Homework Three

Andie Creel

2023-02-23

Quantile Regression

This analysis will use the dataset from Problem Set 1, lalonde_nsw.csv (which I will refer to as NSW), as well as the dataset from Problem Set 2, lalonde_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 $\hat{\pi}_t(C) = Q_\tau(Y|X_j)$, 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 $X =$ education, for $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$

```
# ---------------------------------------------------------------
# read in data
# ---------------------------------------------------------------

myData <- vroom("lalonde_psid.csv") %>%
  mutate(education = as.factor(education)) %>%
  mutate(Y = re78) %>%
  select(Y, education)

# ---------------------------------------------------------------
# p_j, probability of X = x_j
# ---------------------------------------------------------------
p <- summary(myData$education)/length(myData$education)

# p["0"]

# ---------------------------------------------------------------
# pi_j, \tau percentile of Y, conditional on the value of X
# ---------------------------------------------------------------

myPi <- as.data.frame(matrix(nrow = 17, ncol = 9))
colnames(myPi) <- c(".1", ".2", ".3", ".4", ".5", ".6", ".7", ".8", ".9")
rownames(myPi) <- levels(myData$education)

for (i in 1:length(levels(myData$education))) {
  # condition on education level
  myWorking<- myData %>%
    filter(education == levels(myData$education)[i])
```

```r
  # calculate percentiles of outcome
  myPi[i,] <- round(unname(quantile(myWorking$Y, probs = c(.1, .2, .3, .4, .5, .6, .7, .8, .9))),0)
}


# ------------------------------------------------------------
# print tables
# ------------------------------------------------------------
stargazer::stargazer(myPi, type = 'latex', summary = FALSE, title = "pi")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:26

Table 1: pi

	.1	.2	.3	.4	.5	.6	.7	.8	.9
0	2,305	4,315	6,325	8,334	10,344	12,383	14,423	16,462	18,501
2	1,478	2,955	3,694	4,433	6,650	8,866	13,481	18,096	23,087
3	0	0	2,180	9,497	11,083	14,038	17,733	19,949	25,121
4	0	0	2,379	5,911	8,127	10,935	15,402	20,843	22,413
5	0	6,100	9,993	12,107	14,629	14,748	17,437	19,612	22,461
6	0	0	5,615	9,605	11,822	13,595	16,060	20,688	28,077
7	0	2,955	8,877	11,674	13,300	14,777	16,255	19,343	25,121
8	0	1,200	7,102	9,156	13,300	16,994	20,583	25,565	30,316
9	0	7,891	9,058	11,836	13,743	16,839	19,320	24,349	33,101
10	0	7,305	10,060	13,300	17,733	20,777	23,644	25,787	31,032
11	0	3,103	8,314	11,538	14,149	16,255	20,600	26,599	33,142
12	59	11,526	14,777	17,733	20,688	23,644	26,599	29,555	33,988
13	0	6,857	14,777	18,460	22,166	24,698	27,338	31,045	38,421
14	177	13,625	18,605	22,166	25,121	27,338	29,850	33,485	40,416
15	4,404	14,461	18,469	22,195	25,687	28,816	33,471	38,421	45,810
16	4,555	17,733	22,166	25,476	28,816	32,256	35,465	41,010	51,129
17	4,237	19,210	24,530	28,077	31,032	35,465	39,160	44,332	53,937

```r
stargazer::stargazer(t(as.data.frame(p)), type = 'latex', summary = FALSE, title = "p")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:26

Table 2: p

	0	2	3	4	5	6	7	8	9	10	11	12	13
p	0.001	0.002	0.006	0.008	0.012	0.020	0.023	0.059	0.042	0.065	0.065	0.343	0.054

Answers are tables 1 and 2.

