Nonparametric Series Quantile Regression in R: A Vignette*

Michael Lipsitz, Alexandre Belloni, Victor Chernozhukov,
and Iván Fernández-Val†

May 26, 2015

Abstract

Belloni, Chernozhukov, and Fernández-Val (2011) developed nonparametric quantile regression methods to estimate and make inference on conditional quantile models. The R package `quantreg.nonpar` implements these methods for partially linear quantile models. `quantreg.nonpar` obtains point estimates of the conditional quantile function and its derivatives based on series approximations to the nonparametric part of the model. It also provides pointwise and uniform confidence intervals over a region of covariate values and/or quantile indexes for the same functions using analytical and resampling methods. This vignette serves as an introduction to the package and displays basic functionality of the functions contained within.

1 Getting Started

To get started using the package `quantreg.nonpar` for the first time, first issue the command

```
install.packages("quantreg.nonpar")
```

into your R browser to install the package in your computer. Once it has been installed, you can then use the package `quantreg.nonpar` during any R session by simply issuing the command

```
library(quantreg.nonpar)
```

Now you are ready to use the functions and data sets contained in `quantreg.nonpar`. For general questions about the package, type

*We wish to thank Jim Ramsay for assistance with the fda package (Ramsay et al. (2013)), and Roger Koenker for sharing the data used in Koenker (2011). We gratefully acknowledge research support from the NSF.

†BU, Duke, MIT, and BU

```
help(package = "quantreg.nonpar")
```

to view the package help file, or for more questions about a specific function you can type `help(`*function.name*`)`. For example, try

```
help(npqr)
```

2 Nonparametric Series Quantile Regression

Suppose we model an outcome variable of interest, Y, as a function of a vector of observable covariates, X, and an unobserved variable, U, as

$$Y = Q(X, U),$$

where Q is strictly increasing in U and U and X are independent. With the normalization that $U \sim \text{Uniform}(0, 1)$, $Q(x, u)$ is the conditional u-quantile of Y given $X = x$, $Q_{Y|X}(u|x)$. The conditional quantile function can be approximated by a linear combination of series terms,

$$Q_{Y|X}(u|x) \approx Z(x)'\beta(u).$$

The vector $Z(x)$ includes tensor products of transformations of the elements of x such as powers, trigonometrics, indicators, or B-splines, and has a dimension, m, that grows with the sample size, n. The function $u \mapsto \beta(u)$ contains the quantile-specific coefficients of the approximation, where each $\beta(u)$ is defined as the coefficient of the quantile regression (QR) of Y on $Z(X)$ at the quantile u.

The coefficient vectors $\beta(u)$ are estimated using the QR estimator of Koenker and Bassett (1978). Let $\{(Y_i, X_i) : 1 \leq i \leq n\}$ be a random sample from (Y, X) and let $\hat{\beta}(u)$ be the QR estimator of $\beta(u)$, i.e.

$$\hat{\beta}(u) \in \arg\min_{\beta \in \mathbb{R}^m} \sum_{i=1}^{n} \rho_u(Y_i - X_i'\beta), u \in \mathcal{U} \subseteq (0, 1),$$

where $\rho_u(z) = (u - 1\{z < 0\})z$ is the check function. Belloni, Chernozhukov, and Fernández-Val (2011) obtained uniform strong approximations to the empirical series QR coefficient process of increasing dimension $\sqrt{n}(\widehat{\beta}(\cdot) - \beta(\cdot))$ based on:

1. a conditionally pivotal process
2. a gradient bootstrap process
3. a Gaussian process, and

4. a weighted bootstrap process.

Each of these approximations leads to a feasible inference method. The command `npqr` of the `quantreg.nonpar` package implements all these methods for the partially linear quantile model:

$$Q_{Y|X}(u|x) = g(w, u) + v'\gamma(u), \quad X = (W, V),$$

where W is typically the covariate of interest, V is a vector including other controls, and $g(w, u)$ is approximated by a linear combination of series terms $g(w, u) \approx Z(w)'\beta(u)$.

We illustrate the functionality of the package with an empirical application based on data from Koenker (2011) for childhood malnutrition in India, where we model the effect of a child's age and other covariates on the child's height. Here, Y is the child's height in centimeters; W is the child's age in months; U is the unobservable ranking of the child in the height distribution; and V is a vector of 22 controls. These controls include the mother's body mass index (BMI), the number of months the child was breastfed, and the mother's age (as well as the square of the previous three covariates); the mother's years of education and the father's years of education; dummy variables for the child's sex, whether the child was a single birth or multiple birth, whether or not the mother was unemployed, whether the mother's residence is urban or rural, and whether the mother has each of: electricity, a radio, a television, a refrigerator, a bicycle, a motorcycle, and a car; and factor variables for birth order of the child, the mother's religion and quintiles of wealth.

