ECON675 – Assignment 5

Anirudh Yadav

November 25, 2018

Contents

1 Many instruments asymptotics

1.1 Some moments

First,

$$\mathbb{E}[\boldsymbol{u}'\boldsymbol{u}/n] = \frac{1}{n}\mathbb{E}[\boldsymbol{u}'\boldsymbol{u}] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[u_i^2] = \sigma_u^2.$$

An analogous derivation shows that $\mathbb{E}[\boldsymbol{v}'\boldsymbol{v}/n] = \sigma_v^2$.

Next,

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}'\boldsymbol{u}/n] = \frac{1}{n}\mathbb{E}[\boldsymbol{x}'\boldsymbol{u}] &= \frac{1}{n}\mathbb{E}[(\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')\boldsymbol{u}] \\ &= \frac{1}{n}\boldsymbol{\pi}'\boldsymbol{Z}'\mathbb{E}[\boldsymbol{u}] + \frac{1}{n}\mathbb{E}[\boldsymbol{v}'\boldsymbol{u}] \\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[v_i u_i] \\ &= \sigma_{uv}^2,\end{aligned}$$

where I used the assumptions that $\boldsymbol{Z}$ and $\boldsymbol{\pi}$ are nonrandom and $\mathbb{E}[\boldsymbol{u}] = \boldsymbol{0}$.

Now,

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}'\boldsymbol{P}\boldsymbol{u}/n] &= \frac{1}{n}\mathbb{E}[(\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')\boldsymbol{P}\boldsymbol{u}] \\ &= \frac{1}{n}\mathbb{E}[\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{P}\boldsymbol{u}] + \frac{1}{n}\mathbb{E}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{u}] \\ &= \frac{1}{n}\mathbb{E}[\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{u}] + \frac{1}{n}\mathbb{E}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{u}] \\ &= \frac{1}{n}\mathbb{E}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{u}] \\ &= \frac{K}{n}\sigma_{uv}^2\end{aligned}$$

since $\mathbb{E}[v_i u_j] = 0$ for all $i \neq j$ and $\sum_{i=1}^{n} P_{ii} = K$. An analogous derivation proves the last result $\mathbb{E}[\boldsymbol{u}'\boldsymbol{P}\boldsymbol{u}/n] = K/n\sigma_u^2$.

1.2 Some probability limits

First,

$$\begin{aligned}\boldsymbol{x}'\boldsymbol{x}/n &= (\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')(\boldsymbol{Z}\boldsymbol{\pi} + \boldsymbol{v})/n \\ &= \frac{\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{\pi}}{n} + \frac{\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{v}}{n} + \frac{\boldsymbol{v}'\boldsymbol{Z}\boldsymbol{\pi}}{n} + \frac{\boldsymbol{v}'\boldsymbol{v}}{n} \\ &\to_p \mu + \mathbb{E}[\boldsymbol{\pi}'\boldsymbol{z}_i v_i] + \mathbb{E}[\boldsymbol{z}_i'\boldsymbol{\pi} v_i] + \mathbb{E}[v_i^2] \\ &= \mu + \sigma_v^2\end{aligned}$$

Next,

$$\begin{aligned}\boldsymbol{x}'\boldsymbol{P}\boldsymbol{x}/n &= (\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')\boldsymbol{P}(\boldsymbol{Z}\boldsymbol{\pi} + \boldsymbol{v})/n \\ &= \frac{\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{\pi}}{n} + \frac{\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{v}}{n} + \frac{\boldsymbol{v}'\boldsymbol{Z}\boldsymbol{\pi}}{n} + \frac{\boldsymbol{v}'\boldsymbol{P}\boldsymbol{v}}{n} \\ &\to_p \mu + \rho\sigma_v^2.\end{aligned}$$

The above convergence result involves a few steps, which I've suppressed for brevity. First, it uses the assumption that $\boldsymbol{Z}$ and $\boldsymbol{\pi}$ are nonrandom. More importantly, it uses the result that

$$\frac{\boldsymbol{v}'\boldsymbol{P}\boldsymbol{v}}{n} \to_p \mathbb{E}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{v}/n] = \rho\sigma_v^2$$

since $K/n \to \rho$. Note that this is not just a direct application of the WLLN, since we're not dealing with a sum of iid random variables. Rather, you can show that $\mathbb{V}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{v}/n]$ is bounded in probability (i.e. it goes to zero at some rate), and then use the Markov/Chebyshev inequality to get the desired convergence result. This type of result will be used a lot in the following questions too.

An analogous derivation proves the last result, $\boldsymbol{x}'\boldsymbol{P}\boldsymbol{u}/n \to_p \rho\sigma_u^2$.