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 $\hat{W}diag(\hat{p}_1, ..., \hat{p}_J)$. Estimate $\hat{\beta}_\tau$ for the education example above.

```r
# ---------------------------------------------------------------
# Calculate beta
# ---------------------------------------------------------------
X <- as.numeric(levels(myData$education))
W <- diag(p)
matrix_pi <- unname(as.matrix(myPi))

myBeta <- (t(X) %*% W %*% X)^{-1} %*% t(X) %*% W %*% matrix_pi


# clean up beta
myBeta_t <- as.data.frame(round(myBeta, 0))
colnames(myBeta_t) <- c(".1", ".2", ".3", ".4", ".5", ".6", ".7", ".8", ".9")
rownames(myBeta_t) <- c("Beta_t")

# print beta
stargazer::stargazer(myBeta_t, type = 'latex', summary = FALSE, title = "Beta")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:26

Table 3: Beta

	.1	.2	.3	.4	.5	.6	.7	.8	.9
Beta_t	106	916	$1,250$	$1,498$	$1,739$	$1,977$	$2,221$	$2,536$	$3,058$

$$\hat{\beta} = (X'WX)^{-1}X'W\pi$$

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:

$$V = (x'\hat{W}x)^{-1}x'\hat{W}\Sigma\hat{W}x(x'\hat{W}x)^{-1},$$

$$\Sigma = diag(\sigma^2_{\tau,1}/p_1, ..., \sigma^2_{\tau,J}/p_J,$$

$$D = (x'\hat{W}x)^{-1}x'\hat{W}\Delta\hat{W}x(x'\hat{W}x)^{-1},$$
$$\Delta = diag\Big((\pi_{t,1} - x_1\beta_\tau)^2/p_1, ..., (\pi_{t,J} - x_J\beta_\tau)^2/p_J\Big).$$

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

$$b_j = \max\Big\{1, round\Big(\tau N_j - z_{1-\alpha/2}(\tau(1-\tau)N_j)^{1/2}\Big)\Big\}$$

$$t_j = \min\Big\{1, round\Big(\tau N_j + z_{1-\alpha/2}(\tau(1-\tau)N_j)^{1/2}\Big)\Big\}$$

where round is to the closest integer, and $z_{1-\alpha/2} = 1.96$typically, and N_j is the number of observations in the jth bin of X. Then,

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

Report the standard error on you estimates, which is calculated as $((V + D/N))^{1/2}$

```
# --------------------------------------------------------------
# Calculate b_jt and t_jt
# --------------------------------------------------------------

myTau <- c(.1, .2, .3, .4, .5, .6, .7, .8, .9)
myN <- table(myData$education)

b_j_tau <- matrix(nrow = 17, ncol = 9)
t_j_tau <- matrix(nrow = 17, ncol = 9)


for (j in 1:17) {
  for (t in 1:9) {
    b_j_tau[j, t] <- max(1, round(myTau[t]*myN[j]- 1.96 * (myTau[t]*(1-myTau[t])*myN[j])^(1/2),
                                  digits = 0))
    t_j_tau[j,t]<- min(myN[j], round(myTau[t]*myN[j]+ 1.96 * (myTau[t]*(1-myTau[t])*myN[j])^(1/2),
                                  digits = 0))
  }

}

# --------------------------------------------------------------
# Calculating sigma (a 17 x 9 matrix)
# --------------------------------------------------------------
educ_j <- levels(myData$education)

sigma <- matrix(nrow = 17, ncol = 9)

#calculating y_j(t_j)
for (j in 1:17) {
  for (t in 1:9) {
    # calculating y_j()
    y_j <- myData %>%
      filter(education == educ_j[j]) %>%
      select(Y) %>%
      arrange(Y)

    y_j_t <- y_j$Y[t_j_tau[j,t]]
    y_j_b <- y_j$Y[b_j_tau[j,t]]

    sigma[j, t] <- myN[j]*((y_j_t - y_j_b)/2*1.96)^2

  }
}

# --------------------------------------------------------------
```

```r
# Calculating Sigma: nine 17x17 matrices
# ---------------------------------------------------------------
Sigma <- vector("list", 9)

for (t in 1:9) {
  Sigma[[t]] <- matrix(nrow = 17, ncol = 17,
                       data = diag(sigma[,t]))
}

# ---------------------------------------------------------------
# Calculating V
# ---------------------------------------------------------------
V <- vector(length = 9)

for (t in 1:9) {
  V[t] <- (t(X) %*% X)^{-1} %*% t(X) %*% W %*% Sigma[[t]] %*% W %*% X%*% (t(X) %*% W %*% X)^{-1}
}

# ---------------------------------------------------------------
# Calculating Delta
# ---------------------------------------------------------------
#storing each delta in a list
Delta <- vector("list", 9)

#each item is a matrix that is 17x17 with values along diagonal
Delta_vector <- vector(length = 17)

#populating Delta
for (i in 1:9) {
  for (j in 1:17) {
    Delta_vector[j] <- (myPi[j, i] - X[j]*myBeta[i])^2/p[j]
  }
  Delta[[i]] <- diag(Delta_vector)
}

