# Microeconometrics Task - Problem Set 3

Ricardo Semião e Castro

2024-06-06

## 1 Setup

```r
library(tidyverse)
library(furrr)

library(knitr)
library(kableExtra)

set.seed(0306152529)
plan(multisession, workers = 7)
theme_set(theme_bw())
```

## 2 Question 1

Using the Inverse Probability Weighting approach, find an estimand that point-identifies the Average Treatment Effect on the Untreated:

$$ATU = E[Y(1) - Y(0)|D = 0]$$

We need to find expressions for $E[Y(1)|D = 0]$ and $E[Y(0)|D = 0]$, then, we can subtract them to find the ATU.

Lets try to create a guess based on the relation of the ATT and ATE (section 2.3, statement 4. of the lecture notes). Recall the guess at (3), that is: $E[YD]/E[D]$. When dealing with the untreated, we want to consider the $1 - D$ fraction of the individuals, so lets try to alter the guess and see if we find $E[Y(0)|D = 0]$:

$$\frac{E[Y(1-D)]}{E[1-D]} = \frac{E[Y|D=0]P(D=0)}{1-E[D]} = E[Y(0)|D=0]$$

Now, lets use (4), again, substituting $(1 - D)$:

$$\frac{1}{E[1-D]}E\left[\frac{P(X)(1-(1-D))Y}{1-P(X)}\right] = \frac{1}{1-E[D]}E\left[\frac{P(X)}{1-P(X)}DY\right] =$$
$$\frac{1}{1-E[D]}E\left[\frac{P(X)}{1-P(X)}E[DY|X]\right] = \frac{E[\frac{P(X)}{1-P(X)}E[DY|X]]}{1-E[D]} =$$
$$\frac{E[E[1-D|X]E[Y(1)|X]]}{1-E[D]} = \frac{E[E[(1-D)Y(1)|X]]}{1-E[D]} =$$
$$\frac{E[(1-D)Y(1)]}{1-E[D]} = \frac{E[Y(1)|D=0]P(D=0)}{1-E[D]} = E[Y(1)|D=0]$$

Where we used the LEI. Thus, the relation we want is:

$$\frac{1}{E[1-D]}\left[E\left[\frac{P(X)(1-(1-D))Y}{1-P(X)}\right]-E[Y(1-D)]\right]$$

# 3 Question 2

First, lets define the DGPs, saving their relevant functions as a list element:

```r
dgps <- list(
  dgp1 = list(
    d = \(x, u) x[,1]/2 + x[,1]^2/2 >= u,
    y0 = \(x, u, e) 1 + x[,1] + e,
    y1 = \(x, y0) y0 + 2 + x[,1],
    ate = \(x) 2 + mean(x[,1])
  ),
  dgp2 = list(
    d = \(x, u) x[,1]/2 + x[,2]/2 >= u,
    y0 = \(x, u, e) 1 + rowSums(x) + e,
    y1 = \(x, y0) y0 + 2 + rowSums(x),
    ate = \(x) 2 + sum(colMeans(x))
  ),
  dgp3 = list(
    d = \(x, u) x[,1]/2 + x[,2]^2/2 >= u,
    y0 = \(x, u, e) 1 + rowSums(x) + e,
    y1 = \(x, y0) y0 + 2 + rowSums(x),
    ate = \(x) 2 + sum(colMeans(x))
  )
)
```

Recall that the true ATE of each DGP is $Y(1)-Y(0)$.

Lets define functions to generate the actual data:

```r
get_y <- function(x, u, e, d, y0, y1) {
  y0 <- y0(x, u, e)
  d * y1(x, y0) + (1 - d) * y0
}

get_x <- function(n) {
  mvtnorm::rmvnorm(n, sigma = matrix(c(1, 0.9, 0.9, 1), 2, 2)) %>%
    apply(2, pnorm)
}
```

And a function to get the ATE of each of the four given methods:

```r
get_ates <- function(x, y, d) {
  mod_cef <- list(
    control = lm(y ~ I(x[,1] - mean(x[,1])), subset = !d),
    treat = lm(y ~ I(x[,1] - mean(x[,1])), subset = d)
  )

  mod_ipw <- glm(d ~ x[,1] + x[,2], family = binomial)
  p <- predict(mod_ipw, type = "response")

  mod_dr <- list(
    control = lm(y ~ I(x[,1] - mean(x[,1])) + I(x[,2] - mean(x[,2])),
      subset = !d,
      weights = 1 / p
    ),
    treat = lm(y ~ I(x[,1] - mean(x[,1])) + I(x[,2] - mean(x[,2])),
      subset = d,
      weights = 1 / p
    )
  )

  c(
    coef(mod_cef$treat)[1] - coef(mod_cef$control)[1],
    (sum((d * y) / p) / sum(d / p)) - (sum((1 - d) * y / (1 - p)) / sum((1 - d) / (1 - p))),
    coef(mod_dr$treat)[1] - coef(mod_dr$control)[1],
    mean(y[d]) - mean(y[!d])
  ) %>%
    set_names(c("CEF", "IPW", "DR", "Naive"))
}
```

