Problem Set 2 Solutions

Operation IV

EC 425/525: Econometrics

Due *before* midnight (11:59pm) on Wednesday, 29 May 2019

DUE Your solutions to this problem set are due *before* 11:59pm on Wednesday, 29 May 2019 on Canvas. Your problem set **must be typed** with R code beneath your responses. *E.g.*, `knitr` and `R Markdown`.

OBJECTIVE We're going to walk through three classic applications of instrumental variables/two-stage least squares: endogeneity, measurement error, and randomized encouragement designs (REDs).

Part 1: Selection bias

As this problem follows one from Wooldridge, we'll use the `wooldridge` package. You need to install the `wooldridge` package and then load the birthweight data using `data("bwght")`. For (limited) information on the variables, see the help file (*i.e.*, `?wooldridge::bwght`).

1.01 We want to better understand the effect of a number of variables on birth weight (`bwght`)—namely gender (`male`), birth order (`parity`), income (`faminc`), and cigarette smoking during pregnancy (`packs`), *i.e.*,

$$\log(\text{bwght}_i) = \beta_0 + \beta_1 \text{male}_i + \beta_2 \text{parity}_i + \beta_3 \log(\text{faminc}_i) + \beta_4 \text{packs}_i + u_i$$

1.01 Why might you expect amount of smoking (`packs`) to be correlated with u_i?

Answer We're worried about selection into smoking (in other words, omitted-variable bias). Namely, we may expect that children born to women who smoke during pregnancy may have had different birthweights than children born to women who do not smoke during pregnancy, regardless of the number of cigarettes their mothers smoked during pregnancy.

1.02 Suppose that you have data on average cigarette prices in each woman's state of residence. Discuss whether this information is likely to satisfy the properties of a good instrumental variable for `packs`.

Answer Maybe? We have three requirements for our instrument.

1. **First stage** It seems plausible that cigarette prices will affect quantity smoked (something about the law of demand).
2. **Exclusion restriction** We need our instrument to be uncorrelated with other determinants of birthweight (determinants not included in the regression model above). This may not be true—just as quantities respond to prices, prices can respond to quantities (simultaneity). Further, economic shocks may affect cigarette prices *and* birthweight (through channels excluded from our model above). Thus, cigarette prices may not have a super believable exclusion restriction.
3. **Monotonicity** We need need the instrument (cigarette prices) to have a monotone effect on our endogenous regression (smoking behavior). This requirement seems reasonable if we think that price increases will only reduce smoking or will not affect smoking. If we think that some people smoke more when prices are high (*e.g.*, some signal of defiance of a cigarette tax), then we do not have monotonicity.

1.03 Use the data in in `bwght` to estimate equation the equation above. First, use OLS. Then, use 2SLS, where `cigprice` is an instrument for `packs`. Discuss any important differences in the OLS and 2SLS estimates.

Answer

```
# Setup
library(pacman)
p_load(
  wooldridge,
  tidyverse, huxtable,
  ggplot2, ggthemes,
  future, furrr,
  estimatr, magrittr
)
# OLS
est_ols ← lm_robust(bwght ~ male + parity + faminc + packs, data = bwght)
# 2SLS
est_2sls ← iv_robust(
  bwght ~ male + parity + faminc + packs |
  male + parity + faminc + cigprice,
  data = bwght
)
# Table of results
huxreg("OLS" = est_ols, "2SLS" = est_2sls, statistics = "N")
```

	OLS	2SLS
(Intercept)	112.390 ***	96.816 ***
	(1.530)	(18.703)
male	3.163 **	3.636
	(1.068)	(1.933)
parity	1.646 **	-0.211
	(0.597)	(2.845)
faminc	0.102 ***	0.373
	(0.028)	(0.333)
packs	-9.500 ***	91.034
	(1.804)	(123.417)

*** p < 0.001; ** p < 0.01; * p < 0.05.

One major difference: All statistically significant coefficients are no longer significant.

Related: we previously estimated a negative and significant effect of smoking on birthweight. Now we estimate a positive and not significant effect.

1.04 Estimate the reduced form for `packs`. Does it raise any issues? What bearing does this conclusion have on your answer from **1.03**?