1.3 plim of the classical 2SLS estimator

The classical 2SLS estimator is

$$\begin{aligned}\hat{\beta}_{\texttt{2SLS}} &= (\boldsymbol{x}'\boldsymbol{P}\boldsymbol{x})^{-1}(\boldsymbol{x}'\boldsymbol{P}\boldsymbol{y}) \\ &= (\boldsymbol{x}'\boldsymbol{P}\boldsymbol{x})^{-1}\boldsymbol{x}'\boldsymbol{P}(\boldsymbol{x}\boldsymbol{\beta} + \boldsymbol{u}) \\ &= \beta + (\boldsymbol{x}'\boldsymbol{P}\boldsymbol{x})^{-1}(\boldsymbol{x}'\boldsymbol{P}\boldsymbol{u}) \\ &= \beta + (\boldsymbol{x}'\boldsymbol{P}\boldsymbol{x}/n)^{-1}(\boldsymbol{x}'\boldsymbol{P}\boldsymbol{u}/n) \\ &\to_p \beta + \frac{\rho\sigma_u^2}{\mu + \rho\sigma_v^2},\end{aligned}$$

using the CMT and the above results. Thus, $\hat{\beta}_{\texttt{2SLS}} = \beta + \frac{\rho\sigma_u^2}{\mu+\rho\sigma_v^2} + o_p(1)$.

1.4 plim of the bias-corrected 2SLS estimator

The bias-corrected 2SLS estimator is

$$\begin{aligned}\hat{\beta}_{\texttt{2SLS}} &= (\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{x})^{-1}(\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{y}) \\ &= \beta + (\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{x}/n)^{-1}(\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u}/n)\end{aligned}$$

Now,

$$\begin{aligned}\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u}/n &= \frac{1}{n}(\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} \\ &= \frac{\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{u}}{n} - \frac{\frac{K}{n}\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{u}}{n} + \frac{\boldsymbol{v}'\boldsymbol{P}\boldsymbol{u}}{n} - \frac{\frac{K}{n}\boldsymbol{v}'\boldsymbol{u}}{n} \\ &\to_p 0 - 0 + \rho\sigma_{uv}^2 + \rho\sigma_{uv}^2 \\ &= 0.\end{aligned}$$

Thus, $\hat{\beta}_{\texttt{2SLS}} \to_p \beta$.

1.5 Asymptotic normality of the bias-corrected 2SLS estimator

1.5.1

First note that

$$\begin{aligned}
\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u} &= (\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} \\
&= \boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} + \boldsymbol{v}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} \\
&= \boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} + \left(\check{\boldsymbol{v}}' + \frac{\sigma_{uv}^2}{\sigma_u^2}\boldsymbol{u}'\right)(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} \\
&= \boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} + \check{\boldsymbol{v}}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} + \frac{\sigma_{uv}^2}{\sigma_u^2}\boldsymbol{u}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u},
\end{aligned}$$

as required.

1.5.2

Next, note that

$$\mathbb{E}[\boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u}] = \boldsymbol{\pi}'\boldsymbol{Z}'\mathbb{E}[\boldsymbol{u}] - \frac{K}{n}\boldsymbol{\pi}'\boldsymbol{Z}'\mathbb{E}[\boldsymbol{u}] = 0,$$

since $\boldsymbol{Z}$ is nonrandom. Accordingly, the CLT implies that

$$\frac{1}{\sqrt{n}}\boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u} \to_d \mathcal{N}(0, V_1(\rho)),$$

where

$$\begin{aligned}
V_1(\rho) &= \lim_{n\to\infty} \mathbb{V}[1/\sqrt{n}\boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u}] \\
&= \lim_{n\to\infty} \frac{1}{n}\mathbb{E}[\boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{u}\boldsymbol{u}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{Z}\boldsymbol{\pi}] \\
&= \lim_{n\to\infty} \frac{1}{n}\sigma_u^2\left[\boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)(\boldsymbol{P} - \frac{K}{n}\boldsymbol{I}_n)\boldsymbol{Z}\boldsymbol{\pi}\right] \\
&= \lim_{n\to\infty} \frac{1}{n}\sigma_u^2\left[\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{\pi} - 2\frac{K}{n}\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{\pi} + \frac{K^2}{n^2}\boldsymbol{Z}\boldsymbol{\pi}'\boldsymbol{Z}'\boldsymbol{Z}\boldsymbol{\pi}\right] \\
&= \sigma_u^2(1 - \rho^2).
\end{aligned}$$

1.5.3

Now,

$$\begin{aligned}
\mathbb{E}[\check{\boldsymbol{v}}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}] &= \mathbb{E}\left[\left(\boldsymbol{v}' - \frac{\sigma_{uv}^2}{\sigma_u^2}\boldsymbol{u}'\right)\boldsymbol{P}\boldsymbol{u} - \frac{K}{n}\left(\boldsymbol{v}' - \frac{\sigma_{uv}^2}{\sigma_u^2}\boldsymbol{u}'\right)\boldsymbol{u}\right] \\
&= \mathbb{E}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{u}] - \frac{\sigma_{uv}^2}{\sigma_u^2}\mathbb{E}[\boldsymbol{u}'\boldsymbol{P}\boldsymbol{u}] - \frac{K}{n}\mathbb{E}[\boldsymbol{v}'\boldsymbol{u}] + \frac{K}{n}\frac{\sigma_{uv}^2}{\sigma_u^2}\mathbb{E}[\boldsymbol{u}'\boldsymbol{u}]
\end{aligned}$$

Then, plugging in the results from part 1 gives

$$\mathbb{E}[\check{\boldsymbol{v}}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}] = K\sigma_{uv}^2 - \frac{\sigma_{uv}^2}{\sigma_u^2}K\sigma_u^2 - \frac{K}{n}\cdot n\sigma_{uv}^2 + \frac{K}{n}\frac{\sigma_{uv}^2}{\sigma_u^2}\cdot n\sigma_u^2 = 0,$$

as required.

