Journal of Statistical Software

January 2019, Volume 88, Issue 1. *doi: 10.18637/jss.v088.i01*

cdfquantreg: An R Package for CDF-Quantile Regression

Yiyun Shou
The Australian National University

Michael Smithson
The Australian National University

Abstract

The CDF-quantile family of two-parameter distributions with support (0, 1) described in Smithson and Merkle (2014) and recently elaborated by Smithson and Shou (2017), considerably expands the variety of distributions available for modeling random variables on the unit interval. This family is especially useful for modeling quantiles, and also sometimes out-performs the other distributions. The distributions are very tractable, with a location and dispersion parameter, explicit probability distribution functions, cumulative distribution functions, and quantiles. They enable a wide variety of quantile regression models with predictors for the location and dispersion parameters, and simple interpretations of those parameters. The R package **cdfquantreg** (Shou and Smithson 2019) (at least R 3.2.0) presented in this paper includes 36 distributions from the CDF-quantile family. Separate submodels may be specified for the location and for the dispersion parameters, with different or overlapping sets of predictors in each. The package offers maximum likelihood, Bayesian MCMC, and bootstrap estimation methods. Model diagnostics, including the gradient, three types of residuals, and the dfbeta influence measures, are available for evaluating models. The package also provides pseudo-random generators for all of its distributions. Many of its functions and their usage have forms familiar to R users, and the documentation is extensive. We also present a SAS macro for general linear models using the CDF-quantile family that includes many of the same capabilities as the **cdfquantreg** package. The paper provides examples of applications to real data-sets.

Keywords: quantile regression, unit interval, distribution, R.

1. Introduction

The most popular two-parameter distribution for modeling random variables on the (0, 1) interval is the beta distribution (e.g., Ferrari and Cribari-Neto 2004; Smithson and Verkuilen 2006). Less commonly used are the Kumaraswamy (1980), lambda, logit-logistic, simplex, and triangular distributions (e.g., Barndorff-Nielsen and Jørgensen 1991; Kotz and Van Dorp 2004). The R (R Core Team 2018) package **cdfquantreg** (Shou and Smithson 2019) implements general linear models using the recently developed CDF-quantile family of two-parameter distributions with support (0, 1) described in Verkuilen and Smithson (2012) and Smithson and Merkle (2014), and elaborated by Smithson and Shou (2017).

This family may be especially useful for modeling quantiles, and it also sometimes outperforms the other distributions in fitting data. Modeling with quantile functions, of course, has an extensive literature focusing mainly on non-parametric and semi-parametric methods (e.g., Parzen 1979; Gilchrist 2000). More recent relevant semi-parametric transformational techniques include Geraci and Jones (2015), as implemented in Geraci (2016). The approach taken in this paper and by Smithson and Shou (2017) is parametric.

The background to the CDF-quantile family begins with Tadikamalla and Johnson (1982) replacing the standard normal distribution in Johnson's (1949) SB distribution with the standard logistic distribution, thus producing the logit-logistic distribution. A natural extension of this approach is to employ other transformations from (0, 1) to either the real line or nonnegative half of the real line, and to expand the variety of standard distributions as well. Alzaatreh, Lee, and Famoye (2013) defined the so-called T-X family as follows:

$$G\left(x\right) = \int_{a}^{W(S(x))} r\left(t\right)dt, \tag{1}$$

where $r(t)$ is the probability density function (PDF) of a random variable, $T \in [a, b]$, for $-\infty \leq a < b \leq \infty$; and $W(S(x))$ satisfies three properties:

1. $W(S(x)) \in [a, b]$,
2. $W(S(x))$ is differentiable and monotonically non-decreasing, and
3. $W(S(x)) \to a$ as $x \to -\infty$ and $W(S(x)) \to b$ as $x \to \infty$.

The cumulative distribution function (CDF) in Equation 1 can be written in terms of the CDF of T:

$$G(x) = R[W(S(x))]. \tag{2}$$

Aljarrah, Lee, and Famoye (2014) extended this family by proposing that W be the quantile function of a third random variable, Y, say, whose support is the same as T, an idea whose origins date at least as far back as Van Zwet (1964). Independently of these researchers, Verkuilen and Smithson (2012) and Smithson and Merkle (2014, p. 158) described a distribution family that is a special case of Aljarrah et al.'s family and also related to the Johnson SB family.

Smithson and Shou (2017) refer to this family as the "CDF-quantile" family for the simple reason that its members can be written as the composition of a CDF and quantile function, and their paper explicates its characteristics and evaluates its advantages for modeling purposes. The resulting family of distributions has the following useful properties:

1. Tractability, with explicit PDFs, CDFs, and quantiles.
2. Amenability to both maximum likelihood and Bayesian estimation techniques.

3. The family enables a wide variety of quantile regression models for random variables on the (0, 1) interval with predictors for both the location and scale parameters.
4. The family can model four distinct varieties of distribution shapes, with different skew and kurtosis coverage from the beta or the Kumaraswamy.
5. Explicit quantiles render random generation of variates straightforward.

1.1. The distribution family

Let $G(x, \mu, \sigma)$ denote a CDF with support (0, 1), a real-valued location parameter μ and positive scale parameter σ. G is defined as

$$G(x, \mu, \sigma) = F[U(H^{-1}(x), \mu, \sigma)], \tag{3}$$

where F is a standard CDF with support D_1, H is a standard invertible CDF with support D_2 (so that $H^{-1} : (0, 1) \to D_2$ is the corresponding quantile function), and $U : D_2 \to D_1$ is an appropriate transform for imposing the location and scale parameters. D_1 and D_2 are either $(-\infty, \infty)$ or $(0, \infty)$. If $D_1 = D_2 = (-\infty, \infty)$ then

$$U(y, \mu, \sigma) = (y - \mu)/\sigma. \tag{4}$$

All of the CDF-quantile distributions in the **cdfquantreg** package have this form. Explications and examples of members with other combinations of D_1 and D_2 are provided in Smithson and Shou (2017). The distributions in Equation 3 are related to the T-X family in Equation 2 by setting $F = R$ and $U[H^{-1}(S(x), \mu, \sigma)] = W(S(x))$, with H differentiable and $x \in (0, 1)$. This reduces to Equation 3 by restricting S to be the uniform CDF, so that $S(x) = x$. One way of interpreting the CDF-quantile family is that G redistributes X by first transforming it via H^{-1} to a random variable whose domain is D_1, and that variable is rescaled via μ and σ and then fitted by a location-scale distribution, F. Because X and G both share the unit interval as their domain, G provides a redistribution of X. Indeed, if H and F are identical, and $\mu = 0$ and $\sigma = 1$, then G simply returns X.

Lemonte and Bazán (2015) also describe a family of distributions with support (0, 1) as an extension of the Johnson SB family. Theirs is a special case of the CDF-quantile family with H restricted to the logistic CDF (details available from the first author). Lemonte and Bazán (2015) do not cite Alzaatreh *et al.* (2013) or any related papers, so their research efforts seem to have been unconnected with that group of researchers and with Smithson and his co-authors.

