

SOME EXERCISES ON QUANTILE REGRESSION

ROGER KOENKER

INTRODUCTION

These exercises are intended to provide an introduction to quantile regression computing and illustrate some econometric applications of quantile regression methods. For purposes of the course my intention would be to encourage all students to do the first exercise, which gives an overview of the quantile regression software in R in the context of an elementary bivariate Engel curve example. The remaining exercises are more open ended. I would like students to choose one of these exercises according to their own special interests. Given the brief duration of the course, it is probably unrealistic to expect polished answers to these questions at the end of the course, but I would be happy to get responses via email should you choose to continue working on them after the course is finished.

A Word on Software. There is now some quantile regression functionality in most statistical software systems. Not surprisingly, I have a strong preference for the implementation provide by the `quantreg` package of R, since I've devoted a considerable amount of effort to writing it. R is a dialect of John Chambers's S language and provides a very general, very elegant environment for data analysis and statistical research. It is fair to say that R is now the vehicle of choice within the statistical computing community. It remains to be seen whether it can make serious inroads into econometrics, but I firmly believe that it is a worthwhile investment for the younger cohorts of econometricians. R is public domain software and can be freely downloaded from the CRAN website. There is extensive documentation also available from CRAN under the heading manuals. For unix based systems it is usual to download R in source form, but it is also available in binary form for most common operating systems. There are several excellent introductions to R available in published form, in addition to the Introduction to R available in pdf from the CRAN website. I would particularly recommend Dalgaard (2002) and Venables and Ripley (2002). On the CRAN website there are also, under the heading "contributed", introductions to R in Danish, French, German, Spanish Italian, and a variety of other languages all of which can be freely downloaded in pdf.

For purposes of this course a minimal knowledge of R will suffice. R will be available on lab machines and should appear as an icon on the desktop of the laboratory machines. Clicking the icon should produce a window in which R will be running. To quit R, you may type `q()`, you will be prompted to answer whether you want to save the objects that were created during the session; responding "yes" will save the session objects into a file

Version: May 22, 2017. Some of these exercises were first developed for a short course given under the auspices of CEMMAP at UCL, in 2003, and have been expanded for subsequent short courses.

called `.RData`, responding “no” will simply quit without saving. Online help is provided in two modes: if you know what you are looking for, you can type, for example `?rq` and you will get a description of the `rq` command, alternatively you can type `help.start()` and a browser help window should pop up and you can type more general key words or phrases to search for functionality.

R is intended to be a convenient interactive language and you can do many things on the fly by just typing commands into the R console, or even by pointing and clicking at one of the GUI interfaces, but I find that it is often preferable to save R commands into a file and execute a group of commands – this encourages a more reproducible style of research – and can be easily done using the `source("commands.R")` command. Saving output is a bit more complicated since there are many forms of output, graphics are usually saved in either postscript or pdf form, and tables can be saved in latex format for subsequent inclusion in documents. Together with Achim Zeileis, U. of Innsbruck, I’ve written a paper in *J. of Applied Econometrics* on reproducible research strategies that describes some of these things in more detail. The paper and some other ranting and raving about reproducibility are also available from my homepage.