# ---------------------------------------------------------------
# Calculating D
# ---------------------------------------------------------------
D <- vector(length = 9)

for (t in 1:9) {
  D[t] <- (t(X) %*% X)^{-1} %*% t(X) %*% W %*% Delta[[t]] %*% W %*% X%*% (t(X) %*% W %*% X)^{-1}
}

# ---------------------------------------------------------------
# Calculating and reporting SE
# ---------------------------------------------------------------
mySE <- vector(mode = "numeric", length = 9)

for (t in 1:9) {
  mySE[t] <- sqrt((V[t]+D[t])/sum(myN))
}
```

```r
# clean up beta
mySE_t <- as.data.frame(round(t(mySE), 3))
colnames(mySE_t) <- c(".1", ".2", ".3", ".4", ".5", ".6", ".7", ".8", ".9")
rownames(mySE_t) <- c("Stand_Error")

# print beta
stargazer::stargazer(mySE_t, type = 'latex', summary = FALSE, title = "Standard Errors for Betas")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:27

Table 4: Standard Errors for Betas

	.1	.2	.3	.4	.5	.6	.7	.8	.9
Stand_Error	20.519	16.578	12.062	10.197	11.157	13.023	9.272	15.847	27.797

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.

```r
#exported above code as R script to make it a function
# knitr::purl("pset_3.Rmd", documentation = 2)

# Reading in data
myData_2 <- vroom("lalonde_nsw.csv")

# source function
source('getBeta_SE.R')

# get results
results <- getBeta_SE(df = myData_2)

# ---------------------------------------------------------------
# Reporting Beta and SE
# ---------------------------------------------------------------
stargazer::stargazer(results$Beta, type = 'latex', summary = FALSE, title = "Beta for NSW")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:28

Table 5: Beta for NSW

	.1	.2	.3	.4	.5	.6	.7	.8	.9
Beta_t	16	18	50	169	345	508	698	936	1,209

```r
stargazer::stargazer(results$SE, type = 'latex', summary = FALSE, title = "Standard Error for NSW")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:28

Table 6: Standard Error for NSW

	.1	.2	.3	.4	.5	.6	.7	.8	.9
Stand_Error	1.391	2.280	6.608	14.146	19.686	19.224	20.048	22.581	40.058

Poisson Regression

a)

```
# ---------------------------------------------------------------
# Read in data and
# ---------------------------------------------------------------
rm(list = ls())
library(fixest)
library(broom)
library(kableExtra)

myData <- vroom("detroit.csv")

# ---------------------------------------------------------------
# OLS regression
# ---------------------------------------------------------------

regA <- feols(data = myData,
              fml = flows ~ distance_Google_miles | home_ID + work_ID) %>%
  tidy()
```

Coefficient (SE): -0.0929996, (0.0014964)

b)

```
 # ---------------------------------------------------------------
# OLS regression
# ---------------------------------------------------------------

regB <- feols(data = myData,
              fml = log(flows) ~ log(distance_Google_miles) | home_ID + work_ID) %>%
  tidy()
```

```
## NOTE: 994,409 observations removed because of infinite values (LHS: 994,409).
```

Coefficient (SE): -0.4071607, (0.0038916)

This transformation drops 75% of the data set where either the dependent or independent variable is a zero due to the log transformation.

c)

```
regC <- vector("list", 3)

regC[[1]] <- feols(data = myData,
             fml = log(0.1 + flows) ~ log(distance_Google_miles) | home_ID + work_ID) %>%
  tidy()

regC[[2]] <- feols(data = myData,
             fml = log(1 + flows) ~ log(distance_Google_miles) | home_ID + work_ID) %>%
  tidy()

regC[[3]] <- feols(data = myData,
             fml = log(10 + flows) ~ log(distance_Google_miles) | home_ID + work_ID) %>%
  tidy()
```

Coefficient (SE) for 0.1: -0.6779384, (0.0065282)

Coefficient (SE) for 1: -0.3281698, (0.003748)

Coefficient (SE) for 10: -0.0948925, (0.001393)

The addition of c prevents the dropping of observations, however the interpretation of the result is confounded by the addition of c. As c increases, the estimate approached zero.

d

```
regD <- fepois(data = myData,
               fml = flows ~ distance_Google_miles | home_ID + work_ID, vcov = "hetero") %>%
  tidy()
```

```
## NOTE: 0/2 fixed-effects (2,326 observations) removed because of only 0 outcomes.
```

The poisson model (numerically) estimates

$$\frac{dlog(E(Y|X))}{dX}.$$

which can be interpreted as the log of a semi-elasticity. This allows up to keep the zeros and also still have a logged estimand that is a matter of interest. The estimates for a poisson are consistent and the standard error coverage is correct when using heteroskedastically robust standard errors.