If F is invertible, then the distribution has an explicit quantile. If G is differentiable then it has an explicit PDF. All of the distributions in this package share both properties. There is a relation between pairs of these distributions in which F and H exchange roles. These pairs are "quantile-duals" of one another in the sense that one's CDF is the other's quantile, with the appropriate parameterization. We name these distributions with the nomenclature F-H (e.g., Cauchit-logistic and logit-Cauchy). Other relevant properties shared by the members of the family included in this package are as follows (proofs are provided in Smithson and Shou 2017):

1. The PDFs $g(x, \mu, \sigma)$ are self-dual in this respect: $g\left(x, \mu, \sigma\right) = g\left(1 - x, -\mu, \sigma\right)$.
2. When $H = F$ the distribution includes the uniform distribution as a special case. Otherwise, all distributions are symmetrical at $x = \frac{1}{2}$ when $\mu = 0$.
3. The median is a function solely of the location parameter μ.
4. Simple functions of the median and other particular quantiles yield expressions solely in the scale parameter σ.
5. The likelihood function is explicit, as are the gradient and Hessian.

1.2. Example distribution

An example is the arcsinh-Cauchy distribution. This distribution employs the hyperbolic arcsinh CDF

$$F\left(z\right) = \frac{1}{e^{-\sinh^{-1}(z)} + 1} \tag{5}$$

and the Cauchy CDF

$$H\left(z\right) = \frac{\tan^{-1}(z)}{\pi} + \frac{1}{2}. \tag{6}$$

Inverting H and applying it and F to the equation above for $G(x, \mu, \sigma)$ gives the CDF

$$G\left(x, \mu, \sigma\right) = \frac{1}{e^{-\sinh^{-1}\left(\frac{-\mu-\cot(\pi x)}{\sigma}\right)} + 1}, \tag{7}$$

and differentiating it gives the PDF

$$g\left(x, \mu, \sigma\right) = \frac{\pi \csc^2(\pi x) e^{\sinh^{-1}\left(\frac{\mu+\cot(\pi x)}{\sigma}\right)}}{\sigma\sqrt{\frac{\mu^2+\sigma^2+2\mu\cot(\pi x)+\cot^2(\pi x)}{\sigma^2}}\left(e^{\sinh^{-1}\left(\frac{\mu+\cot(\pi x)}{\sigma}\right)} + 1\right)^2}. \tag{8}$$

Inverting F and the appropriate substitutions give us the quantile:

$$G^{-1}\left(\gamma, \mu, \sigma\right) = \frac{\tan^{-1}\left(\frac{\sigma-2\gamma\sigma}{2(\gamma-1)\gamma} + \mu\right)}{\pi} + \frac{1}{2}. \tag{9}$$

As described in property 3 above in Section 1.1, the median is a function of just the location parameter μ:

$$G^{-1}\left(\frac{1}{2}, \mu, \sigma\right) = \frac{\tan^{-1}(\mu)}{\pi} + \frac{1}{2}, \tag{10}$$

and therefore

$$\mu = \tan\left(\pi Q\left(\frac{1}{2}\right) - \frac{1}{2}\right), \tag{11}$$

where $Q(\gamma)$ denotes the quantile at γ. Likewise, as in property 4, the scale parameter σ is a simple function of selected quantiles:

$$G^{-1}\left(\frac{2-\sqrt{2}}{2}, \mu, \sigma\right) = \frac{\tan^{-1}(\mu - \sigma)}{\pi} + \frac{1}{2}, \tag{12}$$

so that

$$\sigma = \mu - \tan\left[\pi\left(2Q\left(\frac{2-\sqrt{2}}{2}\right)-1\right)\Big/2\right]. \tag{13}$$

It also can be shown that this distribution has a finite density in the limits at 0 and 1, unlike most distributions on the unit interval: $\lim_{x\to 0} f(x,\mu,\sigma) = \lim_{x\to 1} f(x,\mu,\sigma) = \pi\sigma/2$.

1.3. CDF-quantile regression

Maximum likelihood inference can be performed for this distribution family, and for all members where the gradient also has an explicit expression. For all of the distributions in this package, the PDF may be written as

$$g(x,\mu,\sigma) = \frac{q(x)\, f\left(\frac{H^{-1}(x)-\mu}{\sigma}\right)}{\sigma}, \tag{14}$$

where f is the PDF corresponding to F, and q is the quantile density function corresponding to H^{-1}. Differentiating the log of g with respect to μ and σ drops q and gives the following:

$$\partial \log(g(x,\mu,\sigma))/\partial\mu = -\frac{\partial f\left(\frac{H^{-1}(x)-\mu}{\sigma}\right)\Big/\partial\mu}{\sigma f\left(\frac{H^{-1}(x)-\mu}{\sigma}\right)}, \tag{15}$$

$$\partial \log(g(x,\mu,\sigma))/\partial\sigma = \frac{(\mu - H^{-1}(x))\left(\partial f\left(\frac{H^{-1}(x)-\mu}{\sigma}\right)\Big/\partial\sigma\right)}{\sigma^2 f\left(\frac{H^{-1}(x)-\mu}{\sigma}\right)} - 1/\sigma. \tag{16}$$

Thus, the requirement for an explicit gradient is that f is differentiable with respect to μ and σ.

The resulting regression model in our framework has two submodels, the "location submodel" for μ and the "dispersion submodel" for σ:

$$\begin{aligned} L_\mu(\hat{\mu}) &= \mathbf{x}^\top\beta, \\ L_\sigma(\hat{\sigma}) &= \mathbf{z}^\top\delta, \end{aligned} \tag{17}$$

where $\mathbf{x}$ and $\mathbf{z}$ are vectors of predictors and β and δ are vectors of coefficients. The location submodel link function L_μ is the identity, and the dispersion submodel link function L_σ is the log. The sets of predictors in $\mathbf{x}$ and $\mathbf{z}$ may or may not overlap.

Note that because this is maximum likelihood estimation, the parameter estimates may be seen as determining the shape of both the PDF (or CDF) and the quantile function G^{-1}. Although this was illustrated in our example distribution, a more general understanding of this point is available by rewriting Equation 3 by assigning a quantile value $G(y,\mu(x),\sigma(z)) = \gamma$, and rearranging it as in Equation 18.

$$\begin{gathered} F\left[(H^{-1}(y) - \mu(\mathbf{x}))/\sigma(\mathbf{z})\right] = \gamma \Rightarrow \\ (H^{-1}(y) - \mu(\mathbf{x}))/\sigma(\mathbf{z}) = F^{-1}(\gamma) \Rightarrow \\ H^{-1}(y) = \mu(\mathbf{x}) + \sigma(\mathbf{z})\, F^{-1}(\gamma) \Rightarrow \\ y = H\left[\mu(\mathbf{x}) + \sigma(\mathbf{z})\, F^{-1}(\gamma)\right] = G^{-1}(\gamma, \mu(\mathbf{x}), \sigma(\mathbf{z})). \end{gathered} \tag{18}$$

Note that the third line in this rearrangement also shows why this model is a generalized linear model (GLM); the parameters are in a linear equation with the inverses of F and H

acting as link functions. Furthermore, if F is a symmetric distribution around 0 (which it always is in **cdfquantreg**) then $F^{-1}(0.5) = 0$. So the median always is a function only of μ and the location submodel provides a GLM for the median. Appropriate combinations with the dispersion submodel provide GLMs for other quantiles. Thus, the predictors in the location submodel influence the locations of all quantiles, whereas predictors in the dispersion submodel influence all of them except the median.

As detailed in Smithson and Shou (2017), the maximum likelihood estimators are well-behaved for the CDF-quantile family. Their sampling distributions closely approximate the normal distribution for modest sample sizes, and they are relatively stable in the presence of outliers. In these respects they compare favorably with the beta and Kumaraswamy distributions.

