Problem Set 6 R Solutions

Matthew Alampay Davis

April 13, 2021

Load the data

```
daron <- read.dta13("maketable4.dta")
```

Question 1

1a

Filter out countries without a European colonial history or that are missing data

```
daron %<>% filter(baseco == 1)
```

Columns 1-2

To do 2SLS, let's use *iv_robust*, which like *lm_robust* comes from the *estimatr* package. Note: for some reason, Stata uses a different type of robust standard error when estimating IV models so the argument for **se_type here is 'HC0' rather than 'HC1' (which is the same as 'stata')**.

The regression formula is as usual except with a bar "|". Regressors go on the left side of the bar and instruments on the right.

```
reg1 <- iv_robust(logpgp95 ~ avexpr | logem4, data = daron, se_type = "HC0")  # different SEs
summary(reg1)
```

```
##
## Call:
## iv_robust(formula = logpgp95 ~ avexpr | logem4, data = daron,
##     se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   1.9097     1.1740   1.627 1.089e-01  -0.4370    4.256 62
## avexpr        0.9443     0.1761   5.362 1.289e-06   0.5923    1.296 62
##
## Multiple R-squared:  0.187 , Adjusted R-squared:  0.1739
## F-statistic: 28.75 on 1 and 62 DF,  p-value: 1.289e-06
```

Now the second model. Note here that we want to model *lat_abst* as exogenous (i.e. we only want to instrument for *avexpr*). So we include *lat_abst* on both sides of the bar "|" in the formula argument. If we hadn't included it on the RHS, then the model would assume *lat_abst* was an endogenous variable and we'd have more endogenous variables than instruments, which would prevent the regression from running.

```
reg2 <- iv_robust(logpgp95 ~ lat_abst + avexpr | lat_abst + logem4,
    data = daron, se_type = "HC0")
summary(reg2)
```

```
##
```

```
## Call:
## iv_robust(formula = logpgp95 ~ lat_abst + avexpr | lat_abst +
##     logem4, data = daron, se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   1.6918     1.4478  1.1686 0.2471284  -1.2032    4.587 61
## lat_abst     -0.6472     1.2270 -0.5275 0.5997834  -3.1008    1.806 61
## avexpr        0.9957     0.2403  4.1431 0.0001073   0.5151    1.476 61
##
## Multiple R-squared:  0.1025 ,    Adjusted R-squared:  0.07305
## F-statistic: 14.17 on 2 and 61 DF,  p-value: 8.846e-06
```

For regressions 1 and 2, no standard errors differ by more than 0.20

1b

1b i) Method 1

This question has you do the first-stage and second-stage regressions yourself.

```
# First stage
mod.1bi.1 <- lm_robust(avexpr ~ logem4, data = daron, se_type = "stata")
# Second stage
mod.1bi.2 <- lm_robust(daron$logpgp95 ~ mod.1bi.1$fitted.values,
    se_type = "stata")
summary(mod.1bi.2)
```

```
##
## Call:
## lm_robust(formula = daron$logpgp95 ~ mod.1bi.1$fitted.values,
##     se_type = "stata")
##
## Standard error type:  HC1
##
## Coefficients:
##                          Estimate Std. Error t value  Pr(>|t|) CI Lower
## (Intercept)                1.9097     0.7569   2.523 1.422e-02   0.3966
## mod.1bi.1$fitted.values    0.9443     0.1203   7.852 7.142e-11   0.7039
##                          CI Upper DF
## (Intercept)                 3.423 62
## mod.1bi.1$fitted.values     1.185 62
##
## Multiple R-squared:  0.4772 ,    Adjusted R-squared:  0.4688
## F-statistic: 61.66 on 1 and 62 DF,  p-value: 7.142e-11
```

Note that the solutions tell us the standard errors that come out of this exercise are incorrect. So even though they're wrong, I used 'stata' standard errors above so they match what's in the Stata solutions. It seems like whenever Stata runs non-IV regressions, they use 'stata' standard errors and when they run IV in one command, they use 'HC0' standard errors. As far as I know, there's no deep reason for this.

