Quantile Regression via the EM algorithm: the QREM package

Haim Bar

April 13, 2021

1 Simulated Data

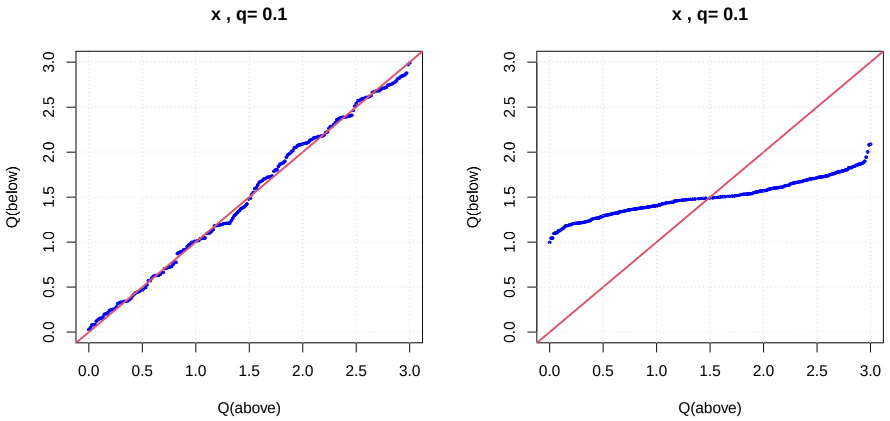
1.1 Assessing Goodness of Fit

The following code shows a simulation scenario[1] in which the relationship between x and y is quadratic, and the error variance increases with x. The model is specified in line 14. We use QREM to fit the data twice – once with the correct model (line 22) and once with an incorrect model, where only a linear relationship is assumed (line 28).

```
1   library(QREM)
2
3   ##########################################################################
4   # Simulation 23 in table A3.
5   # Quadratic mean model, increasing variance.
6   # Compare the fit of the correct model, y~x+I(x^2), with the incorrect,
7   #   linear model. Show diagnostic plots - Q-Q plot when qn=0.1, and
8   #   a flat diagnostic plot for all the quantiles, qns.
9   set.seed(21322)
10  n <- 2000
11  qns <- seq(0.05,0.95,by=0.05)
12  x <- seq(0,3,length.out = n)
13  L <- 20
14  y <- 6*x^2 + x +120 + rnorm(n, 0, 0.1+0.5*x)
15  qrdg <- matrix(0,nrow=n, ncol=length(qns))
16  xqs <- quantile(x, probs = (1:(L-1))/L)
17  names(xqs) <- c()
18  qqp  <- matrix(0, nrow=length(xqs), ncol=length(qns))
19  qqp2 <- matrix(0, nrow=length(xqs), ncol=length(qns))
20  i <- 1
21  for (qn in qns) {
22    qremFit <- QREM(lm, linmod=y~x+I(x^2), df=data.frame(y,x,x^2), qn=qn)
23    qrdg <- QRdiagnostics(x, "x",qremFit$ui, qn,  plot.it = ifelse(abs(qn-0.1) < 1e-6, TRUE, FALSE),
      ↪  filename="tmp/sim23q10correct.pdf")
24    for (j in 1:(L-1)) {
25      qqp[j,i] <- length(which(qrdg$y < xqs[j])) / length(which(qrdg$x < xqs[j]))
26    }
27
28    qremFit <- QREM(lm, linmod=y~x, df=data.frame(y,x,x^2), qn=qn)
29    qrdg <- QRdiagnostics(x, "x",qremFit$ui, qn, plot.it = ifelse(abs(qn-0.1) < 1e-6, TRUE, FALSE),
      ↪  filename="tmp/sim23q10incorrect.pdf")
30    for (j in 1:(L-1)) {
31      qqp2[j,i] <- length(which(qrdg$y < xqs[j])) / length(which(qrdg$x < xqs[j]))
32    }
33    i <- i+1
34  }
35  flatQQplot(x,xqs,qqp,qns, L=21, filename = "tmp/flatQQsim23q10correct.pdf")
36  flatQQplot(x,xqs,qqp2,qns, L=21, filename = "tmp/flatQQsim23q10incorrect.pdf")
```

We use this example to demonstrate two diagnostics functions in the package. `QRdiagnostics` is used to assess goodness of fit for one variable, at one selected quantile. For continuous variables it produces Q-Q plots, such that an appropriate model will results in the residual quantiles to lie along the main diagonal. For categorical variables it produces a spinogram, so that if the model is