First, we load the data:

```
data<-india
```

Next, we construct the variables that will be used in the analysis. Note that the variable prefixes "c" and "m" refer to "child" and "mother". For each factor variable (`csex, ctwin, cbirthorder, munemployed, mreligion, mresidence, wealth, electricity, radio, television, refrigerator, bicycle, motorcycle,` and `car`), we generate a variable "`facvar`" which is the factor version of the variable "`var`". For each quadratic variable (`mbmi, breastfeeding,` and `mage`), we generate a variable "`varsq`" which is the variable squared. For example:

```
faccsex <- factor(csex)

mbmisq <- mbmi^2
```

We also construct the formula to be used for the parametric part of the model, $v'\gamma(u)$:

```
form.par <- cheight ~ mbmi + mbmisq + breastfeeding + breastfeedingsq + mage
+ magesq + medu + edupartner + faccsex + facctwin + faccbirthorder +
facmunemployed + facmreligion + facmresidence + facwealth + facelectricity +
facradio + factelevision + facrefrigerator + facbicycle + facmotorcycle +
faccar
```

Note that this formula does not contain a term for our variable of interest W; namely, the child's age. Let us now construct the nonparametric bases that will be used to estimate the effect of W, i.e., $g(w,u) \approx P(w)'\beta(u)$. For our base case, we construct a cubic B-spline basis with knots at the $\{0, 0.1, 0.2, ..., 0.9, 1\}$ quantiles of the observed values of child's age.

```
basis.bsp <- create.bspline.basis(breaks=quantile(cage, c(0:10)/10))
```

Finally, we set the values of some of the other parameters. For the purposes of this example, we use 500 simulations for the pivotal and Gaussian methods, and 100 repetitions for the weighted and gradient bootstrap methods. The set of analyzed quantile indexes will be $\{0.04, 0.08, ..., 0.96\}$, but we will have `npqr` print only results for quantile indexes contained in the set $\{0.2, 0.4, 0.6, 0.8\}$. Finally, we will use $\alpha = 0.95$ as the level for the confidence intervals.

```
B <- 500

B.boot <- 100

taus <- c(1:24)/25

print.taus <- c(1:4)/5

alpha <- 0.95
```

3 Output of `npqr`

3.1 Comparison of the Inference Processes

Initially, we will focus on the average growth rate, i.e., the average derivative of the conditional quantile function with respect to child's age

$$u \mapsto \int \partial_w g(w,u)\mu(w), \quad u \in \mathcal{U},$$

where μ is a measure and $\mathcal{U}$ is the region of quantile indexes of interest. Inference will be performed uniformly over the set of quantile indexes `taus`, and the standard errors will be computed unconditionally for the pivotal and Gaussian processes; see Section 3.3.

We first construct the 4 inference processes based on the B-spline basis. In each case, we surround the `npqr` call with code to calculate the speed of computation, which we will report later. Instead of invoking a particular process, we may also set `process="none"`. In this case, inference will not be performed, and only point estimates will be reported.

```
start.time.piv<-Sys.time()

piv.bsp <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="pivotal", uniform=T, se="unconditional", printOutput=T)

piv.time<-difftime(Sys.time(),start.time.piv,units="mins")

start.time.gaus<-Sys.time()

gaus.bsp <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="gaussian", uniform=T, se="unconditional", printOutput=T)

gaus.time<-difftime(Sys.time(),start.time.gaus,units="mins")

start.time.wboot<-Sys.time()

wboot.bsp <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B.boot, nderivs=1, average=1, alpha=alpha,
process="wbootstrap", uniform=T, printOutput=T)

wboot.time<-difftime(Sys.time(),start.time.wboot,units="mins")

start.time.gboot<-Sys.time()

gboot.bsp <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B.boot, nderivs=1, average=1, alpha=alpha,
process="gbootstrap", uniform=T, printOutput=T)

gboot.time<-difftime(Sys.time(),start.time.gboot,units="mins")
```

The output for the pivotal method is:

```
Method: pivotal
Standard error estimation is unconditional.

No. of obs.:                                      37623
No. of simulations or bootstrap repetitions:      500


--------------------------------------------------------------------------------
                        Quantile Estimates & Inference
--------------------------------------------------------------------------------
            Point     Standard        Uniform          One-sided 95% CIs
  Quantile  Estimate   Error      95% Conf.Interval     Lower      Upper
     0.2    0.7676  0.008542    0.7442     0.7911    0.7466     0.7887
     0.4    0.7761   0.00751    0.7555     0.7968    0.7577     0.7946
     0.6    0.7967    0.0077    0.7756     0.8179    0.7778     0.8157
     0.8    0.8225  0.008492    0.7992     0.8458    0.8016     0.8434

--------------------------------------------------------------------------------
                            Hypothesis Testing
--------------------------------------------------------------------------------
Null Hypothesis:                                                        p-value
================================================================================
theta(tau, w) <= 0 for all tau, w                                             0
theta(tau, w) >= 0 for all tau, w                                             1
theta(tau, w) = 0 for all tau, w                                        0.02392
================================================================================
```

Additionally, the following results are saved in `piv.bsp`:

- `piv.bsp$CI`: a 1 × `length(taus)` × 2 array: each pair is the lower and upper bounds of the 95% confidence interval for the average derivative of the conditional quantile function at each quantile index in `taus`
- `piv.bsp$CI.oneSided`: a 1 × `length(taus)` × 2 array: each pair contains bounds for two separate one-sided 95% confidence intervals (a lower bound and an upper bound, respectively) for the average derivative of the conditional quantile function at each quantile index in `taus`
- `piv.bsp$point.est`: a 1 × `length(taus)` matrix: each entry is the point estimate for the average derivative of the conditional quantile function at each quantile index in `taus`
- `piv.bsp$std.error`: a 1 × `length(taus)` matrix: each entry is the standard error of the estimator of the average derivative of the conditional quantile function at each quantile index in `taus` (here, unconditional on the sample)

- `piv.bsp$pvalues`: a three item vector containing the p-values reported above: the first tests the null hypothesis that the average derivative is less than zero everywhere (at each quantile index in `taus`); the second tests the null hypothesis that the average derivative is everywhere greater than zero; the third tests the null hypothesis that the average derivative is everywhere equal to zero

- `piv.bsp$taus`: the input vector `taus`, i.e., $\{0.04, 0.08, ..., 0.96\}$

- `piv.bsp$coefficients`: a list of length `length(taus)`: each element of the list contains the estimates of the QR coefficient vector $[\beta(u)', \gamma(u)']'$ at the corresponding quantile index

- `piv.bsp$var.unique`: a vector containing all values of the covariate of interest, W, with no repeated values

- `piv.bsp$load`: the input vector or matrix `load`. If `load` is not input (as in this case), the output `load` is generated based on `average` and `nderivs`. Here, it is a vector containing the average value of the derivative of the regression equation with respect to the variable of interest, not including the estimated coefficients.