To get the convergence result we would do the following. Compute $\mathbb{V}[\check{\boldsymbol{v}}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}]$. Using the assumption $\mathbb{V}[\boldsymbol{u}|\check{\boldsymbol{v}}] = \sigma_u^2\boldsymbol{I}_n$, it can be shown that

$$\lim_{n\to\infty}\mathbb{V}[\check{\boldsymbol{v}}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}] = O(K).$$

Then, we can somehow use the Markov inequality to get the desired convergence result.

1.5.4

Analogous derivations to the above question give the desired results.

1.5.5

Now,

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u}] &= \mathbb{E}[(\boldsymbol{\pi}'\boldsymbol{Z}' + \boldsymbol{v}')(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}] \\
&= \mathbb{E}[\boldsymbol{\pi}'\boldsymbol{Z}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}] + \mathbb{E}[\boldsymbol{v}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}] \\
&= 0 + \mathbb{E}[\boldsymbol{v}'\boldsymbol{P}\boldsymbol{u}] - K/n\mathbb{E}[\boldsymbol{v}'\boldsymbol{u}] \\
&= K\sigma_{uv}^2 - K/n\cdot n\sigma_{uv}^2 \\
&= 0.
\end{aligned}$$

And

$$\begin{aligned}
\vartheta^2 = \mathbb{V}[\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u}/\sqrt{n}] &= \frac{1}{n}\mathbb{E}[\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u}\boldsymbol{u}'\check{\boldsymbol{P}}\boldsymbol{x}] \\
&= \frac{1}{n}\mathbb{E}[\boldsymbol{x}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{u}\boldsymbol{u}'(\boldsymbol{P} - K/n\boldsymbol{I}_n)\boldsymbol{x}] \\
&= \frac{1}{n}\mathbb{E}[(\boldsymbol{x}'\boldsymbol{P}\boldsymbol{u} - K/n\boldsymbol{x}'\boldsymbol{u})(\boldsymbol{u}'\boldsymbol{P}\boldsymbol{x} - K/n\boldsymbol{u}'\boldsymbol{x})]
\end{aligned}$$

1.5.6

Note that

$$\sqrt{n}(\hat{\beta}_{\texttt{2SLS}} - \beta) = (\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{x}/n)^{-1}(\frac{1}{\sqrt{n}}\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u})$$

And we assume that

$$\frac{1}{\sqrt{n}}\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{u} \to_d \mathcal{N}(0, \vartheta^2)$$

Thus,

$$\sqrt{n}(\hat{\beta}_{\texttt{2SLS}} - \beta) \to_d \mathcal{N}(0, \mathbb{E}[\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{x}]^{-1}\vartheta^2\mathbb{E}[\boldsymbol{x}'\check{\boldsymbol{P}}\boldsymbol{x}]^{-1})$$

Intuitively, I think that when $K/n \to \rho = 0$, then the many instruments problem dissipates, so that the bias-corrected 2SLS estimator and the classical 2SLS estimator are asymptotically equivalent.

2 Weak instruments – simulations

Table 1 (overleaf) presents the simulation results, which are are a nice illustration of the weak instruments problem. The following results are worth noting:

- For all values of γ, the OLS estimators of β are very bad: recall that the true value of β is zero and for literally every one of the 20,000 OLS regressions, we reject the null that $\beta = 0$ at the 95% level! This is unsurprising, given that x_i is endogenous.
- For lower values of γ (i.e. when z_i is a "weak" instrument for x_i) the 2SLS estimators of β are also very bad. For higher values of γ, the 2SLS estimator is clearly consistent for β; and for $\gamma = \sqrt{99/n}$, the 2SLS estimator is very precise.

