), "Omitted variable bias of Lasso-based Inference Methods: A finite sample analysis," R.E.Stat., forthcoming.
- Matias Cattaneo, Michael Jannson and Whitney Newey (2018), "Inference in Linear Regression Models with Many Covariates and Heteroscedasticity," JASA, 113(523), 1350-1361.