2. The R package cdfquantreg

Despite the fact that doubly-bounded variables are commonplace in many scientific disciplines, relatively few distributions currently are available in software for fitting models to such data. Beta distribution models for modeling both location and dispersion parameters are available in the well-developed **betareg** package (Cribari-Neto and Zeileis 2010; Grün, Kosmidis, and Zeileis 2012) and the Stata (StataCorp. 2015) package **betafit** (Buis, Cox, and Jenkins 2003). The R package **gamlss** (Rigby and Stasinopoulos 2005; Stasinopoulos and Rigby 2008) also provides beta distribution models, including 0- and 1-inflated models. A recent addition to these resources is the **simplexreg** package (Zhang, Qiu, and Shi 2016), which also estimates two-parameter models, using the simplex distribution.

The **cdfquantreg** package expands the variety of available distributions for such models by including 36 members of the two-parameter CDF-quantile family of distributions for modeling random variables on the (0, 1) interval. Separate submodels may be specified for the location and for the dispersion parameters, with different or overlapping sets of predictors in each. The package offers maximum likelihood, Bayesian MCMC, and bootstrap estimation methods, on a par with the aforementioned packages, except for the regression tree and mixture distribution models available in **betareg**. Package **cdfquantreg** is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=cdfquantreg`.

2.1. General usage

The main function is `cdfquantreg()`, and the basic usage is:

```
cdfquantreg(formula, fd, sd, data)
```

The `formula` has two parts, using the **Formula** package (Zeileis and Croissant 2010). An example formula is `y ~ X1 + X2 | Z1 + Z2`, where `y` on the left of `~` is the dependent variable to be modeled. The two parts separated by `|` on the right-hand side include predictors or independent variables in the model. The `X1 + X2` on the left-hand side of `|` specifies the location submodel, which is linked to the location parameter of `y`, $\hat{\mu}$. The `Z1 + Z2` on the right-hand of `|` specifies the dispersion submodel, which is linked to the dispersion parameter of `y`, $\log(\hat{\sigma})$. A null model part (i.e., intercept only model) can be represented by `1`. For example, `y ~ 1 | Z1 + Z2` specifies the location submodel as a null model.

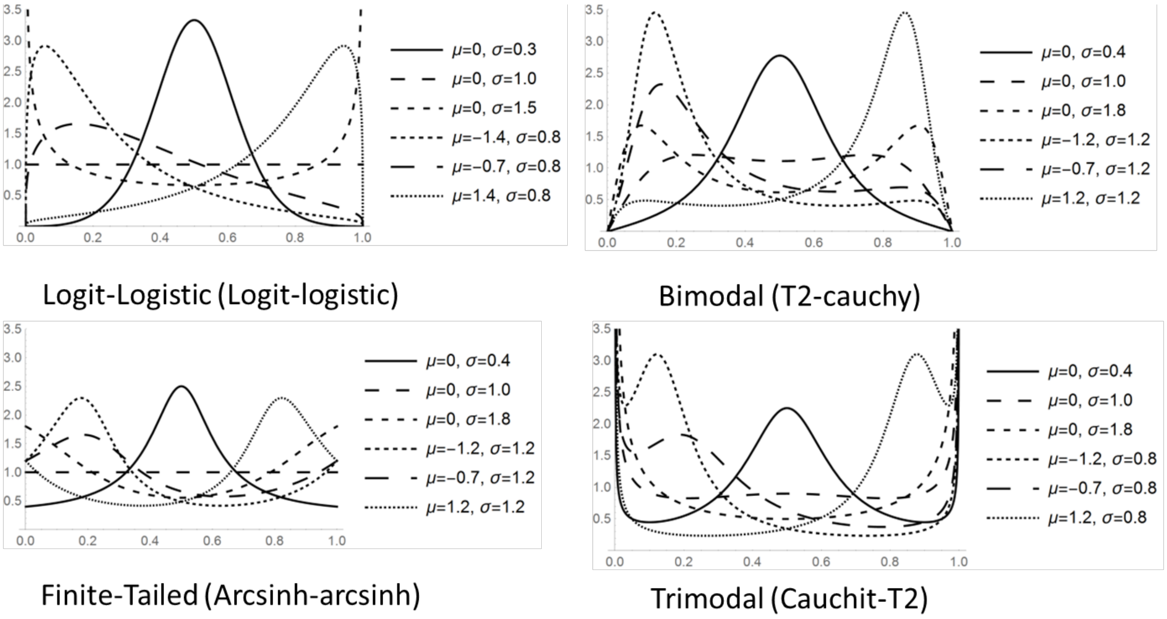
The dependent variable y must have numerical values within the (0, 1) interval. For variables that are on different scales, the function `scaleTR()` can be used to linearly transform the variable into the (0, 1) interval. The user can specify the extent to which the variable's values are pushed away from the boundary (i.e., 0 or 1). `scaleTR` employs the method suggested by Smithson and Verkuilen (2006) and applies a linear transformation to values into the open interval (0, 1). It first transforms the values from their original scale by taking $y' = (y - a)/(b - a)$, where a is the lowest possible value of that variable and b is the highest possible value of that variable. Next, it compresses the range to avoid zeros and ones by taking $y" = (y'(N - 1) + c)/N$, where N is the sample size and c is the compression parameter. The smaller the value c is, the closer the extreme values are to 0 or 1, and the greater is the impact they may have on the estimation of the dispersion parameter in the model. Typically, c is chosen to be 1/2.

A CDF-quantile distribution can be specified by using the arguments `fd` and `sd`, where `fd` refers to the parent distribution while `sd` specifies the child distribution. Specifications of available distributions using `fd` and `sd` are available via the `cdfqrFamily(shape = "all")` command. The help display also describes the shape of each distribution. We briefly review them here; for more details see Smithson and Shou (2017). We have found that these distributions exhibit four kinds of characteristic shapes, which can be described by their density's tail behavior at the boundaries of the (0, 1) interval:

1. Logit-logistic subfamily: For some value of $s > 0$ (depending on the distribution),
   - $\forall \sigma < s$, $\lim_{x \to 0} g\left(x, \mu, \sigma\right) = \lim_{x \to 1} g\left(x, \mu, \sigma\right) = 0$,
   - $\forall \sigma = s$, $\lim_{x \to 0} g\left(x, \mu, \sigma\right) = v\left(-\mu\right)$ and $\lim_{x \to 1} g\left(x, \mu, \sigma\right) = v\left(\mu\right)$, and
   - $\forall \sigma > s$, $\lim_{x \to 0} g\left(x, \mu, \sigma\right) = \lim_{x \to 1} g\left(x, \mu, \sigma\right) = \infty$;
2. Bimodal subfamily: $\lim_{x \to 0} g\left(x, \mu, \sigma\right) = \lim_{x \to 1} g\left(x, \mu, \sigma\right) = 0$;
3. Finite-tailed subfamily: $\lim_{x \to 0} g\left(x, \mu, \sigma\right) = \lim_{x \to 1} g\left(x, \mu, \sigma\right) = u(\sigma)$; and
4. Trimodal subfamily: $\lim_{x \to 0} g\left(x, \mu, \sigma\right) = \lim_{x \to 1} g\left(x, \mu, \sigma\right) = \infty$.

