finc584_ps5

William Lee

09 November, 2022

Problem 1a

```
vrp <- readxl::read_xlsx(file.path(proj, "vrp.xlsx"))

x <- matrix(data = c(rep(1, nrow(vrp)), vrp$OV), ncol = 2)
y <- matrix(data = vrp$VRP)

beta <- solve(t(x) %*% x)%*% (t(x) %*% y)

resid <- y - x %*% beta
bread <- solve(t(x) %*% x)
meat <- t(x) %*% diag(as.vector(resid)^2)  %*% x

beta_vcov_robust <- sqrt(diag(bread %*% meat %*% bread))

knitr::kable(data.frame(name = c('(Intercept)', 'OV'),
                        coefs = beta, se = beta_vcov_robust,
                        t= beta/beta_vcov_robust), digits = 3)
```

name	coefs	se	t
(Intercept)	0.010	0.001	12.393
OV	-0.124	0.052	-2.382

Both variables are significant at the 95% level.

Problem 1b

$$cov(OV_{t-1}, OV_t) = cov(V_{t-1} + \eta_{t-1}, V_t + \eta_t) =$$
$$cov(V_{t-1}, V_t) + cov(V_{t-1}, \eta_t) + cov(V_t, \eta_{t-1}) + cov(\eta_{t-1}, \eta_t) = cov(V_{t-1}, V_t) > 0$$

$$cov(OV_{t-1}, \epsilon_t) = cov(V_{t-1} + \eta_{t-1}, \epsilon_t) = cov(V_{t-1}, \epsilon_t) + cov(\eta_{t-1}, \epsilon_t) = 0$$

```r
z <- matrix(c(rep(1, nrow(vrp) -1), vrp$OV[-nrow(vrp)]), ncol = 2)
x <- matrix(c(rep(1, nrow(vrp) -1), vrp$OV[-1]), ncol = 2)
y <- matrix(vrp$VRP[-1])

fst_stg_hat <- z %*% solve(t(z) %*% z)%*%(t(z) %*% x)
tsls <- solve(t(fst_stg_hat) %*% fst_stg_hat ) %*% (t(fst_stg_hat) %*% y)

resid_tsls <- y - x %*% tsls
bread <- solve(t(x) %*% z)
meat <- t(z) %*% diag(as.vector(resid_tsls)^2)  %*% z

tsls_vcov <- sqrt(diag(bread %*% meat %*% bread))

knitr::kable(data.frame(name = c('(Intercept)', 'OV'),
                        coefs = tsls, se_robust = tsls_vcov), digits = 3)
```

name	coefs	se_robust
(Intercept)	0.012	0.001
OV	-0.195	0.058

The coefficients are different but still fairly close to the original estimate. The new estimates are greater in magnitude which is typical for errors-in-variables problems.

Problem 1c

```r
vrp_gmm <- vrp %>%
  mutate(ov_lag1 = lag(OV), ov_lag5 = lag(OV, 5)) %>% drop_na()

l2_metric <- function(x){
  return(sum(colMeans(x)^2))
}

gmm_moments <- function(betas, data = vrp_gmm){
  e <- (data["VRP"] - betas[1] - betas[2]*data["OV"]) %>% as.matrix()
  z <- data[c("ov_lag1", "ov_lag5")] %>% as.matrix()
  #moment_conditions
  moment1 <- e
  moment2 <- e * z[,1]
  moment3 <- e * z[,2]
  return(cbind(moment1, moment2, moment3))
}

naive <- optim(par = c(0,0), fn = function(betas)
  l2_metric(gmm_moments(betas, data = vrp_gmm)))

w_optimal <- solve(t(gmm_moments(naive$par)) %*% gmm_moments(naive$par))

optimal_beta <- optim(par = c(0,0), fn = function(betas)
  t(colMeans(gmm_moments(betas))) %*% w_optimal %*% colMeans(gmm_moments(betas)))
```

```r
kable(data.frame(var = c('(intercept)', 'OV'),
                 naive = naive$par, optimal = optimal_beta$par), digits = 4)
```

var	naive	optimal
(intercept)	0.0136	0.0117
OV	-0.2747	-0.1528

The optimal GMM estimates are similar to what we found in the 2SLS estimates from part B.