Table 1: Weak Instrument Summary Statistics

(a) $\gamma^2 = 0/n$ ($F \approx 1$)

	mean	**st.dev.**	**quantiles**		
			0.1	0.5	0.9
OLS					
$\hat{\beta}$	0.99	0.010	0.977	0.99	1.003
$SE(\hat{\beta})$	0.01	0.001	0.009	0.01	0.011
$\mathbf{1}_{\texttt{rej}}$	1.00	0.000	1.000	1.00	1.000
2SLS					
$\hat{\beta}$	0.985	0.491	0.798	0.991	1.175
$SE(\hat{\beta})$	1.153	40.031	0.070	0.144	0.610
$\mathbf{1}_{\texttt{rej}}$	0.873	0.333	0.000	1.000	1.000
$\hat{F}$	1.027	1.442	0.017	0.468	2.805

(b) $\gamma^2 = 0.25/n$ ($F \approx 1.25$)

	mean	**st.dev.**	**quantiles**		
			0.1	0.5	0.9
OLS					
$\hat{\beta}$	0.989	0.010	0.975	0.989	1.002
$SE(\hat{\beta})$	0.010	0.001	0.009	0.010	0.011
$\mathbf{1}_{\texttt{rej}}$	1.00	0.000	1.000	1.00	1.000
2SLS					
$\hat{\beta}$	0.875	1.207	0.411	0.826	1.392
$SE(\hat{\beta})$	3.506	105.963	0.085	0.228	1.328
$\mathbf{1}_{\texttt{rej}}$	0.697	0.460	0.000	1.000	1.000
$\hat{F}$	1.273	1.770	0.023	0.584	3.467

(c) $\gamma^2 = 9/n$ ($F \approx 10$)

	mean	**st.dev.**	**quantiles**		
			0.1	0.5	0.9
OLS					
$\hat{\beta}$	0.947	0.018	0.924	0.947	0.970
$SE(\hat{\beta})$	0.017	0.001	0.016	0.017	0.019
$\mathbf{1}_{\texttt{rej}}$	1.00	0.000	1.000	1.00	1.000
2SLS					
$\hat{\beta}$	-0.002	0.614	-0.440	0.104	0.363
$SE(\hat{\beta})$	0.502	2.888	0.147	0.283	0.743
$\mathbf{1}_{\texttt{rej}}$	0.150	0.357	0.000	0.000	1.000
$\hat{F}$	10.073	6.470	2.760	9.004	19.087

(d) $\gamma^2 = 99/n$ ($F \approx 100$)

	mean	**st.dev.**	**quantiles**		
			0.1	0.5	0.9
OLS					
$\hat{\beta}$	0.662	0.035	0.619	0.661	0.707
$SE(\hat{\beta})$	0.034	0.002	0.031	0.034	0.037
$\mathbf{1}_{\texttt{rej}}$	1.00	0.000	1.000	1.00	1.000
2SLS					
$\hat{\beta}$	-0.002	0.104	-0.140	0.009	0.121
$SE(\hat{\beta})$	0.103	0.023	0.077	0.099	0.134
$\mathbf{1}_{\texttt{rej}}$	0.057	0.232	0.000	0.000	0.000
$\hat{F}$	100.842	25.222	70.239	98.868	133.923

3 Weak instruments – empirical studies

3.1 Angrist and Krueger (1991)

Table 2: OLS and 2SLS Estimates of Return to Schooling

	(1)	(2)	(3)	(4)
	l_w_wage	l_w_wage	l_w_wage	l_w_wage
educ	0.0632	0.0632	0.0806	0.0600
	(0.000377)	(0.000377)	(0.0164)	(0.0290)
non_white	-0.257	-0.257	-0.230	-0.263
	(0.00459)	(0.00459)	(0.0261)	(0.0458)
married	0.248	0.248	0.244	0.249
	(0.00358)	(0.00358)	(0.00487)	(0.00726)
SMSA	-0.176	-0.176	-0.158	-0.180
	(0.00305)	(0.00305)	(0.0174)	(0.0305)
age_q		-0.0760		-0.0741
		(0.0601)		(0.0626)
age_sq		0.000770		0.000743
		(0.000667)		(0.000712)
N	329509	329509	329509	329509

Standard errors in parentheses

Table 2 replicates columns (5), (7), (6) and (8) from Angrist and Krueger (1991, AK) as required. The OLS and 2SLS estimates are quite similar, suggesting that there is little bias in the OLS estimates. From these models, a reasonable estimate of the return to school is around 0.06 (i.e. each additional year of schooling increases wages by around 6%, on average).

Bound, Jaeger and Baker (1995, BJB) outline a number of potential problems with these estimates. First, the association between quarter of birth and years of schooling is very weak (so we're in weak instruments territory). Furthermore, quarter of birth may affect wages through channels other than its affect on educational attainment ('other seasonal effects' on wages). BJB argue that the weak association between educational attainment and quarter of birth indicates that even if other seasonal effects are weak, they could still have large effects on the estimated return to schooling.

3.2 Bound, Jaeger and Baker (1995)

Table 3, below, replicates the first two columns of Table 3 in BJB. As BJB note, it is striking that these results look very similar to AK's results, even though the simulated instruments contain no information about educational attainment. The estimated standard deviations are also pretty close to the estimated standard errors from AK's 2SLS regressions. These results imply that AK's 2SLS results suffer badly from the weak instruments problem; in particular, they show that when the correlation between the instruments and the endogenous variable is small, then even very large sample sizes do not guarantee that quantitatively important finite sample bias will be eliminated from 2SLS estimates.