PROBLEM 1: A FAMILY OF ENGEL CURVES

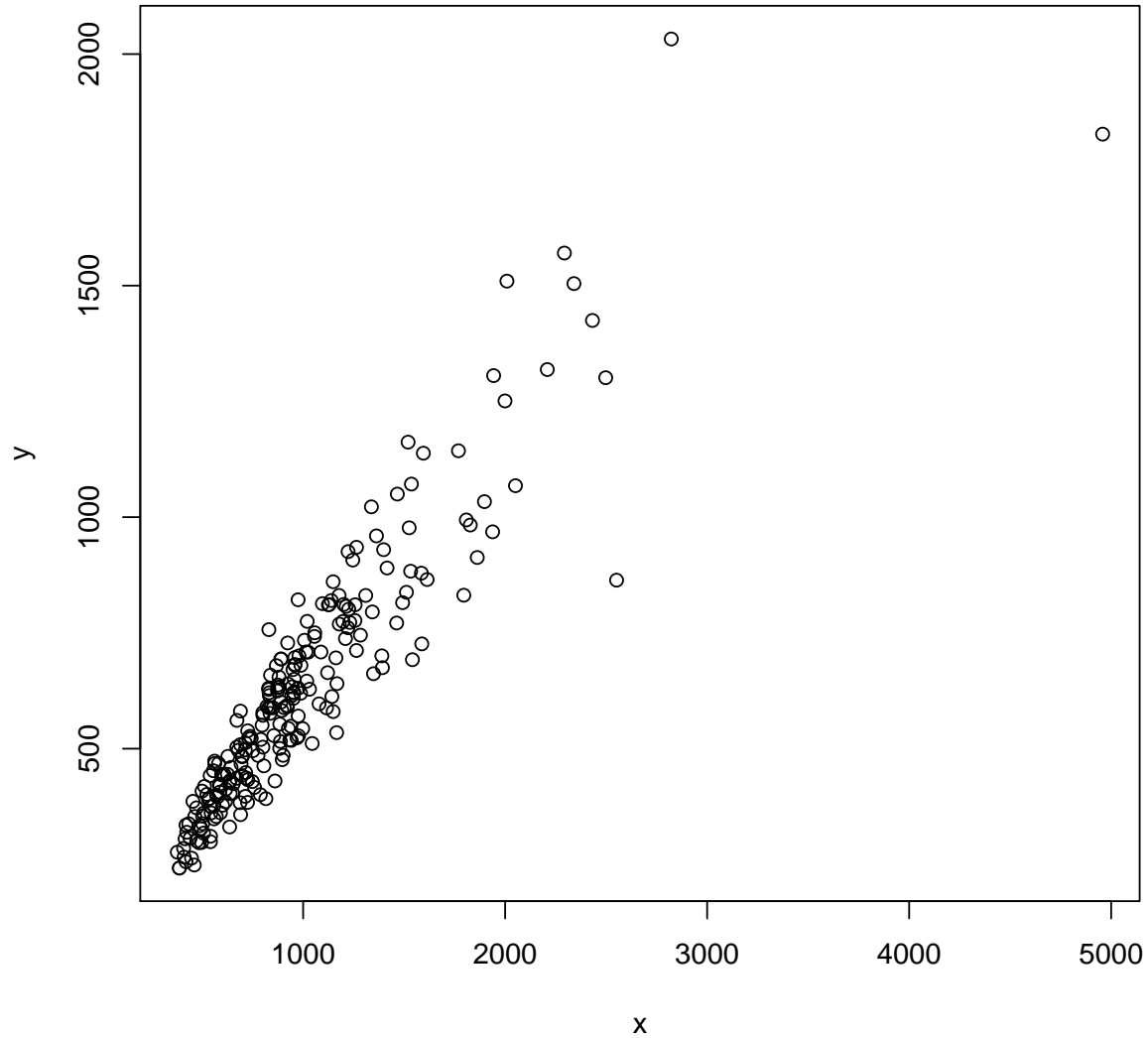
This is a simple bivariate linear quantile regression exercise designed to explore some basic features of the `quantreg` software in R. The data consists of observations on household food expenditure and household income of 235 working class Belgian families taken from the well-known study of Ernst Engel (1857).

1. Read the data. The data can be downloaded from the website specified in class. You will see that it has a conventional ascii format with a header line indicating the variable names, and 235 lines of data, one per household. This can be read in R by the command

```
url <- "http://www.econ.uiuc.edu/~roger/courses/Copenhagen/data/engel.data"
d <- read.table(file = url, header=TRUE) #data is now a matrix "d"
attach(d) # makes the individual variables accessible by name.
```

2. Plot the data. After the `attach` command the data is available using the names in the header, so we can plot the scatter diagram as:

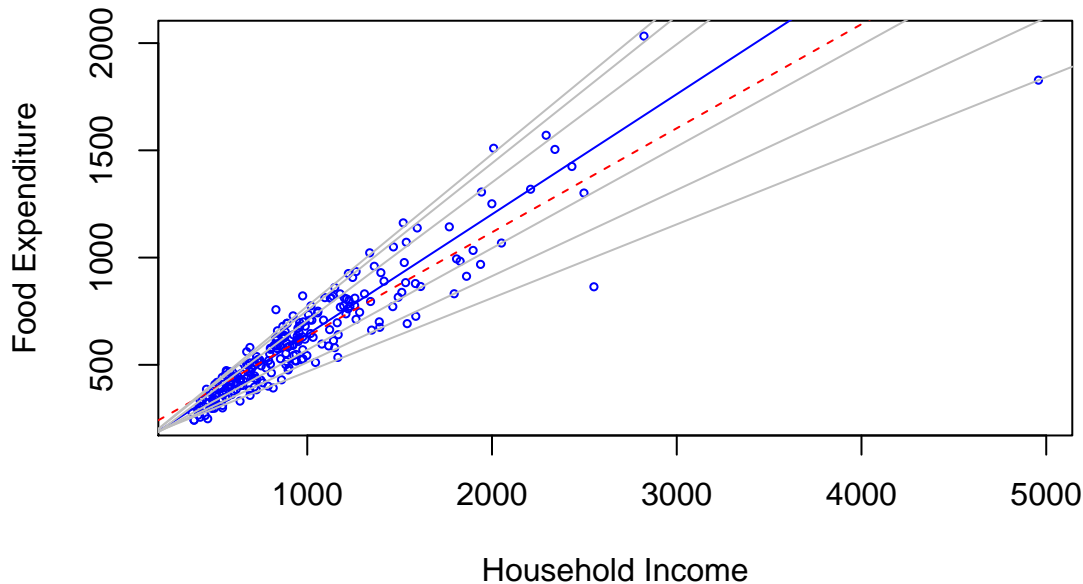
```
plot(x,y)
```



3. Replot with some better axis labels and superimpose some quantile regression lines on the scatter plot.

```
require(quantreg)
plot(x,y,cex=.25,type="n",xlab="Household Income", ylab="Food Expenditure")
points(x,y,cex=.5,col="blue")
abline(rq(y~x,tau=.5),col="blue")
abline(lm(y~x),lty=2,col="red") #the dreaded ols line
taus <- c(.05,.1,.25,.75,.90,.95)
```

```
f <- rq(y ~ x, tau = taus)
for( i in 1:length(taus)){
  abline(coef(f)[,i],col="gray")
}
```



