Treatment Effects Practical Session #2: Gaussian MTEs

Frank DiTraglia

Oxford Econometrics Summer School

Introduction

This practical session is based on Heckman, Tobias, & Vytlacil (2001) and Angrist (2004). You may find it helpful to consult these papers. You will *definitely* find it helpful to consult the lecture slides: the paper and or the lecture slides. For a review of R basics, see Hands-On Programming with R for a review of basic R that you will need below. My notes on this book are available here.

Throughout this session, we will work with the following model:

$$
\begin{aligned}
Y &= (1-D)Y_0 + DY_1 \\
D &= 1\{\gamma_0 + \gamma_1 Z > V\} \\
Y_0 &= \mu_0 + U_0 \\
Y_1 &= \mu_1 + U_1
\end{aligned}
$$

$$
\begin{bmatrix} V \\ U_0 \\ U_1 \end{bmatrix} \sim \text{Normal}(0, \Sigma), \quad \Sigma = \begin{bmatrix} 1 & \sigma_0\rho_0 & \sigma_1\rho_1 \\ & \sigma_0^2 & \sigma_{01} \\ & & \sigma_1^2 \end{bmatrix}
$$

$$
Z \sim \text{Bernoulli}(q), \text{ indep. of } (V, U_0, U_1)
$$

In real life, we would observe only (Y, D, Z) but in some of the exercises below we will also work with the unobserved variables (Y_0, Y_1, V) directly.

Exercises

1. Write a function to simulate n iid draws of (Y_0, Y_1, V, Z, D) from the multivariate normal distribution described above, fixing $\mu_0 = \mu_1 = 0$, $\sigma_0 = \sigma_1 = 1$, $\sigma_{01} = 1/2$, and $q = 1/2$. Your function should take five arguments—`n`, `rho0`, `rho1`, `gamma0`, and `gamma1`—and return a data frame (or tibble) with named columns `Y0`, `Y1`, `V`, `Z`, `D`, and `Y`. In real life we can only observe (Y, D, Z) but fortunately for us, simulations aren't real life! In

this example there is no need to store (U_0, U_1) since they coincide with (Y_0, Y_1) when $\mu_0 = \mu_1 = 0$.

2. Use your function from the preceding part to make and store 1,000,000 simulation draws with $\rho_0 = 0.5$, $\rho_1 = 0.2$, $\gamma_0 = -1$ and $\gamma_1 = 1.5$. Use your simulation draws to calculate the LATE, TOT, and TUT at these parameter values.

3. In the previous exercise you used simulation to approximate the values of the LATE, TOT, and TUT at particular parameter values. Use the following formulas from the lecture slides to check your simulations against the *analytical* formulas that apply in the case where $q = 1/2$ and $\sigma_0 = \sigma_1 = 1$. Recall that we use the shorthand $\delta = (\rho_1 - \rho_0)$.

$$\text{LATE} = -\delta \left[\frac{\varphi(\gamma_0 + \gamma_1) - \varphi(\gamma_0)}{\Phi(\gamma_0 + \gamma_1) - \Phi(\gamma_0)}\right]$$

$$\text{TOT} = -\delta \left[\frac{\varphi(\gamma_0) + \varphi(\gamma_0 + \gamma_1)}{\Phi(\gamma_0) + \Phi(\gamma_0 + \gamma_1)}\right]$$

$$\text{TUT} = \delta \left[\frac{\varphi(\gamma_0) + \varphi(\gamma_0 + \gamma_1)}{\{1 - \Phi(\gamma_0)\} + \{1 - \Phi(\gamma_0 + \gamma_1)\}}\right]$$

4. Consult Section 2.1 of Angrist (2004). Angrist's notation is slightly different from ours: what he calls η we call V, what he calls ρ_{01} we call γ, and what he calls TT we call TOT. Other than that, everything is the same. Figure 1 of this paper plots the TOT and LATE over a range of values for $\mathbb{P}(D = 1|Z = 0)$. Use the formulas from the previous question to reproduce panel (a) of this figure. Add in the TUT effect for good measure if you're feeling ambitious! You'll need to read a few short extracts of the paper to determine how to set the parameters γ_0 and γ_1 when making your plot.