Using `piv.bsp$taus`, `piv.bsp$CI`, and `piv.bsp$point.est`, as well as the corresponding objects for the Gaussian, weighted bootstrap, and gradient bootstrap methods, we construct plots containing the estimated average quantile derivatives, as well as 95% uniform confidence bands over the quantile indexes in `taus`:

```
par(mfrow=c(2,2))

yrange <- c(.65,.95)

xrange <- c(0,1)

plot(xrange,yrange,type="n",xlab="",ylab="Average Growth (cm/month)",
ylim=yrange)

lines(piv.bsp$taus,piv.bsp$point.est)

lines(piv.bsp$taus,piv.bsp$CI[1, ,1],col="blue")

lines(piv.bsp$taus,piv.bsp$CI[1, ,2],col="blue")

title("Pivotal")

plot(xrange,yrange,type="n",xlab="",ylab="",ylim=yrange)

lines(gaus.bsp$taus,gaus.bsp$point.est)

lines(gaus.bsp$taus,gaus.bsp$CI[1, ,1],col="blue")
```

```
lines(gaus.bsp$taus,gaus.bsp$CI[1, ,2],col="blue")

title("Gaussian")

plot(xrange,yrange,type="n",xlab="Quantile",ylab="Average Growth (cm/month)",
ylim=yrange)

lines(wboot.bsp$taus,wboot.bsp$point.est)

lines(wboot.bsp$taus,wboot.bsp$CI[1, ,1],col="blue")

lines(wboot.bsp$taus,wboot.bsp$CI[1, ,2],col="blue")

title("Weighted Bootstrap")

plot(xrange,yrange,type="n",xlab="Quantile",ylab="",ylim=yrange)

lines(gboot.bsp$taus,gboot.bsp$point.est)

lines(gboot.bsp$taus,gboot.bsp$CI[1, ,1],col="blue")

lines(gboot.bsp$taus,gboot.bsp$CI[1, ,2],col="blue")

title("Gradient Bootstrap")

title("Avg Growth Rate with 95% CI", outer=T)
```

As we can see in Figure 1, the confidence bands generated are roughly similar. Note that the point estimates are the same for all the methods.

We can compare the computation times of each of the approximations using the timings recorded earlier. Additionally, we compare the p-values generated by each of the four inference methods. Note that computation times may vary widely depending on the machine in use. However, the relative computation times will be approximately constant across different machines. The computation times in the table below were obtained on a computer with two eight-core 2.6 GHz processors (note: `npqr` does not make use of parallel computing).

```
pval.dimnames<-vector("list", 2)

pval.dimnames[[1]]<-c("Pivotal", "Gaussian", "Weighted Bootstrap", "Gradient
Bootstrap")

pval.dimnames[[2]]<-c("H0:  Growth Rate <= 0","H0:  Growth Rate >= 0", "H0:
Growth Rate = 0", "Computation Minutes")

pvals<-matrix(NA,nrow=4, ncol=4, dimnames=pval.dimnames)

pvals[1,]<-c(round(piv.bsp$pvalues,digits=4),round(piv.time,digits=0))

pvals[2,]<-c(round(gaus.bsp$pvalues,digits=4),round(gaus.time,digits=0))
```