Note that you have to load the `quantreg` package before invoking the `rq()` command. Careful inspection of the plot reveals that the ols fit is severely biased at low incomes due to a few outliers. The plot command has a lot of options to fine tune the plot. There is a convenient looping structure, but beware that it can be slow in some applications. In `rq()` there are also many options: the first argument is a “formula” that specifies the model that is desired, in this case we want to fit the simple bivariate linear model so it is just $y \sim x$ if we had two covariates we could say, e.g. $y \sim x+z$.

4. If we wanted to see all the distinct quantile regression solutions for this example we could specify a `tau` outside the range $[0,1]$, e.g.

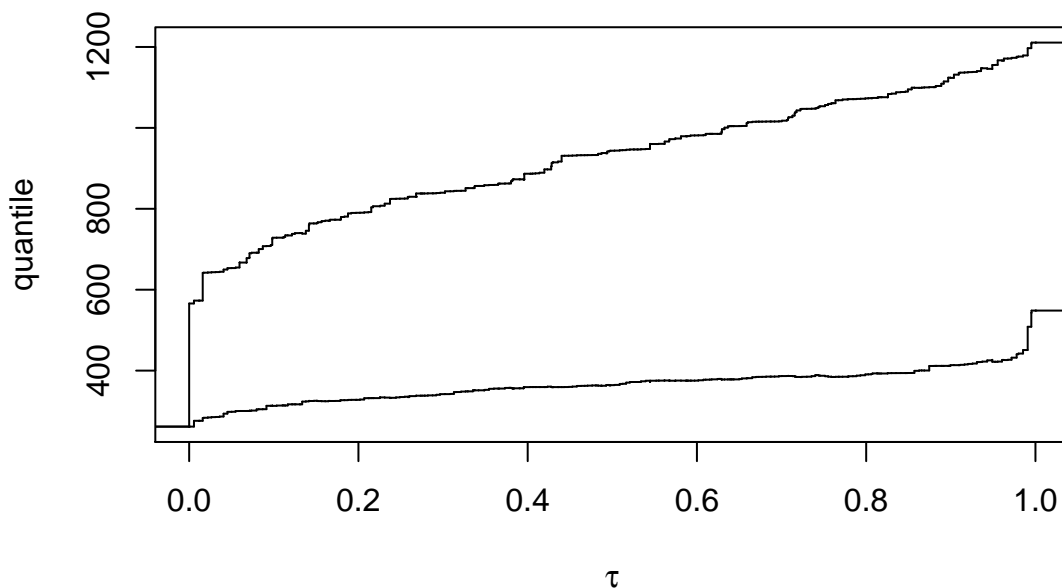
```
z <- rq(y ~ x, tau=-1)
```

Now if you look at components of the structure `z` that are returned by the command, you can see for example the primal solution in `z$sol`, and the dual solution in `z$dsol`. In interactive mode just typing the name of some R object causes the program to print the object in some more or less easily intelligible manner. Now, if you want to estimate the conditional quantile function of y at a specific value of x and plot it you can do something like this:

```

x.poor <- quantile(x,.1) #Poor is defined as the .1 quantile of income
x.rich <- quantile(x,.9) #Rich is defined as the .9 quantile of income
ps <- z$sol[1,]
qsp <- c(c(1,x.poor)%*%z$sol[4:5,])
qsr <- c(c(1,x.rich)%*%z$sol[4:5,])
#now plot the two quantile functions to compare
plot(c(ps,ps),c(qsp,qsr),type="n",xlab=expression(tau),ylab="quantile")
plot(stepfun(ps,c(qsp[1],qsp)),do.points=FALSE,add=TRUE)
plot(stepfun(ps,c(qsr[1],qsr)),do.points=FALSE,add=TRUE)

```



A nice feature of R is that documentation of functions usually includes some examples of their usage. These examples can be “run” by simply typing `example(SomeFunctionName)`, so for example when you type `example(rq)` you get a plot somewhat like the one you have just done “by hand.” In a second plot you get a pair of coefficient plots that depict the estimate intercept and slope coefficients as a function of τ and provide a confidence band. More on this later. If you look carefully at the code being executing by in these examples you will see that you didn’t need to download the data from the url specified, the Engel data is available directly from the **quantreg** package using the statement `data(engel)`. But it is often handy to be able to download data from the web. There are quite a lot of tools for handling web data sources, but this is another story entirely.

