**Introduction**

Today we are going to talk about quantile regression. When we use the lm command in R we are fitting a linear regression using Ordinary Least Squares (OLS), which has the interpretation of a model for the conditional mean of yon x. However, sometimes we may need to look at more than the conditional mean to understand our data and quantile regressions may be a good alternative. Instead of looking at the mean, quantile regressions will establish models for particular quantiles as chosen by the user. The most simple case when quantile regressions are good is when you have outliers in your data because the median is much less affected by extreme values than the mean (0.5 quantile). But there are other cases where quantile regression may be used, for example to identify some heterogeneous effects of some variable or even to give more robustness to your results.

**Outliers**

The package we will be using for quantile regressions is the quantreg, which is very easy to use if you are already familiar with the lm function.

library(quantreg)

Our data is going to be very simple. The sample size will be 300 and we will have 10 variables. The \betas will be either 1 or -1 depending on the index of the variable. This is our dgp:

\displaystyle y_i =  -x_{1,i}+x_{2,i}-x_{3,i}+\dots+x_{10,i} + \varepsilon_i

where x_iand \varepsilon_iare all generated from a N(0,1)distribution.

The first step is to generate the data and run a linear regression by OLS. As you can see in the results below, the estimates are very precise. The signals are always correct and the coefficients are close to 1 or -1.

N = 300 #sample size

P = 10 # number of variable

### Generate Data ###

betas = (-1)^(1:P)

set.seed(1) # Seed for replication

error = rnorm(N)

X = matrix(rnorm(N\*P), N, P)

y = X%\*%betas + error

summary(lm(y~X))

##

## Call:

## lm(formula = y ~ X)

##

## Residuals:

## Min 1Q Median 3Q Max

## -2.94301 -0.58806 -0.03508 0.61445 2.55389

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 0.03501 0.05739 0.61 0.542

## X1 -0.98513 0.05534 -17.80 <2e-16 \*\*\*

## X2 1.04319 0.05321 19.60 <2e-16 \*\*\*

## X3 -1.01520 0.05577 -18.20 <2e-16 \*\*\*

## X4 0.92930 0.05705 16.29 <2e-16 \*\*\*

## X5 -0.97971 0.05404 -18.13 <2e-16 \*\*\*

## X6 1.01255 0.05376 18.83 <2e-16 \*\*\*

## X7 -1.02357 0.05622 -18.21 <2e-16 \*\*\*

## X8 0.97456 0.05946 16.39 <2e-16 \*\*\*

## X9 -0.97097 0.05295 -18.34 <2e-16 \*\*\*

## X10 1.03386 0.05312 19.46 <2e-16 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 0.9747 on 289 degrees of freedom

## Multiple R-squared: 0.9274, Adjusted R-squared: 0.9249

## F-statistic: 369 on 10 and 289 DF, p-value: < 2.2e-16

Now we are going to add some outliers to the data. We have 300 observations and only 5 are going to be outliers. The code below introduces the outliers and run OLS again. Estimates are now considerably worst than the first case with no outliers.

### Introducing outliers ###

set.seed(1)

outliers = rnorm(5, 0, 50)

error\_o = error

error\_o[1:5] = outliers

y\_o = X%\*%betas + error\_o

## OLS ###

summary(lm(y\_o~X))

##

## Call:

## lm(formula = y\_o ~ X)

##

## Residuals:

## Min 1Q Median 3Q Max

## -41.216 -1.105 -0.027 0.828 77.040

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 0.1951 0.3367 0.579 0.562717

## X1 -1.3165 0.3247 -4.055 6.45e-05 \*\*\*

## X2 1.1355 0.3122 3.637 0.000326 \*\*\*

## X3 -1.0541 0.3272 -3.222 0.001421 \*\*

## X4 0.9136 0.3347 2.729 0.006733 \*\*

## X5 -1.2849 0.3171 -4.052 6.52e-05 \*\*\*

## X6 1.2834 0.3154 4.069 6.10e-05 \*\*\*

## X7 -0.7777 0.3298 -2.358 0.019055 \*

## X8 1.4859 0.3488 4.260 2.77e-05 \*\*\*

## X9 -1.0915 0.3107 -3.513 0.000513 \*\*\*

## X10 0.5756 0.3117 1.847 0.065797 .

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 5.719 on 289 degrees of freedom

## Multiple R-squared: 0.3223, Adjusted R-squared: 0.2989

## F-statistic: 13.75 on 10 and 289 DF, p-value: < 2.2e-16

If we run a quantile regression for the median like in the code below we can get good results once again. Note that in the OLS case the coefficients were not as close to 1 or -1 as in the quantile case below.