[1] In the examples in this documentation, the simulation numbers refer to Tables A3 and A4 in our paper



(a) correct (quadratic) model

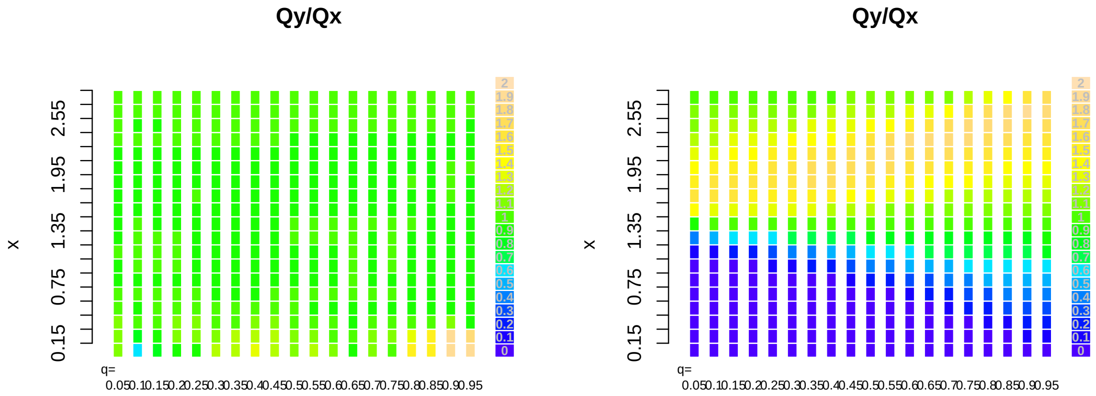
(b) incorrect (linear) model

Figure 1: Simulation 23 – Diagnostic plot using the `QRdiagnostics` function. The true model is $y \sim N(6x^2 + x + 120, (0.1 + 0.5x)^2))$

appropriate the percentage of points below the regression line is the chosen quantile for each level of the factor. The function `flatQQplot` can be used to plot a heatmap which represents the Q-Q plots for multiple quantiles in one graph. This can reveal whether a model fits well in all quantiles, or if it is misspecified in some.

In lines 23 and 29 we generate the diagnostic plot for the linear term, x, at $q = 0.1$ for the correct and incorrect models, respectively. We see in Figure 1 (a) that the points lie along the diagonal, indicating a good fit of the quadratic model, while in Figure 1 (b) there is substantial deviation from the diagonal, suggesting that the linear model is inadequate.

In lines 35-36 we generate the 'flat Q-Q plot' for the linear term, for all the quantiles in our analysis. Figure 2 (a) shows a near-ideal situation. The heatmap represents the average ratio between the x and y axes in the Q-Q plot. The green color represents a value close to 1, which means that the points are almost exactly along the main diagonal. Figure 2 (b) shows that for all quantiles, the linear model is inadequate, since each column contains cells which represents ratios which are quite different from 1.



(a) correct (quadratic) model

(b) correct (quadratic) model

Figure 2: Simulation 23 – Diagnostic plot using the `flatQQplot` function.

1.2 Estimation of Standard Deviation of Regression Estimates

A second example is provided in the code below. In this case, the true model (line 12) involves an interaction between two continuous variables, and a variance which increases with the first variable. The dataset also contains a categorical variables with levels C, T1 and T2.

```
 1  library(QREM)
 2  
 3  ##########################################################################
 4  # Simulation 24 in table A3
 5  # A two-way interaction model between two continuous variables, and an
 6  #  increasing variance. A categorical variable is also included in the data.
 7  
 8  n <- 10000
 9  x <- seq(0,1,length.out = n)
10  x2 <- x[sample(n)]
11  x3 <- factor(c(rep("C",5000), rep("T1", 3000), rep("T2",2000)))
12  y <- 4*x*x2 + rnorm(n,0,0.1+0.2*x)
```