If you look carefully at the plots of the two estimated quantile functions that you made you will see minor violations of the expected monotonicity of these functions. This may

or may not be regarded as a mortal sin, depending on your religious convictions. One way to deal with this, recently suggested by Chernozhukov et al. (2009) is to “rearrange” the estimated functions. See the the documentation for this function using the usual strategy: `?rearrange`. To see an example of how this works try typing: `example(rearrange)`.

5. Now let’s consider some formal testing. For starters suppose we just estimate two quartile fits and look at the default output:

```
fit.25 <- rq(y~x,tau=.25)
summary(fit.25)
##
## Call: rq(formula = y ~ x, tau = 0.25)
##
## tau: [1] 0.25
##
## Coefficients:
##           coefficients lower bd upper bd
## (Intercept)  95.48         73.79  120.10
## x              0.47           0.42   0.49
fit.75 <- rq(y~x,tau=.75)
summary(fit.75)
##
## Call: rq(formula = y ~ x, tau = 0.75)
##
## tau: [1] 0.75
##
## Coefficients:
##           coefficients lower bd upper bd
## (Intercept)  62.40         32.74  107.31
## x              0.64           0.58   0.69
```

By default the confidence intervals that are produced use the rank inversion method. This is fine for judging whether covariates are significant at particular quantiles but suppose that we wanted to test that the slopes were the same at the two quantiles? This is done with the `anova` command as follows:

```
anova(fit.25,fit.75)
## Quantile Regression Analysis of Deviance Table
##
## Model: y ~ x
## Joint Test of Equality of Slopes: tau in { 0.25 0.75 }
##
##   Df Resid Df F value Pr(>F)
## 1 1      469    30.9 4.6e-08
```

This is an example of a general class of tests proposed in Koenker and Bassett (1982). It is instructive to look at the code for the command `anova.rq` to see how this test is carried out. The Wald approach is used and the asymptotic covariance matrix is estimated using the approach of Hendricks and Koenker (1991). It also illustrates a general syntax for testing in R adapted to the QR situation. If you have two models that are nested, with fits say `f0` and `f1`, then `anova(f0,f1)` should test whether the restricted model is correct. One needs to be careful however to check that the hypothesis that is intended, is really the one that the `anova` command understands, see `?anova.rq` for further details on the QR version of this. If you have more than two quantiles and want to do a joint test that all the slope coefficients are the same at all the quantiles you can use `anova(ft1,ft2,ft3,ft4)`.

In very large problems the rank inversion approach to confidence intervals is quite slow, and it is better to use another method. There are several choices. By default the computational method employs a variant of the Barrodale and Roberts (simplex-like) algorithm, for problems with sample size greater than about 5000 it is preferable to use interior point methods by using the `method="fn"`, flag in the call to `rq`. When this "Frisch-Newton" version of the algorithm is used, rank test confidence intervals are not provided by summary instead a form of the Wald test is returned. Various options can be specified to produce various estimates of the standard errors as described below. These Wald forms of estimating standard errors are also possible to achieve with the default `method="br"` setting by adding for example the flag `se=nid`. Details of the algorithms are provided in Koenker and d'Órey (1987), Koenker and d'Órey (1993), for the "BR" method and Portnoy and Koenker (1997) for the "Frisch-Newton" method.

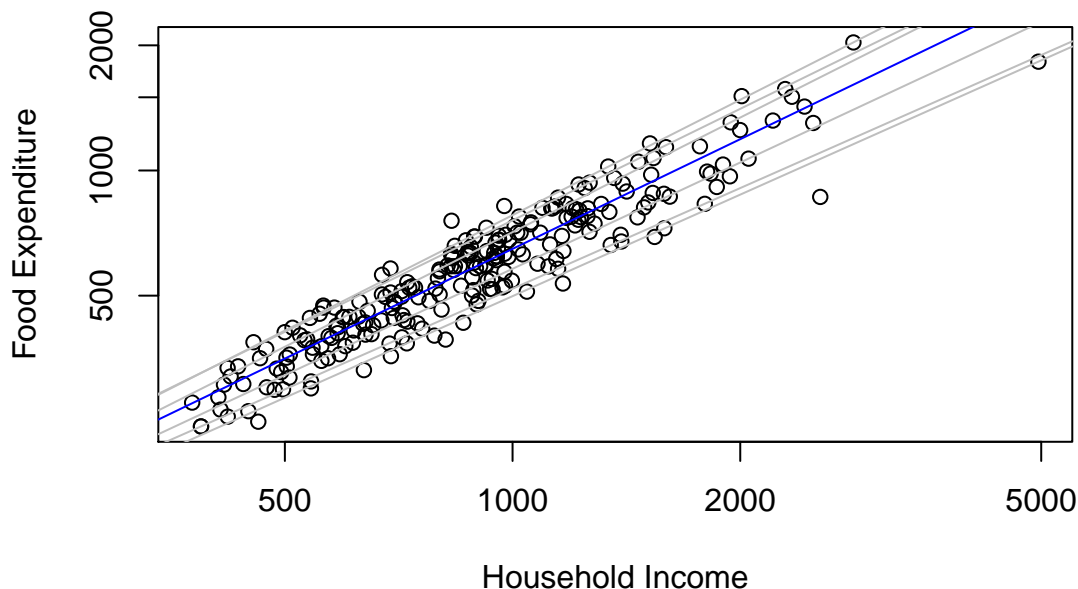
```
fit <- rq(y~x,tau=.27,method="fn")
summary(fit)
##
## Call: rq(formula = y ~ x, tau = 0.27, method = "fn")
##
## tau: [1] 0.27
##
## Coefficients:
##           coefficients lower bd upper bd
## (Intercept)  94.19         81.53  127.51
## x              0.48          0.43   0.50
```