## Quantile Regression - Median ##

summary(rq(y\_o ~ X, tau = 0.5), se = "boot", bsmethod= "xy")

##

## Call: rq(formula = y\_o ~ X, tau = 0.5)

##

## tau: [1] 0.5

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) -0.03826 0.08202 -0.46642 0.64127

## X1 -0.95730 0.07394 -12.94750 0.00000

## X2 0.99394 0.07612 13.05731 0.00000

## X3 -0.97981 0.07995 -12.25523 0.00000

## X4 0.92966 0.09319 9.97573 0.00000

## X5 -0.92781 0.08520 -10.88998 0.00000

## X6 0.98021 0.08114 12.08030 0.00000

## X7 -1.00993 0.06727 -15.01266 0.00000

## X8 1.03572 0.08043 12.87722 0.00000

## X9 -0.96732 0.07502 -12.89374 0.00000

## X10 1.02600 0.08775 11.69171 0.00000

**Treatment effects**

Suppose now that we have the same model, without outliers, plus a treatment that is positive for lower values of yand negative for bigger values of y. Only half of the sample will be treated and we want to see what we get when we try to estimate the effects of this treatment. The figure below plots the original yagainst the treated y. The points in the 45 degrees line are the untreated observations.

Our treatment was simple adding 5 to the equation if the deterministic part of the equation is negative and subtracting 5 if it is positive. If we run OLS we can se below that the treatment effects are very poorly estimated and they are also not significant. That is because we are looking at the average effect of the treatment, which is 0 because half of the treated sample had an increase of 5 and the other half had a decrease of five, which is 0 on average.

#### More complicated data #####

set.seed(1)

treatment = sample(c(0,1),N,replace = TRUE)

aux = X%\*%betas

treatment\_error = rnorm(N)\*treatment

y\_tr = aux + ifelse(aux0,-5,0)\*treatment + error + treatment\_error

X\_tr = cbind(treatment,X)

## Treated X Untreated

plot(y,y\_tr)



Now lets try quantile regression for multiple quantiles (0.1 ,0.2 ,…,0.8, 0.9). The results are presented below. When we look at the middle quantiles like 0.5 and 0.6 we find that the treatment is not significant just like in the OLS case. However, as we move further to 0.1 or 0.9 we obtain significant results with estimates very close to the real treatment, which would be 5 or -5. The model is telling us that bigger values of yare either untreated samples or treated samples that were previously small but became big because of the treatment of 5, that is why the treatment effect is close to 5 in the 0.9 quantile. Similarly, smaller values of yare either untreated samples or treated samples that were previously big and became small because of the treatment of -5. **We are modeling the quantiles of the treated y.**

## OLS ##

summary(lm(y\_tr ~ X\_tr))

##

## Call:

## lm(formula = y\_tr ~ X\_tr)

##

## Residuals:

## Min 1Q Median 3Q Max

## -7.5913 -2.0909 0.0168 1.8840 8.8915

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -0.08713 0.24512 -0.355 0.722502

## X\_trtreatment -0.24555 0.35698 -0.688 0.492093

## X\_tr -0.40117 0.17283 -2.321 0.020973 \*

## X\_tr 0.55826 0.16594 3.364 0.000872 \*\*\*

## X\_tr -0.61606 0.17365 -3.548 0.000454 \*\*\*

## X\_tr 0.56658 0.17766 3.189 0.001584 \*\*

## X\_tr -0.37539 0.16829 -2.231 0.026477 \*

## X\_tr 0.43680 0.16788 2.602 0.009752 \*\*

## X\_tr -0.29718 0.17523 -1.696 0.090972 .

## X\_tr 0.39646 0.18585 2.133 0.033751 \*

## X\_tr -0.34905 0.16499 -2.116 0.035233 \*

## X\_tr 0.83088 0.16593 5.007 9.64e-07 \*\*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 3.035 on 288 degrees of freedom