QREM provides two methods to estimate the standard deviation of regression parameters in fixed effects models – one is based on the bootstrap (line 2, and the apply statement in line 3), and the other is based on an asymptotic approximation (Bahadur-type estimate, through the bcov function in line 3.) The following code is used to generate both:

```
1  qremFit <- QREM(lm, linmod=y~x*x2 +x3, df=data.frame(y,x,x2, x3), qn=qn)
2  estBS <- boot.QREM(lm, linmod=y~x*x2+x3, df=data.frame(y,x,x2,x3), qn=0.2, n=length(y), B=50)
3  ests <- rbind(qremFit$coef$beta,apply(estBS,2, sd), sqrt(diag(bcov(qremFit,linmod=y~x*x2+x3,
   ↪  df=data.frame(y,x,x2,x3), qn=0.2))))
4  rownames(ests) <- c("Estimate","Bootstrap","Bahadur")
5  print(xtable(ests))
```

As can be seen in the following table, the estimates of the quantile regression ($q = 0.2$) model are accurate, and the standard deviation estimates from the two methods are similar. The Bahadur-type estimates are slightly more conservative:

	(Intercept)	x	x2	x3T1	x3T2	x:x2
Estimate	0.15	0.38	0.02	-0.02	-0.05	4.01
Bootstrap	0.01	0.03	0.01	0.01	0.01	0.03
Bahadur	0.02	0.04	0.03	0.02	0.02	0.05

1.3 Mixed Models

QREM also allows to fit mixed models (as well as generalized additive models, or 'gam'). The following code shows how we generate the data and fit the quantile regression model.

```
library(QREM)

##########################################################################
# Simulation 25 in table A3 - A mixed model

n <- 100
tm <- 4
z <- kronecker(diag(1,n),rep(1,tm))
set.seed(71371)
x <- rep(c(1:tm)/tm,n) + rnorm(n*tm,0,2e-2)
u <- rnorm(n,0,0.5)
e <- rnorm(tm*n,0,0.1)
y <- 2 + x + z%*%u + e
dframe <- data.frame(y,x,as.factor(rep(1:tm, n)),as.factor(kronecker(1:n,rep(1,tm))))
colnames(dframe) <- c("y","X","Z","S")
```

The sample size is $n = 100$ and each subject is observed at four time points. The subjects are independent, but observations within subject are correlated (via the variable u in lines 11 and 13). To fit the quantile regression model we use the code below. Notice that the only difference, compared with the previous examples, is that we use the `lmer` function and the appropriate formula to specify the model to be fitted.

```
linmod <- y~X+(1|S)
qn <- 0.2
qremFit <- QREM(lmer,linmod, dframe, qn)
estBS <- boot.QREM(lmer,linmod, dframe, qn, n=nrow(dframe), B=50)
ests <- rbind(qremFit$coef$beta,apply(estBS,2, sd))
rownames(ests) <- c("Estimate","Bootstrap s.d.")
library(xtable)
print(xtable(ests))
```

The bootstrap is the only valid estimation approach for the standard deviations for mixed models (and it may take a couple of minutes to complete the previous code segment). The results are shown in the following table.

	(Intercept)	X
Estimate	1.81	1.03
Bootstrap s.d.	0.02	0.03

1.4 Quantile Regression with Variable Selection

It is also possible to perform variable selection in quantile regression by combining functionality from QREM with the `fitSEMMS` function in the SEMMS package [1]. In the following code, we generate data according to scenario 5 in Table A4. The response is a function of five variables, but the total number of predictors is 500. The additional 495 variables are added in line 13 below. Note also that the variance of errors is not constant. It increases with the first predictor.

```
1  library(SEMMS)
2  library(QREM)
3  ##########################################################################
4  # Simulation 5 in table A4 - variable selection with SEMMS
5
6  coefs <- c(1, -3,2,2,-1,-2)
7  n <- 200
8  X <- matrix(runif(n*6),nrow=n, ncol=6)
9  X[,1] <- rep(1,n)
10 colnames(X) <- c("const", paste("X",1:5,sep=""))
11 set.seed(11002)
12 y <- X%*%as.matrix(coefs,ncol=1,nrow=6) + rnorm(n, 0, 0.1+X[,2])
13 dframe <- data.frame(y,X[,-1], matrix(rnorm((500-ncol(X)+1)*n,0,0.1), nrow=n,
   ↪  ncol=(500-ncol(X)+1)))
14 save(dframe, file="tmp/sim05.RData")
```

We then use the `QREM_vs` function in the QREM package to select variables associated with the qth quantile. The first argument to the function is a data set that can be read by the SEMMS package via the `readInputFile` function. It may be a simple csv file, or an Rdata file which contains a data frame with variables in the columns and observations in rows, as in the code above. See the SEMMS documentation for more details.

The second argument is the column number in the data where the response is stored. The third argument is a vector of column number containing the putative predictors, and the forth one is the quantile.