5. In this problem you will apply the Heckman two-step estimator to the simulated values of (Y, D, Z) from question 2 above to estimate the parameters μ_1, μ_1, $\delta_0 \equiv \sigma_0\rho_0$ and $\delta_1 \equiv \sigma_1\rho_1$. In the simulation we know that $\mu_1 = \mu_0 = 0$, $\sigma_0 = \sigma_1 = 1$, $\rho_0 = 0.5$ and $\rho_1 = 0.2$, so you'll check the estimator against these values. Follow these steps:
   (a) Use the simulated values of (D, Z) to estimate (γ_0, γ_1). Call your estimates $(\widehat{\gamma}_0, \widehat{\gamma}_0)$. Check that your estimates match the true values that you used to generate the data: $\gamma_0 = -1$ and $\gamma_1 = 1.5$.
   (b) Define the shorthand $\widehat{\lambda}(z) = \varphi(\widehat{\gamma}_0 + \widehat{\gamma}_1 z)/[1 - \Phi(\widehat{\gamma}_0 + \widehat{\gamma}_1 z)]$. Add a column called `lambda` to the dataframe containing your simulated values of (Y, D, Z) that evaluates the function $\widehat{\lambda}(\cdot)$ at the observed values of Z.

   (c) Define the shorthand $\widehat{\kappa}(z) = -\varphi(\widehat{\gamma}_0 + \widehat{\gamma}_1 z)/\Phi(\widehat{\gamma}_0 + \widehat{\gamma}_1 z)$. Add a column called `kappa` to the dataframe containing your simulated values of (Y, D, Z) that evaluates the function $\widehat{\lambda}(\cdot)$ and at the observed values of Z.

(d) For the subset of observations with $D = 0$, run a regression of Y on `lambda` and a constant. The intercept should be approximately equal to μ_0 and the slope approximately equal to $\delta_0 = \sigma_0 \rho_0$.

(e) For the subset of observations with $D = 1$, run a regression of Y on `kappa` and a constant. The intercept should be approximately equal to μ_1 and the slope approximately equal to $\delta_1 = \sigma_1 \rho_1$.

Solutions

Solution to Exercise 1

```r
library(mvtnorm)
library(dplyr)
draw_sims <- function(n, rho0, rho1, gamma0, gamma1) {

  S <- matrix(c(1, rho0, rho1,
                rho0, 1, 0.5,
                rho1, 0.5, 1), 3, 3, byrow = TRUE)

  sims <- rmvnorm(n, sigma = S)
  V <- sims[,1]
  Y0 <- sims[,2]
  Y1 <- sims[,3]
  Z <- rbinom(n, size = 1, prob = 1/2)
  sims <- tibble(Y0 = Y0, Y1 = Y1, V = V, Z = Z)
  sims %>%
    mutate(D = 1 * (gamma0 + gamma1 * Z > V),
           Y = (1 - D) * Y0 + D * Y1)
}
```

Solution to Exercise 2

```r
set.seed(1983)
sims <- draw_sims(n = 1e6, rho0 = 0.5, rho1 = 0.2, gamma0 = -1, gamma1 = 1.5)
# Two different ways to calculate the LATE. Using the observables:
LATE_obs <- sims %>%
  summarize(cov(Z, Y) / cov(Z, D)) %>%
  pull()
# Or the unobservables:
LATE <- sims %>%
  mutate(complier = (V >= -1) & (V < -1 + 1.5)) %>%
  filter(complier) %>%
  summarize(mean(Y1 - Y0)) %>%
```

```r
  pull()

# They're not identical because the first is only an estimate given that
# we can't observe the counterfactual outcomes in real life!
c(LATE1 = LATE_obs, LATE2 = LATE) %>% round(3)
```

```
## LATE1 LATE2
## 0.064 0.060
```

```r
TOT <- sims %>%
  filter(D == 1) %>%
  summarize(mean(Y1 - Y0)) %>%
  pull()

TUT <- sims %>%
  filter(D == 0) %>%
  summarize(mean(Y1 - Y0)) %>%
  pull()

simulation <- c(LATE = LATE, TOT = TOT, TUT = TUT)
simulation %>% round(3)
```

```
##   LATE    TOT    TUT
##  0.060  0.207 -0.154
```