## Multiple R-squared: 0.2454, Adjusted R-squared: 0.2166

## F-statistic: 8.514 on 11 and 288 DF, p-value: 5.243e-13

# Quantilie Regression for q = 0.1, 0.2, ... , 0.8, 0.9

summary(rq(y\_tr ~ X\_tr, tau = seq(0.1,0.9,0.1)), se = "boot", bsmethod= "xy")

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.1

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) -1.59418 0.22484 -7.09019 0.00000

## X\_trtreatment -4.13074 0.47280 -8.73680 0.00000

## X\_tr -0.85152 0.17088 -4.98320 0.00000

## X\_tr 1.10326 0.14030 7.86384 0.00000

## X\_tr -0.85055 0.18554 -4.58421 0.00001

## X\_tr 0.68881 0.21612 3.18719 0.00159

## X\_tr -0.88047 0.16396 -5.36995 0.00000

## X\_tr 0.68033 0.15287 4.45051 0.00001

## X\_tr -1.00234 0.15569 -6.43819 0.00000

## X\_tr 0.99443 0.19484 5.10369 0.00000

## X\_tr -0.94634 0.17999 -5.25759 0.00000

## X\_tr 0.92124 0.17348 5.31048 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.2

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) -0.87790 0.18851 -4.65701 0.00000

## X\_trtreatment -4.04359 0.43354 -9.32695 0.00000

## X\_tr -0.87484 0.17336 -5.04648 0.00000

## X\_tr 0.90953 0.16734 5.43510 0.00000

## X\_tr -0.92386 0.15931 -5.79905 0.00000

## X\_tr 0.84438 0.14318 5.89740 0.00000

## X\_tr -0.79882 0.16353 -4.88478 0.00000

## X\_tr 0.86351 0.14477 5.96469 0.00000

## X\_tr -0.89046 0.14533 -6.12699 0.00000

## X\_tr 0.80481 0.16133 4.98842 0.00000

## X\_tr -0.85986 0.14983 -5.73877 0.00000

## X\_tr 1.05532 0.14247 7.40725 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.3

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) -0.48650 0.15254 -3.18941 0.00158

## X\_trtreatment -3.55483 0.39081 -9.09596 0.00000

## X\_tr -0.81732 0.15516 -5.26750 0.00000

## X\_tr 0.97204 0.15176 6.40526 0.00000

## X\_tr -0.88749 0.12320 -7.20353 0.00000

## X\_tr 0.80248 0.13892 5.77649 0.00000

## X\_tr -0.63208 0.15928 -3.96825 0.00009

## X\_tr 0.85960 0.12409 6.92695 0.00000

## X\_tr -0.73083 0.13347 -5.47540 0.00000

## X\_tr 0.80175 0.14581 5.49859 0.00000

## X\_tr -0.73295 0.14156 -5.17769 0.00000

## X\_tr 0.96849 0.14074 6.88120 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.4

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) -0.20518 0.17305 -1.18566 0.23673

## X\_trtreatment -3.12524 0.82009 -3.81083 0.00017

## X\_tr -0.75969 0.17512 -4.33802 0.00002

## X\_tr 0.75137 0.18705 4.01697 0.00008

## X\_tr -0.89673 0.15018 -5.97081 0.00000

## X\_tr 0.81794 0.15558 5.25741 0.00000

## X\_tr -0.60392 0.18754 -3.22019 0.00143

## X\_tr 0.76221 0.16089 4.73733 0.00000

## X\_tr -0.63207 0.24626 -2.56665 0.01077

## X\_tr 0.70476 0.24297 2.90064 0.00401

## X\_tr -0.72050 0.21575 -3.33951 0.00095

## X\_tr 0.97243 0.14277 6.81125 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.5

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) 0.01435 0.17259 0.08317 0.93377