Table 3: 2SLS Estimates of Return to Schooling Using Permuted Quarter of Birth*

	(1)	(2)
	l_w_wage	l_w_wage
Mean	.0646165	.0646411
Std. dev.	.0387673	.0387972

*500 replications

4 Appendix

4.1 R code

4.1.1 Question 2

```
## ECON675: ASSIGNMENT 5
## Q2: WEAK INSTRUMENTS SIMULATIONS
## Anirudh Yadav
## 11/19/2018

#########################################################################
# Load packages, clear workspace
#########################################################################
rm(list = ls())             #clear workspace
library(foreach)            #for looping
library(data.table)         #for data manipulation
library(Matrix)             #fast matrix calcs
library(ggplot2)            #for pretty plots
library(sandwich)           #for variance-covariance estimation
library(xtable)             #for latex tables
library(boot)               #for bootstrapping
library(mvtnorm)            #for MVN stuff
library(AER)                #for IV regressions
options(scipen = 999)       #forces R to use normal numbers instead of scientific notation

#########################################################################
# Generate random data for each simulation
#########################################################################
N     = 200
M     = 5000
SIGMA = matrix(c(1,0,0,0,1,0.99,0,0.99,1),3,3)

set.seed(1234)

# Generate Z, U, V
W     = replicate(M,rmvnorm(N, mean = c(0,0,0), sigma = SIGMA, method="chol"))

# Get Y (assuming that beta=0, Y=U)
Y     = W[,2,]

# Generate X matrix for each value of gamma
gamma.vec = sqrt((1/N)*c(0,0.25,9,99))
X         = lapply(1:length(gamma.vec),function(i) gamma.vec[i]*W[,1,]+W[,3,])

#########################################################################
# Compute OLS statistics for each gamma, and simulation
#########################################################################

# Run OLS for each gamma and each simulation -- this spits out 5000 lm's for each gamma
ols.big = foreach(j=1:length(gamma.vec)) %do%
  lapply(1:M, function(i) lm(Y[,i]~X[[j]][,i]-1))

# Extract point estimates, standard errors, t-stats
ols.beta = foreach(j=1:length(gamma.vec)) %do%
  sapply(1:M, function(i) ols.big[[j]][[i]]$coefficients)

ols.se   = foreach(j=1:length(gamma.vec)) %do%
  sapply(1:M, function(i) coef(summary(ols.big[[j]][[i]]))[,"Std. Error"])

ols.t    = sapply(1:length(gamma.vec),function(j) ols.beta[[j]]/ols.se[[j]])

ols.rej  = ifelse(ols.t>1.96,1,0)

# Compute desired summary statistics across the simulations (spits out a list containing results for each gamma)
ols.results = foreach(j=1:length(gamma.vec)) %do%
  rbind(c(mean(ols.beta[[j]]),sd(ols.beta[[j]]),quantile(ols.beta[[j]], probs = c(0.1, 0.5 ,0.9))),
```

```
  c(mean(ols.se[[j]]),sd(ols.se[[j]]),quantile(ols.se[[j]], probs = c(0.1, 0.5 ,0.9))),
  c(mean(ols.rej[,j]),sd(ols.rej[,j]),quantile(ols.rej[,j], probs = c(0.1, 0.5 ,0.9))))

# Remove big objects!
rm(ols.big,ols.beta,ols.se,ols.t,ols.rej)

########################################################################
# Compute 2SLS statistics for each gamma, and simulation
########################################################################

# Run 2SLS for each gamma and each simulation -- this spits out 5000 ivreg's for each gamma
# WATCH OUT: this takes a minute or so!
iv.big = foreach(j=1:length(gamma.vec)) %do%
  lapply(1:M, function(i) ivreg(Y[,i]~X[[j]][,i]-1|W[,1,i]))

# Extract point estimates, standard errors, t-stats
iv.beta = foreach(j=1:length(gamma.vec)) %do%
  sapply(1:M, function(i) iv.big[[j]][[i]]$coefficients)

iv.se   = foreach(j=1:length(gamma.vec)) %do%
  sapply(1:M, function(i) summary(iv.big[[j]][[i]])[["coefficients"]][,"Std. Error"])

iv.t    = sapply(1:length(gamma.vec),function(j) iv.beta[[j]]/iv.se[[j]])

iv.rej  = ifelse(iv.t>1.96,1,0)

# Run first-stage regression and extract F-statistics
iv.f       = foreach(j=1:length(gamma.vec)) %do%
  sapply(1:M, function(i) summary(lm(X[[j]][,i]~W[,1,i]-1))$fstatistic[1])

# Combine results for each gamma
iv.results = foreach(j=1:length(gamma.vec)) %do%
  rbind(c(mean(iv.beta[[j]]),sd(iv.beta[[j]]),quantile(iv.beta[[j]], probs = c(0.1, 0.5 ,0.9))),
        c(mean(iv.se[[j]]),sd(iv.se[[j]]),quantile(iv.se[[j]], probs = c(0.1, 0.5 ,0.9))),
        c(mean(iv.rej[,j]),sd(iv.rej[,j]),quantile(iv.rej[,j], probs = c(0.1, 0.5 ,0.9))),
        c(mean(iv.f[[j]]),sd(iv.f[[j]]),quantile(iv.f[[j]], probs = c(0.1, 0.5 ,0.9))))

# Remove big objects!
rm(iv.big,iv.beta,iv.se,iv.t,iv.rej,iv.f)
```