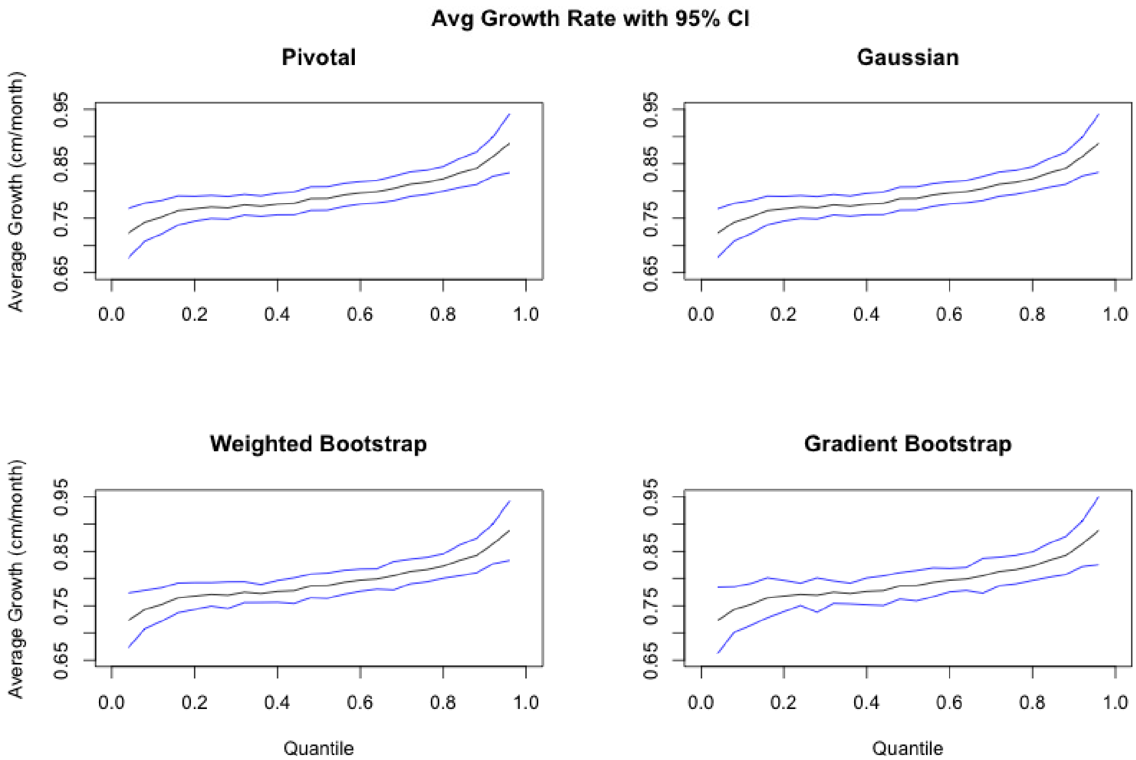

Figure 1: Growth Rate Chart: point estimates and 95% uniform confidence bands for the average derivative of the conditional quantile function of height with respect to age based on B-spline series approximation.

```
pvals[3,]<-c(round(wboot.bsp$pvalues,digits=4),round(wboot.time,digits=0))
pvals[4,]<-c(round(gboot.bsp$pvalues,digits=4),round(gboot.time,digits=0))
print(pvals)
```

These commands generate the output:

```
                     H0:  Growth  H0:  Growth  H0:  Growth  Computation
                      Rate <= 0    Rate >= 0    Rate = 0      Minutes
 Pivotal                  0            1         0.0237         0.9
 Gaussian                 0            1         0.0234         0.6
 Weighted Bootstrap       0            1         0.0221         30
 Gradient Bootstrap       0            1         0.0221         346
```

As expected, we reject at the 5% level the null hypothesis that the growth rate is negative and the null hypothesis that the growth rate is equal to zero in all cases, and we fail to reject the null hypothesis that the growth rate is positive in all cases. For the one-sided tests, the relevant null hypothesis is that the average growth rate is less than or equal to zero (greater than or equal to zero) at all the quantile indexes in `taus`. For the two-sided test, the relevant null hypothesis is that the average growth rate is equal to zero at all the quantile indexes in `taus`. Additionally, note that the pivotal and Gaussian methods are substantially faster than the two bootstrap methods.

3.2 Comparison of Series Bases

Another option is to take advantage of the variety of bases available in the `quantreg.nonpar` package. Here, we consider two bases: the B-spline basis used in the analysis above and an orthogonal polynomial basis of degree 12, chosen to yield the same number of basis terms as the B-spline basis[1]. We compare the estimates of the average quantile derivative function generated by using each of these bases. We construct the orthogonal polynomial basis of degree 12 for `cage` with the command:

```
basis.poly <- poly(cage, degree=12)
```

In this section, we focus on pivotal and Gaussian methods for inference. We run `npqr` for the orthogonal polynomial basis using each of the two methods, mimicking the analysis run above for the B-spline basis.

[1]The `npqr` command will also admit a basis generated by the `fda` package of type `"fourier"`. We do not illustrate this capability in this vignette since the periodic nature of Fourier bases will generate unrealistic estimates in this nonperiodic setting.

```
piv.poly <- npqr(formula=form.par, basis=basis.poly, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="pivotal", uniform=T, printOutput=T)
```

```
gaus.poly <- npqr(formula=form.par, basis=basis.poly, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="gaussian", uniform=T, printOutput=T)
```

Similar to Section 3.1, we plot the point estimates with their uniform 95% confidence bands for each method - basis pair:

```
par(mfrow=c(2,2))
yrange<-c(0.65,0.95)
xrange<-c(0,1)
plot(xrange,yrange,type="n",xlab="",ylab="Average Growth (cm/month)")
lines(piv.bsp$taus,piv.bsp$point.est)
lines(piv.bsp$taus,piv.bsp$CI[1, ,1],col="blue")
lines(piv.bsp$taus,piv.bsp$CI[1, ,2],col="blue")
title("Pivotal Approximation, B-Spline Basis")
plot(xrange,yrange,type="n",xlab="",ylab="")
lines(piv.poly$taus,piv.poly$point.est)
lines(piv.poly$taus,piv.poly$CI[1, ,1],col="blue")
lines(piv.poly$taus,piv.poly$CI[1, ,2],col="blue")
title("Pivotal Approximation, Polynomial Basis")
plot(xrange,yrange,type="n",xlab="Quantile",ylab="Average Growth (cm/month)")
lines(gaus.bsp$taus,gaus.bsp$point.est)
lines(gaus.bsp$taus,gaus.bsp$CI[1, ,1],col="blue")
lines(gaus.bsp$taus,gaus.bsp$CI[1, ,2],col="blue")
title("Gaussian Approximation, B-Spline Basis")
plot(xrange,yrange,type="n",xlab="Quantile",ylab="")
lines(gaus.poly$taus,gaus.poly$point.est)
```