## X\_trtreatment -0.63174 0.97917 -0.64518 0.51933

## X\_tr -0.60768 0.19231 -3.15992 0.00175

## X\_tr 0.54861 0.19838 2.76543 0.00605

## X\_tr -0.80869 0.16209 -4.98908 0.00000

## X\_tr 0.73479 0.18775 3.91377 0.00011

## X\_tr -0.41427 0.21412 -1.93477 0.05400

## X\_tr 0.64703 0.21863 2.95946 0.00334

## X\_tr -0.24361 0.22809 -1.06801 0.28641

## X\_tr 0.37166 0.23587 1.57566 0.11620

## X\_tr -0.45650 0.19626 -2.32605 0.02071

## X\_tr 0.92603 0.17611 5.25825 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.6

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) 0.25546 0.18342 1.39270 0.16478

## X\_trtreatment 1.36457 1.24271 1.09806 0.27310

## X\_tr -0.60693 0.21221 -2.86008 0.00455

## X\_tr 0.64917 0.21300 3.04782 0.00252

## X\_tr -0.83220 0.17072 -4.87459 0.00000

## X\_tr 0.73617 0.18348 4.01219 0.00008

## X\_tr -0.45447 0.19288 -2.35628 0.01913

## X\_tr 0.76270 0.21047 3.62385 0.00034

## X\_tr -0.45186 0.25197 -1.79332 0.07397

## X\_tr 0.46886 0.22511 2.08276 0.03816

## X\_tr -0.56117 0.22729 -2.46900 0.01413

## X\_tr 0.94549 0.19311 4.89600 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.7

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) 0.52520 0.14537 3.61291 0.00036

## X\_trtreatment 3.30124 0.61742 5.34682 0.00000

## X\_tr -0.67172 0.15957 -4.20960 0.00003

## X\_tr 0.74401 0.15205 4.89331 0.00000

## X\_tr -0.93090 0.15068 -6.17788 0.00000

## X\_tr 0.78792 0.15054 5.23389 0.00000

## X\_tr -0.72815 0.13243 -5.49819 0.00000

## X\_tr 0.84332 0.14158 5.95641 0.00000

## X\_tr -0.71329 0.16296 -4.37707 0.00002

## X\_tr 0.59229 0.17063 3.47111 0.00060

## X\_tr -0.76839 0.16055 -4.78593 0.00000

## X\_tr 0.98971 0.15318 6.46091 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.8

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) 0.79126 0.14003 5.65083 0.00000

## X\_trtreatment 3.70886 0.36971 10.03188 0.00000

## X\_tr -0.73646 0.14104 -5.22148 0.00000

## X\_tr 0.71236 0.12350 5.76817 0.00000

## X\_tr -0.86668 0.13143 -6.59424 0.00000

## X\_tr 0.75392 0.13379 5.63521 0.00000

## X\_tr -0.76165 0.10571 -7.20496 0.00000

## X\_tr 0.86711 0.11852 7.31598 0.00000

## X\_tr -0.73505 0.13053 -5.63128 0.00000

## X\_tr 0.80531 0.13801 5.83513 0.00000

## X\_tr -0.79344 0.12981 -6.11250 0.00000

## X\_tr 1.00810 0.13979 7.21169 0.00000

##

## Call: rq(formula = y\_tr ~ X\_tr, tau = seq(0.1, 0.9, 0.1))

##

## tau: [1] 0.9

##

## Coefficients:

## Value Std. Error t value Pr(>|t|)

## (Intercept) 1.16385 0.16462 7.07006 0.00000

## X\_trtreatment 4.08872 0.37080 11.02661 0.00000

## X\_tr -0.93217 0.15900 -5.86256 0.00000

## X\_tr 0.82984 0.12705 6.53140 0.00000

## X\_tr -0.93147 0.13285 -7.01143 0.00000

## X\_tr 0.70112 0.12903 5.43359 0.00000

## X\_tr -0.90992 0.13283 -6.85018 0.00000

## X\_tr 0.84127 0.14036 5.99349 0.00000

## X\_tr -0.66466 0.15369 -4.32462 0.00002

## X\_tr 0.84578 0.17838 4.74134 0.00000

## X\_tr -0.84474 0.13833 -6.10682 0.00000

## X\_tr 0.98760 0.16193 6.09902 0.00000