QUANTILE SELECTION MODELS: AN R VIGNETTE

ROGER KOENKER

Abstract. An R implementation of an estimator of the the quantile selection model proposed recently by Arellano and Bonhomme (2017) is described. Method of moments estimation of the model's copula parameters, however, is contrasted with a somewhat unconventional profile likelihood approach.

1. Introduction

Arellano and Bonhomme (2017) have recently proposed a new approach to analysing sample selection effects in the context of a general quantile regression model, thereby extending the classical parametric selection methods of Heckman (1979). A central feature of their approach is the estimation of the parameters of a copula function that determines the dependence between the random components of the (latent) selection model and the observable outcome model. This note contrasts a profile likelihood alternative with the method of moments strategy for estimating the copula parameters proposed by Arellano and Bonhomme (2017). The possibility of likelihood based alternatives was already suggested by Arellano and Bonhomme (2017), so this can be considered a tentative first step in this direction.

2. A QRious Likelihood

Quantile regression, at least as it was originally conceived, posits a *local* statistical model focussed on estimating a single conditional quantile function while professing total indifference about the form of adjacent conditional quantile functions. However, when one writes,

$$Q_{Y|X}(\tau|x) = x^\top \beta(\tau) \quad \tau \in (0,1),$$

a global model for the entire conditional distribution of Y given X has been specified, so it is natural to ask: Can we compute a global likelihood value for fitted models of this form?

To achieve this dubious objective we obviously need estimates of the conditional density of Y at each observed setting of the conditioning covariate vector, X. Given estimates of

Version: January 27, 2017. Code and data to reproduce the results reported here will be available from the binary version of the **quantreg** package downloadable from a CRAN mirror near you. The R function `browseVignettes("quantreg")` should bring up a browser window with links to the pdf version of this document and the file with the R code that generates the computational results described.

the conditional quantile function at each $X = x_i$, this is just a matter of smoothing. In problems of moderate size all of the solutions to the problem,

$$\hat{\beta}(\tau) = \text{argmin}_b \sum_{i=1}^{n} \rho_\tau(y_i - x_i b),$$

can be very efficiently computed by parametric linear programming, and thus for any x we have $\hat{Q}(\tau|x)$ as a piecewise constant (CAGLAD) function on (0,1). In the **quantreg** package for R the invocation,

```
data(stackloss)
fit <- rq(stack.loss ~ stack.x, tau = -1)
fhat <- predict(fit, type = "fhat")
```

loads the infamous stackloss data, fits a model for the entire quantile regression process and computes a list of n conditional densities, one for each of the original observations. Smoothing is done with the function `akj`, which implements Silverman's well-known adaptive kernel density estimator. Given these conditional densities it is easy to define a function to compute the log likelihood:

```
logLik.rq.process <- function(fit){
    y <- model.response(model.frame(fit))
    fhat <- predict(fit, type = "fhat")
    fy <- mapply(function(f,y) f(y), fhat, y)
    sum(log(fy))
}
```

To illustrate we can compare the log likelihoods for the unconditional and conditional quantile models as follows,

```
f0 <- rq(stack.loss ~ 1, tau=-1)
f1 <- rq(stack.loss ~ stack.x, tau=-1)
l0 <- logLik(f0)
l1 <- logLik(f1)
```

which yields `l1 - l0` $= -34.446 - -70.736 = 36.289$. It remains to be seen if Professor Wilks ghost can be persuaded to divulge a limiting distribution theory for such log likelihood ratio statistics. It is not at all obvious how one might count degrees of freedom for such global quantile regression models. It is also evident that in applications with much larger sample sizes it is impractical, and obviously superfluous to compute *all* the distinct solutions of the QR process. Portnoy (1991) shows that the expected number of distinct solutions is of order $\mathcal{O}(n \log n)$. In what follows we adopt the pragmatic attitude that a grid of a few hundred $\tau \in (0, 1)$ is sufficient to produce a reasonable estimate of conditional densities provided a reasonable choice of the initial bandwidth for the pilot estimate of the Silverman procedure is employed.