4.1.2 Question 3

```
## ECON675: ASSIGNMENT 5
## Q3: WEAK INSTRUMENTS -- EMPIRICAL STUDIES
## Anirudh Yadav
## 11/19/2018

########################################################################
# Load packages, clear workspace
########################################################################
rm(list = ls())             #clear workspace
library(foreach)            #for looping
library(data.table)         #for data manipulation
library(Matrix)             #fast matrix calcs
library(ggplot2)            #for pretty plots
library(sandwich)           #for variance-covariance estimation
library(xtable)             #for latex tables
library(boot)               #for bootstrapping
library(mvtnorm)            #for MVN stuff
library(AER)                #for IV regressions
options(scipen = 999)       #forces R to use normal numbers instead of scientific notation

########################################################################
# Input data
########################################################################
ak <- fread('PhD_Coursework/ECON675/HW5/Angrist_Krueger.csv')
```

```
#########################################################################
# [3.1] Angrist-Krueger
#########################################################################

# make YOB dummies
ak[, .N, "YoB_ld"]
for(year_i in unique(ak$YoB_ld)){

  ak[ ,temp := 0]
  ak[YoB_ld == year_i ,temp := 1]
  setnames(ak, "temp", paste0("d_YOB_ld_", year_i))

}
# get a list of all year dummies but one. Exclude the proper one to match coeffs
year_dummies <- setdiff(grep("d_YOB", colnames(ak), value = TRUE), "d_YOB_ld_0")

# make QoB dummies
for(qob_i in unique(ak$QoB)){

  ak[ ,temp := 0]
  ak[QoB == qob_i ,temp := 1]
  setnames(ak, "temp", paste0("d_QoB_", qob_i))

}

# get qob dummy list.  Exclude the proper one to match coeffs
qob_dummies <- setdiff(grep("d_QoB", colnames(ak), value = TRUE), "d_QoB_1")

# make cross variables of year dummies and qob
#note there is almost certainly a better way to do this but here we are
inter_list <- NULL
for(d_year in year_dummies){

  for(d_qob in qob_dummies){

    ak[, temp := get(d_qob)*get(d_year)]
    setnames(ak, "temp", paste0(d_year, "X", d_qob))
    inter_list<- c(inter_list, paste0(d_year, "X", d_qob))
  }
}


# standard controls
# (i) race, (ii) marrital status, (iii) SMSA, (iv) dummies for
# region, and (iv) dummies for YoB ld.
std_cont <- c("non_white","married", "SMSA",
              "ENOCENT","ESOCENT", "MIDATL",
              "MT", "NEWENG", "SOATL", "WNOCENT",
              "WSOCENT",  year_dummies) # get year dummies but leave one out

# save extra controls
extra_cont <- c("age_q", "age_sq")

#================#
# ==== ols 1 ====
#================#

# make the formula
ols1_form <- as.formula(paste0("l_w_wage~educ +", paste(std_cont, collapse = " + ")))

# run ols
out_ols1 <- data.table(tidy(lm(ols1_form, data = ak)))

# keep what I need
out_ols1 <- out_ols1[term %chin% c("educ"), c("term", "estimate", "std.error")]
out_ols1[, model := "OLS 1"]

#================#
```

```
# ==== OlS 2 ====
#================#

# make the formula
ols2_form <- as.formula(paste0("l_w_wage~educ +", paste(std_cont, collapse = " + "), " + ", paste0(extra_cont, collapse = " + ")))

#run ols
out_ols2 <- data.table(tidy(lm(ols2_form, data = ak)))

# keep what I need
out_ols2 <- out_ols2[term %chin% c("educ"), c("term", "estimate", "std.error")]
out_ols2[, model := "OLS 2"]

#===============#
# ==== 2sls ====
#===============#

wrap_2sls <- function(in_data){

  #=================#
  # ==== 2sls 1 ====
  #=================#

  iv_form <- as.formula(paste0("l_w_wage~educ +", paste(std_cont, collapse = " + "),
                               "| ",
                               paste(std_cont, collapse = " + "), " + ", paste0(inter_list, collapse = " + ")))
  iv_reg1 <- data.table(tidy(ivreg(iv_form , data = in_data)))

  # keep what I need
  iv_reg1 <- iv_reg1[term %chin% c("educ"), c("term", "estimate", "std.error")]
  iv_reg1[, model := "2sls 1"]

  #=================#
  # ==== 2sls 2 ====
  #=================#