Following the solutions, you have to go back to using the all-in-one-step method from Part A using 'HC0' standard errors to get the proper standard errors:

```
mod.1bii <- iv_robust(logpgp95 ~ avexpr | logem4, data = daron,
    se_type = "HC0")
summary(mod.1bii)
```

```
##
## Call:
## iv_robust(formula = logpgp95 ~ avexpr | logem4, data = daron,
##     se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   1.9097     1.1740   1.627 1.089e-01  -0.4370    4.256 62
## avexpr        0.9443     0.1761   5.362 1.289e-06   0.5923    1.296 62
##
## Multiple R-squared:  0.187 , Adjusted R-squared:  0.1739
## F-statistic: 28.75 on 1 and 62 DF,  p-value: 1.289e-06
```

1b ii) Method 2

```
cov(select(daron, logpgp95, avexpr, logem4))
```

```
##            logpgp95     avexpr     logem4
## logpgp95  1.0885987  1.1261454 -0.9067352
## avexpr    1.1261454  2.1569245 -0.9602404
## logem4   -0.9067352 -0.9602404  1.5825228
```

1b iii) Method 3

```
# First stage regression
mod.1biii.1 <- lm_robust(avexpr ~ logem4, data = daron, se_type = "stata")
# Second stage regression
mod.1biii.2 <- lm_robust(logpgp95 ~ logem4, data = daron, se_type = "stata")
# Divide the coefficient in the second by the coefficient in
# the first
coef(mod.1biii.2)[2]/coef(mod.1biii.1)[2]
```

```
##    logem4
## 0.9442794
```

$$\beta_3 = \frac{-0.9067352}{-0.9602404} \approx 0.9443$$

1c

Repeat for regressions 3 and 4, excluding the four countries with exceptional GDPs

```
daron2 <- filter(daron, rich4 != 1)
reg3 <- iv_robust(logpgp95 ~ avexpr | logem4, data = daron2,
    se_type = "HC0")
summary(reg3)
```

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

```
## iv_robust(formula = logpgp95 ~ avexpr | logem4, data = daron2,
##     se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)  -0.1412     2.5832 -0.05467  0.95659  -5.3120    5.030 58
## avexpr        1.2812     0.4019  3.18800  0.00231   0.4768    2.086 58
##
## Multiple R-squared:  -0.6877 ,   Adjusted R-squared:  -0.7168
## F-statistic: 10.16 on 1 and 58 DF,  p-value: 0.00231
```

```
reg4 <- iv_robust(logpgp95 ~ avexpr + lat_abst | logem4 + lat_abst,
    data = daron2, se_type = "HC0")
summary(reg4)
```

```
##
## Call:
## iv_robust(formula = logpgp95 ~ avexpr + lat_abst | logem4 + lat_abst,
##     data = daron2, se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   0.1442     2.4541 0.05877 0.953345  -4.7701    5.059 57
## avexpr        1.2118     0.3921 3.09023 0.003091   0.4265    1.997 57
## lat_abst      0.9385     1.1910 0.78800 0.433964  -1.4465    3.324 57
##
## Multiple R-squared:  -0.4918 ,   Adjusted R-squared:  -0.5442
## F-statistic: 6.768 on 2 and 57 DF,  p-value: 0.002305
```

Repeat for regressions 5 and 6, excluding the African countries

```
daron3 <- filter(daron, africa != 1)
reg5 <- iv_robust(logpgp95 ~ avexpr | logem4, data = daron3,
    se_type = "HC0")
reg6 <- iv_robust(logpgp95 ~ avexpr + lat_abst | logem4 + lat_abst,
    data = daron3, se_type = "HC0")
```

For regressions 3 through 6, two standard errors differ by more than 0.20: t > he SE on latitude in the first stage in regression 4 and in regression 6.

1d

No exclusions. Add continent indicators as controls

Create a variable called "other" that equals one if the observation is not in the base group, not coded Africa or Asia, and not in the Americas:

```
daron %<>% mutate(other = shortnam %in% c("AUS", "NZL", "MLT"))
table(daron$other)
```

```
##
## FALSE  TRUE
##    61     3
```