Now, we can run the simulation. Using `evalq`, we access the functions of each dgp, and get the bias with `get_ates(x, y, d) - ate(x)`. The purrr functions compile the results in a data frame.

```
m <- 1000
n <- 10000

biases <- map(dgps, function(dgp) {
  future_map_dfr(seq_len(m),
    function(iter) {
      evalq(envir = dgp, {
        u <- runif(n)
        e <- runif(n)
        x <- get_x(n)
        d <- d(x, u)
        y <- get_y(x, u, e, d, y0, y1)
        ate <- ate(x)
        (get_ates(x, y, d) - ate) / ate
      })
    },
    .options = furrr_options(seed = TRUE)
  )
})

data_bias <- map(biases, colMeans) %>%
  bind_rows()
```

| DGP | CEF | IPW | DR | Naive |
|---|---:|---:|---:|---:|
| 1 | -0.0006 | 1.6295 | 0.0038 | 21.72 |
| 2 | 2.4378 | 2.4372 | -0.0054 | 31.54 |
| 3 | 2.6214 | 2.4546 | 0.0027 | 34.23 |

We have a few problems that the estimators are facing:

- The variables $X_1$ and $X_2$ (except at DGP1) are relevant, and must be included (in some way).
  - Relevant as the affect the outcome, and are related to the treatment.

- The relation between $X$s and the treatment is different at each DGP. The estimators must be, in some sense, flexible.

The naive estimator is the worst, as it does not account for the missing variables. It present the highest bias. The bias of DGP1 is lower than at the other two, as there are, in some sense, less omitted variables.

The CEF estimator is relatively unbiased on DGP1, as it accounts for the only relevant variable. On the other hand, it presents a bias on the other two DGPs, as it does not account for $X_2$. Still, we can see that it is better than the naive, as it does control for part of the effect.

The IPW estimator is similar to the CEF, but it wrongly includes $X_2$ at DGP1, causing a bias.

The DR brings the needed flexibility. As it is consistent if either the conditional expectation function or the propensity score are correctly specified, it is unbiased in DGP1, in all DGPs.

# 4 Question 3

Lets load the data from the GH repo of the quantreg package. Then, map the quantiles to create different quantile regressions, and save the coefficients in a data frame.

```r
load(url("https://raw.githubusercontent.com/cran/quantreg/master/data/engel.rda"))

quantiles <- seq(0.05, 0.95, 0.025)

data_qr <- map_dfr(quantiles, function(q) {
  mod <- quantreg::rq(foodexp ~ income, engel, tau = q)
  coef(mod)[1:2]
})
```

| | mean | sd | median | min | max | se |
|---|---|---|---|---|---|---|
| (Intercept) | 86.826 | 19.580 | 85.019 | 50.721 | 124.880 | 3.219 |
| income | 0.548 | 0.099 | 0.556 | 0.343 | 0.709 | 0.016 |

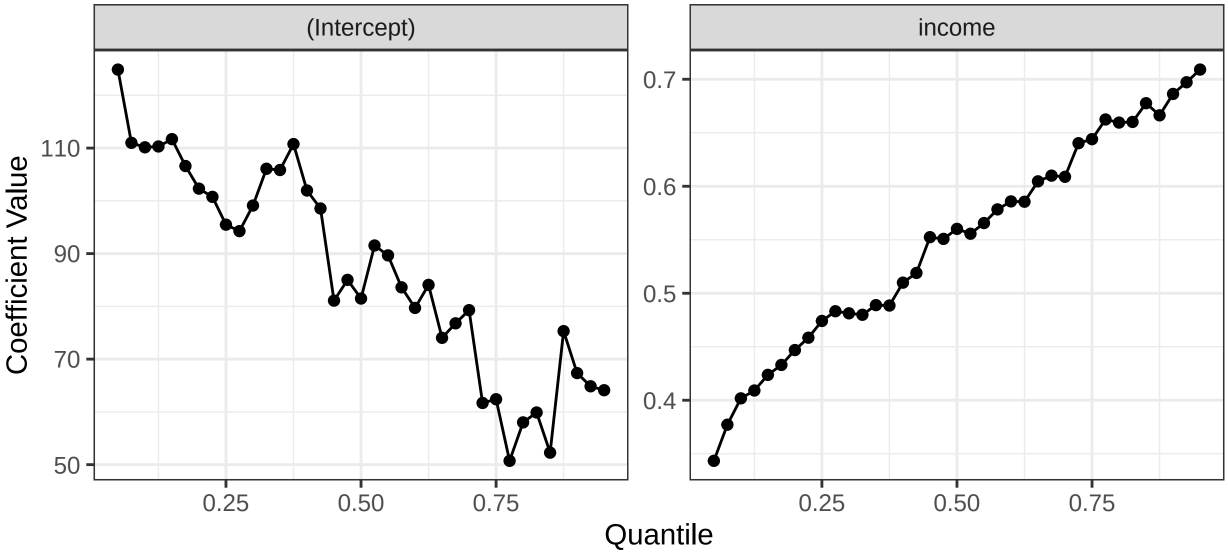
For the median regression, we can see the results:

```r
quantreg::rq(foodexp ~ income, engel, tau = 0.5)
```

| | (1) |
|---|---|
| (Intercept) | 81.482 |
| | (19.251) |
| income | 0.560 |
| | (0.028) |
| Num.Obs. | 235 |
| R2 | 0.810 |
| RMSE | 120.33 |

We can see that an increase in income is associated with a statistically significant increase of 0,56 units of food consumption.

Now, we can plot the full results:



We can see that the intercept, representing the standard expenditure, independent of income, is bigger for poorer people, and smaller for wealthier. The opposite is seen for the slopes. This result is expected, given the basic need for food at lower incomes.