  iv_form2 <- as.formula(paste0("l_w_wage~educ +", paste(std_cont, collapse = " + "), "+", paste0(extra_cont, collapse = " + "),
                                "| ",
                                paste(std_cont, collapse = " + "),
                                " + ", paste0(inter_list, collapse = " + "),
                                "+", paste0(extra_cont, collapse = " + ")))
  iv_reg2 <- data.table(tidy(ivreg(iv_form2 , data = in_data)))

  # keep what I need
  iv_reg2 <- iv_reg2[term %chin% c("educ"), c("term", "estimate", "std.error")]
  iv_reg2[, model := "2sls 2"]

  # stack 2sls
  out_2sls <- rbind(iv_reg1, iv_reg2)

  return(out_2sls)

}#end 2sls function

# run function
ak_2sls <- wrap_2sls(ak)

#========================#
# ==== output tables ====
#========================#

output_3.1 <- rbind(out_ols1, out_ols2, ak_2sls)
setcolorder(output_3.1, c("model", "term", "estimate", "std.error"))



#================#
```

```
# ==== Q 3.2 ====
#================#


#==================================#
# ==== Fast 2sls function ==========
#==================================#

fast_2sls <- function(in_data){

  #===============#
  # ==== reg1 ====
  #===============#

  # make x z and y matrices
  y <- as.matrix(in_data[, l_w_wage])
  x <- as.matrix(in_data[, educ])
  cont <- as.matrix(in_data[, c( std_cont, 'const'), with = FALSE])
  z <- as.matrix(in_data[, c(inter_list, std_cont, "const"), with = FALSE])

  first_stage_fit <-  z%*%Matrix::solve(Matrix::crossprod(z))%*%(Matrix::crossprod(z, x))

  # make x' matrix
  x_prime <- cbind(first_stage_fit, cont)


  form_2nd <-  Matrix::solve(Matrix::crossprod(x_prime))%*%(Matrix::crossprod(x_prime, y))

  reg1 <- data.table( term = "educ", estimate = form_2nd[1,1], model = "2sls 1")

  #===============#
  # ==== reg2 ====
  #===============#

  cont <- as.matrix(in_data[, c( std_cont, extra_cont, 'const'), with = FALSE])
  z <- as.matrix(in_data[, c(inter_list, std_cont, extra_cont, "const"), with = FALSE])

  first_stage_fit <-  z%*%Matrix::solve(Matrix::crossprod(z))%*%(Matrix::crossprod(z, x))

  # make x' matrix
  x_prime <- cbind(first_stage_fit, cont)


  form_2nd <-  Matrix::solve(Matrix::crossprod(x_prime))%*%(Matrix::crossprod(x_prime, y))

  reg2 <- data.table( term = "educ", estimate = form_2nd[1,1], model = "2sls 2")

  # stack results and retur n
  out_results <- rbind(reg1, reg2)
}


#=========================#
# ==== run simulation ====
#=========================#

# copy data for permutation
ak_perm <- copy(ak)

# add constant
ak_perm[, const := 1]

# write a function
sim_warper <- function(sim_i, in_data = ak_perm ){

  # get random sampel
  perm <- sample(c(1:nrow(in_data)))
```

```
  # purmute data
  in_data[, QoB := QoB[perm]]

  # clear out dummy variables
  in_data <- in_data[, -c(grep("d_QoB", colnames(in_data), value = TRUE), inter_list), with = FALSE]

  # redo dummy vars
  for(qob_i in unique(in_data$QoB)){

    in_data[ ,temp := 0]
    in_data[QoB == qob_i ,temp := 1]
    setnames(in_data, "temp", paste0("d_QoB_", qob_i))

  }
  # recalculate interactions
  inter_list <- NULL
  for(d_year in year_dummies){

    for(d_qob in qob_dummies){

      in_data[, temp := get(d_qob)*get(d_year)]
      setnames(in_data, "temp", paste0(d_year, "X", d_qob))
      inter_list<- c(inter_list, paste0(d_year, "X", d_qob))
    }
  }

  # run 2sls funciton on new data
  ak_2sls_i <- fast_2sls(in_data)

  # add simulation
  ak_2sls_i[, sim := sim_i]

  # retunrn it
  return(ak_2sls_i)

} # end funciton

# run simulations in parallel
output_list <- foreach(sim = 1 : 5000,
                       .inorder = FALSE,
                       .packages = "data.table",
                       .options.multicore = list(preschedule = FALSE, cleanup = 9)) %dopar% sim_warper(sim_i = sim)

# stop clusters
stopCluster(cl)



#=========================#
# ==== organize output ====
#=========================#

# stack data
sim_res3.2 <- rbindlist(output_list)

# make table
output3.2 <- sim_res3.2[, list(mean = mean(estimate), std.dev = sd(estimate)), "model"]
```