Standard inference results are obtained by calling `summary`, e.g. by default `summary` produces estimates of the asymptotic covariance matrix based on the approach described in Hendricks and Koenker (1991), an alternative approach suggested by Powell (1989) can be obtained by specifying `se="ker"`. There are further details and options regarding bandwidth and controlling the nature of what is returned by the `summary` command, see `?summary.rq` for these details.

At this point it would be useful to compare and contrast the various estimation and inference options that are available. Try estimating the simple model used above with

both the `method = "br")` and `method = "fn")` choices, and then compare some of the `se` options in `summary.rq`.

```
plot(x,y,log="xy",xlab="Household Income", ylab="Food Expenditure")
taus <- c(.05,.1,.25,.75,.90,.95)
abline(rq(log10(y)~log10(x),tau=.5),col="blue")
for( i in 1:length(taus)){
  abline(rq(log10(y)~log10(x),tau=taus[i]),col="gray")
}
```



6. The magic of logarithms. Thus far we have considered Engel functions that are linear in form, and the scatter as well as the QR testing has revealed a strong tendency for the dispersion of food expenditure to increase with household income. This is a particularly common form of heteroscedasticity. If one looks more carefully at the fitting, one sees interesting departures from symmetry that would not be likely to be revealed by the typical textbook testing for heteroscedasticity, however. One common remedy for symptoms like this would be to reformulate the model in log linear terms. It is interesting to compare what happens after the log transformation with what we have already seen. Consider the following plot: Note that the flag `log="xy"` produces a plot with log-log axes, and for convenience of axis labeling these logarithms are base 10, so the subsequent fitting is also specified as base 10 logs for plotting purposes, even though base 10 logarithms are *unnatural* and would never be used in reporting numerical results. This looks much more like a classical iid error regression model, although again some departure from symmetry

is visible. An interesting exercise is to conduct some formal testing for departures from the iid assumption of the type already considered above. This is left as an exercise for the reader.

PROBLEM 2: NONPARAMETRIC QUANTILE REGRESSION

Nonparametric quantile regression is most easily considered within a locally polynomial framework. Locally linear fitting can be carried out by the following function, provided in the `quantreg` package:

```
"lprq" <-
function(x, y, h, m=50 , tau=.5)
{
  xx <- seq(min(x),max(x),length=m)
  fv <- xx
  der <- xx
  for(i in 1:length(xx)) {
    z <- x - xx[i]
    wx <- dnorm(z/h)
    r <- rq(y~z, weights=wx, tau=tau)
    fv[i] <- r$coef[1.]
    der[i] <- r$coef[2.]
  }
  list(xx = xx, fv = fv, der = der)
}
```

If you read through the function carefully you will see that it is just a matter of computing a quantile regression fit at each of m equally spaced x -values over the support of the observed x points. The function value estimates are returned as `fv` and the first derivative estimates at the m points are returned as `der`. As usual you can specify τ , but now you also need to specify a bandwidth `h`.

1. Begin by exploring the effect of the `h` and `tau` arguments for fitting the motorcycle data. Note that fitting derivatives requires larger bandwidth and larger sample size to achieve the same precision obtainable by function fitting. You are encouraged to substitute a more economic data set for the ubiquitous motorcycle data, its only advantage in the current context is that you can easily find examples to compare in the nonparametric regression literature.

2. Adapt `lprq` so that it does locally quadratic rather than linear fitting and compare performance.

3. Another general strategy for nonparametric quantile regression that is relatively simple to adapt to R uses regression splines. The function `bs()` in the package `splines` gives a very flexible way to construct B-spline basis expansions. For example you can fit a model like this:

```
require(splines)
## Loading required package: splines
url <- "http://www.econ.uiuc.edu/~roger/courses/Copenhagen/data/motorcycle.data"
d <- read.table(file = url, header=TRUE)
attach(d)
fit <- rq(accel ~ bs(time,df=5),tau=.33)
```

This fits a piecewise cubic polynomial with knots (breakpoints in the third derivative) at quintiles of the x 's. You can also explicitly specify the knot sequence and the order of the spline. One advantage of this approach is that it is very easy to add a partially linear model component. So if there is another covariate, say z , we can add a parametric component like this:

```
fit <- rq(y~bs(x,df=5)+z,tau=.33)
```

This avoids complications of backfitting when using kernel methods for partially linear models. Compare some fitting using the spline approach with that obtained with the local polynomial kernel approach.

