

R OLS and Instrumental Variable Regression M Outcomes and N RHS Alternatives

Fan Wang

2020-04-01

Contents

1	IV Loop over RHS	1
1.1	Construct Program	1
1.2	Prepare Data	2
1.3	Program Testing	3
1.3.1	Test Program OLS Z-Stat	3
1.3.2	Test Program IV T-stat	3
1.3.3	Test Program OLS Coefficient	3
1.3.4	Test Program IV coefficient	4
1.3.5	Test Program OLS Return All	4
1.3.6	Test Program IV Return All	4
1.4	Program Line by Line	5
1.4.1	Lapply	5
1.4.2	Nested Lapply Test	5
1.4.3	Nested Lapply All	6
1.4.4	Nested Lapply Select	6

1 IV Loop over RHS

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](#)).

Regression with a Variety of Outcome Variables and Right Hand Side Variables. There are M outcome variables, and there are N alternative right hand side variables. Regress each M outcome variable and each N alternative right hand side variable, with some common sets of controls and perhaps shared instruments. The output file is a M by N matrix of coefficients, with proper variable names and row names. The matrix stores coefficients for this key endogenous variable.

- Dependency: *R4Econ/linreg/ivreg/ivregdfrow.R*

1.1 Construct Program

The program relies on double lapply. lapply is used for convenience, not speed.

```
ff_reg_mbyn <- function(list.vars.y, list.vars.x,
                        vars.c, vars.z, df,
                        return_all = FALSE,
                        stats_ends = 'value', time = FALSE) {

  # regf.iv() function is from C:\Users\fan\R4Econ\linreg\ivreg\ivregdfrow.R
```

```

if (time) {
  start_time <- Sys.time()
}

if (return_all) {
  df.reg.out.all <-
    bind_rows(lapply(list.vars.x,
                      function(x) (
                        bind_rows(
                          lapply(list.vars.y, regf.iv,
                                vars.x=x, vars.c=vars.c, vars.z=vars.z, df=df))
                        )))
} else {
  df.reg.out.all <-
    (lapply(list.vars.x,
            function(x) (
              bind_rows(
                lapply(list.vars.y, regf.iv,
                      vars.x=x, vars.c=vars.c, vars.z=vars.z, df=df)) %>%
                select(vars_var.y, starts_with(x)) %>%
                select(vars_var.y, ends_with(stats_ends))
              ))) %>% reduce(full_join)
}

if (time) {
  end_time <- Sys.time()
  print(paste0('Estimation for all ys and xs took (seconds):',
               end_time - start_time))
}

return(df.reg.out.all)
}

```

1.2 Prepare Data

```

# Library
library(tidyverse)
library(AER)

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

# Source Dependency
source('C:/Users/fan/R4Econ/linreg/ivreg/ivregdfrow.R')

# Setting
options(repr.matrix.max.rows=50, repr.matrix.max.cols=50)

```

Parameters.

```

var.y1 <- c('hgt')
var.y2 <- c('wgt')

```

```

var.y3 <- c('vil.id')
list.vars.y <- c(var.y1, var.y2, var.y3)

var.x1 <- c('prot')
var.x2 <- c('cal')
var.x3 <- c('wealthIdx')
var.x4 <- c('p.A.prot')
var.x5 <- c('p.A.nProt')
list.vars.x <- c(var.x1, var.x2, var.x3, var.x4, var.x5)

vars.z <- c('indi.id')
vars.c <- c('sex', 'wgt0', 'hgt0', 'svymthRound')

```

1.3 Program Testing

1.3.1 Test Program OLS Z-Stat

```

vars.z <- NULL
suppressWarnings(suppressMessages(
  ff_reg_mbyn(list.vars.y, list.vars.x,
    vars.c, vars.z, df,
    return_all = FALSE,
    stats_ends = 'value')))) %>%
kable() %>%
kable_styling_fc_wide()

```

vars_var.y	prot_tvalue	cal_tvalue	wealthIdx_tvalue	p.A.prot_tvalue	p.A.nProt_tvalue
hgt	18.8756010031786	23.4421863484661	13.508899618216	3.83682180045518	32.5448257554855
wgt	16.3591125056062	17.3686031309332	14.1390521528113	1.36958319982295	12.0961557911467
vil.id	-14.9385580468907	-19.6150110809452	34.0972558327347	8.45943342783186	17.7801422421419

1.3.2 Test Program IV T-stat

```

vars.z <- c('indi.id')
suppressWarnings(suppressMessages(
  ff_reg_mbyn(list.vars.y, list.vars.x,
    vars.c, vars.z, df,
    return_all = FALSE,
    stats_ends = 'value')))) %>%
kable() %>%
kable_styling_fc_wide()

