Exercises: Week 3

Econometrics Prof. Conlon

Ulrich Atz

2021-02-16

```r
library(tidyverse)
library(broom)
library(AER)
library(gmm)
library(stargazer)
data("PSID1976")
df <- subset(PSID1976, participation=="yes")
```

Estimate the relationship between:

$$\log(wage_i) = \beta_0 + \beta_1 educ_i + \beta_2 exper_i + \beta_3 exper_i^2 + \varepsilon_i$$

First we ignore the endogeneity of "education"

```r
iv_results1 <- lm(log(wage) ~ education + experience + I(experience^2), data = df)

exog_ols <- cbind(df$education, df$experience, I(df$experience^2))
gmm_results1 <- gmm(log(wage) ~ education + experience + I(experience^2),
              x = exog_ols, data = df)

summary(iv_results1)
```

```
##
## Call:
## lm(formula = log(wage) ~ education + experience + I(experience^2),
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08404 -0.30627  0.04952  0.37498  2.37115
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.5220406  0.1986321  -2.628  0.00890 **
## education        0.1074896  0.0141465   7.598 1.94e-13 ***
## experience       0.0415665  0.0131752   3.155  0.00172 **
## I(experience^2) -0.0008112  0.0003932  -2.063  0.03974 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6664 on 424 degrees of freedom
## Multiple R-squared:  0.1568, Adjusted R-squared:  0.1509
```

```
## F-statistic: 26.29 on 3 and 424 DF,  p-value: 1.302e-15
```

```r
summary(gmm_results1)
```

```
##
## Call:
## gmm(g = log(wage) ~ education + experience + I(experience^2),
##     x = exog_ols, data = df)
##
##
## Method:  twoStep
##
## Kernel:  Quadratic Spectral
##
## Coefficients:
##                   Estimate     Std. Error   t value      Pr(>|t|)
## (Intercept)      -5.2204e-01   2.0342e-01  -2.5663e+00   1.0279e-02
## education         1.0749e-01   1.3689e-02   7.8525e+00   4.0781e-15
## experience        4.1567e-02   1.4317e-02   2.9033e+00   3.6923e-03
## I(experience^2)  -8.1119e-04   3.9241e-04  -2.0672e+00   3.8713e-02
##
## J-Test: degrees of freedom is 0
##                  J-test                P-value
## Test E(g)=0:    2.98455924400836e-24  *******
```

1. How come the point estimates $\widehat{\beta}$ are the same but the standard errors are different?

In this simple example, the moment condition for the point estimates are the same as the first order condition in the OLS minimization problem. This is why OLS is also a method of moment estimator.

The standard errors, however, are calculated as follows. In the GMM case, we use a sandwich estimator for the variance of our parameters. The OLS specification here assumes constant variance and is hence the most efficient estimator. We can check if a H(A)C covariance matrix gets us closer to the GMM results.

```r
sqrt(diag(vcov(gmm_results1)))
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.2034203733    0.0136885823    0.0143168804    0.0003924069
```

```r
# As per gmm::gmm documentation
sqrt(diag(kernHAC(iv_results1)))
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.2043776509    0.0137529995    0.0143842543    0.0003942535
```