Answer

```
# The reduced form
est_rf ← lm_robust(bwght ~ male + parity + faminc + cigprice, data = bwght)
# The reduced form
est_fs ← lm_robust(packs ~ male + parity + faminc + cigprice, data = bwght)
# Table
huxreg(
    "OLS" = est_ols, "2SLS" = est_2sls, "Red. form" = est_rf, "1st stage" = est_fs,
    statistics = "N"
)[c(1,10:14),] %>%
insert_row(c("Dep. Var.:", rep("Birth Weight", 3), "# Cig.")) %>%
merge_cells(c(1,1), c(2,4)) %>%
insert_row(c("", 1:4 %>% paste0("(", ., ")"))) %>%
add_footnote("Note: I'm not showing all coefficients to preserve space.") %>%
set_all_borders(0) %>%
set_top_border(c(1,8), everywhere, 1) %>%
set_top_border(4, 2:5, 0.5) %>%
set_top_border(3, 2:4, 0.5 )
```

	(1)	(2)	(3)	(4)
Dep. Var.:	Birth Weight			# Cig.
	OLS	2SLS	Red. form	1st stage
packs	-9.500 ***	91.034		
	(1.804)	(123.417)		
cigprice			0.072	0.001
			(0.052)	(0.001)

*** p < 0.001; ** p < 0.01; * p < 0.05.

Note: I'm not showing all coefficients to preserve space.

We see that our reduced form shows a positive and not statistically significant effect of cigarette prices on birth weight. This is the sign we might expect in the absence of exclusion-restriction violations, as higher prices should reduce smoking and increase birth weight.

On the other hand, our 2SLS estimate is positive, which we might not have expected (again, not significant, so don't put too much weight on this result). We know that the 2SLS coefficient is the ration of the reduced-form coefficient and the first-stage coefficent. Thus, we know that the first-stage coefficient is also positive (though tiny and not significant)—implying higher prices lead to higher consumption. Again, this relationship is not statistically significant, so I wouldn't interpret this results as saying cigarettes are a Giffen good. Finally, because the first-stage coefficient is so small, it is magnifying our (not-significant) effect from the reduced form.

Note: You did not have to estimate the first stage.

Part 2: Randomized encouragement designs

Another common implementation of IV/2SLS is a randomized encouragement design (RED), in which we randomly select individuals to receive "encouragement" (*e.g.*, we call them to tell them about an exciting new program) in order to try to induce an exogenous change in program participation.

Let's imagine we want estimate the effect of solar-panel installation on household electricity consumption.

2.01 What would be the problem with comparing average electricity consumption for houses with solar panels to average electricity consumption without solar panels?

Answer Selection bias. There are a lot of reasons why households with solar panels might differ in observable and unobservable ways from households without solar panels.

2.02 We randomly select 200 homes that have not yet installed solar panels. Within this sample, we randomly assign 100 houses to our "encouragement" group and 100 houses to our "non-encouragement" group. For the 100 houses in the encouragement group, we call/visit the households and tell them how awesome solar panels are—and how much money they could save with solar.[†]

What do we need for our encouragement to be a valid instrument for solar panel installation? Do you think it is satisfied?

Answer We need

1. **First stage** If our encouragement increases purchases of solar panels, then we will have a first stage. If the encouragement doesn't affect solar panel purchasing, then we will not have a valid instrument.
2. **Exclusion restriction** We need our instrument—randomly showing up at someone's home to talk about solare panels—to only affect energy consumption through solar-panel purchases. This requirement should be fine as long as our encouragement doesn't directly affect households' energy consumption. We should make sure the encouragement doesn't actually talk about conserving energy. One related concern is that our instrument may increase the salience of energy consumption/costs for treated households.
3. **Monotonicity** We need assignment to encouragement to move folks from no-panel to panel or no-panel to no-panel. This is fine, as none of our encouragement group has panels. We also need assignment to non-encouragement to not cause someone to buy a solar panel. This requirement seems plausible, as non-encouragement folks likely do not know they are part of the non-encouragement group.

[†] Tangent: In case you haven't seen it, you should check out Google's Project Sunroof.

2.03 A year later, we conduct a survey and find that in the encouragement group, 15/100 homes now have solar panels. In the non-encouragement group, 5/100 homes now have solar panels. If we estimated the first stage, (regressing an indicator for solar panel on an intercept and an indicator for encouragement group), what would our estimates be?

Answer The first stage

$$\mathbb{I}(\text{Solar Panel})_i = \gamma_0 + \gamma_1 \mathbb{I}(\text{Encouraged})_i + u_i$$

compares the share of solar panels in the encouragement group and control group, so $\hat{\gamma}_1 = 0.10$.

2.04 Imagine that average monthly electricity consumption in the encouragement group is 900 kWh (kilowatthours), while the average in the non-encouragement is 870 kWh. Based upon these numbers, what are the reduced-form (the effect of encouragement on energy consumption) and 2SLS estimates?