4.2 STATA code

4.2.1 Question 2

```
clear all
set more off
cap log close

program define weak_IV, rclass
```

```
    syntax [, obs(integer 200) f_stat(real 10) ]
drop _all

set obs `obs'

* DGP
gen u = rnormal()
gen v = 0.99 * u + sqrt(1-0.99^2) * rnormal()
gen z = rnormal()

local gamma_0 = sqrt((`f_stat' - 1) / `obs')
gen x = `gamma_0' * z + v
gen y = u

* OLS
qui reg y x, robust
return scalar OLS_b   = _b[x]
return scalar OLS_se  = _se[x]
return scalar OLS_rej = abs(_b[x]/_se[x]) > 1.96

* 2SLS
qui ivregress 2sls y (x = z)
return scalar TSLS_b   = _b[x]
return scalar TSLS_se  = _se[x]
return scalar TSLS_rej = abs(_b[x]/_se[x]) > 1.96
qui reg x z
return scalar TSLS_F   = e(F)
end

* simulation 1: F = 1
simulate OLS_b=r(OLS_b) OLS_se=r(OLS_se) OLS_rej=r(OLS_rej) ///
    TSLS_b=r(TSLS_b) TSLS_se=r(TSLS_se) TSLS_rej=r(TSLS_rej) TSLS_F=r(TSLS_F), ///
    reps(5000) seed(123) nodots: ///
    weak_IV, f_stat(1)

local k = 1
matrix Results = J(7, 5, .)

qui sum OLS_b, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
local k = `k' + 1

qui sum OLS_se, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
local k = `k' + 1

qui sum OLS_rej, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
local k = `k' + 1

qui sum TSLS_b, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
```

```
local k = `k' + 1

qui sum TSLS_se, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
local k = `k' + 1

qui sum TSLS_rej, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
local k = `k' + 1

qui sum TSLS_F, detail
matrix Results[`k',1] = r(mean)
matrix Results[`k',2] = r(sd)
matrix Results[`k',3] = r(p10)
matrix Results[`k',4] = r(p50)
matrix Results[`k',5] = r(p90)
local k = `k' + 1

mat2txt, matrix(Results) saving(result1.txt) format(%9.4f) replace
```

4.2.2 Question 3

```
*******************************************************************************
* ECON675: ASSIGNMENT 5
* Q3: WEAK INSTRUMENTS -- EMPIRICAL STUDIES
* Anirudh Yadav
* 11/19/2018
*******************************************************************************


*******************************************************************************
* Preliminaries
*******************************************************************************
clear all
set more off

* Set working directory
global dir "/Users/Anirudh/Desktop/GitHub"


*******************************************************************************
* Import AK data
*******************************************************************************

use "$dir/PhD_Coursework/ECON675/HW5/Angrist_Krueger.dta"

*******************************************************************************
* [3.1] Run AK regressions
*******************************************************************************
eststo ols1: reg l_w_wage educ non_white married SMSA i.region i.YoB_ld, r
eststo ols2: reg l_w_wage educ non_white married SMSA i.region i.YoB_ld age_q age_sq, r
eststo  iv1: ivregress 2sls l_w_wage non_white married SMSA i.region i.YoB_ld (educ = i.QoB##i.YoB_ld),r
eststo  iv2: ivregress 2sls l_w_wage non_white married SMSA i.region i.YoB_ld age_q age_sq (educ = i.QoB##i.YoB_ld), r

esttab ols1 ols2 iv1 iv2 using "$dir/PhD_Coursework/ECON675/HW5/q3_ak_results.tex", keep(educ non_white SMSA married age_q age_sq) s

*******************************************************************************
* [3.2] Run BJB permutation regressions
```

```
*************************************************************************
capture program drop IV_quick
program define IV_quick, rclass
    syntax varlist(max=1) [, model(integer 1) ]
local x "‘varlist’"

if (‘model’ == 1) {
    capture drop educ_hat
qui reg educ non_white married SMSA i.region i.YoB_ld i.YoB_ld##i.‘x’
predict educ_hat
qui reg l_w_wage educ_hat non_white married SMSA i.region i.YoB_ld
return scalar beta = _b[educ_hat]
}
if (‘model’ == 2) {
    capture drop educ_hat
qui reg educ non_white married SMSA age_q age_sq i.region i.YoB_ld i.YoB_ld##i.‘x’
predict educ_hat
qui reg l_w_wage educ_hat non_white married SMSA age_q age_sq i.region i.YoB_ld
return scalar beta = _b[educ_hat]
}
end


permute QoB TSLS_1_b = r(beta), reps(500) seed(123) saving("$dir/PhD_Coursework/ECON675/HW5/premute1.dta", replace): ///
    IV_quick QoB, model(1)

permute QoB TSLS_2_b = _b[educ], reps(500) seed(123) saving("$dir/PhD_Coursework/ECON675/HW5/premute2.dta", replace): ///
    IV_quick QoV, model(2)

clear all
use "$dir/PhD_Coursework/ECON675/HW5/premute1.dta"
sum TSLS_1_b

clear all
use "$dir/PhD_Coursework/ECON675/HW5/premute2.dta"
sum TSLS_2_b
```