Where we have mother's and father's education as instruments for the endogenous variable (education):

```r
iv_results <- ivreg(log(wage) ~ education + experience + I(experience^2) |
                   .-education + feducation + meducation, data = df)

exog <- cbind(df$feducation, df$meducation, df$experience, I(df$experience^2))
gmm_results <- gmm(log(wage) ~ education + experience + I(experience^2),
              x = exog, data = df)

summary(iv_results)
```

```
##
## Call:
```

```
## ivreg(formula = log(wage) ~ education + experience + I(experience^2) |
##     . - education + feducation + meducation, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0986 -0.3196  0.0551  0.3689  2.3493
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.0481003  0.4003281   0.120  0.90442
## education        0.0613966  0.0314367   1.953  0.05147 .
## experience       0.0441704  0.0134325   3.288  0.00109 **
## I(experience^2) -0.0008990  0.0004017  -2.238  0.02574 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6747 on 424 degrees of freedom
## Multiple R-Squared: 0.1357,  Adjusted R-squared: 0.1296
## Wald test: 8.141 on 3 and 424 DF,  p-value: 2.787e-05
```

```
summary(gmm_results)
```

```
##
## Call:
## gmm(g = log(wage) ~ education + experience + I(experience^2),
##     x = exog, data = df)
##
##
## Method:  twoStep
##
## Kernel:  Quadratic Spectral(with bw =  0.28778 )
##
## Coefficients:
##                  Estimate     Std. Error   t value      Pr(>|t|)
## (Intercept)       0.00310758   0.46562511   0.00667400   0.99467496
## education         0.06430017   0.03689420   1.74282594   0.08136402
## experience        0.04549283   0.01436735   3.16640345   0.00154337
## I(experience^2)  -0.00093366   0.00039618  -2.35663420   0.01844140
##
## J-Test: degrees of freedom is 1
##                 J-test   P-value
## Test E(g)=0:    0.37641  0.53953
##
## Initial values of the coefficients
##     (Intercept)       education      experience I(experience^2)
##    0.0481003046    0.0613966279    0.0441703943   -0.0008989696
```

2. Why do both the point estimates and the standard errors differ now?

We are estimating 2SLS with `ivreg`; the number of instruments is greater than the number of predictors. The two parameter estimates should be equivalent if the weighting matrix is chosen as $(Z^T Z)^{-1}$. Here for GMM, however, it is the inverse of the HAC covariance matrix. The remaining difference in s.e. could be by assuming the sample mean as 0 (p. 14)?

```
gmm_results_iid <- gmm(log(wage) ~ education + experience + I(experience^2),
    x = exog, data = df, vcov = "iid")

# Compare estimates
gmm_results_iid$coefficients
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.0481003046    0.0613966279    0.0441703943   -0.0008989696
```

```
iv_results$coefficients
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.0481003046    0.0613966279    0.0441703943   -0.0008989696
```

```
# Compare standard errors
sqrt(diag(vcov(gmm_results_iid)))
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.3984529940    0.0312894503    0.0133695596    0.0003998042
```

```
sqrt(diag(vcov(iv_results)))
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.4003280773    0.0314366956    0.0134324755    0.0004016856
```

3. Let's write our own linear IV GMM estimator

Below is the one-step estimator.

a. a function that recovers $\widehat{\beta}$

```
# Could also use `crossprod`
gmm_estimates<- function(Y, X, Z, W = "inverse"){
  X = cbind(rep(1, nrow(X)), X)
  Z = cbind(1, Z) # cbind recycles
  if (is.matrix(W))  W = W
    else if (W == "inverse")    W = solve(t(Z)%*% Z)
    else if (W == "identity")   W = diag(ncol(Z))

  A = solve(t(X) %*% Z %*% W %*% t(Z) %*% X)
  B = t(X) %*% Z %*% W %*% t(Z) %*% Y
  b_gmm <- A %*% B
  b_gmm
}

# Test
gmm_estimates(log(df$wage), exog_ols, exog) %>% t()
```

```
##            [,1]       [,2]       [,3]          [,4]
## [1,] 0.0481003 0.06139663 0.04417039 -0.0008989696
```

```
gmm_results_iid$coefficients
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.0481003046    0.0613966279    0.0441703943   -0.0008989696
```

b. a function that returns the GMM objective function $Q(\theta)$

```r
gmm_obj<- function(Y, X, Z, W = "inverse", beta){
  n = nrow(X)
  X = cbind(1, X)
  Z = cbind(1, Z)
  if (is.matrix(W))  W = W
    else if (W == "inverse")    W = solve(t(Z)%*% Z)
    else if (W == "identity")   W = diag(ncol(Z))

  G = t(Z) %*% (Y - X %*% beta) / n
  Q = t(G) %*% W %*% G * n # hm
  Q
}

# Test
gmm_obj(log(df$wage), exog_ols, exog, beta = gmm_results_iid$coefficients)
```

```
##              [,1]
## [1,] 0.0003983719
```

```r
gmm_results_iid$objective
```

```
##              [,1]
## [1,] 0.0008833445
```