Solution to Exercise 3

```r
LATE <- function(delta, gamma0, gamma1) {
  -delta * (dnorm(gamma0 + gamma1) - dnorm(gamma0)) /
    (pnorm(gamma0 + gamma1) - pnorm(gamma0))
}
TOT <- function(delta, gamma0, gamma1) {
  -delta  * (dnorm(gamma0) + dnorm(gamma0 + gamma1)) /
    (pnorm(gamma0) + pnorm(gamma0 + gamma1))
}
TUT <- function(delta, gamma0, gamma1) {
  delta * (dnorm(gamma0) + dnorm(gamma0 + gamma1)) /
    (1 - pnorm(gamma0) + 1 - pnorm(gamma0 + gamma1))
}

analytical <- c(LATE = LATE(delta = -0.3, gamma0 = -1, gamma1 = 1.5),
  TOT =  TOT(delta = -0.3, gamma0 = -1, gamma1 = 1.5),
  TUT = TUT(delta = -0.3, gamma0 = -1, gamma1 = 1.5))

simulation %>% round(3)
```

```
##   LATE    TOT    TUT
##  0.060  0.207 -0.154
```

```
analytical %>% round(3)
```

```
##   LATE    TOT    TUT
##  0.062  0.210 -0.155
```

Solution to Exercise 4

Figure 1 of Angrist (2004) computes the LATE and TOT as $\mathbb{P}(D=1|Z=0) = \Phi(\gamma_0)$ varies while holding the *first stage* fixed at 0.07. In other words:

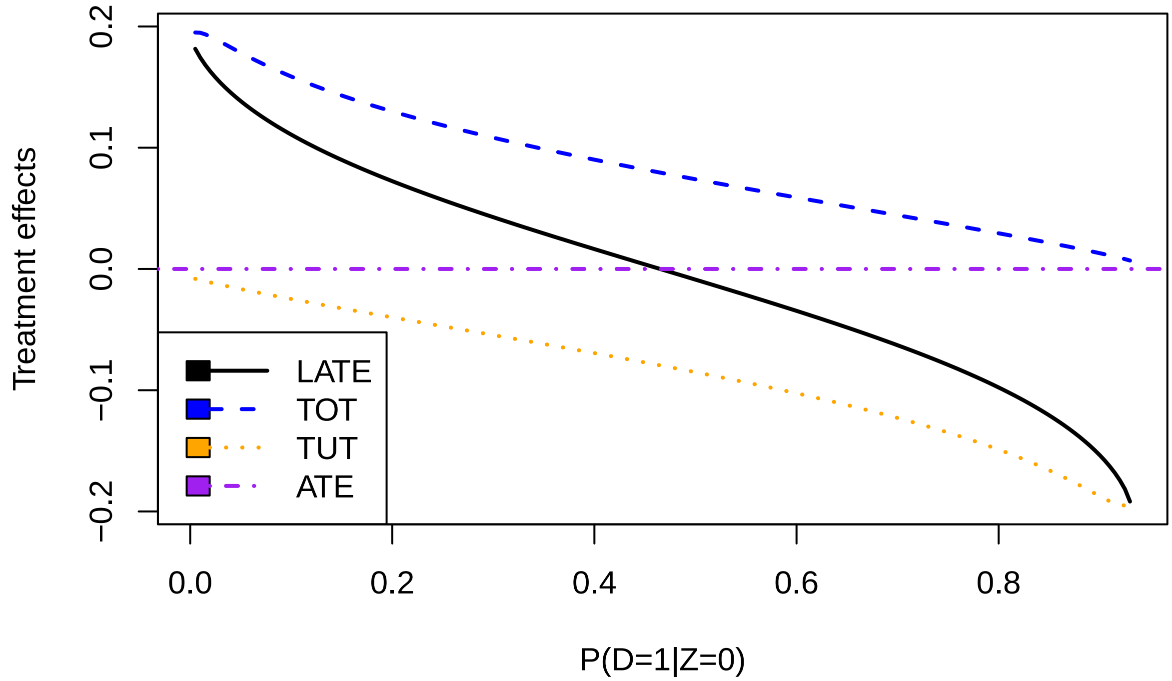
$$
\begin{aligned}
\mathbb{E}[D|Z=1] - \mathbb{E}[D|Z=0] &= \mathbb{P}(D=1|Z=1) - \mathbb{P}(D=1|Z=0) \\
&= \Phi(\gamma_0 + \gamma_1) - \Phi(\gamma_0) = 0.07
\end{aligned}
$$