Coefficient (SE) for poisson: -0.0885796, (2.3868959×10^{-4})

The estimate is very close to part A, but the standard errors are much smaller. Estimates in B and C (0.1) and C (1) were far larger. C (10) was a similar estimate, but the interpretation of the estimand in C is maybe impossible.

Duration Modeling

a

The unconditional probability that a household stays in a home for T or more yeas is $1 - F(t)$, where $F(t) = Pr(T < t)$ which is the probability of a duration shorter than t.

```r
# ----------------------------------------------------------------
# Data set up
# ----------------------------------------------------------------

rm(list = ls())

myData_og <- vroom("https://raw.githubusercontent.com/paulgp/applied-methods-phd/main/homework/Homework5


# ----------------------------------------------------------------
# prob stays in house longer than 7 years set up
# ----------------------------------------------------------------
myData <- myData_og %>%
  mutate(T_less_7 = (moving_approx < 7))
```

$1 - F(7) = 0.6324$

b

I calculate the hazard rate as

$$\theta(t) = -\frac{\log(1 - F(t))}{t}$$

(page 17/30 in the duration modeling slides).

```r
# ----------------------------------------------------------------
# Different hazard rates
# ----------------------------------------------------------------
move_levels <- as.numeric(levels(as.factor(myData$moving_approx)))

myHazard_rates <- vector(mode = "numeric", length = 7)

for (i in 1:7) {
  myTemp <- myData %>%
    mutate(T_less = (moving_approx < move_levels[i]))

  F_t <- sum(myTemp$T_less)/length(myTemp$T_less)

  myHazard_rates[i] <- -log(1 - F_t)/ move_levels[i]
}

# ----------------------------------------------------------------
# print results
# ----------------------------------------------------------------
myHazard_rates_t <- as.data.frame(round(t(myHazard_rates), 5))
colnames(myHazard_rates_t) <- move_levels
rownames(myHazard_rates_t) <- c("Hazard Rate")

# print beta
stargazer::stargazer(myHazard_rates_t, type = 'latex', summary = FALSE, title = "Hazard Rate")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:34

Table 7: Hazard Rate

	0.5	1.5	3	7	15	25	40
Hazard Rate	0	0.082	0.069	0.065	0.048	0.052	0.048

```
t_7 <- myHazard_rates_t$`7`
```

The hazard rate for T = 7 is 0.06546.

C

```
# -------------------------------------------------------------
# hazard rates for home and rent
# -------------------------------------------------------------
myHazard_rates_home <- vector(mode = "numeric", length = 7)
myHazard_rates_rent <- vector(mode = "numeric", length = 7)

for (i in 1:7) {
  # home
  myTemp_home <- myData %>%
    filter(homeowner) %>%
    mutate(T_less = (moving_approx < move_levels[i]))

  F_t_home <- sum(myTemp_home$T_less)/length(myTemp_home$T_less)

  myHazard_rates_home[i] <- -log(1 - F_t_home)/ move_levels[i]

  # rents
  myTemp_rent <- myData %>%
    filter(renter) %>%
    mutate(T_less = (moving_approx < move_levels[i]))

  F_t_rent <- sum(myTemp_rent$T_less)/length(myTemp_rent$T_less)

  myHazard_rates_rent[i] <- -log(1 - F_t_rent)/ move_levels[i]
}

# -------------------------------------------------------------
# print results
# -------------------------------------------------------------
# home
myH_home <- as.data.frame(round(t(myHazard_rates_home), 5))
colnames(myH_home) <- move_levels
rownames(myH_home) <- c("Homeowner")

# rent
myH_rent <- as.data.frame(round(t(myHazard_rates_rent), 5))
colnames(myH_rent) <- move_levels
rownames(myH_rent) <- c("Renter")
```

```r
myH_both <-bind_rows(myH_home, myH_rent)

# print beta
stargazer::stargazer(myH_both, type = 'latex', summary = FALSE, title = "Hazard Rates")
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Sun, Feb 26, 2023 - 20:12:35

Table 8: Hazard Rates

	0.5	1.5	3	7	15	25	40
Homeowner	0	0.040	0.037	0.041	0.033	0.041	0.041
Renter	0	0.199	0.165	0.155	0.116	0.110	0.089

Renters always have a higher hazard rate than homeowners. The hazard values for homeowners are 0.04 and 0.15 for renters.