4. Yet another even more appealing approach to univariate nonparametric smoothing involves penalty methods as described for example in Koenker et al. (1994) In recent work, Koenker and Mizera (2004), this approach has been extended to bivariate nonparametric regression, and more recently to a general class of additive models. Again, partially linear models are easily adapted, and there are easy ways to impose monotonicity and convexity on the fitted functions. In large problems it is essential to take advantage of the sparsity of the linear algebra. This is now feasible using special versions of the interior point algorithm for quantile regression and the SparseM package, Koenker and Ng (2003). The paper Koenker (2011) describes some recent developments of inference apparatus for these models. Further development of these methods would be aided by some additional experience with real data. An important feature of these additive models is that it is possible to impose monotonicity and/or convexity/concavity on the individual components. There are also relatively new methods for doing inference and prediction as well as plotting. As usual you can experiment with these methods by trying the `example()` function on methods like `summary.rqss`, `plot.rqss`, and `predict.rqss`. But more interesting would be to try new examples based on real data.

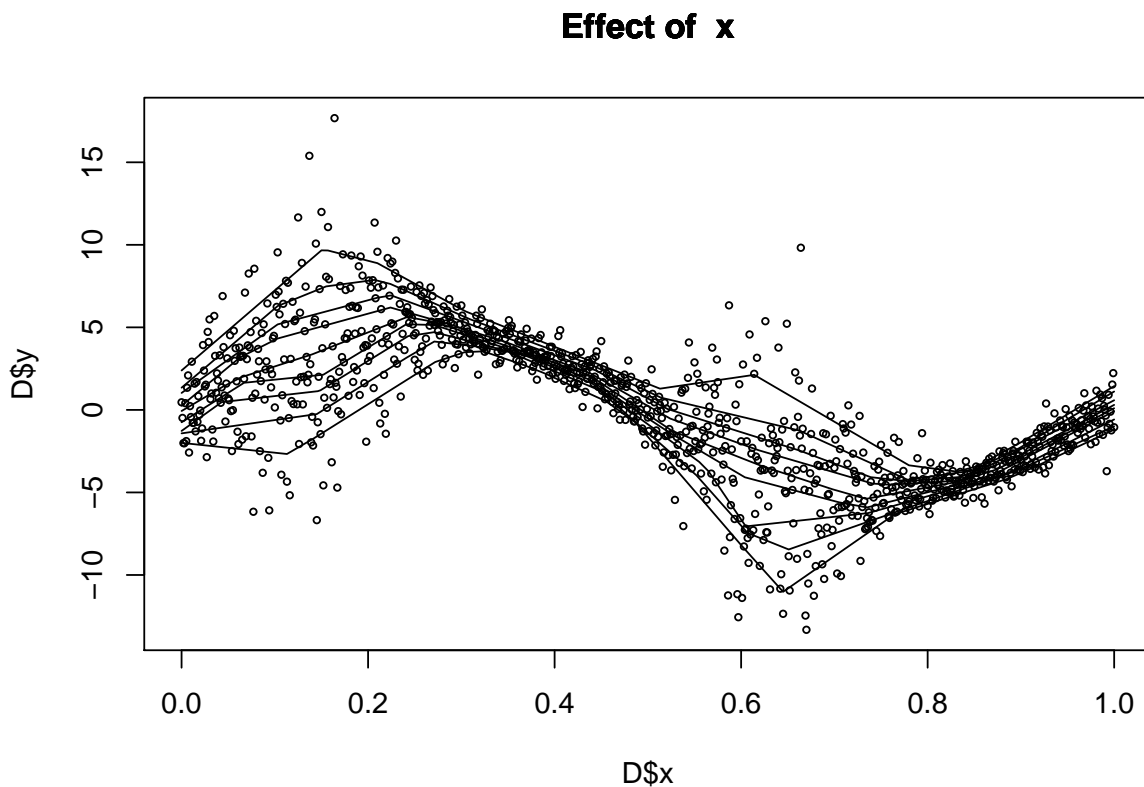
As a toy example consider the following recent simulation exercise from Polson and Scott (2015)

```
dgp <- function(n) {
  x <- 0:n/n
  y <- rnorm(n+1, 5 * sin(2 * pi * x), 0.5 + exp(1.5 * sin(4 * pi * x)))
  data.frame(x = x, y = y)
}
D <- dgp(1000)
```

```

plot(D$x, D$y, cex = .5)
taus <- 1:9/10
for(i in 1:length(taus))
  plot(rqss(y ~ qss(x, lambda = 1/10), tau = taus[i], data = D),
       rug = FALSE, add = TRUE)
## Warning in rq.fit.sfn(x, y, tau = tau, rhs = rhs, control = control, ...):

```



It is curious that the estimation strategy of Polson and Scott, which is based on a proximal gradient descent method, produces a smooth solution to this total variation penalized problem when the form of the solution can be easily shown to be piecewise linear. This “feature” seems to be entirely due to premature stopping of the iterations of the algorithm. A pervasive difficulty in such problems involves the choice of the smoothing parameter, λ . This is exacerbated here since it would appear that different λ 's might be needed for different τ 's. Experiment a bit with this choice, and try to suggest a reasonable criterion for choosing λ .