It may be noted at this point that a fully efficient likelihood procedure would require that we weight the usual QR objective by estimates of the local conditional density. See Koenker (2005) Section 5.3.1. We reserve this (rather utopian) diversion for future exploration.

3. Estimation of the QR Selection Model

Leaving aside for the moment (or millennium) the distribution theory for tests based on such LRTs, our motivating application for the QR likelihood is estimation of the quantile regression selection model of Arellano and Bonhomme (2017). Their approach involves GMM estimation of a parametric copula model that captures the dependence between the random components of the latent selection model and the outcome model. The objective function involves evaluating indicator functions along a grid of τ's evaluated at "rotated" quantile regression estimates. It seemed worth exploring whether the foregoing likelihood approach might provide a more direct way to implement estimation of their model.

Our implementation of the Arellano and Bonhomme (2017) estimator exploits general features of the R protocol for fitting linear models. The model is specified as a generalized formula as introduced in Zeileis and Croissant (2010),

$$y|D \sim X|Z,$$

where y is the observable outcome variable, D is the binary selection indicator, X specifies the conditioning covariates of the outcome model, and the union of X and Z specifies the conditioning covariates of the selection model. The outer wrapper of the fitting functions looks like this:

```
rqsel <- function(formula, data, taus = 1:99/100, rhodom = c(-5,1),
        grid = 0, copula = frankCopula, link = "probit", rhometh = Liksel){
    # Generic Formula is y|D ~ X|Z
    Form <- Formula(formula)
    oform <- formula(Form, lhs = 1, rhs = 1)
    sform <- formula(Form, lhs = 2, rhs = 1:2, collapse = TRUE)
    v <- glm(sform, family = binomial(link = link), data = data)
    D <- model.response(model.frame(v))
    v <- v$fitted[D == 1]
    fit <- rq(oform, tau = taus, data = data, method = "fnb", eps = 1e-4)
    if(grid){
        rhos <- seq(rhodom[1], rhodom[2], length = grid)
        objs <- sapply(rhos,function(x, fit, copula, v)
             rhometh(x, fit, copula, v), fit = fit, copula=copula, v=v)
        return(list(x = rhos, y = objs))
    }
    else
        rhohat <- optimize(rhometh, rhodom, fit = fit, v = v,
          copula = copula)$minimum
    rq.fit.sel(rhohat, fit, copula, v)
```

```
}
```

The formula is first parsed into its outcome and selection pieces, and a binary response model is then estimated for the selection variable, D. A propensity score, v is then extracted and evaluated for all the selected observations, i.e., those with $D = 1$. In the next step a naive QR model is estimated that ignores the sample selection effect; the output of this step serves only as a repository for subsequent information for fitted objects that do account for the selection. Note that the fitting of the QR model automatically drops the $D = 0$ observations since their response variable is coded as "NA", i.e., missing. QR fitting is done on a grid of $\tau \in (0, 1)$, by default at the percentiles.

At this stage we are almost ready to optimize over the dependence parameter of the copula function that links the selection and outcome equations. We will restrict attention to settings with a scalar copula parameter, by default with the Frank copula, but this could be easily extended to more general parametric settings such as the generalized Frank specification of Arellano and Bonhomme (2017).[1]

We will consider two criteria for estimating the copula parameter: the method of moments criterion of Arellano and Bonhomme (2017), and the profile likelihood criterion described in the previous section as implemented in the functions `Momsel` and `Liksel` respectively.

```
Liksel <- function(rho, fit, copula, v) {
    f <- rq.fit.sel(rho, fit, copula, v)
    y <- model.response(model.frame(fit))
    fhats <- predict(f, type = "fhat")
    fy <- mapply(function(f,y) f(y), fhats, y)
    -sum(log(fy))
}
Momsel <- function(rho, fit, copula, v) {
    R <- rq.fit.sel(rho, fit, copula, v)$resid
    sum(apply(R,2,mean))^2
}
```

Both approaches rely on the novel QR estimation strategy introduced by Arellano and Bonhomme (2017) that modifies the conventional fixed τ weighting of the objective function with a observation specific $\hat{\tau}_i$ depending upon the estimated propensity score and the copula dependence relation,

$$\hat{\tau}_i = C(\tau, v_i; \rho)/v_i,$$

where C denotes the distribution function of the chosen copula function evaluated at the proposed τ, the propensity score, v_i of the ith observation and the trial value of the copula parameter, ρ. This is implemented in the function `rq.fit.sel`.