```

vars_var.y	prot_zvalue	cal_zvalue	wealthIdx_zvalue	p.A.prot_zvalue	p.A.nProt_zvalue
hgt	8.87674929300964	12.0739764947235	4.62589553677969	26.6373587567312	32.1162192385744
wgt	5.60385871756365	6.1225187008946	5.17869536991717	11.9295584469998	12.3509307017263
vil.id	-9.22106223347162	-13.0586007975839	-51.5866689219593	-29.9627476577329	-38.3528894620707

1.3.3 Test Program OLS Coefficient

```

vars.z <- NULL
suppressWarnings(suppressMessages(
  ff_reg_mbyn(list.vars.y, list.vars.x,

```

```

vars.c, vars.z, df,
return_all = FALSE,
stats_ends = 'Estimate')))) %>%
kable() %>%
kable_styling_fc_wide()

```

vars_var.y	prot_Estimate	cal_Estimate	wealthIdx_Estimate	p.A.prot_Estimate	p.A.nProt_Estimate
hgt	0.049431093806755	0.00243408846205622	0.21045655488185	3.86952250259526e-05	0.00542428867316449
wgt	16.5557424523585	0.699072500364623	106.678721085969	0.00521731297924587	0.779514232050632
vil.id	-0.0758835879205584	-0.00395676177098486	0.451733304543324	0.000149388430455142	0.00526237555581024

1.3.4 Test Program IV coefficient

```

vars.z <- c('indi.id')
suppressWarnings(suppressMessages(
  ff_reg_mbyn(list.vars.y, list.vars.x,
    vars.c, vars.z, df,
    return_all = FALSE,
    stats_ends = 'Estimate')))) %>%
kable() %>%
kable_styling_fc_wide()

```

vars_var.y	prot_Estimate	cal_Estimate	wealthIdx_Estimate	p.A.prot_Estimate	p.A.nProt_Estimate
hgt	0.859205733632614	0.0238724384575419	0.144503490136948	0.00148073028434642	0.0141317656200726
wgt	98.9428234201406	2.71948246216953	69.1816142883022	0.221916473012486	2.11856940494335
vil.id	-6.02451379136132	-0.168054407187466	-1.91414470908345	-0.00520794333267238	-0.0494468877742109

1.3.5 Test Program OLS Return All

```

vars.z <- NULL
t(suppressWarnings(suppressMessages(
  ff_reg_mbyn(list.vars.y, list.vars.x,
    vars.c, vars.z, df,
    return_all = TRUE,
    stats_ends = 'Estimate')))) %>%
kable() %>%
kable_styling_fc_wide()

```

vars_var.y	prot_Estimate	cal_Estimate	wealthIdx_Estimate	p.A.prot_Estimate	p.A.nProt_Estimate
hgt	0.859205733632614	0.0238724384575419	0.144503490136948	0.00148073028434642	0.0141317656200726
wgt	98.9428234201406	2.71948246216953	69.1816142883022	0.221916473012486	2.11856940494335
vil.id	-6.02451379136132	-0.168054407187466	-1.91414470908345	-0.00520794333267238	-0.0494468877742109

1.3.6 Test Program IV Return All

[illegible]

```

                                mean(df[[y]], na.rm=TRUE))))))
# lapplytwice

```

1.4.3 Nested Lapply All

```

df.reg.out.all <- bind_rows(
  lapply(list.vars.x,
    function(x) (
      bind_rows(
        lapply(list.vars.y, regf.iv,
          vars.x=x, vars.c=vars.c, vars.z=vars.z, df=df))
      )))

```

```

# df.reg.out.all %>%
#   kable() %>%
#   kable_styling_fc_wide()

```

1.4.4 Nested Lapply Select

```

df.reg.out.all <-
  (lapply(list.vars.x,
    function(x) (
      bind_rows(lapply(list.vars.y, regf.iv,
        vars.x=x, vars.c=vars.c, vars.z=vars.z, df=df)) %>%
        select(vars_var.y, starts_with(x)) %>%
        select(vars_var.y, ends_with('value'))
      ))) %>% reduce(full_join)

```

```

df.reg.out.all %>%
  kable() %>%
  kable_styling_fc_wide()

```

vars_var.y	prot_tvalue	cal_tvalue	wealthIdx_tvalue	p.A.prot_tvalue	p.A.nProt_tvalue
hgt	18.8756010031786	23.4421863484661	13.508899618216	3.83682180045518	32.5448257554855
wgt	16.3591125056062	17.3686031309332	14.1390521528113	1.36958319982295	12.0961557911467
vil.id	-14.9385580468907	-19.6150110809452	34.0972558327347	8.45943342783186	17.7801422421419