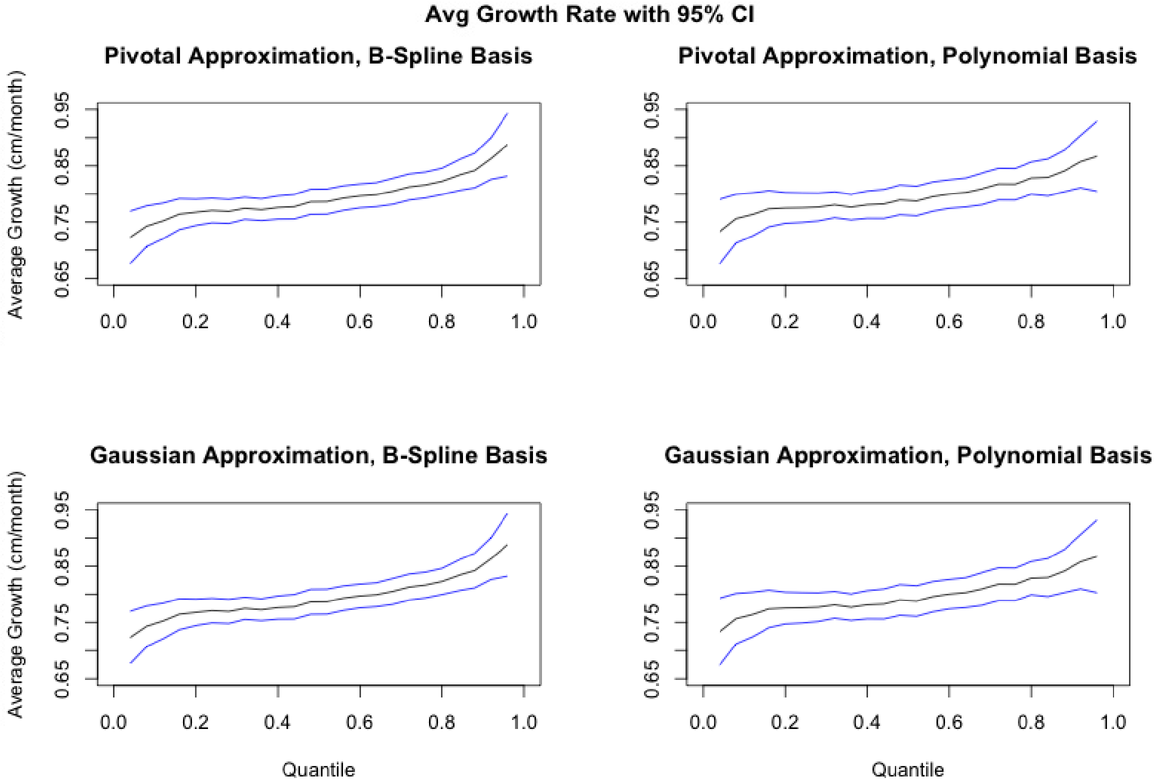

Figure 2: Growth Rate Chart: point estimates and 95% uniform confidence bands for the average derivative of the conditional quantile function of height with respect to age based on B-spline and polynomial series approximations.

```
lines(gaus.poly$taus,gaus.poly$CI[1, ,1],col="blue")
lines(gaus.poly$taus,gaus.poly$CI[1, ,2],col="blue")
title("Gaussian Approximation, Polynomial Basis")
title("Avg Growth Rate with 95% CI", outer=T)
```

Figure 2 shows that the choice of basis does not have an important impact on the estimation and inference on the growth rate charts. The p-values associated with the hypothesis tests for each method - basis pair are largely similar as well:

```
pval2.dimnames<-vector("list",2)
```

```
pval2.dimnames[[1]]<-c("Pivotal, B-spline", "Pivotal, Polynomial", "Gaussian,
B-spline", "Gaussian, Polynomial")

pval2.dimnames[[2]]<-c("H0:  Growth Rate <= 0","H0:  Growth Rate >= 0", "H0:
Growth Rate = 0")

pvals2<-matrix(NA,nrow=4,ncol=3,dimnames=pval2.dimnames)

pvals2[1,]<-round(piv.bsp$pvalues,digits=4)

pvals2[2,]<-round(piv.poly$pvalues,digits=4)

pvals2[3,]<-round(gaus.bsp$pvalues,digits=4)

pvals2[4,]<-round(gaus.poly$pvalues,digits=4)

print(pvals2)
```

These commands yield:

	H0: Growth Rate <= 0	H0: Growth Rate >= 0	H0: Growth Rate = 0
Pivotal, B-Spline	0	1	0.0239
Pivotal, Polynomial	0	1	0.0334
Gaussian, B-Spline	0	1	0.0264
Gaussian, Polynomial	0	1	0.0325

In all cases, the conclusion of the tests are the same across choice of basis and method at the 5% level.

3.3 Confidence Intervals and Standard Errors

Now, we illustrate two additional options available to the user. First, to perform inference pointwise over a region of covariate values and/or quantile indexes instead of uniformly, and second, to estimate the standard errors conditional on the values of the covariates X in the sample. When inference is uniform, the test statistic used in construction of the confidence interval is the maximal t-statistic across all covariate values and quantile indexes in the region of interest, whereas pointwise inference uses the t-statistic at each covariate value and quantile index. When standard errors are estimated unconditionally, a correction term is used to account for the fact that the empirical distribution of X is an estimator of the distribution of X. The option to estimate standard errors conditionally or unconditionally is not available for the bootstrap methods. The inference based on these methods is always unconditional.

We will use only the pivotal method with a B-spline basis for this illustration. First, we run `npqr` for each combination of options mentioned above:

```
piv.bsp <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="pivotal", uniform=T, se="unconditional", printOutput=T)

piv.bsp.cond <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="pivotal", uniform=T, se="conditional", printOutput=T)

piv.bsp.point <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="pivotal", uniform=F, se="unconditional", printOutput=T)

piv.bsp.point.cond <- npqr(formula=form.par, basis=basis.bsp, var="cage",
taus=taus, print.taus=print.taus, B=B, nderivs=1, average=1, alpha=alpha,
process="pivotal", uniform=F, se="conditional", printOutput=T)
```

We obtain Figure 3 using the graphing techniques described in Sections 3.1 and 3.2. As is visible in this figure, usage of conditional standard errors changes the confidence bands only minimally in our example. As expected, the pointwise confidence bands are narrower than the uniform confidence bands.