[1]This would simply entail replacing `optimize` by `optim`, and the interval `rhodom` by a vector of starting values.

```
rq.fit.sel <- function(rho, fit, copula, v){
    taus <- fit$tau
    fit$frho <- rho
    x <- model.matrix(terms(fit), model.frame(fit))
    y <- model.response(model.frame(fit))
    cop <- copula(rho)
    for(j in 1:length(taus)) {
        u <- pCopula(cbind(taus[j], v), cop)/v
        rhs <- t(x) %*% (1 - u)
        f <- rq.fit.fnb(x, y, taus[j], rhs = rhs, eps = 1e-4)
        fit$coefficients[,j] <- f$coef
        fit$residuals[,j] <- v * ((f$resid <= 0) - u)
    }
    fit
}
```

In the dual formulation of the QR problem that is typically used to construct algorithms, the introduction of this observation specific τ vector is trivially accommodated by changing the right hand side of the dual equality constraints. Instead of a scalar τ with dual constraint $X^\top a = (1-\tau)X^\top 1$, we can write, $X^\top a = X^\top(1-u)$, where u denotes the new τ specific vector. Such problems are efficiently solved with the so-called Frisch-Newton linear programming algorithm invoked by `rq.fit.fnb`, and described in detail in Portnoy and Koenker (1997). This is a fortran implementation of the same algorithm used by Arellano and Bonhomme (2017), in Matlab.

Estimated coefficients in our repository `fit` object are replaced by the coefficients of this new solution for each τ on our grid, and residuals are replaced by a vector of scaled and centered residual signs needed to evaluate the moment criterion. This new fitted object is passed either to `Momsel` or `Liksel` which evaluate the copula fitting criteria. In the former case, estimated conditional densities are evaluated at the observed response, logged and summed. In the latter case we simply sum mean discrepancies from the hypothesized moment condition and square the sum.

4. A Replication Exercise

To illustrate the foregoing approach we reconsider the model posited in Arellano and Bonhomme (2017).[2] The `grid` argument in `rqsel` can be used to evaluate the criterion function on a grid of ρ's equally spaced on the domain `rhodom`. The number of evaluation points in this interval is set by specifying `grid` as a positive integer. This option is primarily intended as an exploratory device for determining an appropriate domain for the

[2]The Appendix describes some code that transforms the data files distributed from `https://sites.google.com/site/stephanebonhommeresearch/` into R data frames that are more efficiently stored and loaded for R.