Now define the shorthand $p_0 \equiv \mathbb{P}(D=1|Z=0) = \Phi(\gamma_0)$. Solving for γ_1 and γ_0

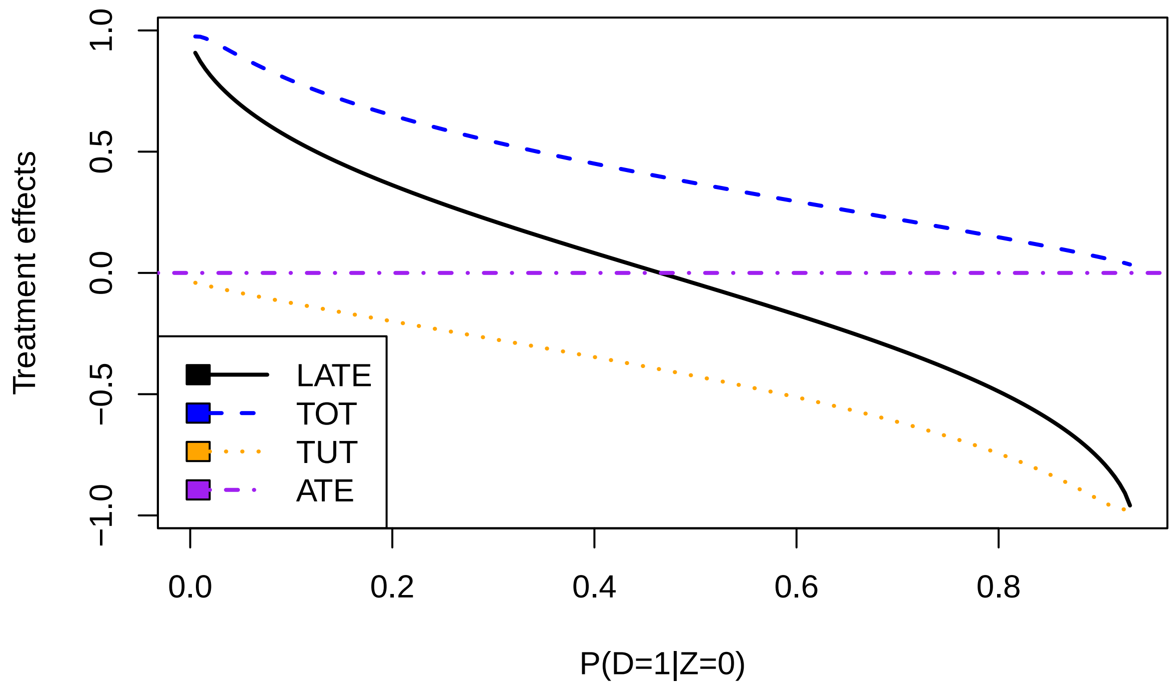
$$
\begin{aligned}
\gamma_1 &= \Phi^{-1}\left(0.07 + p_0\right) - \Phi^{-1}(p_0) \\
\gamma_0 &= \Phi^{-1}(p_0)
\end{aligned}
$$

Now we have a way to calculate the required values of (γ_0, γ_1) for any p_0 to ensure a first-stage of 0.07. We can combine this with our functions from above as follows.