Answer The reduced form

$$\text{Consumption}_i = \pi_0 + \pi_1 \mathbb{I}(\text{Encouraged})_i + v_i$$

compares the mean consumption in the encouragement group and control group, so $\hat{\pi}_1 = 30$.

We know the 2SLS estimate is the ratio of the reduced-form estimate and the first-stage estimate, so $\hat{\beta}_1 = \hat{\pi}/\hat{\gamma} = 30/0.10 = 300$.

2.05 What does the LATE in this setting mean—*i.e.*, what does *local* mean in this setting?

Answer Recall that the LATE is for determined by *compliers*. In this setting, compliers are individuals who install solar panels when they are part of our encouragement group—folks who respond to our encouragement (learning about solar panel/energy savings causes them to install a solar panel).

Part 3: Measurement error

Now for a good, old-fashioned simulation.

3.01 Set up a data-generating process such that

$$y_i = 3 + 7x_i + u_i$$

where $x_i \overset{\text{iid}}{\sim} N(5, 5)$ and $u_i \overset{\text{iid}}{\sim} N(0, 3)$.

In this simulation, we want to imagine what would happen if we could not observe x_i (or if x_i is measured with error/noise).

Thus, we want to create two additional variables: w_{1i} and w_{2i}, such that

$$\begin{aligned} w_{1i} &= x_i + \varepsilon_i \\ w_{2i} &= x_i + \nu_i \end{aligned}$$

where ε_i is i.i.d. standard Normal and ν_i is i.i.d. $N(0, 7)$. For this problem, the sample size will be 50.

This setting is *classical* measurement error—the error (or noise) in measurement (*i.e.*, ε_i and ν_i) is uncorrelated with the true variable (x_i).

Note: No results for this part of the problem. Just make sure you've set up the DGP.

Answer

```
# Function: DGP
fun_dgp ← function(i, n = 50) {
  # Generate x and u
  x ← rnorm(n, mean = 5, sd = sqrt(5))
  y ← 3 + 7 * x + rnorm(n, sd = sqrt(3))
  # Generate w1 and w2
  w1 ← x + rnorm(n)
  w2 ← x + rnorm(n, sd = sqrt(7))
  # Return tibble of variables
  return(tibble(y, x, w1, w2))
}
# Generate a dataset
set.seed(12345)
gen_df ← fun_dgp(1)
```

3.02 Imagine you cannot observe x_i and are stuck with our noisily measured versions w_{1i} and/or w_{2i}. Regress y_i on w_{1i}. What do you get? What if you regress y_i on both w_{1i} *and* w_{2i}?

Answer When we use w_1 instead of x, we have an attenuated effect (less than 7). Adding w_2 to the regression does not appear to improve anything. Notice that the 95% confidence intervals for columns 2–4 would reject the true effect of x on y (*i.e.*, $\beta_1 = 7$).

Note I included additional regression (columns 1 and 3) that you did not need to include.

```
# Regress y on x
est_x ← lm_robust(y ~ x, data = gen_df)
# Regress y on w1
est_w1 ← lm_robust(y ~ w1, data = gen_df)
# Regress y on w2
est_w2 ← lm_robust(y ~ w2, data = gen_df)
# Regress y on w1 and w2
est_w1w2 ← lm_robust(y ~ w1 + w2, data = gen_df)
# Table
huxreg(est_x, est_w1, est_w2, est_w1w2)
```

	(1)	(2)	(3)	(4)
(Intercept)	3.546 ***	11.199 ***	20.671 ***	10.225 ***
	(0.724)	(1.826)	(3.088)	(1.829)
x	6.999 ***			
	(0.120)			
w1		5.595 ***		4.437 ***
		(0.309)		(0.302)
w2			3.644 ***	1.271 ***
			(0.461)	(0.301)
N	50	50	50	50

*** p < 0.001; ** p < 0.01; * p < 0.05.

3.03 Now what happens if you *instrument* w_{1i} with w_{2i}?

Answer When we instrument w_1 with w_2 (meaning we use w_2 as our instrument), our point estimate is much closer to the true value. In fact, our 95% confidence interval now contains the true value. Also notice that our standard errors have substantially increased with IV.

```
# Instrument w1 with w2
iv_w1_w2 <- iv_robust(y ~ w1 | w2, data = gen_df)
# Table
huxreg("OLS" = est_x, "OLS" = est_w1, "IV"= iv_w1_w2)
```

	OLS	OLS	IV
(Intercept)	3.546 ***	11.199 ***	4.627
	(0.724)	(1.826)	(2.950)
x	6.999 ***		
	(0.120)		
w1		5.595 ***	6.815 ***
		(0.309)	(0.585)
N	50	50	50

*** p < 0.001; ** p < 0.01; * p < 0.05.

3.04 Confirm your results form **3.02** and **3.03** were not anomalies. In other words, run a simulation (with at least 1,000 iterations). In each iteration, record the results of

- regressing y_i on w_{1i}
- regressing y_i on w_{2i}
- instrumenting w_{1i} with w_{2i}
- instrumenting w_{2i} with w_{1i}

Report the results of your simulation. Do you see anything interesting? Does IV outperform OLS in the presence of measurement error (in terms of bias in $\hat{\beta}_1$)? What happens in your inference (look at the share of estimates in which you reject the null)?

