

PROBLEM SET 4

MGMT 737

You should have gotten this homework assignment from the Github classroom environment (<https://classroom.github.com/classrooms/192971645-yale-mgmt-737-spring-2025-classroom>). In submitting your problem set, you should have two files in your Github repository:

1. `homework4-code.R`, which contains your code,
2. `homework4-writeup.pdf`, which contains your writeup

You may have other files / folders if necessary when there are images or auxiliary files. My very strong preference is for you to write this up in R. If this is not possible, you can use Python. Please let me know if you're planning on this. (This is not a coding preference, but mainly a grading issue.)

1. **Quantile Regression.** This analysis will use the dataset from Problem Set 1, `lalde_nsw.csv` (which I will refer to as NSW), as well as the dataset from Problem Set 2, `lalde_psid.csv` (which I will call PSID).

- (a) We will begin by defining an estimation approach for doing quantile regression that doesn't require linear programming. This approach comes from Gary Chamberlain (in Chamberlain (1994), and discussed in Angrist et al. (2006)).

Let X be a (discrete) right hand side variable with J discrete values. For each j value of $X = x_j$, calculate $\hat{\pi}_\tau(x) = Q_\tau(Y|X_j)$, which is the τ percentile of the outcome variable, conditional on the value of X , and \hat{p}_j , which is the empirical probability of $X = x_j$. Do so using the PSID dataset for $X = \text{education}$, for $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$, using `re78` as the outcome variable.

→ Label the calculated value for $\tau = 0.5$ value as `tau_med_1a` in your code.

- (b) Given these inputs, the quantile regression slope estimates is just

$$\hat{\beta}_\tau = \arg \min_b \sum_j (\hat{\pi}_\tau(x_j) - x_j b)^2 \hat{p}_j.$$

This is a simple (weighted) linear regression (or minimum distance problem), with the diagonal weight matrix $\hat{W} = \text{diag}(\hat{p}_1, \dots, \hat{p}_J)$. Estimate $\hat{\beta}_\tau$ for the education example above.

→ Label the calculated value for $\tau = 0.5$ value as `tau_med_beta_1b` in your code.

- (c) Our variance estimator is the sum of two terms (coming from uncertainty in the QCF, and the estimation of the slope conditional on those terms), V and D :

$$\begin{aligned} V &= (\mathbf{x}' \hat{W} \mathbf{x})^{-1} \mathbf{x}' \hat{W} \Sigma \hat{W} \mathbf{x} (\mathbf{x}' \hat{W} \mathbf{x})^{-1}, \\ \Sigma &= \text{diag}(\sigma_{\tau,1}^2/p_1, \dots, \sigma_{\tau,J}^2/p_J) \\ D &= (\mathbf{x}' \hat{W} \mathbf{x})^{-1} \mathbf{x}' \hat{W} \Delta \hat{W} \mathbf{x} (\mathbf{x}' \hat{W} \mathbf{x})^{-1}, \\ \Delta &= \text{diag}((\pi_{\tau,1} - x_1 \beta_\tau)^2/p_1, \dots, (\pi_{\tau,J} - x_J \beta_\tau)^2/p_J). \end{aligned}$$

Everything here should be straight forward to estimate, except for $\sigma_{\tau,j}^2$. To do this, define the following order statistics:

$$\begin{aligned} b_j &= \max \left\{ 1, \text{round} \left(\tau N_j - z_{1-\alpha/2} \sqrt{\tau(1-\tau)N_j} \right) \right\} \\ t_j &= \min \left\{ N_j, \text{round} \left(\tau N_j + z_{1-\alpha/2} \sqrt{\tau(1-\tau)N_j} \right) \right\}, \end{aligned}$$

where $\text{round}(a)$ rounds to the closest integer, and $z_{1-\alpha/2} = 1.96$, typically, and N_j is the number of observations in the j th bin of X . Then,

$$\hat{\sigma}_{\tau,j}^2 = N_j \left(\frac{y_j(t_j) - y_j(b_j)}{2z_{1-\alpha}} \right)^2. \quad (1)$$

Report the standard error on your estimates, which is calculated as $\sqrt{(V + D)/N}$

→ Label the calculated value for $\tau = 0.5$ value as `se_med.1c` in your code.

- (d) Finally, using the NSW dataset, calculate the $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ treatment effects, and their standard errors.

→ Label the calculated values for $\tau = 0.5$ value as `tau_med.1d_nsw` and `se_med.1d_nsw` in your code.

2. **Unconditional Quantile Effects** This analysis will walk through an alternative way of approaching quantile effects, using results from Firpo, Fortin and Lemieux (2009). Instead of focusing on conditional quantile functions, we will consider the *unconditional* (or marginal) distribution of the outcome, Y , and how that changes when you shift an exogeneous covariate.