The `QREM_vs` function starts by obtaining an initial set of candidates from the putative variables. Then, it iteratively alternates between the variable selection step, and the quantile regression step using only the selected predictors.

The result (the variable res below) contains the output from SEMMS from the last iteration of the variable selection algorithm, and the QREM output for the final model. We then extract the selected variables (line 3) by using the variable containing the columns of the non-null predictors from the complete data set (by using `1+res$fittedSEMMS$gam.out$nn`). In line 4, we fit the QR model with the selected variables, and proceed as before (e.g. estimating the standard errors of the estimates, or producing a diagnostic plot).

```
1  qn <- 0.25
2  res <- QREM_vs("tmp/sim05.RData", 1, 2:501, qn=qn)
3  dfsemms <- dframe[,c(1, 1+res$fittedSEMMS$gam.out$nn)]
4  qremFit <- QREM(lm, y~., dfsemms, qn=qn)
5  ests <- rbind(qremFit$coef$beta, sqrt(diag(bcov(qremFit,linmod=y~., df=dfsemms, qn=0.2))))
6  rownames(ests) <- c("Estimate","s.d. (Bahadur)")
7  library(xtable)
8  print(xtable(ests, digits=3), file="tmp/SimA4_5.tex")
9
10 qrdg <- QRdiagnostics(dfsemms[,2], "X1",qremFit$ui, qn, filename="tmp/simVS5.pdf")
```

The following table shows the parameter estimates. The five predictors were selected by SEMMS, and no false predictors were chosen in the process, and Figure 3 shows the Q-Q plot for the first predictor (note that column 1 in the dfsemms matrix is the response). The model fits well. The points lie approximately along the diagonal, and the deviation of the points in the middle could be due to the relatively small sample size (although $n = 200$, since $q = 0.25$ the effective sample size for this diagram is 50.)

	(Intercept)	X1	X2	X3	X4	X5
Estimate	0.696	-3.491	1.960	2.032	-0.903	-1.732
s.d. (Bahadur)	0.162	0.139	0.139	0.133	0.141	0.144

2 Case Study – Daily Temperature Data

We demonstrate fitting a quantile regression mixed-model using our approach to daily temperature data from Las Vegas, NV. Minimum and maximum temperatures (in Fahrenheit) from January 1, 1960 to December 31, 2010 were obtained from the National Oceanic and Atmospheric Administration website (https://www.noaa.gov/). We fit a model which allows for the temperature characteristics to change linearly over time, while accounting for the cyclical nature of temperatures. To allow for extra variability in the data in addition to between-days variability, we include a random intercept for each month of the year.

Let y_{ijk} denote the daily maximum (minimum) temperature in month j and day k of year i and define $S_k = \sin(2\pi(k-11)/365 - \pi/2)$, for $k = 1, \ldots, 365$ (or 366), to represent the annual sinusoidal variation of daily temperatures, with a minimum at (approximately) the shortest day of the year in the northern hemisphere, December 21, and a maximum at around June 21. Our main objective

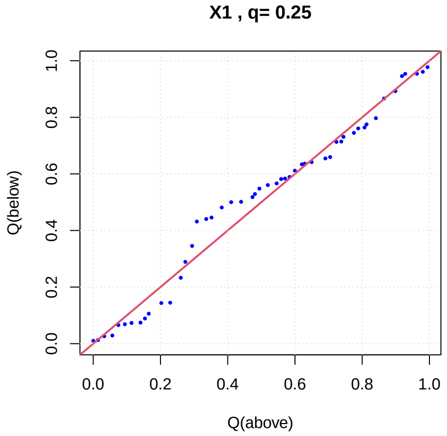


Figure 3: Simulation 5 in Table A4 – Diagnostic plot for the first predictor using the `flatQQplot` function.

is to estimate linear trends in temperatures over the 50 year period. We fit six models with linear predictors of the form

$$\eta_{ijk} = \mu + \beta_1 i + \beta_2 S_k + v_j \, .$$

for the 10-th percentile, the mean, or the 90-th percentile of the daily minimum or maximum temperature, and where v_j, $j = 1, \ldots, 12$ is a (random) effect of month j.