The objective function in gmm specifies $\|var(\bar{g})^{-1/2}\bar{g}\|^2$, so I'm not sure how to replicate it exactly.

c. a function that returns the sandwich standard errors $SE(\widehat{\beta})$

```r
gmm_se <- function(Y, X, Z, W = "inverse", beta){
  n = nrow(X)
  X = cbind(1, X)
  Z = cbind(1, Z)
  if (is.matrix(W))  W = W
    else if (W == "inverse")    W = solve(t(Z) %*% Z)
    else if (W == "identity")   W = diag(ncol(Z))

  res = (Y - X %*% beta)
  S = t(Z) %*% Z * as.numeric(t(res) %*% res) / n
  D =  t(X) %*% Z
  bread = solve(D %*% W %*% t(D))
  fill = D %*% W %*% S %*% t(W) %*% t(D)
  V =  bread %*% fill %*% bread
# V = solve(D %*% solve(S) %*% t(D))
  se = sqrt(diag(V))
  se
}


gmm_se(log(df$wage), exog_ols, exog, beta = gmm_results_iid$coefficients)
```

```
## [1] 0.3984529940 0.0312894503 0.0133695596 0.0003998042
```

```r
sqrt(diag(gmm_results_iid$vcov))
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.3984529940    0.0312894503    0.0133695596    0.0003998042
```

d. a function that returns an updated weighting matrix $\hat{W}$.

```
gmm_W <- function(Y, X, Z, beta){
  n = nrow(X)
  X = cbind(1, X)
  Z = cbind(1, Z)

  g_bar = t(Z) %*% (Y - X %*% beta) / n

  g <-  matrix(Z[1,] * c(Y[1] - X[1,] %*% beta)) - g_bar
  for (i in 2:nrow(X)) {
    g_i <- matrix(Z[i,] * c(Y[i] - X[i,] %*% beta)) - g_bar
    g <- cbind(g, g_i)
  }

  W_hat = solve(g %*% t(g) / n)
  W_hat
}

W_hat <- gmm_W(log(df$wage), exog_ols, exog, beta = gmm_results_iid$coefficients)

gmm_estimates(log(df$wage), exog_ols, exog, W = W_hat) %>% t()
```

```
##            [,1]       [,2]       [,3]          [,4]
## [1,] 0.04765346 0.06105225 0.04513615 -0.0009312341
```

```
gmm_results$coefficients
```

```
##     (Intercept)       education      experience I(experience^2)
##    0.0031075799    0.0643001733    0.0454928251   -0.0009336585
```

4. Put your GMM estimates in a table with the following:

a. OLS estimates

```r
tidy(iv_results1) %>% mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.5220	0.1986	-2.6282	0.0089
education	0.1075	0.0141	7.5983	0.0000
experience	0.0416	0.0132	3.1549	0.0017
I(experience^2)	-0.0008	0.0004	-2.0628	0.0397

b. OLS (GMM) estimates

```r
tidy(gmm_results1) %>% mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.5220	0.2034	-2.5663	0.0103
education	0.1075	0.0137	7.8525	0.0000
experience	0.0416	0.0143	2.9033	0.0037
I(experience^2)	-0.0008	0.0004	-2.0672	0.0387

c. IV estimates

```r
tidy(iv_results) %>% mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.0481	0.4003	0.1202	0.9044
education	0.0614	0.0314	1.9530	0.0515
experience	0.0442	0.0134	3.2883	0.0011
I(experience^2)	-0.0009	0.0004	-2.2380	0.0257

d. IV (GMM) estimates