We can also compare how much of the differences (or lack thereof) in the confidence bands are driven by differences in the standard errors versus the test statistics. Here, we compare the estimated standard errors at the median for conditional versus unconditional inference:

```
piv.bsp.med <- npqr(formula=form.par, basis=basis.bsp, var="cage", taus=0.5,
B=B, nderivs=1, average=1, alpha=alpha, process="pivotal", uniform=T,
se="unconditional", printOutput=F)

piv.bsp.cond.med <- npqr(formula=form.par, basis=basis.bsp, var="cage",
taus=0.5, B=B, nderivs=1, average=1, alpha=alpha, process="pivotal", uniform=T,
se="conditional", printOutput=F)

stderr.dimnames<-vector("list",2)

stderr.dimnames[[1]]<-c("Unconditional","Conditional")

stderr.dimnames[[2]]<-c("Standard Error")

stderr<-matrix(NA,nrow=2,ncol=1,dimnames=stderr.dimnames)

stderr[1,]<-piv.bsp.med$std.error[1]

stderr[2,]<-piv.bsp.cond.med$std.error[1]

print(stderr)
```

These commands yield the output:

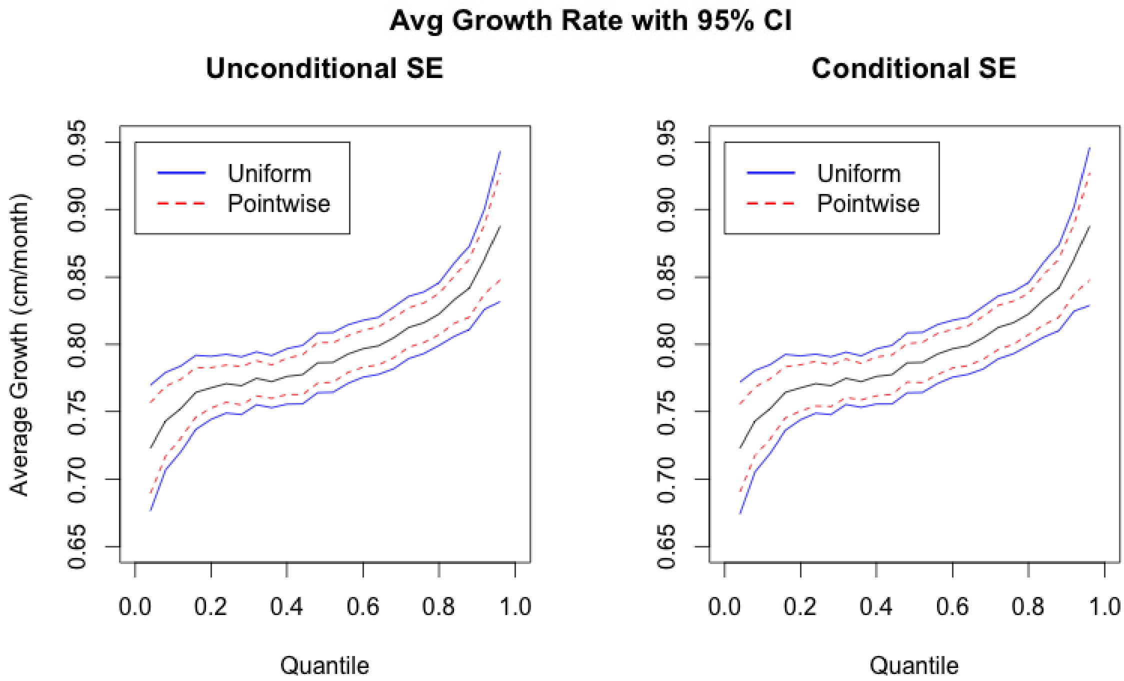

Figure 3: Growth Rate Chart: 95% uniform and pointwise confidence bands for the average derivative of the conditional quantile function of height with respect to age based on B-spline series approximation.

	Standard Error
Unconditional	0.008104
Conditional	0.007663

Finally, we compare p-values generated by each of the option choices:

	H0: Growth Rate <= 0	H0: Growth Rate >= 0	H0: Growth Rate = 0
Uniform, Unconditional	0	1	0.0239
Uniform, Conditional	0	1	0.0243
Pointwise, Unconditional	0	1	0.0267
Pointwise, Conditional	0	1	0.0222

In all cases, we find the expected results. Indeed, conditional versus unconditional standard errors and uniform versus pointwise inference have little impact on the estimated p-values in this example where the sample size is large, about 38,000 observations.

3.4 Estimation and Uniform Inference on Linear Functionals

Finally, we illustrate how to estimate and make uniform inference on linear functionals of the conditional quantile function over a region of covariate values and quantile indexes. These functionals include the function itself and derivatives with respect to the covariate of interest. The `quantreg.nonpar` package is able to perform estimation and inference on the conditional quantile function, its first derivative, and its second derivative over a region of covariate values and/or quantile indexes. We also illustrate how to report the estimates using three dimensional plots.

