

# R OLS and Instrumental Variable Regression

Fan Wang

2020-04-01

## Contents

OLS and IV Regression . . . . . 1

Back to [Fan's R4Econ Homepage](#) [Table of Content](#)

## OLS and IV Regression

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) Package, [R Code Examples](#) Repository ([bookdown site](#)), or [Intro Stats with R](#) Repository ([bookdown site](#)).

IV regression using AER package. Option to store all results in dataframe row for combining results from other estimations together. Produce Row Statistics.

```
# IV regression function
# The code below uses the AER library's regresison function
# All results are stored in a single row as data_frame
# This functoin could work with dplyr do
# var.y is single outcome, vars.x, vars.c and vars.z are vectors of endogenous variables, controls and
regf.iv <- function(var.y, vars.x,
                    vars.c, vars.z, df, transpose=TRUE) {

  # A. Set-Up Equation
  str.vars.x <- paste(vars.x, collapse='+')
  str.vars.c <- paste(vars.c, collapse='+')

  df <- df %>%
    select(one_of(var.y, vars.x, vars.c, vars.z)) %>%
    drop_na() %>% filter_all(all_vars(!is.infinite()))

  if (length(vars.z) >= 1) {
    # library(AER)
    str.vars.z <- paste(vars.z, collapse='+')
    equa.iv <- paste(var.y,
                     paste(paste(str.vars.x, str.vars.c, sep='+'),
                           paste(str.vars.z, str.vars.c, sep='+'),
                           sep='|'),
                     sep='~')

    # print(equa.iv)

    # B. IV Regression
    ivreg.summ <- summary(ivreg(as.formula(equa.iv), data=df),
                             vcov = sandwich, df = Inf, diagnostics = TRUE)
```

```

# C. Statistics from IV Regression
#     ivreg.summ$coef
#     ivreg.summ$diagnostics

# D. Combine Regression Results into a Matrix
df.results <- suppressWarnings(suppressMessages(
  as_tibble(ivreg.summ$coef, rownames='rownames') %>%
    full_join(as_tibble(ivreg.summ$diagnostics, rownames='rownames')) %>%
    full_join(tibble(rownames=c('vars'),
                      var.y=var.y,
                      vars.x=str.vars.x,
                      vars.z=str.vars.z,
                      vars.c=str.vars.c))))

} else {

  # OLS regression
  equa.ols <- paste(var.y,
                    paste(paste(vars.x, collapse='+'),
                          paste(vars.c, collapse='+'), sep='+'),
                    sep='~')

  lmreg.summ <- summary(lm(as.formula(equa.ols), data=df))

  lm.diagnostics <- as_tibble(
    list(df1=lmreg.summ$df[[1]],
         df2=lmreg.summ$df[[2]],
         df3=lmreg.summ$df[[3]],
         sigma=lmreg.summ$sigma,
         r.squared=lmreg.summ$r.squared,
         adj.r.squared=lmreg.summ$adj.r.squared)) %>%
    gather(variable, value) %>%
    rename(rownames = variable) %>%
    rename(v = value)

  df.results <- suppressWarnings(suppressMessages(
    as_tibble(lmreg.summ$coef, rownames='rownames') %>%
      full_join(lm.diagnostics) %>%
      full_join(tibble(rownames=c('vars'),
                        var.y=var.y,
                        vars.x=str.vars.x,
                        vars.c=str.vars.c))))

}

# E. Flatten Matrix, All IV results as a single tibble
# row to be combined with other IV results
df.row.results <- df.results %>%
  gather(variable, value, -rownames) %>%
  drop_na() %>%
  unite(esti.val, rownames, variable) %>%
  mutate(esti.val = gsub(' ', '', esti.val))

if (transpose) {
  df.row.results <- df.row.results %>% spread(esti.val, value)
}

```

```

}

# F. Return
return(data.frame(df.row.results))
}

```

## Construct Program

### Program Testing Load Data

```

# Library
library(tidyverse)
library(AER)

# Load Sample Data
setwd('C:/Users/fan/R4Econ/_data/')
df <- read_csv('height_weight.csv')