The first group is typified by the logit-logistic distribution, and thus is labeled "logit-logistic". Its tail behavior depends on σ and for $s < 1$, the densities at 0 and 1 go to 0, for $s > 1$ both densities go to infinity, and for $s = 1$ the density at 0 is $\exp(-\mu)$ while at 1 it is $\exp(\mu)$. The second group is the "bimodal" distributions, which are capable of having two modes within the (0, 1) interval because the densities at 0 and 1 always are 0. The third group is labeled "finite-tailed" distributions, which tend to be unimodal but with finite, identical, densities at 0 and 1 that are a function of σ. The arcsinh-Cauchy distribution described in Section 1.2 is a member of this subfamily, with density of $\pi\sigma/2$ at 0 and 1. Finally, the fourth group contains the "trimodal" distributions, which have one mode in the interior of (0, 1) and modes at 0 and 1 with infinite densities at the limit. Examples of the four subfamilies are displayed in Figure 1 (reproduced from Smithson and Shou 2017, Figures 1–4).

As shown in Table 1, `data` takes the data set, including both the dependent and predictor variables, in either matrix or data frame format. The variable (or corresponding column) names in the data object should correspond to the variable names in the formula. In some cases, users might use `start` to specify starting values for the mean and dispersion to improve convergence. By default, the empirical median and σ of y are used as the starting values for


Logit-Logistic (Logit-logistic)

Bimodal (T2-cauchy)

Finite-Tailed (Arcsinh-arcsinh)

Trimodal (Cauchit-T2)

Figure 1: Four CDF-quantile subfamilies.

Argument	Description
`formula`	A formula object, with the dependent variable (DV) on the left of a `~` operator, and predictors on the right. For the part on the right of '`~`', the location and dispersion submodels are separated by '|'. So `y ~ X1 + X2 | Z1 + Z2` specifies that the DV is `y`, `X1 + X2` specifies the location submodel, and `Z1 + Z2` specifies the predictors in the dispersion submodel.
`fd`, `sd`	Arguments that specify the distribution. `fd` indicates the parent distribution, while `sd` indicates the child distribution.
`data`	Specifies the data object which is in a `data.frame` format. The columns are variables while the rows are observations.
`start`	User-specified starting values for estimation of the distribution mean and dispersion.
`control`	Other specifications for the estimation.

Table 1: Overview arguments for `cdfquantreg()`.

the intercepts of the location and dispersion submodels. Finally, the `control` argument can be used to specify a list of parameters in the optimization procedure, such as the maximal number of iterations. The function `cdfquantreg()` returns a model object of S3 class '`cdfqr`'. Model outputs can be extracted via common generic functions, which are described in Table 2.

2.2. Random number generator

The **cdfquantreg** package also contains probability functions that parallel `rnorm()` (i.e., `rq`), `qnorm()` (i.e., `qq`), `pnorm()` (i.e., `pq`), and `dnorm()` (i.e., `dq`). With user-specified μ and σ, these functions can be used to generate random variates, density values, quantiles, and cumulative density values for a given distribution. These functions can be useful in simulations.

Function	Description
`summary(), print()`	Display the main results of the model fit, including model coefficients, log-likelihood, and gradient.
`coef()`	Extracts the coefficient values of the model object.

`deviance(), logLik(),`
`AIC(), BIC()`
	Extract common model fit indices.

`vcov()`	Extracts the variance-covariance matrix.
`predict(), fitted()`	Extract predicted and fitted values.
`residuals()`	Extracts residuals, including raw, Pearson's, and deviance residuals.
`influence(), dfbeta()`	Tests the influence of cases, via dfbetas as an influence measure.

Table 2: Generic functions that can be used to extract output.

```
R> library("cdfquantreg")
R> (x <- rq(5, mu = 0.5, sigma = 1, "arcsinh", "cauchy"))

[1] 0.1048779 0.9005961 0.8874756 0.9287983 0.7308099

R> dq(x, mu = 0.5, sigma = 1, "arcsinh", "cauchy")

[1] 0.920862 1.580366 1.577277 1.580416 1.262578

R> (cpv <- pq(x, mu = 0.5, sigma = 1, "arcsinh", "cauchy"))

[1] 0.1251272 0.8432766 0.8225590 0.8878642 0.5931846

R> qq(cpv, mu = 0.5, sigma = 1, "arcsinh", "cauchy")

[1] 0.1048779 0.9005961 0.8874756 0.9287983 0.7308099
```

3. Examples

We present five examples applying the **cdfquantreg** package to real data-analyses. The first example compares the performance of the CDF-quantile model with a beta-regression model. The second example illustrates the incorporation of a continuous predictor into a CDF-quantile model with a categorical moderator. The third example demonstrates how multivariate models of dependent observations on the (0, 1) interval can be constructed using the CDF-quantile family and copulas. The fourth applies CDF-quantile distributions to modeling data from an experiment, and this is reprised using the SAS (SAS Institute Inc. 2013) macro in Appendix A. The fifth example demonstrates the use of the **cdfquantreg** package's random-sampling capabilities for a simulation problem.

3.1. Interpretation of uncertainty phases in the IPCC report

Budescu, Broomell, and Por (2009) conducted an experimental study of lay interpretations of verbal phrases such as “likely” and “unlikely” to describe uncertainties. They used 13 sentences from the Intergovernmental Panel on Climate Change (IPCC) report (e.g., “The Greenland ice sheet and other Arctic ice fields likely contributed no more than 4 m of the observed sea level rise.”). They asked participants to provide lower, “best”, and upper numerical estimates of the probabilities to which they believed each sentence referred.

The IPCC data-set includes the lower, best, and upper estimates for the phrases “likely” and “unlikely” in six IPCC report sentences. Half of the six sentences used the term “likely” and the remaining used “unlikely”. The “likely” sentences are categorized as having a “positive” term while the “unlikely” sentences are categorized as having a “negative” term. A dummy variable named `valence` codes the responses in the positive term condition as 1, and those in the negative term condition as 0.

Using beta regression, Smithson, Budescu, Broomell, and Por (2012) reported two main findings. The first was that the “best” estimates were nearer to the middle of the [0, 1] interval in the negative-term condition than in the positive-term condition (estimated coefficient in the mean submodel = 0.150, $p = .004$, indicating that the mean is further from 1/2 in the positive-term condition). In addition, the responses were more variable (i.e., there was less consensus among respondents) in the negative-term than in the positive-term conditions (estimated coefficient in the precision submodel = 0.603, $p < .001$, indicating less variability in the positive-term condition).

We retest both findings by modeling the data with members of the CDF-quantile distribution family. The raw estimates themselves are in the variable named `prob`. The estimates for the negative-valence sentences were subtracted from 1 to render them directly comparable to the estimates for the positive-valence sentences. This variable was then transformed into a new variable named `probm` for shifting values away from the boundary values of 0 and 1.