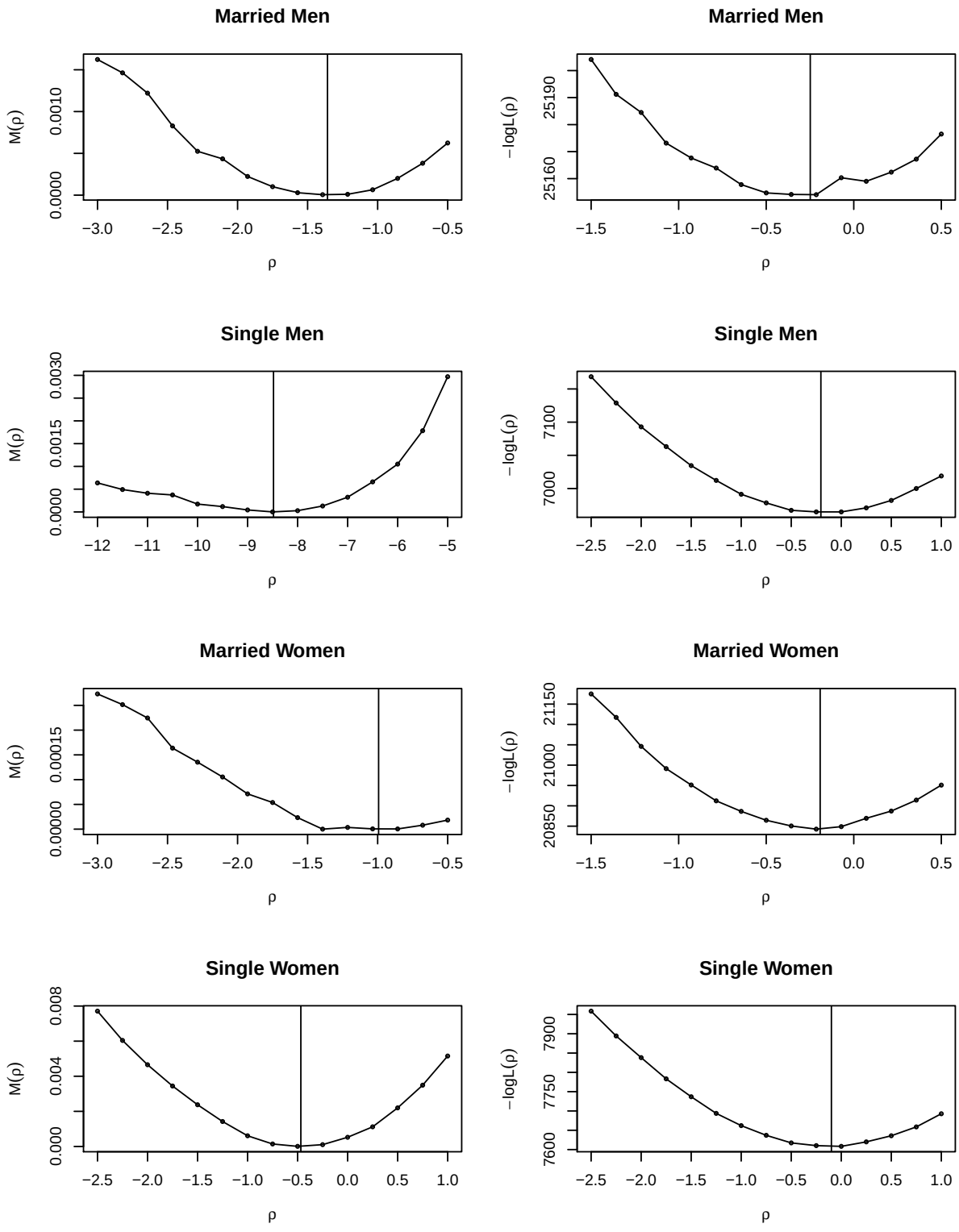
optimization for the copula parameter. We will illustrate with a plot of both criteria for all four subsamples, together with an point estimate of ρ based on optimization.

```
Mfit <- as.list(1:4)
Lfit <- as.list(1:4)
MForm <- lw | work ~ ed + age + region + trend + I(trend^2) + I(trend^3) +
    kids_d1 + kids_d2 + kids_d3 + kids_d4 + kids_d5 + kids_d6 |
    I(married * ben)
SForm <- lw | work ~ ed + age + region + trend + I(trend^2) + I(trend^3) +
    kids_d1 + kids_d2 + kids_d3 + kids_d4 + kids_d5 + kids_d6 |
    I((1-married) * ben)
load("M.Rda")
G <- G[G[,"married"] == 1,] # Married men
Mfit[[1]] <- rqsel(MForm, data = G, grid = 15, rhometh = Momsel,
                   rhodom = c(-3, -0.5))
Lfit[[1]] <- rqsel(MForm, data = G, grid = 15, rhometh = Liksel,
                   rhodom = c(-1.5, 0.5))
load("M.Rda")
G <- G[G[,"married"] == 0,] # Single men
Mfit[[2]] <- rqsel(SForm, data = G, grid = 15, rhometh = Momsel,
                   rhodom = c(-12, -5))
Lfit[[2]] <- rqsel(SForm, data = G, grid = 15, rhometh = Liksel,
                   rhodom = c(-2.5, 1))
load("F.Rda")
G <- G[G[,"married"] == 1,] # Married women
Mfit[[3]] <- rqsel(MForm, data = G, grid = 15, rhometh = Momsel,
                   rhodom = c(-3, -0.5))
Lfit[[3]] <- rqsel(MForm, data = G, grid = 15, rhometh = Liksel,
                   rhodom = c(-1.5, 0.5))
load("F.Rda")
G <- G[G[,"married"] == 0,] # Single women
Mfit[[4]] <- rqsel(SForm, data = G, grid = 15, rhometh = Momsel,
                   rhodom = c(-2.5, 1))
Lfit[[4]] <- rqsel(SForm, data = G, grid = 15, rhometh = Liksel,
                   rhodom = c(-2.5, 1))
```

```
Mf <- as.list(1:4)
Lf <- as.list(1:4)