```

```

# One Instrumments
var.y <- c('hgt')
vars.x <- c('prot')
vars.z <- NULL
vars.c <- c('sex', 'hgt0', 'wgt0')
# Regression
regf.iv(var.y, vars.x, vars.c, vars.z, df, transpose=FALSE) %>%
  kable() %>%
  kable_styling_fc()

```

### Example No Instrument, OLS

```

# One Instrumments
var.y <- c('hgt')
vars.x <- c('prot')
vars.z <- c('momEdu')
vars.c <- c('sex', 'hgt0', 'wgt0')
# Regression
regf.iv(var.y, vars.x, vars.c, vars.z, df, transpose=FALSE) %>%
  kable() %>%
  kable_styling_fc()

```

### Example 1 Instrumment

```

# Multiple Instrumments
var.y <- c('hgt')
vars.x <- c('prot')
vars.z <- c('momEdu', 'wealthIdx', 'p.A.prot', 'p.A.nProt')
vars.c <- c('sex', 'hgt0', 'wgt0')
# Regression
regf.iv(var.y, vars.x, vars.c, vars.z, df, transpose=FALSE) %>%
  kable() %>%
  kable_styling_fc()

```

esti.val	value
(Intercept) _Estimate	52.1186286658651
prot _Estimate	0.374472386357917
sexMale _Estimate	0.611043720578292
hgt0 _Estimate	0.148513781160842
wgt0 _Estimate	0.00150560230505631
(Intercept) _Std.Error	1.57770483608693
prot _Std.Error	0.00418121191133815
sexMale _Std.Error	0.118396259120659
hgt0 _Std.Error	0.0393807494783186
wgt0 _Std.Error	0.000187123663624397
(Intercept) _tvalue	33.0344608660332
prot _tvalue	89.5607288744356
sexMale _tvalue	5.16100529794248
hgt0 _tvalue	3.77122790013449
wgt0 _tvalue	8.04602836377991
(Intercept) _Pr(> t )	9.92126150975783e-233
prot _Pr(> t )	0
sexMale _Pr(> t )	2.48105505495642e-07
hgt0 _Pr(> t )	0.000162939618371183
wgt0 _Pr(> t )	9.05257561534111e-16
df1 _v	5
df2 _v	18958
df3 _v	5
sigma _v	8.06197784622979
r.squared _v	0.319078711001325
adj.r.squared _v	0.318935041565942
vars _var.y	hgt
vars _vars.x	prot
vars _vars.c	sex+hgt0+wgt0

## Example Multiple Instrucments

```
# Multiple Instrucments
var.y <- c('hgt')
vars.x <- c('prot', 'cal')
vars.z <- c('momEdu', 'wealthIdx', 'p.A.prot', 'p.A.nProt')
vars.c <- c('sex', 'hgt0', 'wgt0')
# Regression
regf.iv(var.y, vars.x, vars.c, vars.z, df, transpose=FALSE) %>%
  kable() %>%
  kable_styling_fc()
```

## Example Multiple Endogenous Variables

**Examples Line by Line** The examples are just to test the code with different types of variables.

```
# Selecting Variables
var.y <- c('hgt')
vars.x <- c('prot', 'cal')
vars.z <- c('momEdu', 'wealthIdx', 'p.A.prot', 'p.A.nProt')
vars.c <- c('sex', 'hgt0', 'wgt0')
```

