Simulation

Sam Lee

Data-Generating Process for Simulation

Bootstrap samples by cluster

```
DGP <- function(clusters, Y, cluster=F){
  G = length(clusters)
  if(cluster){
    return(Y[unlist(sapply(1:G, function(g){
      lag(cumsum(clusters), default=0)[g]+
        floor(clusters[g]*runif(clusters[g]))+1
    }))])
  } else {
    return(Y[floor(length(Y)*runif(length(Y)))+1])
  }
```

```
}


g.map <- women.cleaned %>% group_by(countryid) %>%
  summarize(
    total = n()
  ) %>% ungroup() %>% mutate(
    g.index = cumsum(total)
  )
```

Two-Stage Inference with Homoskedasticity Assumption

```
twosls <- function(X, Y, Z){
  n <- nrow(X)
  if(ncol(X) == ncol(Z)){
    #Just Identified
    beta.hat <- solve(t(Z)%*%X)%*%t(Z)%*%Y
  } else {
    #Assume homoskedasticity
    W <- solve(t(Z)%*%Z)
    beta.hat <- solve(t(X)%*%Z%*%W%*%t(Z)%*%X)%*%
```

```
      t(X)%*%Z%*%W%*%t(Z)%*%Y
  }
  sigma2 = (1/n)*sum((Y-X%*%beta.hat)^2)
  #Variance-Covariance Matrix
  V <- sigma2*solve((t(X)%*%Z)%*%(solve(t(Z)%*%Z))%*%
                    t(t(X)%*%Z))

  #95% C.I.
  beta.hat[1,]+c(-1,1)*qnorm(0.975)*sqrt(V[1,1])
}

#These two are the same via the Frisch-Waugh Theorem
#twosls(X, Z, Y)
twosls(X.tilde, Y.tilde, Z.tilde)
```

```
[1] -133.36956  -82.31544
```

Cluster Robust Variance Function

```
cluster.robust.iv <- function(X, Y, Z, clusters){ #Assumes data is sorted
  n <- nrow(X)
```

```
k <- ncol(X)

G <- length(clusters)

if(ncol(X) == ncol(Z)){

  #Just Identified

  beta.hat <- solve(t(Z)%*%X)%*%t(Z)%*%Y

}

u.hat <- X%*%beta.hat - Y


xtz <- t(X)%*%Z

Sigma <- matrix(0, k, k)

for(g in 1:G){

  cluster = (lag(cumsum(clusters), default = 0)[g]+1):cumsum(clusters)[g]

  M.z <- solve(diag(1, clusters[g])-

                Z[cluster,]%*%solve(t(Z)%*%Z)%*%t(Z[cluster,]))

  zeta.g <- t(Z[cluster,])%*%M.z%*%u.hat[cluster]

  Sigma <- Sigma+zeta.g%*%t(zeta.g)

}

V <- (G-1)/G*solve(n^-1*xtz%*%solve(n^-1*Sigma)%*%t(n^-1*xtz))


#95% C.I.
```

```
  beta.hat[1,]+c(-1,1)*qnorm(0.975)*sqrt(n^-1*V[1,1])
}
```

3. Cluster Robust Variance #3 (Jackknife)

$$\frac{G-1}{G}(\hat{X}'\hat{X})^{-1}(\sum_{g=1}^{G}\acute{s}_g\acute{s}_g')(\hat{X}'\hat{X})^{-1}$$

$$\acute{s}_g = \hat{X}_g' M_{gg}^{-1}\hat{u}_g$$

Proposed CRVE for Two-Stage Least Squares

$$\sqrt{n}(\hat{b}-\beta) \underset{d}{\to} N(0, (\mathbb{E}[X_i Z_i']\Sigma^{-1}\mathbb{E}[X_i Z_i]')^{-1})$$