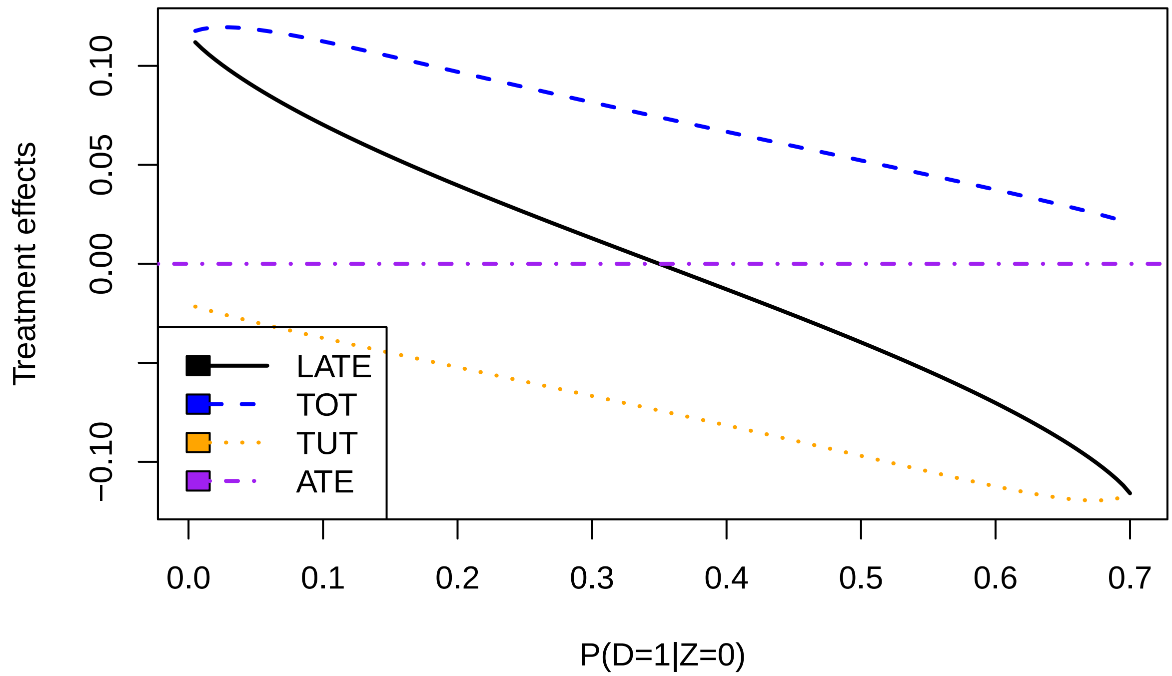
```
make_plot <- function(delta, first_stage = 0.07) {
  p0_seq <- seq(from = 0.005, to = 1 - first_stage, by = 0.005)
  g0_seq <- qnorm(p0_seq)
  g1_seq <- qnorm(first_stage + p0_seq) - g0_seq
  LATE_seq <- LATE(delta, g0_seq, g1_seq)
  TUT_seq <- TUT(delta, g0_seq, g1_seq)
  TOT_seq <- TOT(delta, g0_seq, g1_seq)
  mywd <- 2
  mycols <- c('black', 'blue', 'orange', 'purple')
  matplot(p0_seq, cbind(LATE_seq, TOT_seq, TUT_seq),
          xlab = 'P(D=1|Z=0)', ylab = 'Treatment effects', type = 'l',
          lwd = mywd, col = mycols[1:3])
  abline(h = 0, lty = 4, lwd = mywd, col = mycols[-c(1:3)])
  legend('bottomleft', legend = c('LATE', 'TOT', 'TUT', 'ATE'),
         lty = 1:4, fill = mycols, col = mycols, lwd = mywd)
}
make_plot(-0.1, 0.07) # Fig 1a
```