First, we consider the first and second derivatives of the conditional quantile function. In the application they correspond to the growth rate and growth acceleration of height with respect to age as a function of age (from 0 to 59 months) and the quantile index. To do so, we use the output of `npqr` called `var.unique`, which contains a vector with all the distinct values of the covariate of interest (`cage` here). To generate this output, we estimate the first and second derivatives of the conditional quantile function using a B-spline series approximation over the covariate values in `var.unique` and the quantile indexes in `taus`:

```
piv.bsp.firstderiv <- npqr(formula=form.par, basis=basis.bsp, var="cage",
taus=taus, print.taus=print.taus, B=B, nderivs=1, average=0, alpha=alpha,
process="none", se="conditional", printOutput=F, method="fn")
```

```
piv.bsp.secondderiv <- npqr(formula=form.par, basis=basis.bsp, var="cage",
taus=taus, print.taus=print.taus, B=B, nderivs=2, average=0, alpha=alpha,
process="none", se="conditional", printOutput=F, method="fn")
```

Next, we generate vectors containing the region of covariate values and quantile indexes of interest:

```
xsurf1<-as.vector(piv.bsp.firstderiv$taus)

ysurf1<-as.vector(piv.bsp.firstderiv$var.unique)

zsurf1<-t(piv.bsp.firstderiv$point.est)

xsurf2<-as.vector(piv.bsp.secondderiv$taus)

ysurf2<-as.vector(piv.bsp.secondderiv$var.unique)

zsurf2<-t(piv.bsp.secondderiv$point.est)
```

Finally, we create the three dimensional plots for:

$$(w, u) \mapsto \partial^k_{w^k} g(w, u), \quad (w, u) \in I$$

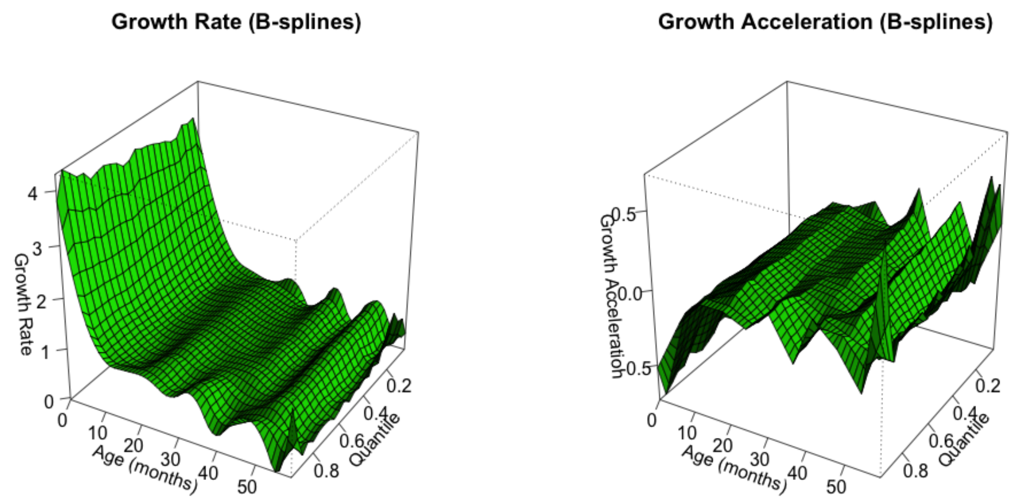

Figure 4: Growth Rate and Acceleration Charts: estimates of the first and second derivatives of the conditional quantile function of height with respect to age.

where $\partial^k_{w^k}$ denotes the k-th partial derivative with respect to w, $k \in \{1, 2\}$, and I is the region of interest.

```
par(mfrow=c(1,2))
```

```
persp(xsurf1, ysurf1, zsurf1, xlab="Quantile", ylab="Age (months)",
zlab="Growth Rate", ticktype="detailed", phi=30,theta=120, d=5, col="green",
shade=0.75, main="Growth Rate (B-splines)")
```

```
persp(xsurf2, ysurf2, zsurf2, xlab="Quantile", ylab="Age (months)",
zlab="Growth Acceleration", ticktype="detailed", phi=30,theta=120, d=5,
col="green", shade=0.75, main="Growth Acceleration (B-splines)")
```

These commands produce Figure 4. Here, we see that the growth rate is positive at all ages and quantile indexes. The growth rate decreases in the first few months of life and stabilizes afterwards, which can also be seen in the graph of growth acceleration. Growth acceleration is negative at young ages but stabilizes around zero at about 15 months. Both growth rate and growth acceleration are relatively homogeneous across quantiles at all ages. Saved in `piv.bsp.firstderiv$pvalues` and `piv.bsp.secondderiv$pvalues` are the p-values from hypothesis tests to determine whether the first and second derivatives, respectively, are negative, positive, and equal to zero uniformly over the region of ages and quantile indexes:

Order of Derivative	H0: Growth Rate <= 0	H0: Growth Rate >= 0	H0: Growth Rate = 0
First Derivative	0	1	0.042
Second Derivative	1	0	0.061

Thus, we reject at the 5% level the null hypotheses that growth rate is negative, that growth rate is equal to zero, and that growth acceleration is positive over all the first five years of the children's lives at all the quantiles of interest. We come close to rejecting at the 5% level the null hypothesis that growth acceleration is equal to zero over all the first five years of the children's lives at all the quantiles of interest.