First, we load the data:

```r
# quantile regression application - temperature data, Las Vegas, NV

library("QREM")
dat <- read.csv("LasVegas.csv", header=TRUE)
dat$DATE <- as.Date(dat$DATE, "%Y-%m-%d")
dat$year <- as.numeric(format(dat$DATE,"%Y"))
dat$month <- factor(months(dat$DATE), levels = month.name)
dat$yday <- as.numeric(format(dat$DATE,"%j"))
dat$sinedoy <- sin(2*pi*(dat$yday-11)/365-pi/2)
```

Here, we only show the change in terms of the coefficient β_1 for two reponses:

- The 10-th percentile of the minimum temperature, and
- The 90-th percentile of the maximum temperature.

The models are specified and fitted as follows:

```r
linmodMin <- TMIN ~ sinedoy + year + (1|month)
linmodMax <- TMAX ~ sinedoy + year + (1|month)

qremFit10min <- QREM(lmer,linmodMin, dat,qn = 0.1)
qrdg10 <- QRdiagnostics(dat$year, "year (min. temp.)", qremFit10min$ui, qn=0.1, filename =
↪  "tmp/LVmintemp.pdf")

qremFit90max <- QREM(lmer,linmodMax, dat,qn = 0.9)
qrdg90 <- QRdiagnostics(dat$year, "year (max. temp.)", qremFit90max$ui, qn=0.9, filename =
↪  "tmp/LVmaxtemp.pdf")
```

The 10-th percentile of the minimum daily temperature increased by almost 9 degrees ($\hat{\beta}_1 = 0.174$), while the 90-th percentile of the maximum daily temperature increased by only 1 degree, ($\hat{\beta}_1 = 0.0194$). This means that the 10% 'cooler than usual for the time of year' days were quite warmer in 2010 than in 1960, but there was no significant change in the 10% 'warmer than usual for the time of year' days (the 90th percentile.)

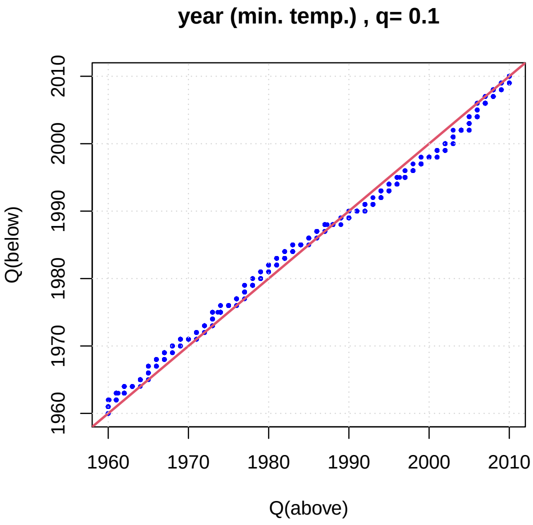
Figure 4 shows the diagnostic plot for the two responses. Both look excellent, and in fact, this is the case for all quantiles for both the maximum and minimum temperatures.

We can also use the generalized additive model (gam), as in the following example. The results in this case is similar to the mixed-model above, but gam takes considerably longer to run.

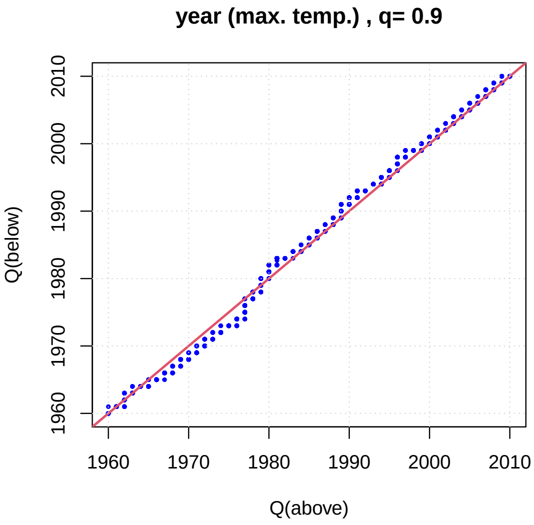
```r
#  gam10fit <- QREM(gam, TMIN ~ s(yday,4) + year + month, data=dat, qn=0.1)
```

References

[1] Bar, H. Y., Booth, J. G., and Wells, M. T. (2020). A Scalable Empirical Bayes Approach to Variable Selection in Generalized Linear Models. *Journal of Computational and Graphical Statistics*, **0**(0), 1–12.



(a) 10th percentile of the minimum temperature



(b) 90th percentile of the maximum temperature

Figure 4: Diagnostic plot for the Las Vegas temperature data.