Answer First we write a function to perform our desired analyses

```
# Function: Generate data and analyze
fun_analysis ← function(i, i_df) {
  # Regress y on w1; grab the coefficient
  ols1 ← lm_robust(y ~ w1, data = i_df) %>% tidy() %>% filter(term == "w1")
  # Regress y on w2; grab the coefficient
  ols2 ← lm_robust(y ~ w2, data = i_df) %>% tidy() %>% filter(term == "w2")
  # IV w1 with w2; grab the coefficient
  iv1 ← iv_robust(y ~ w1 | w2, data = i_df) %>% tidy() %>% filter(term == "w1")
  # IV w2 with w1; grab the coefficient
  iv2 ← iv_robust(y ~ w2 | w1, data = i_df) %>% tidy() %>% filter(term == "w2")
  # Results with extra columns for model and iteration
  res_df ← bind_rows(ols1, ols2, iv1, iv2) %>% mutate(
    # Variable for the model
    model = c("OLS w1", "OLS w2", "IV w1|w2", "IV w2|w1"),
    # Iteration
    iter = i
  )
  # Return results
  return(res_df)
}
```

Now a function that puts the two individual functions together.

```
fun_iter ← function(i, n = 50) {
  # Generate and analyze the data
  fun_analysis(i = i, i_df = fun_dgp(i, n))
}
```

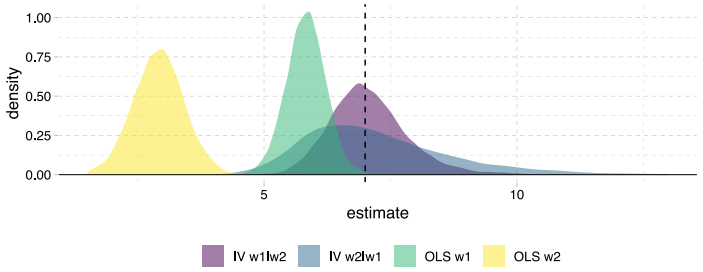
Now run the function 10,000 times (you needed at least 1,000).

```
# Set the seed
set.seed(12345)
# Tell R (furrr) to parallelize
plan(multiprocess, workers = 8)
# Usin map_dfr from furrr: Parallelize and bind resulting data frames
sim_df ← future_map_dfr(
  # The argument to our function: The iteration number (10,000)
  1:1e4,
  # The function for each iteration
  fun_iter,
  # Tell map_dfr to use the set seed
  .options = future_options(seed = T)
)
```

Figures on the next page(s).

Answer, continued

```
ggplot(data = sim_df, aes(x = estimate, fill = model)) +
geom_density(color = NA, alpha = 0.5) +
geom_vline(xintercept = 7, size = 0.5, linetype = "dashed") +
geom_hline(yintercept = 0, size = 0.1) +
xlim(1.5, 13) +
scale_fill_viridis_d("") +
theme_pander() + theme(legend.position = "bottom")
```

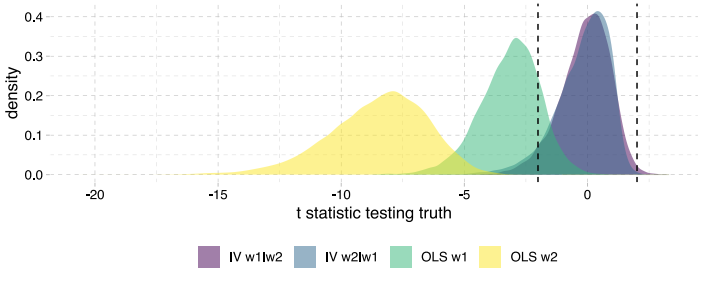


Two items to notice:

1. More measurement error (w_2) leads to more attenuation bias.
2. Instrumenting w_1 (less noisy) with w_2 (more noisy) is more efficient than the reverse.