Similarly, we estimate the conditional quantile function over a region of covariate values and quantile indexes, which corresponds to a growth chart in our application. Here, we use a fully saturated indicator basis for the series approximation to the nonparametric part of the model. We also compare the original estimates of the resulting growth chart to rearranged estimates that impose that the conditional quantile function of height is monotone in age and the quantile index. In this example, the conditional quantile function estimated using all data is nearly monotone without rearrangement. To illustrate the power of rearrangement when estimates are not monotone, we use a subset of the data containing the first 1,000 observations:

```
data.subset <- data[1:1000,]

detach(data)

attach(data.subset)
```

Now, we create the fully saturated indicator basis for `cage`:

```
faccage <- factor(cage)
```

To perform estimation using this basis, we input `faccage` for `basis`:

```
piv.fac.fun <- npqr(formula=form.par, basis=faccage, var="cage", taus=taus,
print.taus=print.taus, B=B, nderivs=0, average=0, alpha=alpha, process="none",
rearrange=F, se="conditional", printOutput=F, method="fn")
```

We also obtain the rearranged estimates with respect to age and the quantile index using the options of the command `npqr`. Note that we input `"both"` for `rearrange.vars`. This option performs rearrangement over quantiles and age. Other allowable options are `"quantile"` (for monotonization over quantiles only) and `"var"` (for monotonization over the variable of interest only).

```
piv.fac.fun.re <- npqr(formula=form.par, facvar="faccage", var="cage",
taus=taus, print.taus=print.taus, B=B, nderivs=0, average=0, alpha=alpha,
```

```
process="none", rearrange=T, rearrange.vars="both", se="conditional", printOutput=F, method="fn")
```

Now, we construct three dimensional plots for the estimates of the conditional quantile function:

$$(w, u) \mapsto Q_{Y|X}(u|x) = g(w, u) + v'\gamma(u), \quad (w, u) \in I,$$

where v are evaluated at the sample mean for cardinal variables (`mbmi`, `breastfeeding`, `mage`, `medu`, and `edupartner`) and the sample mode for unordered factor variables (`faccsex`, `facctwin`, `faccbirthorder`, `facmunemployed`, `facmreligion`, `facmresidence`, `facwealth`, `facelectricity`, `facradio`, `factelevision`, `facrefrigerator`, `facbicycle`, `facmotorcycle`, and `faccar`).

```
xsurf<-as.vector(piv.bsp.fun$taus)

ysurf<-as.vector(piv.bsp.fun$var.unique)

zsurf.fac<-t(piv.fac.fun$point.est)

zsurf.fac.re<-t(piv.fac.fun.re$point.est)

par(mfrow=c(1,2))

persp(xsurf, ysurf, zsurf.fac, xlab="Quantile", ylab="Age (months)",
zlab="Height", ticktype="detailed", phi=30, theta=40, d=5, col="green",
shade=0.75, main="Growth Chart (Indicators)")

persp(xsurf, ysurf, zsurf.fac.re, xlab="Quantile", ylab="Age (months)",
zlab="Height", ticktype="detailed", phi=30, theta=40, d=5, col="green",
shade=0.75, main="Growth Chart (Indicators, Rearranged)")
```

Figure 5 shows that the rearrangement fixes the non-monotonic areas of the original estimates.

References

[1] Belloni, A., Chernozhukov, V., and I. Fernandez-Val (2011), "Conditional quantile processes based on series or many regressors," *arXiv:1105.6154*.

[2] Koenker, R (2011), "Additive models for quantile regression: Model selection and confidence bandaids," *Brazilian Journal of Probability and Statistics* 25(3), pp. 239–262.

[3] Koenker, R. and G. Bassett (1978): "Regression Quantiles," *Econometrica* 46, pp. 33-50.

[4] Ramsay, J. O., Wickham, H., Graves, S., and G. Hooker (2013), "fda: Functional Data Analysis," R package version 2.3.6, `http://CRAN.R-project.org/package=fda`.

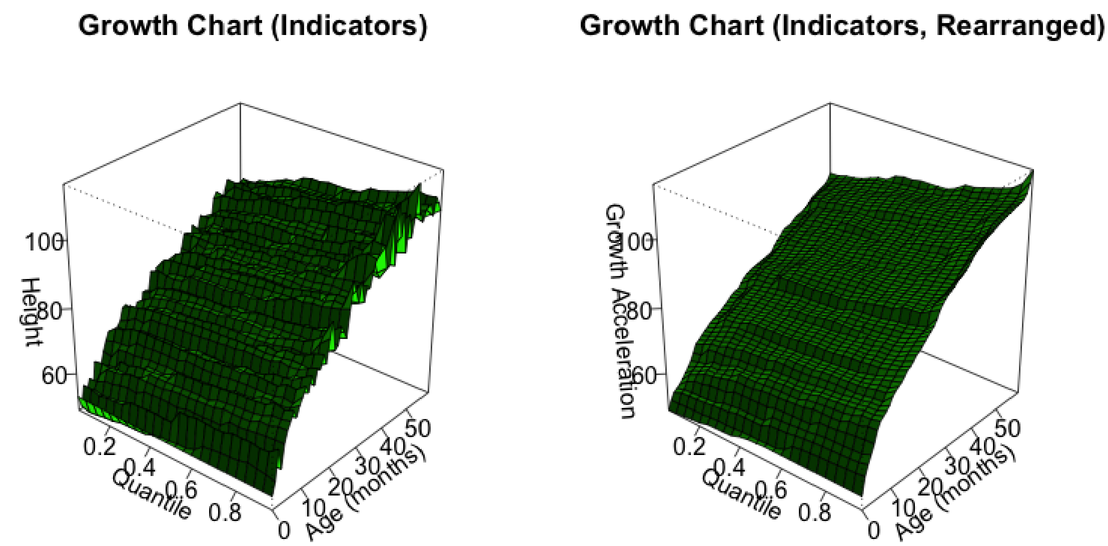

Figure 5: Growth Chart: estimates of the conditional quantile function of height based on a fully saturated indicator approximation with respect to age.