A CDF-quantile model can be fitted in the similar way as a regression model that uses `lm()`. Here, we estimate a model using the t2-t2 distribution (where both parent and child distributions are t-distributions with 2 degrees of freedom). The t2-t2 CDF is

$$G\left(x, \mu, \sigma\right) = 1/2 + \frac{\left(w\left(x\right) - \mu\right)/\sigma}{2\sqrt{2 + \left(\left(w\left(x\right) - \mu\right)/\sigma\right)^2}}, \tag{19}$$

where $w\left(x\right) = q \cdot \sqrt{\left(1 - 2x\right)^2} \Big/ \sqrt{2}\sqrt{(1 - x)x}$, with $x \in \left(0, 1\right)$ and $q = -1$ when $x < 1/2$ and $q = 1$ when $x \geq 1/2$. The t2-t2 distribution is a member of the finite-tailed subfamily and it has density σ^2 at 0 and 1.

```
R> library("cdfquantreg")
R> data("cdfqrExampleData", package = "cdfquantreg")
R> dataipcc <- subset(IPCC, mid == 1 & high == 0)
R> fit <- cdfquantreg(probm ~ valence | valence, fd = "t2", sd = "t2",
+    data = dataipcc)
```

The summary of model fit results are extracted by using `summary(fit)`:

```
R> summary(fit)
```

```
Family:  t2 t2
Call:  cdfquantreg(formula = probm ~ valence | valence, data = dataipcc,
    fd = "t2", sd = "t2")

Mu coefficients (Location submodel)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.79843    0.03436  23.240  < 2e-16 ***
valence     -0.18600    0.04120  -4.514 6.35e-06 ***

Sigma coefficients (Dispersion submodel)
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.36789    0.04500  -8.176 2.22e-16 ***
valence     -0.42067    0.06228  -6.755 1.43e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Converge:  successful completion
Log-Likelihood:  435.2941

Gradient:  -0.0114 -0.01 0.0294 0.0203
```

The first part of the output labeled as “Mu coefficients” displays the parameter estimation results for the location submodel. The results show that valence had a significant influence on the median of participants’ probability estimates. The median of the probability estimates in the positive-term condition is found to be lower than in the negative-term condition (estimated coefficient $= -0.186$, $p < .001$). This is the opposite of the finding in Smithson *et al.* (2012) (for a detailed comparison of their model with this one, see Smithson and Shou 2017). The second part of the output shows the estimation results for the dispersion submodel. Our model agrees with the finding reported by Smithson *et al.* (2012), because the probability estimates in the positive-term condition have less dispersion (i.e., more consensus) than those in the negative-term condition (estimated coefficient $= -0.4207$, $p < .001$). Also noteworthy is that the log-likelihood for the beta regression model is 264.7 whereas the log-likelihood for the t2-t2 model is 435.3, a substantially better fit than the beta.

General fit plots can be obtained via `plot()` (see Figure 2). For evaluating the fit for each of the two conditions, we can examine the probability distribution given $\hat{\mu}$ and $\hat{\sigma}$ estimated by the model. For the negative-term condition (valence $= 0$) $\hat{\mu} = 0.7984$, and $\hat{\sigma} = \exp(-0.3679)$. For the positive-term condition (valence $= 1$) $\hat{\mu} = 0.7984 - 0.1860$, and $\hat{\sigma} = \exp(-0.3679 - 0.4207)$. Figure 3 shows the histograms of the responses in both conditions generated by the `truhist()` function of the **MASS** package (Venables and Ripley 2002), and the PDFs estimated by the CDF-quantile model using the t2-t2 distribution.

```
R> plot(fit)
R> ind <- which(dataipcc$valence == 0)
R> coefs <- coef(fit); fitv <- fitted(fit)
R> dneg <- dq(fitv[ind], coefs[1], exp(coefs[3]), "t2", "t2")
R> dpos <- dq(fitv[-ind], sum(coefs[1:2]), exp(sum(coefs[3:4])), "t2", "t2")
R> neg <- data.frame(x = fitv[ind], d = dneg)[order(fitv[ind]), ]
```

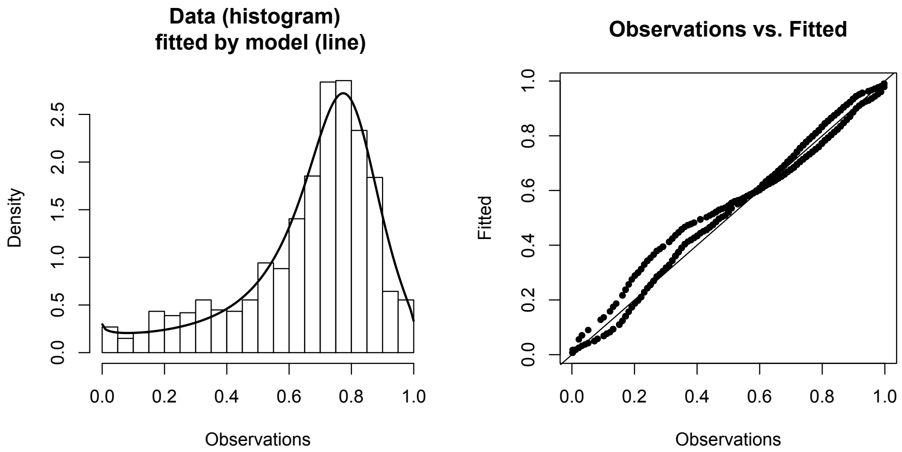


Figure 2: Compare model estimates and a histogram whose area equals 1.

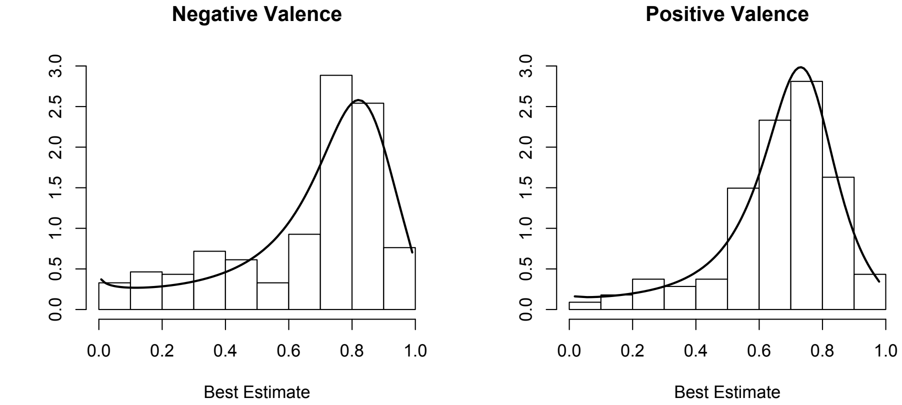


Figure 3: Compare conditional model estimates and histograms whose area equals 1.

```
R> pos <- data.frame(x = fitv[-ind], d = dpos)[order(fitv[-ind]), ]
R> library("MASS")
R> par(mfrow = c(1, 2), mar = c(2, 2, 3, 0), oma = c(2, 2, 0, 0))
R> truehist(dataipcc$probm[ind], col = "white", ylim = c(0, 3),
+    main = "Negative Valence", ylab = "Density")
R> lines(neg$x, neg$d, lty = 1, lwd = 2)
R> truehist(dataipcc$probm[-ind], col = "white", ylim = c(0, 3),
+    main = "Positive Valence", ylab = "Density")
R> lines(pos$x, pos$d, lty = 1, lwd = 2)
R> mtext("Best Estimate", side = 1, outer = TRUE, line = 0.5)
R> mtext("Density", side = 2, outer = TRUE, line = 0.5)
```