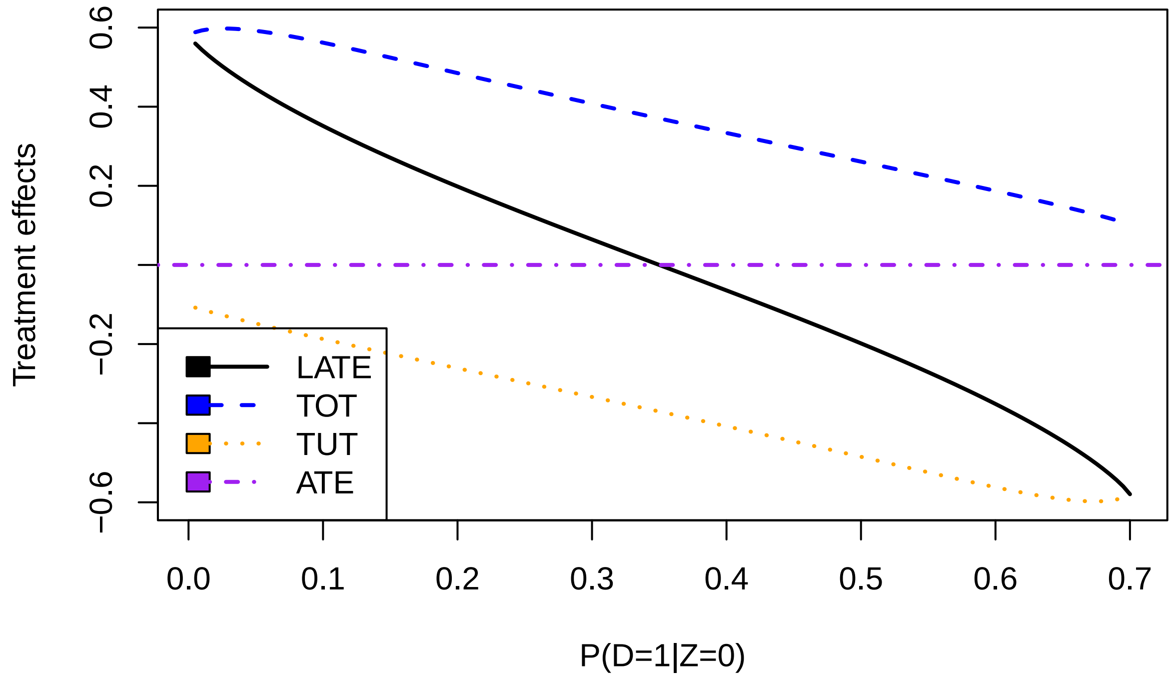

```r
make_plot(-0.5, 0.07) # Fig 1b
```




```r
make_plot(-0.1, 0.3) # Fig 2a
```




```
make_plot(-0.5, 0.3) # Fig 2b
```



Solution to Exercise 5

```
# Only keep the observables since I'm fastidious!
dat <- sims %>%
  select(Y, D, Z)
```

```r
# Part (a) - estimate gamma0 and gamma1
p0_est <- with(dat, mean(D[Z == 0]))
p1_est <- with(dat, mean(D[Z == 1]))
gamma0_est <- qnorm(p0_est)
gamma1_est <- qnorm(p1_est) - gamma0_est

c(g0_est = gamma0_est, g0_true = -1, g1_est = gamma1_est, g1_true = 1.5) %>%
  round(3)
```

```
##  g0_est g0_true  g1_est g1_true
##  -0.997  -1.000   1.494   1.500
```

```r
# Parts (a)-(c): calculate lambda and kappa
dat <- dat %>% mutate(gZ = gamma0_est + gamma1_est * Z,
                      phi = dnorm(gZ),
                      Phi = pnorm(gZ),
                      kappa = -phi / Phi,
                      lambda = phi / (1 - Phi)) %>%
  select(Y, D, Z, lambda, kappa)

# Part (d)
reg0 <- lm(Y ~ lambda, data = dat, subset = (D == 0))
coef(reg0)
```

```
##  (Intercept)       lambda
## -0.001969749  0.502145306
```

```r
# Part (e)
reg1 <- lm(Y ~ kappa, data = dat, subset = (D == 1))
coef(reg1)
```

```
##   (Intercept)         kappa
## -0.0005315038  0.2012802222
```

```r
# Display everything in one place
coefs0 <- coef(reg0)
coefs1 <- coef(reg1)
mu0_est <- coefs0['(Intercept)']
delta0_est <- coefs0['lambda']
mu1_est <- coefs0['(Intercept)']
delta1_est <- coefs1['kappa']
estimates <- c(mu0_est, mu1_est, delta0_est, delta1_est, gamma0_est, gamma1_est)
names(estimates) <- c('mu0', 'mu1', 'delta0', 'delta1', 'gamma0', 'gamma1')
estimates %>% round(3)
```

```
##    mu0    mu1 delta0 delta1 gamma0 gamma1
## -0.002 -0.002  0.502  0.201 -0.997  1.494
```