Then the regression:

```
reg7 <- iv_robust(logpgp95 ~ avexpr + africa + asia + other |
    logem4 + africa + asia + other, data = daron, se_type = "HC0")
summary(reg7)
```

```
##
## Call:
## iv_robust(formula = logpgp95 ~ avexpr + africa + asia + other |
##     logem4 + africa + asia + other, data = daron, se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   2.0324     2.1905  0.9278 0.357278   -2.351   6.4156 59
## avexpr        0.9822     0.3309  2.9679 0.004327    0.320   1.6444 59
## africa       -0.4643     0.3131 -1.4830 0.143388   -1.091   0.1622 59
## asia         -0.9242     0.3705 -2.4947 0.015424   -1.666  -0.1829 59
## otherTRUE    -0.9405     0.9121 -1.0312 0.306667   -2.766   0.8846 59
##
## Multiple R-squared:  0.2286 ,    Adjusted R-squared:  0.1763
## F-statistic: 9.094 on 4 and 59 DF,  p-value: 8.607e-06
```

```
reg8 <- iv_robust(logpgp95 ~ avexpr + africa + asia + other +
    lat_abst | logem4 + africa + asia + other + lat_abst, data = daron,
    se_type = "HC0")
summary(reg8)
```

```
##
## Call:
## iv_robust(formula = logpgp95 ~ avexpr + africa + asia + other +
##     lat_abst | logem4 + africa + asia + other + lat_abst, data = daron,
##     se_type = "HC0")
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
## (Intercept)   1.4405     3.0736  0.4687  0.64107  -4.7120  7.59294 58
## avexpr        1.1071     0.5029  2.2014  0.03170   0.1004  2.11372 58
## africa       -0.4373     0.3775 -1.1584  0.25145  -1.1929  0.31833 58
## asia         -1.0471     0.5049 -2.0738  0.04254  -2.0578 -0.03642 58
## otherTRUE    -0.9904     1.0594 -0.9349  0.35371  -3.1109  1.13014 58
## lat_abst     -1.1782     1.7917 -0.6576  0.51342  -4.7647  2.40834 58
##
## Multiple R-squared:  0.01082 ,   Adjusted R-squared:  -0.07445
## F-statistic: 5.689 on 5 and 58 DF,  p-value: 0.000246
```

```
reg9 <- iv_robust(loghjypl ~ avexpr | logem4, data = daron, se_type = "HC0")
summary(reg9)
```

```
##
## Call:
## iv_robust(formula = loghjypl ~ avexpr | logem4, data = daron,
##     se_type = "HC0")
```

```
##
## Standard error type:  HC0
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper DF
## (Intercept)  -8.3229     1.3066  -6.370 3.121e-08 -10.9373   -5.708 59
## avexpr        0.9808     0.1963   4.996 5.533e-06   0.5879    1.374 59
##
## Multiple R-squared:  -0.1518 ,   Adjusted R-squared:  -0.1714
## F-statistic: 24.96 on 1 and 59 DF,  p-value: 5.533e-06
```

1e

Any recap, highlights, points of interest, opinions, or assertions gets points.

1f

We'll modify Regression 9

IV-GMM requires the *gmm* package

```
y <- daron$loghjypl  # outcome
d <- daron$avexpr  # endogenous regressor
z <- daron$logem4  # instrument

iv.gmm <- gmm(y ~ d, x = z)
```

```
## Warning in getDat(object$g, object$x, data = object$data): There are
## missing values. Associated observations have been removed
```

```
summary(iv.gmm)
```

```
##
## Call:
## gmm(g = y ~ d, x = z)
##
##
## Method:  twoStep
##
## Kernel:  Quadratic Spectral
##
## Coefficients:
##              Estimate     Std. Error   t value      Pr(>|t|)
## (Intercept)  -8.3229e+00   1.4764e+00  -5.6372e+00   1.7287e-08
## d             9.8076e-01   2.2103e-01   4.4373e+00   9.1115e-06
##
## J-Test: degrees of freedom is 0
##                 J-test                P-value
## Test E(g)=0:    1.71609813591347e-27  *******
```

IV-LIML requires the *ivmodel* package

```
iv.liml <- ivmodel(y, d, z)
LIML(iv.liml) %>% unlist
```

```
##    point.est      std.err    test.stat      p.value          ci1
## 9.807598e-01 1.709027e-01 5.738702e+00 3.494678e-07 6.387842e-01
##          ci2            k
```

```
## 1.322735e+00 1.000000e+00
```

Question 2

Does not require R so see the official solutions