MForm <- lw | work ~ ed + age + region + trend + I(trend^2) + I(trend^3) +
    kids_d1 + kids_d2 + kids_d3 + kids_d4 + kids_d5 + kids_d6 |
    I(married * ben)
SForm <- lw | work ~ ed + age + region + trend + I(trend^2) + I(trend^3) +
```

```
    kids_d1 + kids_d2 + kids_d3 + kids_d4 + kids_d5 + kids_d6 |
    I((1-married) * ben)
load("M.Rda")
G <- G[G,"married"] == 1,] # Married men
Mf[[1]] <- rqsel(MForm, data = G, rhometh = Momsel,
                 rhodom = c(-3, -0.5))
Lf[[1]] <- rqsel(MForm, data = G, rhometh = Liksel,
                 rhodom = c(-1.5, 0.5))
load("M.Rda")
G <- G[G,"married"] == 0,] # Single men
Mf[[2]] <- rqsel(SForm, data = G,  rhometh = Momsel,
                 rhodom = c(-12, -5))
Lf[[2]] <- rqsel(SForm, data = G, rhometh = Liksel,
                 rhodom = c(-2.5, 1))
load("F.Rda")
G <- G[G,"married"] == 1,] # Married women
Mf[[3]] <- rqsel(MForm, data = G, rhometh = Momsel,
                 rhodom = c(-3, -0.5))
Lf[[3]] <- rqsel(MForm, data = G, rhometh = Liksel,
                 rhodom = c(-1.5, 0.5))
load("F.Rda")
G <- G[G,"married"] == 0,] # Single women
Mf[[4]] <- rqsel(SForm, data = G, rhometh = Momsel,
                 rhodom = c(-2.5, 1))
Lf[[4]] <- rqsel(SForm, data = G, rhometh = Liksel,
                 rhodom = c(-2.5, 1))
```

```
par(mfrow = c(4,2))
status <- c("Single", "Married")
gender <- c("Men", "Women")
for(i in 1:4){
    main <- paste(status[1 + (i %% 2)],gender[1 + (i > 2)])
    plot(Mfit[[i]], xlab = expression(rho), ylab = expression(M(rho)),
        main = main, cex = 0.5)
    lines(Mfit[[i]])
    abline(v = Mf[[i]]$frho)
    plot(Lfit[[i]], xlab = expression(rho), ylab = expression(-logL(rho)),
        main = main, cex = 0.5)
    lines(Lfit[[i]])
    abline(v = Lf[[i]]$frho)
}
```