PROBLEM 3: QUANTILE REGRESSION SURVIVAL ANALYSIS

Quantile regression as proven to be a particularly attractive approach for univariate survival analysis (aka duration modeling). The classical accelerated failure time model

$$\log(T_i) = x_i^\top \beta + u_i$$

with iid errors u_i , can be easily extended to consider,

$$(1) \quad Q_{\log(T_i)}(\tau|x_i) = x_i^\top \beta(\tau),$$

yielding a flexible, yet parametrically parsimonious, approach.

In this problem you are asked to explore such models in the context of the Pennsylvania reemployment bonus experiment conducted in 1988-89. In this period new claimants for unemployment insurance were randomized into one of several treatment groups or a control group. Control participants abided by the usual rules of the unemployment insurance system; treatment participants were offered a cash bonus to be awarded if the claimant was certifiably reemployed within a specified qualification period. For simplicity we will focus on only one of the treatment groups, those offered a bonus of 6 times their weekly benefit provided reemployment was established within 12 weeks. For this group the bonus averaged about \$ 1000 for those collecting it. The data will be available in the form of an R data set called `Penn46.data` in the same directory as we have indicated for the prior datasets. This can be read into R using the same procedure as was used for the Engel data. For a more detailed analysis incorporating the other treatments, see Koenker and Biliias (2001). See Koenker and Xiao (2002) for further details on approaches to inference for these models.

In this application interest naturally focuses on the effect of the binary, randomized treatment. How does the bonus influence the distribution of the duration of unemployment? The Lehmann quantile treatment effect (QTE) is a natural object of empirical attention.

1. Explore some specifications of the QR model (1) and compare to fitting the Cox proportional hazard specification. See `require(survival)` for functions to estimate the corresponding Cox models. Note that covariate effects in the Cox models are necessarily scalar in nature, so for example the treatment effect must either increase, or decrease unemployment durations over the whole range of the distribution, but it cannot decrease durations in the lower tail and increase them in the upper tail – unless the model is specified with distinct baseline hazard functions for the two groups. See Koenker and Geling (2001) for some further details on the relationship between the QR survival model and the Cox model.

2. Explore some formal inference options to try to narrow the field of interesting specifications. See for example the discussion in Koenker and Xiao (2002) on tests based on the whole QR process.

PROBLEM 4: DISTRIBUTIONAL REGRESSION

Direct estimation of conditional distribution functions has been recently proposed as an alternative to quantile regression for some applications. There are a variety of interpretations of this approach, but I have begun to explore one such alternative in a very brief note available from the class webpage as `dreg.pdf`. It provides an explicit comparison of performance both from an asymptotic viewpoint and some simulations. It is rather difficult, it seems, to do general asymptotic efficiency comparisons, but the simulations suggest a modest advantage of QR over DR. I would be very interested in any further work along these lines that would help to illuminate the advantages and disadvantages of the two approaches. Of particular interest, given the results so far, would be situations in which DF performs better than QR. As an aid to getting started with such a question, I've also posted the code for the simulations in the class problems directory. This code follows my usual simulation protocol: there are two files `sim1.R` and `sim2.R` that do the simulations, and two other files `sim1a.R` and `sim2a.R` that produce the two tables that appear in the paper.

PROBLEM 5: QUANTILE AUTOREGRESSION

Consider a simple linear QAR model,

$$y_t = \alpha_1(u_t)y_{t-1} + \alpha_0(u_t) \quad t = 0, 1, \dots, T$$

where u_t is iid $U[0, 1]$. Suppose that $\alpha_1(u) = 0.85 + 0.25u$ and $\alpha_0(u) = \Phi^{-1}(u)$ with Φ denoting the standard normal distribution function. Simulate a realization of this process with $T = 1000$ and estimate and plot the QAR coefficients, comparing them to the usual OLS estimates.

Verify whether or not the process is stationary. In your realization of the process check to see whether y_{t-1} stays in the region for which the conditional quantile function of y_t is monotone. What is the usual OLS estimate of the AR(1) model estimating in this case? Check the residuals from the OLS fit to see if they exhibit any suspicious features that would reveal what is unusual here.

PROBLEM 6: PORTFOLIO CHOICE

This problem deals with the “pessimistic portfolio allocation” proposed in Bassett et al. (2004). The paper employs a highly artificial example. Your task, should you decide to accept it, is to produce a more realistic example using real data. Software implementing the methods of the paper is available in `quantreg` using the function `qrisk`. The function `qrisk` in this package computes optimal portfolio weights based on a matrix of observed, or simulated, asset returns using a specified form of pessimistic Choquet preferences.