esti.val	value
(Intercept)_Estimate	43.4301969117558
prot_Estimate	0.130833343849446
sexMale_Estimate	0.868121847262411
hgt0_Estimate	0.412093881817148
wgt0_Estimate	0.000858630042617921
(Intercept)_Std.Error	1.82489550971182
prot_Std.Error	0.0192036220809189
sexMale_Std.Error	0.13373016700542
hgt0_Std.Error	0.0459431912927002
wgt0_Std.Error	0.00022691057702563
(Intercept)_zvalue	23.798730766023
prot_zvalue	6.81295139521853
sexMale_zvalue	6.49159323361366
hgt0_zvalue	8.96963990141069
wgt0_zvalue	3.7840018472164
(Intercept)_Pr(> z )	3.4423766196876e-125
prot_Pr(> z )	9.56164541643828e-12
sexMale_Pr(> z )	8.49333228172763e-11
hgt0_Pr(> z )	2.97485394526792e-19
wgt0_Pr(> z )	0.000154326676608523
Weakinstruments_df1	1
Wu-Hausman_df1	1
Sargan_df1	0
Weakinstruments_df2	16394
Wu-Hausman_df2	16393
Weakinstruments_statistic	935.817456612075
Wu-Hausman_statistic	123.595856606729
Weakinstruments_p-value	6.39714929178024e-200
Wu-Hausman_p-value	1.30703637796748e-28
vars_var.y	hgt
vars_vars.x	prot
vars_vars.z	momEdu
vars_vars.c	sex+hgt0+wgt0

*# A. create Equation*

```
str.vars.x <- paste(vars.x, collapse='+')
str.vars.c <- paste(vars.c, collapse='+')
str.vars.z <- paste(vars.z, collapse='+')
print(str.vars.x)
```

```
## [1] "prot+cal"
```

```
print(str.vars.c)
```

```
## [1] "sex+hgt0+wgt0"
```

```
print(str.vars.z)
```

```
## [1] "momEdu+wealthIdx+p.A.prot+p.A.nProt"
```

```
equa.iv <- paste(var.y,
  paste(paste(str.vars.x, str.vars.c, sep='+'),
    paste(str.vars.z, str.vars.c, sep='+'),
```

esti.val	value
(Intercept)_Estimate	42.2437613555242
prot_Estimate	0.26699945194704
sexMale_Estimate	0.695548488812932
hgt0_Estimate	0.424954881263031
wgt0_Estimate	0.000486951420329484
(Intercept)_Std.Error	1.85356686789642
prot_Std.Error	0.0154939347964083
sexMale_Std.Error	0.133157977814374
hgt0_Std.Error	0.0463195803786233
wgt0_Std.Error	0.000224867994873235
(Intercept)_zvalue	22.7905246296649
prot_zvalue	17.2325142357597
sexMale_zvalue	5.22348341593581
hgt0_zvalue	9.17441129192849
wgt0_zvalue	2.16549901022595
(Intercept)_Pr(> z )	5.69294074735747e-115
prot_Pr(> z )	1.51424021931607e-66
sexMale_Pr(> z )	1.75588197502565e-07
hgt0_Pr(> z )	4.54048595587756e-20
wgt0_Pr(> z )	0.030349491114332
Weakinstruments_df1	4
Wu-Hausman_df1	1
Sargan_df1	3
Weakinstruments_df2	14914
Wu-Hausman_df2	14916
Weakinstruments_statistic	274.147084958343
Wu-Hausman_statistic	17.7562545747101
Sargan_statistic	463.729664547249
Weakinstruments_p-value	8.61731956233366e-228
Wu-Hausman_p-value	2.52567249124181e-05
Sargan_p-value	3.45452874915475e-100
vars_var.y	hgt
vars_vars.x	prot
vars_vars.z	momEdu+wealthIdx+p.A.prot+p.A.nProt
vars_vars.c	sex+hgt0+wgt0

```

                sep='| '),
                sep='~')
print(equa.iv)

## [1] "hgt~prot+cal+sex+hgt0+wgt0|momEdu+wealthIdx+p.A.prot+p.A.nProt+sex+hgt0+wgt0"
# B. regression
res.ivreg <- ivreg(as.formula(equa.iv), data=df)
coef(res.ivreg)

##      (Intercept)      prot      cal      sexMale      hgt0      wgt0
## 44.0243196254 -1.4025623247 0.0651048958 0.1208327876 0.2865254380 0.0008504814
# C. Regression Summary
ivreg.summ <- summary(res.ivreg, vcov = sandwich, df = Inf, diagnostics = TRUE)