Married Men
M(ρ)
0.0000
0.0010
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
ρ
Married Men
−logL(ρ)
25160
25190
−1.5
−1.0
−0.5
0.0
0.5
ρ
Single Men
M(ρ)
0.0000
0.0015
0.0030
−12
−11
−10
−9
−8
−7
−6
−5
ρ
Single Men
−logL(ρ)
7000
7100
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
ρ
Married Women
M(ρ)
0.00000
0.00015
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
ρ
Married Women
−logL(ρ)
20850
21000
21150
−1.5
−1.0
−0.5
0.0
0.5
ρ
Single Women
M(ρ)
0.000
0.004
0.008
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
ρ
Single Women
−logL(ρ)
7600
7750
7900
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
ρ

```
SForm <- lw | work ~ ed + age + region + trend + I(trend^2) + I(trend^3) +
    kids_d1 + kids_d2 + kids_d3 + kids_d4 + kids_d5 + kids_d6 |
    I((1-married) * ben)
load("M.Rda")
G <- G[G[,"married"] == 0,] # Single men
Mc <- rqsel(SForm, data = G,  rhometh = Momsel,
                    link = "cauchit", rhodom = c(-8, -1))
```

The method of moments estimates of ρ are quite close to those obtained by Arellano and Bonhomme (2017), however, there are several puzzles that have yet to be resolved. Foremost among these is the fact that the estimated $\hat{\rho}$'s are consistently more negative for the method of moments criterion than for the log likelihood criterion. For single men this difference is quite substantial. Sensitivity to choices of copula and propensity score models is also of considerable interest. As a very small step in this direction, I tried replacing the probit link with the cauchit for the sample of single men. This had little impact on the likelihood estimate of ρ, but increased the method of moments estimate from -8.482 to -4.327.

Appendix A. Data Management

In this appendix we document the procedure employed to simplify the data sources required for the replication of the Arellano and Bonhomme (2017) results. Data from `https://sites.google.com/site/stephanebonhommeresearch/` takes the form of four comma separated value (csv) files. Only two of these are used as detailed in the code below. These files are distinguished by gender, but have substantial overlap in the variables they contain. These files collapsed into two files stored in R `save` format which is considerably compressed and more efficiently loaded. In the process we have also collapsed groups of binary indicator variables into single R factor variables. It may be noted that the omitted category of these factor variables is given a null name.

```
undummy <- function(A) { # Convert dummy variables to factor
    if(!all(A %in% c(0,1))) stop("not dummies")
    A <- as.matrix(A)
    a <- A %*% rep(1,ncol(A))
    if(sum(a) < length(a)) A <- cbind(1-a, A)
    as.factor(colnames(A)[A %*% 1:ncol(A)])
}
MakeRda <-function(){
Ddir <- "Bonhomme/Codes_for_replication/Data_files/"
files <- c("M", "F")
for(i in 1:2){
    file <- paste(Ddir, "datasub_", i, ".out", sep = "")
    G <- as.matrix(read.csv(file, header = TRUE))
    ed <- undummy(G[,6:7])
```

```
    trend <- G[,8]
    age <- undummy(G[,11:14])
    region <- undummy(G[,15:25])
    kids <- G[,26:31]
    ben <- G[,4] + G[,5]
    ofile <- paste(files[i],".Rda", sep = "")
    G <- data.frame(G[,1:5], ben, ed, age, region, trend, G[,26:31])
    save(G, file = ofile)
    }
}
```

References

Arellano M, Bonhomme S. 2017. Quantile selection models. *Econometrica* Forthcoming.

Heckman JJ. 1979. Sample selection bias as a specification error. *Econometrica* **47**: 153–161.

Koenker R. 2005. *Quantile Regression*. Cambridge U. Press.

Portnoy S. 1991. Asymptotic behavior of the number of regression quantile breakpoints. *SIAM J. Scientific and Statistical Computing* **12**: 867–883.

Portnoy S, Koenker R. 1997. The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators. *Statistical Science* **12**: 279–300.

Silverman BW. 1986. *Density estimation for statistics and data analysis*. Chapman & Hall.

Zeileis A, Croissant Y. 2010. Extended model formulas in R: Multiple parts and multiple responses. *Journal of Statistical Software* **34**: 1–13.
URL `http://www.jstatsoft.org/v34/i01/`