PROBLEM 7: INEQUALITY DECOMPOSITION

The extensive literature on the measurement of inequality has devoted considerable attention to the question of how to decompose changes in measurements of inequality. If we observe increases in the Gini coefficient in a particular region over some sample period,

can we attribute these changes in some way to underlying changes in covariates, or to changes in the effects of these covariates? QR offers a convenient general approach to this question. Suppose that we have estimated a garden variety wage equation model in QR form,

$$(2) \quad Q_{\log y}(\tau|x) = x^\top \beta(\tau),$$

and we would like to compute a conditional Gini coefficient.

Recall that the Lorenz function of a univariate distribution with quantile function, Q , is given by,

$$\lambda(t) = \mu^{-1} \int_0^t Q(s) ds$$

where $\mu = \int_0^1 Q(s) ds$ is the mean of the distribution. The Gini coefficient is simply twice the area between $\lambda(t)$ and the 45 degree line,

$$\gamma = 1 - 2 \int_0^1 \lambda(t) dt.$$

1. Given the linear decomposition of the conditional quantile function in (2) and the fact that the Gini coefficient is a linear functional of the quantile function, formulate a conditional Gini decomposition for log wages, and interpret it.

2. Over time we may wish to “explain” changes in *the* Gini coefficient by considering changes in the wage structure – which we can interpret as $\beta(\tau)$ in (2) – and changes in the characteristics of the population – which are captured by the evolution of the distribution of x . This way of thinking enables us to consider thought experiments such as, “How would Gini have evolved if the wage structure were fixed at some initial condition, but population characteristics changed according to some specified pattern, historical or otherwise”. Or alternatively, suppose that we fix population characteristics and consider the evolution of the the conditional components of Gini as $\beta_t(\tau)$ changes over time. Decompositions of this type have been considered in recent work of Machado and Mata (2005). The Gini decomposition has also been recently considered by Aaberge et al. (2006) I would love to see a further applications along these lines.

REFERENCES

- Aaberge R, Bjerre S, Doksum K. 2006. Modeling inequality and spread in multiple regression. *Lecture Notes-Monograph Series, 2nd Lehmann Symposium* **49**: 120–130.
- Bassett G, Koenker R, Kordas G. 2004. Pessimistic portfolio allocation and Choquet expected utility. *J. Financial Econometrics* **2**: 477–92.
- Chernozhukov V, Fernandez-Val I, Galichon A. 2009. Improving point and interval estimators of monotone functions by rearrangement. *Biometrika* **96**: 559–570.
- Dalgaard P. 2002. *Introductory Statistics with R*. Springer.
- Hendricks W, Koenker R. 1991. Hierarchical spline models for conditional quantiles and the demand for electricity. *J. of Am. Stat. Assoc.* **87**: 58–68.

- Koenker R. 2011. Additive models for quantile regression: Model selection and confidence band-aids. *Brazilian J. of Probability and Statistics* **25**: 239–262.
- Koenker R, Bassett G. 1982. Robust tests for heteroscedasticity based on regression quantiles. *Econometrica* **50**: 43–61.
- Koenker R, Biliias Y. 2001. Quantile regression for duration data: A reappraisal of the pennsylvania reemployment bonus experiments. *Empirical Economics* **26**: 199–220.
- Koenker R, d’Orey V. 1987. Computing regression quantiles. *Applied Statistics* **36**: 383–393.
- Koenker R, d’Orey V. 1993. A remark on computing regression quantiles. *Applied Statistics* **36**: 383–393.
- Koenker R, Geling O. 2001. Reappraising medfly longevity: A quantile regression survival analysis. *J. of Am. Stat. Assoc.* **96**: 458–468.
- Koenker R, Mizera I. 2004. Penalized triograms: Total variation regularization for bivariate smoothing. *Journal of the Royal Statistical Society: Series B* **66**: 145–163.
- Koenker R, Ng P. 2003. SparseM: A sparse linear algebra package for R. *J. Stat. Software* **8**: 1–9.
- Koenker R, Ng P, Portnoy S. 1994. Quantile smoothing splines. *Biometrika* **81**: 673–80.
- Koenker R, Xiao Z. 2002. Inference on the quantile regression process. *Econometrica* **70**: 1583–1612.
- Machado J, Mata J. 2005. Counterfactual decomposition of changes in wage distributions using quantile regression. *Journal of Applied Econometrics* **20**: 445–466.
- Polson NG, Scott JG. 2015. Mixtures, envelopes and hierarchical duality. *Journal of the Royal Statistical Society: Series B* Forthcoming.
- Portnoy S, Koenker R. 1997. The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators, with discussion. *Stat. Science* **12**: 279–300.
- Powell JL. 1989. Estimation of monotonic regression models under quantile restrictions. In Barnett W, Powell J, Tauchen G (eds.) *Nonparametric and Semiparametric Methods in Econometrics*. Cambridge U. Press: Cambridge.
- Venables W, Ripley B. 2002. *Modern Applied Statistics with S*. Springer.