Although this is not a statistics paper, given that non-parametric and semi-parametric quantile regression methods are the conventional default approach to modeling quantiles, a brief comparison of quantile regression with the CDF-quantile model in this example may be worthwhile. The quantile regression model for this example requires a unique intercept and regression coefficient for every quantile we wish to estimate, i.e., $2 \times k$ where k is the number of quantiles. For example, for the 25th, 50th, and 75th percentiles, the quantile regression intercepts estimated via the **quantreg** package (Koenker 2018) are $\{0.470, 0.749, 0.829\}$ and the coefficients for the valence variable are $\{0.119, -0.050, -0.040\}$. Of course, in this situation these simply reproduce the empirical quantiles. The CDF-quantile model, on the other hand, estimates all (uncountably infinitely many) quantiles with just the 4 parameters in its location and dispersion submodels. Plotting the two models' CDFs provides an illustration of their similarities and differences adequate for our purposes.

```
R> xx <- seq(0.005,0.995,.005)
R> cdf_neg <- pq(xx, coefs[1], exp(coefs[3]), "t2", "t2")
R> cdf_pos <- pq(xx, sum(coefs[1:2]), sum(coefs[1:2]), "t2", "t2")
R> fitqtot <- quantreg::rq(probm ~ valence, tau = xx, data = dataipcc)
```

Figure 4 shows that the noticeable difference between the two models is in the lower half of the distribution when valence is negative. The quantile regression model is capturing a "bump" in the distributions that is not able to be accounted for with the limited number of parameters in the CDF-quantile model. But this is simply because, with its arbitrarily many parameters, in the two-sample case the quantile regression model is simply repeating the empirical CDFs for the two samples. Moreover, we also note that there is very close agreement between the two models for the upper half of the distribution when valence is negative and for the entire distribution when valence is positive. All this is achieved with just two parameters in each of the CDF-quantile submodels. The user must then decide whether to prefer the more parsimonious parametric model or the slightly better-fitting so-called non-parametric model.

```
R> par(mfrow = c(1, 2))
R> plot(xx, cdf_neg, type = "l", lty = 1, main = "negative valence",
+    xlab = "probability", ylab = "cdf")
R> lines(coef(fitqtot)[1, ], xx, lty = 2)
R> plot(xx, cdf_pos, type = "l", lty = 1, main = "positive valence",
+    xlab = "probability", ylab = "cdf")
R> lines(colSums(coef(fitqtot)[1:2, ]), xx, lty = 2)
R> legend("topleft", c("Cdf-quantreg", "Quantreg"), lty = 1:2)
```

The relative influence of individual observations can be assessed via `influence()`, which returns dfbetas for each of the parameters in the model. A dfbeta value for an observation is the scaled difference between a parameter value estimated in a model M_1 by using the entire sample and a parameter value estimated in a model M_2 by using the sample with this particular observation deleted. The scale factor is the standard error of the parameter when it is estimated in M_2 (Belsley, Kuh, and Welsch 2005). Figure 5 shows the dfbeta values for each of the four parameters.

```
R> infs <- influence(fit, plot = TRUE)
```

3.2. A continuous predictor and a categorical moderator

To supplement the preceding example, here we present a model that involves a continuous predictor whose effect is moderated by a categorical covariate. The data are, as in the previous

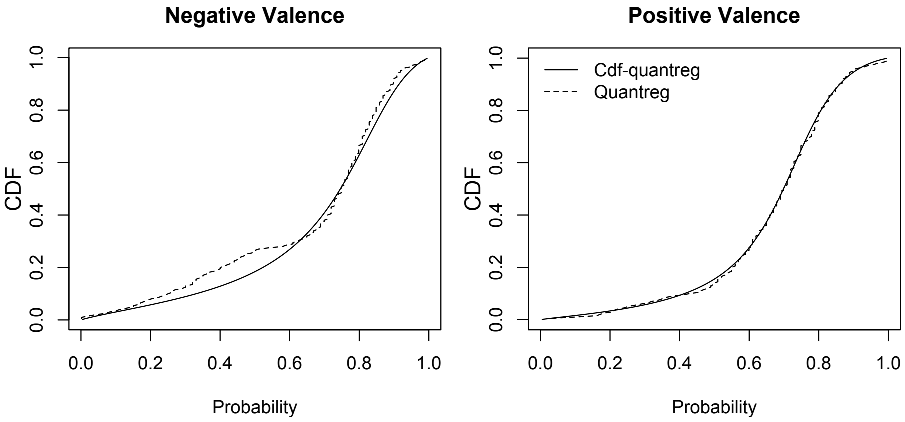


Figure 4: Compare quantile regression and CDF-quantile model CDFs.

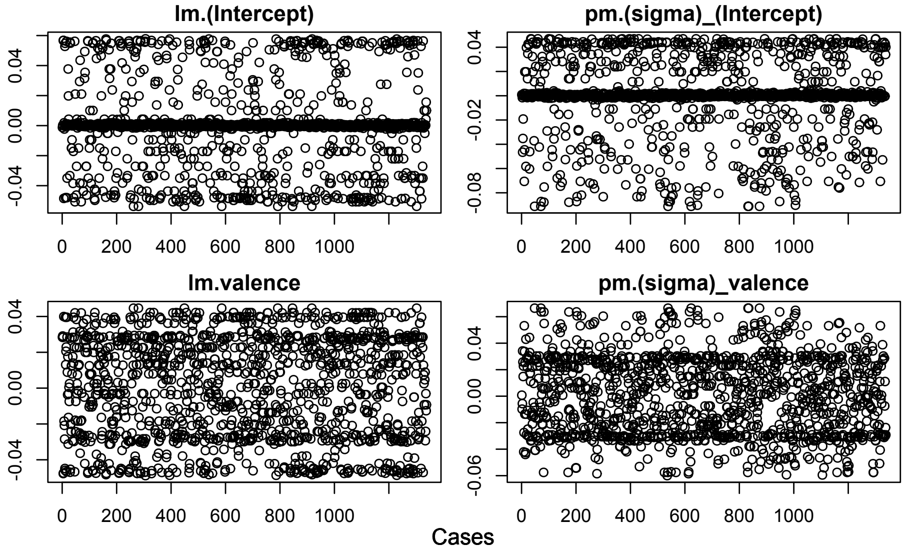


Figure 5: Influence diagnosis by dfbeta values for IPCC data analysis.

example, people's interpretations of the IPCC report verbal uncertainty phrase "likely" in the sentence "Temperatures of the most extreme hot nights, cold nights and cold days are likely to have increased due to anthropogenic forcing." However, these data are from a different study involving samples from 27 countries reported by Budescu, Por, Broomell, and Smithson (2014). Our illustration uses the Australian sample of 393 respondents. The (reduced) dataset contains each participant's age, gender (0 = male, 1 = female), the probability that they consider corresponds to their most typical use of "likely", and their best estimate of the probability intended in the sentence from the IPCC report.