You have options to look at inference. Let's look at the distribution of *t* statistics testing that our estimate differs from the true value of 7:

```
ggplot(data = sim_df, aes(x = (estimate - 7)/std.error, fill = model)) +
geom_density(color = NA, alpha = 0.5) +
geom_vline(xintercept = qt(p = 0.975, df = 48), size = 0.5, linetype = "dashed") +
geom_vline(xintercept = qt(p = 0.025, df = 48), size = 0.5, linetype = "dashed") +
xlab("t statistic testing truth") +
scale_fill_viridis_d("") +
theme_pander() + theme(legend.position = "bottom")
```



Answer, continued Alterantively, we could just create a table for the share of iterations that do reject $\hat{\beta}_1 = 7$ (truth) for each model...

```
sim_df %>%
  group_by(model) %>%
  summarize(mean(!(conf.low < 7 & conf.high > 7))) %>%
  hux()
```

IV w1|w2	0.0501
IV w2|w1	0.0647
OLS w1	0.833
OLS w2	1

3.05 Now let x_i **positively** correlate with ε_i and **negatively** correlate with ν_i, *i.e.*, $\text{Cov}(x_i, \varepsilon_i) = 1$ and $\text{Cov}(x_i, \nu_i) = -2$. What happens to your results from **3.04**?

Hint You can use `mvrnorm()` from `MASS` to draw correlated variables from a multivariate Normal distribution (which you can assume here). See our simulation lab for details.

Answer First we need to slightly modify our DGP. We need to define a 3×3 variance-covariance matrix for x, ε, and ν.

```
# Function: DGP
fun_dgp2 ← function(i, n = 50) {
  # Define the covariance matrix for x, ε, and v
  Σ ← matrix(c(
     5,  1, -2,
     1,  1,  0,
    -2,  0,  7
  ), byrow = T, ncol = 3)
  # Vector means for
  μ ← c(5, 0, 0)
  # Generate x, ε, and v (and convert to tibble)
  i_df ← MASS::mvrnorm(n = n, Sigma = Σ, mu = μ) %>% as_tibble()
  names(i_df) ← c("x", "ε", "v")
  # Calculate w1, w2, and y
  i_df %<>% mutate(
    y = 3 + 7 * x + rnorm(n, sd = sqrt(3)),
    w1 = x + ε,
    w2 = x + v
  )
  # Return tibble of variables
  return(i_df)
}
```

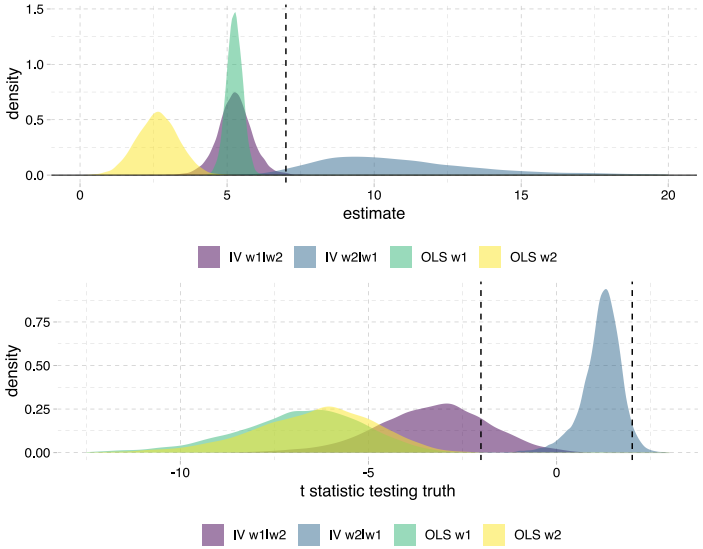
Answer, continued Now we create a new function to run one iteration and then run the simulation.

```
fun_iter2 ← function(i, n = 50) {
  # Generate and analyze the data
  fun_analysis(i = i, i_df = fun_dgp2(i, n))
}
```

```
# Set the seed
set.seed(12345)
# Tell R (furrr) to parallelize
plan(multiprocess, workers = 8)
# Usin map_dfr from furrr: Parallelize and bind resulting data frames
sim2_df ← future_map_dfr(
  # The argument to our function: The iteration number (10,000)
  1:1e4,
  # The function for each iteration
  fun_iter2,
  # Tell map_dfr to use the set seed
  .options = future_options(seed = T)
)
```

Plotting the distributions of point estimates and *t* statistics (as before), things are a mess. None of the distributions are anywhere near the true estimate, and the inference is mostly a mess, as well.

The takeaway: Correcting measurement error with IV requires classical measurement error.



Extra credit For our simple linear regression setup, show (analytically) why OLS estimates for β_1 are biased toward zero. How does IV help?