```

esti.val	value
(Intercept) _Estimate	44.0243196254297
prot _Estimate	-1.4025623247106
cal _Estimate	0.065104895750151
sexMale _Estimate	0.120832787571818
hgt0 _Estimate	0.286525437984517
wgt0 _Estimate	0.000850481389651033
(Intercept) _Std.Error	2.75354847244082
prot _Std.Error	0.198640060273635
cal _Std.Error	0.00758881298880996
sexMale _Std.Error	0.209984580636303
hgt0 _Std.Error	0.0707828182888255
wgt0 _Std.Error	0.00033711210444429
(Intercept) _zvalue	15.9882130516502
prot _zvalue	-7.06082309267581
cal _zvalue	8.57906181719737
sexMale _zvalue	0.575436478267434
hgt0 _zvalue	4.04795181812859
wgt0 _zvalue	2.52284441418383
(Intercept) _Pr(> z )	1.54396598126854e-57
prot _Pr(> z )	1.65519210848649e-12
cal _Pr(> z )	9.56500648203187e-18
sexMale _Pr(> z )	0.564996139463599
hgt0 _Pr(> z )	5.16677787108928e-05
wgt0 _Pr(> z )	0.0116409892837831
Weakinstruments(prot) _df1	4
Weakinstruments(cal) _df1	4
Wu-Hausman _df1	2
Sargan _df1	2
Weakinstruments(prot) _df2	14914
Weakinstruments(cal) _df2	14914
Wu-Hausman _df2	14914
Weakinstruments(prot) _statistic	274.147084958343
Weakinstruments(cal) _statistic	315.036848606231
Wu-Hausman _statistic	94.7020085425169
Sargan _statistic	122.081979628898
Weakinstruments(prot) _p-value	8.61731956233366e-228
Weakinstruments(cal) _p-value	1.18918641220866e-260
Wu-Hausman _p-value	1.35024050408262e-41
Sargan _p-value	3.09196773720398e-27
vars_var.y	hgt
vars_vars.x	prot+cal
vars_vars.z	momEdu+wealthIdx+p.A.prot+p.A.nProt
vars_vars.c	sex+hgt0+wgt0

```
ivreg.summ$coef
```

```
##              Estimate   Std. Error   z value   Pr(>|z|)
## (Intercept) 44.0243196254 2.7535484724 15.9882131 1.543966e-57
## prot        -1.4025623247 0.1986400603 -7.0608231 1.655192e-12
## cal          0.0651048958 0.0075888130  8.5790618 9.565006e-18
```

```
## sexMale      0.1208327876 0.2099845806 0.5754365 5.649961e-01
## hgt0         0.2865254380 0.0707828183 4.0479518 5.166778e-05
## wgt0         0.0008504814 0.0003371121 2.5228444 1.164099e-02
## attr(,"df")
## [1] 0
```

```
ivreg.summ$diagnostics
```

```
##              df1   df2 statistic      p-value
## Weak instruments (prot)  4 14914 274.14708 8.617320e-228
## Weak instruments (cal)  4 14914 315.03685 1.189186e-260
## Wu-Hausman              2 14914  94.70201 1.350241e-41
## Sargan                  2    NA 122.08198 3.091968e-27
```

```
# D. Combine Regression Results into a Matrix
```

```
df.results <- suppressMessages(as_tibble(ivreg.summ$coef, rownames='rownames') %>%
  full_join(as_tibble(ivreg.summ$diagnostics, rownames='rownames')) %>%
  full_join(tibble(rownames=c('vars'),
                    var.y=var.y,
                    vars.x=str.vars.x,
                    vars.z=str.vars.z,
                    vars.c=str.vars.c)))
```

```
# E. Flatten Matrix, All IV results as a single tibble row to be combined with other IV results
```

```
df.row.results <- df.results %>%
  gather(variable, value, -rownames) %>%
  drop_na() %>%
  unite(esti.val, rownames, variable) %>%
  mutate(esti.val = gsub(' ', '', esti.val))
```

```
# F. Results as Single Column
```

```
# df.row.results
```

```
# G. Results as Single Row
```

```
# df.row.results
```

```
# t(df.row.results %>% spread(esti.val, value)) %>%
# kable() %>%
# kable_styling_fc_wide()
```