Our model investigates the influence that a person's own probability that they associate with "likely" may have had on the probability they nominated for the IPCC report sentence containing that term. The hypothesis being tested is that the personal probability is positively related to the nominated probability. The first model tested (`mod0` below) fits a conditional logit-logistic distribution and verifies that this relationship exists and is positive (the location submodel coefficient for `cfprob`, the personal probability, is 2.1666).

```
R> mod0 <- cdfquantreg(bestprob ~ cfprob | 1, fd = "logit",
+    sd = "logistic", data = IPCCAUS)
R> summary(mod0)

Family:  logit logistic
Call:  cdfquantreg(formula = bestprob ~ cfprob|1, data=IPCCAUS,
fd="logit", sd="logistic")
Mu coefficients (Location submodel)
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.6791     0.1518  -4.474 7.69e-06 ***
cfprob        2.1666     0.2431   8.912  < 2e-16 ***

Sigma coefficients (Dispersion submodel)
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.67413    0.04301  -15.67   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Converge:  successful completion
Log-Likelihood:  138.7996
Gradient:  3e-04 3e-04 -6e-04
```

The second model (`mod1` below) tests whether gender moderates the influence of personal probability on the probability given as an interpretation of “likely” in the IPCC report sentence. The interpretation of the moderator effect from gender is facilitated by using the `qq` function to plot the model predicted median for values of `cfprob` from one standard deviation below to one standard deviation above its mean. The conditional median for the logit-logistic distribution in our model is $1/(1 + \exp(-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \beta_{12} x_1 x_2))$, where x_1 is `cfprob`, x_2 is `gender` (rescaled to take values 0 and 1), $\beta_0 = -0.3385$, $\beta_1 = 1.5887$, $\beta_2 = -0.7258$, and $\beta_3 = 1.2204$. The females in this sample were more strongly influenced than the males by their personal probability, as the slope of their medians is steeper than the males’ slope (see Figure 6).

```
R> mod1 <- update(mod0, . ~ . + cfprob * gender)
R> anova(mod0, mod1)

Likelihood ratio tests

Resid. Df -2Loglik Df LR stat Pr(>Chi)
1       390  -277.60
2       388  -283.92  2  6.3229  0.04236 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> summary(mod1)
```

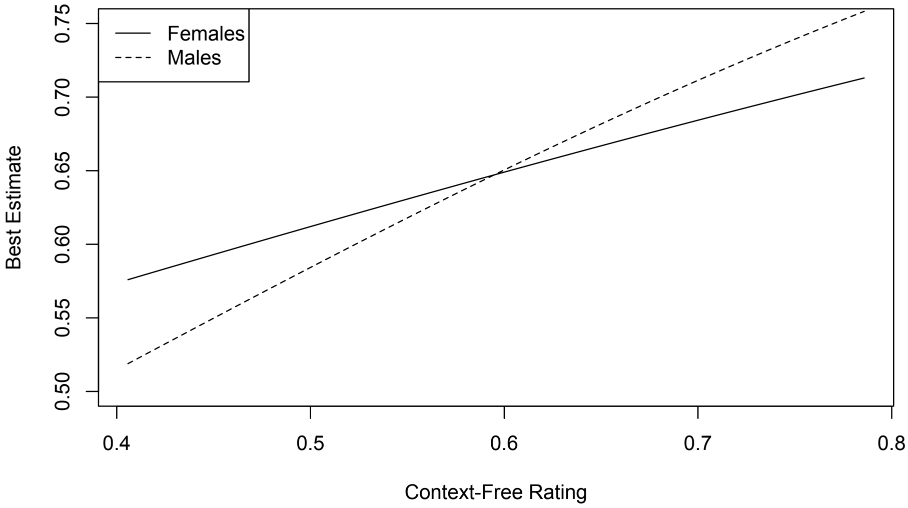


Figure 6: Effect of personal probability for "likely" on IPCC sentence interpretation by females and males.

```
Family:  logit logistic
Call:  cdfquantreg(formula = bestprob ~ cfprob + gender + cfprob:gender | 1,
data = IPCCAUS, fd = "logit", sd = "logistic")
Mu coefficients (Location submodel)
Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.3385     0.2069  -1.636   0.1018
cfprob          1.5887     0.3336   4.763 1.91e-06 ***
gender         -0.7258     0.3028  -2.397   0.0165 *
cfprob:gender   1.2204     0.4851   2.516   0.0119 *

Sigma coefficients (Dispersion submodel)
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.68075    0.04296  -15.85   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Converge:  successful completion
Log-Likelihood:  141.9611
Gradient:  0 0 0 0 0
```

3.3. Multivariate models of the IPCC data using copulas

An attractive approach to constructing multivariate distributions uses copulas, which are functions of CDFs and quantile functions. The CDF-quantile family has explicit CDFs, so copulas may be used to construct multivariate models of dependent doubly-bounded random variables. The most direct method is to select a copula, C (e.g., T, Clayton, or Frank), and derive an explicit expression for $C(G_1, G_2, \ldots)$, where C is the appropriate copula function and G_j is the jth CDF. If C is differentiable in all of its parameters, then it has an explicit log-likelihood function and thereby (in principle) is amenable to maximum likelihood estimation of its parameters. In R the **copula** package (Hofert, Kojadinovic, Maechler, and Yan 2017; Yan 2007; Kojadinovic and Yan 2010) obtains maximum likelihood estimates of the marginal distribution and copula parameters simultaneously. Moreover, user-defined marginal distributions can be used in **copula**, so long as the CDF, PDF, and quantile functions (vectorized) are available. The **cdfquantreg** package provides all three functions.

We illustrate the application of **cdfquantreg** and **copula** to modeling multivariate distributions of variables with support on the unit interval by constructing a trivariate copula for the IPCC example data. The trivariate copula models the data from questions 4–6 (the positive-valence probability expression "likely"). The **copula** package offers two alternative estimation methods: a one-stage procedure in which the package estimates the marginal distribution and association parameters simultaneously, and a two-stage procedure in which the marginal distribution parameters are estimated first and the association parameters thereafter. In the latter procedure, the quantile functions G_j^{-1} are applied using the marginal parameter estimates to generate pseudo-observations with uniform marginals, and the association parameters are then estimated from those pseudo-observations. We compare both procedures, using **cdfquantreg** to estimate the marginal parameters in the two-stage process. The marginal distributions are t2-t2 and the copula is the T copula, with degrees of freedom as a free parameter.

The dialog below displays the way to form a trivariate t-copula that allows three association parameters.

```
R> library("cdfquantreg")
R> library("lmtest")
R> library("copula")
R> dqt2 <- function(x, mu, sigma) dq(x, mu, sigma, "t2", "t2")
R> qqt2 <- function(x, mu, sigma) qq(x, mu, sigma, "t2", "t2")
R> pqt2 <- function(x, mu, sigma) pq(x, mu, sigma, "t2", "t2")
R> tCop1 <- tCopula(c(0.6, 0.3, 0.3), dim = 3, dispstr = "un",
+    df.fixed = FALSE)
R> margin1 <- list(mu = 1.5, sigma = 1)
R> MvdPV1 <- mvdc(copula = tCop1, margins = c("qt2", "qt2", "qt2"),
+    paramMargins = list(margin1, margin1, margin1))
R> start <- c(1.5, 1, 1.5, 1, 1.5, 1, 0.6, 0.3, 0.3, 4)
R> copfitPV1 <- fitMvdc(data = as.matrix(IPCC_Wide[, 1:3]),
+    mvdc = MvdPV1, start = start,
+    optim.control = list(trace = TRUE, maxit = 2000))
```

It is also possible to form a model that restricts the associations to a single parameter (as in sphericity).

```
R> tCop2 <- tCopula(0.5, dim = 3, dispstr = "ex", df.fixed = FALSE)
R> MvdPV2 <- mvdc(copula = tCop2, margins = c("qt2", "qt2", "qt2"),
+    paramMargins = list(margin1, margin1, margin1))
R> start <- c(1.5, 1, 1.5, 1, 1.5, 1, 0.5, 4)
R> copfitPV2 <- fitMvdc(as.matrix(IPCC_Wide[, 1:3]),
+    MvdPV2, start = start,
+    optim.control = list(trace = TRUE, maxit = 2000))
```