We define the *unconditional* (marginal) distribution of Y :

$$F_Y(y) = \int F_{Y|X}(y|X = x) \cdot dF_X(x),$$

and consider small changes to F_X , holding fixed $F_{Y|X}$. Intuitively, we consider the effect of changing F_X to G_X using a simple additive shift, and consider the effect that has on the marginal distribution of F . This will give us the *unconditional quantile partial effect* (UQPE), analogous to the unconditional average partial effect (UAPE), $E(dE(Y|X)/dx)$. The estimation approach for this uses influence functions. I will walk you through the estimation approach, using our example above and the approach they call RIF-OLS.

- (a) Our goal is to estimate $UQPE(\tau)$. We need to estimate three pieces. Let Y_i be income (`re78`) and X_i be years of education using the PSID dataset.
- (b) First, we estimate the τ quantile of Y_i (unconditionally), which we denote as \hat{q}_τ
- (c) Second, we need to estimate a constant $c_{1,\tau} = 1/f_Y(q_\tau)$. To do so, we need an estimate of the density of \hat{q}_τ . Calculate this directly using the following estimator:

$$\hat{f}_Y(\hat{q}_\tau) = \frac{1}{Nb} \sum_{i=1}^N K\left(\frac{Y_i - \hat{q}_\tau}{b}\right)$$

Assume $b = 2534.263$ (this bandwidth depending on the outcome), and let K be the Gaussian kernel,

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$$

- (d) Finally, we need to estimate $E(dPr(Y > q_\tau|X)/dX)$. Firpo, Fortin and Lemieux (2009) show that you can do this as follows. Calculate $R\hat{I}F(Y_i, \hat{q}_\tau) = \hat{c}_{1,\tau} \cdot 1(Y_i > \hat{q}_\tau) + \hat{c}_{2,\tau}$, where $\hat{c}_{2,\tau} = \hat{q}_\tau - \hat{c}_{1,\tau}(1 - \tau)$. Regress $R\hat{I}F(Y_i, \hat{q}_\tau)$ against X_i , and extract the coefficient on X_i . This is the estimate for the *UQPE* at the τ quantile.
- (e) Implement the above estimation procedure for $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$. Compare these results to your results in Part I.

→ Label the calculated values for $\tau = 0.5$ value as `tau_med.2.uncond` and `se_med.2.uncond` in your code.

3. **Debiased Lasso Regression.** We'll now consider estimation of Double/De-biased Lasso. Using the treated units in the NSW dataset for the treatment group, and PSID data for the control group, let Y denote `re78`, D denote the treatment indicator, and X denote a matrix of indicator variables for all values of `education`, `hispanic`, `black`, an indicator variable if `re74` is zero, an indicator if `re75` is zero, and a linear, quadratic, and cubic term for `age`, `re74` and `re75`. This should be $K = 30$ terms.
 - (a) Using `rlasso` (available in R in the package "hdm" and in Stata in Lassopack), implement the DML codebook to identify the subset of X to control for in the regression of Y on D and controls. Report the coefficient on D .
 - Label the calculated coefficient on D as `tau_3` in your code.
 - (b) Now reimplement this using NSW dataset. What variables get selected in X ?
4. **Puffer Lasso** Finally, we consider a new dataset, `health_ins`. This dataset has commuting zone level measures of health insurance (`has_insurance`), and additional explanatory variables (the remaining variables in the dataset, except for `czone`). We will now consider how our estimates change when we use Puffer vs. not in Lasso. Denote $Y = \text{has_insurance}$ and let the matrix X be the remaining variables in the dataset (except for `czone`).
 - (a) Estimate the OLS regression of Y on X and report the coefficient on `cs_frac_hisp`
 - Label the calculated coefficient as `beta_4a` in your code.
 - (b) Now estimate Lasso (using `rLasso` as before), and report the coefficient on `cs_frac_hisp`, and the number of non-zero coefficients
 - Label the calculated coefficient as `beta_4b` in your code.
 - (c) Implement the Puffer transformation of Y and X using the singular value decomposition of X (you should use a built-in package to get these values) to construct a pre-multiplying matrix F . Report the coefficient on `cs_frac_hisp` and the number of non-zero coefficients of the regression of FY on FX .
 - Label the calculated coefficient as `beta_4c` in your code.
 - (d) Take the non-zero coefficients from the Puffer lasso regression, and rerun OLS using those selected coefficients (using the non-puffered data). Report the coefficient on `cs_frac_hisp`, and the standard error, and compare to the original OLS regression and standard error.
 - Label the calculated coefficient as `beta_4d` in your code.