$$\implies \hat{V} = (\frac{1}{n}X'Z\hat{\Sigma}^{-1}(\frac{1}{n}X'Z)')^{-1}; \quad \Sigma^{-1} = \mathbb{E}[Z_i Z_i' u_i^2] \implies \hat{\Sigma}^{-1} = (\frac{1}{n}\sum_{g=1}^{G} Z_g' Z_g \hat{u_g}^2)^{-1}$$

$$\frac{G-1}{G}(X'Z)^{-1}(\sum_{g=1}^{G}\hat{\zeta}_g\hat{\zeta}_g')(X'Z)^{-1}$$

$$\hat{\zeta}_g = Z_g'\overset{z}{M}_{gg}^{-1}\hat{u}_g; \quad \overset{z}{M}_{gg} = I_{N_g} - Z_g(Z'Z)^{-1}Z'$$

```
cluster.robust.iv(X.tilde, Y.tilde, Z.tilde, g.map$total)
```

```
[1] -166.85855  -48.82644
```

Simulated CRVE Intervals

```
simulate.CRVE <- function(n, bootstrap.cluster=F){
  n.sims <- n
  sims.ci <- matrix(0, nrow=n.sims, ncol=2*2)
  for(i in 1:n.sims){
    Y.new = DGP(g.map$total, Y.tilde, cluster=bootstrap.cluster) %>% as.matrix()
    for(j in 1:2){
      if(j == 2){
        sims.ci[i,(j+(j-1)):(j+(j-1)+1)] = twosls(X.tilde, Y.new, Z.tilde)
      } else {
        sims.ci[i,(j+(j-1)):(j+(j-1)+1)] =
          cluster.robust.iv(X.tilde, Y.new, Z.tilde, g.map$total)
      }
    }
  }
  colnames(sims.ci) <- c("JackKnife.025", "JackKnife.975",
                         "2SLS.025", "2SLS.975")
  return(sims.ci)
}
```

```r
coverage <- function(sims){
  coverages = c()
  k = 1
  for(i in 1:(ncol(sims)/2)){
    coverages[i] = sum(0 < sims[,k] & 0 < sims[,k+1] |
      (0 > sims[,k] & 0 > sims[,k+1]))/nrow(sims)
    k = k+2
  }
  names(coverages) = unlist(lapply(str_split(colnames(sims), "[.]"), first)) %>%
    unique()
  return(coverages)
}
```

```r
simulate.CRVE(10) %>% round(2) %>% knitr::kable()
```

JackKnife.025	JackKnife.975	2SLS.025	2SLS.975
-52.84	34.26	-28.84	10.26
-27.50	29.14	-15.54	17.18
-24.78	37.16	-13.10	25.48
-27.70	52.43	-11.39	36.12

JackKnife.025	JackKnife.975	2SLS.025	2SLS.975
-32.18	43.12	-15.83	26.77
-27.09	15.15	-28.52	16.58
-17.45	9.31	-20.79	12.64
-22.20	13.02	-24.25	15.06
-13.52	20.22	-14.28	20.98
-13.06	23.20	-16.02	26.15

Simulated CRVE Intervals with Clustered Bootstrap Sampling

```
sims.cluster = simulate.CRVE(1000, bootstrap.cluster = F)
```

```
sims.cluster %>%
  round(2) %>% head(10) %>% knitr::kable()
```

JackKnife.025	JackKnife.975	2SLS.025	2SLS.975
-13.22	25.85	-12.49	25.13
-42.16	52.77	-14.56	25.18
-36.76	16.55	-30.92	10.71
-34.08	2.39	-33.56	1.87
-21.61	3.07	-27.12	8.58
-28.63	36.88	-17.26	25.51
-18.35	22.95	-16.44	21.04
-15.58	28.74	-12.59	25.74
-77.46	28.70	-45.39	-3.38
-12.51	27.83	-11.68	27.00

```
coverage(sims.cluster)
```

```
JackKnife      2SLS
```

```
0.016      0.047
```