

Software code in R for performing instrumental variable analyses for Mendelian randomization investigations

maintained by Stephen Burgess

September 4, 2015

This is a non-traditional publication to provide software code for the Mendelian randomization community in a single document. It will be updated when necessary as new methods are developed. Hopefully, this will become a collaborative resource than can be authored by the community rather than a single-author manuscript. However, I (Stephen Burgess, sb452@cam.ac.uk) retain the prerogative to exert editorial control.

Currently, it only contains R code. If someone wants to write Stata code or code for any other software package, this can also be included.

Contents

1	Introduction and notation	3
2	Standard Mendelian randomization analysis with individual-level data	4
2.1	Ratio (Wald) method – single instrument (SNP or score)	4
2.2	Two-stage least squares method	6

1 Introduction and notation

Dimensions

N # sample size

K # number of genetic variants

Individual-level data

g # genetic variant(s), matrix dimension $N \times K$

x # risk factor/exposure, vector length N

y # outcome, vector length N

Summary data

bx # genetic associations with exposure, vector length K

by # genetic associations with outcome, vector length K

bxse # standard errors of genetic associations with exposure

byse # standard errors of genetic associations with outcome

2 Standard Mendelian randomization analysis with individual-level data

2.1 Ratio (Wald) method – single instrument (SNP or score)

```
## A. Ratio estimate (continuous outcome)

bx  = lm(x~g)$coef[2]
bxse = summary(lm(x~g))$coef[2,2]
by  = lm(y~g)$coef[2]
byse = summary(lm(y~g))$coef[2,2]

beta_ratio = lm(y~g)$coef[2]/lm(x~g)$coef[2]
beta_ratio = by/bx

## B. Asymptotic standard error (poor with weak instruments)

# 1. Delta method approximation (summarized data)

se_ratio_approx = summary(lm(y~g))$coef[2,2]/lm(x~g)$coef[2]
se_ratio_approx = byse/bx
# first order approximation
se_ratio_approx = sqrt(summary(lm(y~g))$coef[2,2]^2/lm(x~g)$coef[2]^2 +
  lm(y~g)$coef[2]^2*summary(lm(x~g))$coef[2,2]^2/lm(x~g)$coef[2]^4 -
  2*theta*lm(y~g)$coef[2]/lm(x~g)$coef[2]^3)
se_ratio_approx = sqrt(byse^2/bx^2 + by^2*bxse^2/bx^4 - 2*theta*by/bx^3)
# second order approximation
# theta is correlation between numerator
# and denominator in ratio estimate

# 2. Two-stage least squares method for standard error (individual-level
  data)

library(sem)
se_tsls = sqrt(tsls(y, cbind(x, rep(1,N)), cbind(g, rep(1,N)),
  w=rep(1,N))$V[1,1])

## C. Valid confidence intervals with weak instruments

# 1. Fieller's theorem
f0 = by^2 - qt(0.975, N)^2 * byse^2
f1 = bx^2 - qt(0.975, N)^2 * bxse^2
f2 = by*bx
D = f2^2 - f0*f1
```

```

if(D>0) {
  r1 = (f2-sqrt(D))/f1
  r2 = (f2+sqrt(D))/f1
if(f1>0) { cat("Confidence interval is a closed interval [a,b]: \n a=",
  r1, ", b=", r2, sep="") }
if(f1<0) { cat("Confidence interval is the union of two open intervals
  (-Inf, a], [b, +Inf): \n a=", r2, ", b=", r1, sep="") }
}
if(D<0|D==0) { cat("No finite confidence interval exists other than the
  entire real line.") }

```

2. Anderson--Rubin

```

library(ivpack)
ivmodel = ivreg(y~x|g, x=TRUE)
anderson.rubin.ci(ivmodel)
  # As with Fieller's theorem, interval may be a closed interval,
  # the union of two open intervals, or undefined

```

D. Binary outcome, logistic-linear model (assuming case--control data)

```

bx  = lm(x[y==0]~g[y==0])$coef[2]
bxse = summary(lm(x[y==0]~g[y==0]))$coef[2,2]
by  = glm(y~g, family=binomial)$coef[2]
byse = summary(lm(y~g))$coef[2,2]

```

Genetic associations with the risk factor are estimated in control participants only (see Bowden and Vansteelandt [2011]).

2.2 Two-stage least squares method

References

Bowden, J. and Vansteelandt, S. 2011. Mendelian randomisation analysis of case-control data using structural mean models. *Statistics in Medicine*, 30(6):678–694.