```r
tidy(gmm_results) %>% mutate(across(where(is.numeric), round, 4))  %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.0031	0.4656	0.0067	0.9947
education	0.0643	0.0369	1.7428	0.0814
experience	0.0455	0.0144	3.1664	0.0015
I(experience^2)	-0.0009	0.0004	-2.3566	0.0184

e. Your estimates of one-step GMM using Identity weights

```r
est_1s <- gmm_estimates(log(df$wage), exog_ols, exog, W = "identity")
se_1s <- gmm_se(log(df$wage), exog_ols, exog, beta = est_1s)

tidy(gmm_results) %>%
  mutate(
    estimate = est_1s,
```

```r
    std.error = se_1s,
    statistic = est_1s / se_1s,
    p.value = pnorm(abs(statistic), lower.tail = F)
  ) %>%
  mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.9703	0.3964	-2.4477	0.0072
education	0.1285	0.0311	4.1274	0.0000
experience	0.0639	0.0133	4.8025	0.0000
I(experience^2)	-0.0014	0.0004	-3.4381	0.0003

f. Your estimates of two-step GMM starting at Identity weights

```r
w_hat <- gmm_W(log(df$wage), exog_ols, exog, beta = est_1s)
est_2s <- gmm_estimates(log(df$wage), exog_ols, exog, W = w_hat)
se_2s <- gmm_se(log(df$wage), exog_ols, exog, beta = est_2s)

tidy(gmm_results) %>%
  mutate(
    estimate = est_2s,
    std.error = se_2s,
    statistic = est_2s / se_2s,
    p.value = pnorm(abs(statistic), lower.tail = F)
  ) %>%
  mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.0391	0.3984	0.0980	0.4610
education	0.0617	0.0313	1.9708	0.0244
experience	0.0454	0.0134	3.3999	0.0003
I(experience^2)	-0.0009	0.0004	-2.3546	0.0093

g. Your estimates of one-step GMM using 2SLS weights

```r
est_1s <- gmm_estimates(log(df$wage), exog_ols, exog, W = "inverse")
se_1s <- gmm_se(log(df$wage), exog_ols, exog, beta = est_1s)

tidy(gmm_results) %>%
  mutate(
    estimate = est_1s,
    std.error = se_1s,
    statistic = est_1s / se_1s,
    p.value = pnorm(abs(statistic), lower.tail = F)
  ) %>%
  mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.0481	0.3985	0.1207	0.4520
education	0.0614	0.0313	1.9622	0.0249
experience	0.0442	0.0134	3.3038	0.0005
I(experience^2)	-0.0009	0.0004	-2.2485	0.0123

h. Your estimates of two-step GMM starting at 2SLS weights

```
w_hat <- gmm_W(log(df$wage), exog_ols, exog, beta = est_1s)
est_2s <- gmm_estimates(log(df$wage), exog_ols, exog, W = w_hat)
se_2s <- gmm_se(log(df$wage), exog_ols, exog, beta = est_2s)

tidy(gmm_results) %>%
  mutate(
    estimate = est_2s,
    std.error = se_2s,
    statistic = est_2s / se_2s,
    p.value = pnorm(abs(statistic), lower.tail = F)
  ) %>%
  mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.0477	0.3985	0.1196	0.4524
education	0.0611	0.0313	1.9508	0.0255
experience	0.0451	0.0134	3.3754	0.0004
I(experience^2)	-0.0009	0.0004	-2.3288	0.0099

i. Use the (GMM) package to estimate continuously updating GMM (type='cue')

```
gmm_cue <- gmm(log(wage) ~ education + experience + I(experience^2),
               x = exog, data = df, type = 'cue')
tidy(gmm_cue) %>% mutate(across(where(is.numeric), round, 4)) %>%
  kableExtra::kbl(booktabs = T)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.0043	0.4657	0.0093	0.9926
education	0.0642	0.0369	1.7399	0.0819
experience	0.0455	0.0144	3.1669	0.0015
I(experience^2)	-0.0009	0.0004	-2.3579	0.0184