The likelihood-ratio tests using the **lmtest** package (Zeileis and Hothorn 2002) and AIC values indicate that a single-parameter association model fits the data as well as its three-parameter counterpart.

```
R> lrtest(copfitPV1, copfitPV2)

Likelihood ratio test

Model 1: copfitPV1
Model 2: copfitPV2
#Df LogLik Df  Chisq Pr(>Chisq)
1  10 336.36
2   8 336.25 -2 0.2359     0.8887

R> AIC(copfitPV1, copfitPV2)

          df       AIC
copfitPV1 10 -652.7272
copfitPV2  8 -656.4913
```

The next dialog shows the two-stage estimation procedure. It begins with the **cdfquantreg** estimates of the marginal parameters, generates the uniformly-distributed pseudo-observations, and then estimates a one-parameter trivariate copula using the pseudo-observations.

```
R> fit4 <- cdfquantreg(Q4 ~ 1 | 1, fd = "t2", sd = "t2",  data = IPCC_Wide)
R> fit5 <- update(fit4, Q5 ~ 1 | 1)
R> fit6 <- update(fit4, Q6 ~ 1 | 1)
R> udat <- cbind(pqt2(IPCC_Wide[, 1], coef(fit4)[1], exp(coef(fit4)[2])),
+    pqt2(IPCC_Wide[, 2], coef(fit5)[1], exp(coef(fit5)[2])),
+    pqt2(IPCC_Wide[, 3], coef(fit6)[1], exp(coef(fit6)[2])))
R> copfitudatPV <- fitCopula(tCop2, udat, start = c(0.5, 2.5))
R> loglikMvdc(c(coef(fit4)[1], exp(coef(fit4)[2]),
+    coef(fit5)[1], exp(coef(fit5)[2]), coef(fit6)[1],
+    exp(coef(fit6)[2]), coef(copfitudatPV)),
+    as.matrix(IPCC_Wide)[, 1:3], MvdPV2)

[1] 334.5089
```

Table 3 displays the estimates from the one- and two-stage models, along with their respective log-likelihoods. The $\hat{\rho}$ estimate denotes the Spearman's correlation coefficient and the $\hat{\psi}$ estimate denotes the degrees of freedom in the T copula. Both models have similar log-likelihoods and fairly similar parameter estimates. The standard errors for the **cdfquantreg** estimates are somewhat larger than they are for the **copula** estimates. However, differences such as these are not surprising given that these two packages are using different estimation algorithms as well as different methods.

Positive valence	**1-stage**		**2-stage**	
Log-likelihood	336.25		334.51	
Copula:				
Parameter	Estim.	S.err.	Estim.	S.err.
$\hat{\rho}$	0.430	0.047	0.438	0.186
$\hat{\psi}$	3.265	0.474	3.299	—
Marginal dist.:			cdfquant.	
Parameter	Estim.	S.err.	Estim.	S.err.
$\hat{\mu}_4$	0.634	0.039	0.643	0.045
$\hat{\sigma}_4$	0.480	0.034	0.517	0.039
$\hat{\mu}_5$	0.582	0.032	0.553	0.035
$\hat{\sigma}_5$	0.385	0.027	0.401	0.030
$\hat{\mu}_6$	0.632	0.035	0.653	0.039
$\hat{\sigma}_6$	0.417	0.030	0.451	0.034

Table 3: Estimates for one-stage and two-stage models.

3.4. Probability estimates under ambiguity and conflict

Smithson, Priest, Shou, and Newell (2018) conducted a study to examine lay people's probability judgments when receiving ambiguous or conflicting information. In each of four scenarios, participants were presented with two pairs of expert forecasts regarding the number of days on which it would rain or the probability of raining. The forecasts were presented in pairs. For example, one expert predicts that 4 to 6 days out of the next 7 days will have rain, while another predicts that 2 to 5 days out of the next 7 days will have rain. Based on this pair of estimates (i.e., [4, 6] and [2, 5]), participants were requested to provide their own estimates of how many days out of the 7 days will have rain. There were four scenarios in the study, in which the first pair of estimates in Scenarios 1, 3 and 4 had identical intervals. One of the research questions was whether the identical intervals yielded identical distributions of best estimates by participants even though they were presented in different contexts.

To answer this question, we fitted the response variable using `cdfquantreg()` with several distributions. We examined whether the responses in different scenarios had significantly different location and/or dispersion parameters. The outputs below show the likelihood-ratio tests assessing these effects via model comparisons. For all of the distributions fitted, the location parameters did not significantly differ across the scenarios, but the dispersion parameters did. We illustrate these findings with three of the distributions and their respective models.

The results of model comparison for models using t2-t2 distribution are displayed below. The likelihood-ratio tests show that the model with a scenario effect in the location submodel does not improve model fit over the null model, but a model with scenario effects in both the location and dispersion submodels does. This is due to the effect of scenario on dispersion.

```
R> Ambdata$prob <- scaleTR(Ambdata$value, high = 70, low = 0, N = 1570,
+    scale = 0.5)
R> fit1_t2t2 <- cdfquantreg(prob ~ 1 | 1, fd = "t2", sd = "t2",
+    data = Ambdata)
R> fit2_t2t2 <- update(fit1_t2t2, . ~ + scenario | 1)
```

```
R> fit3_t2t2 <- update(fit2_t2t2, . ~ . | + scenario)
R> anova(fit1_t2t2, fit2_t2t2, fit3_t2t2)

Likelihood ratio tests

Resid. Df -2Loglik Df LR stat Pr(>Chi)
1      4708  -5226.1
2      4706  -5226.4  2  0.3285  0.84854
3      4704  -5233.5  2  7.0555  0.02937 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model comparisons for models using the logit-logistic distribution are shown next. As with the t2-t2 distribution, there appears to be a dispersion submodel effect but no effect in the location submodel.

```
R> fit1_loglog <- cdfquantreg(prob ~ 1 | 1, fd = "logit", sd = "logistic",
+    data = Ambdata)
R> fit2_loglog <- update(fit1_loglog, . ~ + scenario | 1)
R> fit3_loglog <- update(fit2_loglog, . ~ . | + scenario)
R> anova(fit1_loglog, fit2_loglog, fit3_loglog)

Likelihood ratio tests

Resid. Df -2Loglik Df LR stat Pr(>Chi)
1      4708  -3237.5
2      4706  -3240.4  2  2.9604 0.227592
3      4704  -3251.0  2 10.5527 0.005111 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model comparisons for models using the arcsinh-t2 distribution, displayed below, yield the same pattern in the model comparisons as the preceding distributions.

```
R> fit1_ac <- cdfquantreg(prob ~ 1 | 1, fd = "arcsinh", sd = "t2",
+    data = Ambdata)
R> fit2_ac <- update(fit1_ac, . ~ + scenario | 1)
R> fit3_ac <- update(fit2_ac, . ~ . | + scenario)
R> anova(fit1_ac, fit2_ac, fit3_ac)

Likelihood ratio tests

Resid. Df -2Loglik Df LR stat Pr(>Chi)
1      4708    -5800
2      4706    -5801  2    1.01      0.6
3      4704    -5830  2   29.31  4.3e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

However, the three distribution models provide different conclusions regarding where the effect in the dispersion submodel occurred (as shown below). The results of the t2-t2 model suggest that the dispersion is similar in Scenarios 3 and 1 (Scenario 1 is the base comparison group), whereas the dispersion in Scenario 4 is significantly smaller than in Scenario 1. The logit-logistic model, on the other hand, finds that Scenarios 4 and 1 have similar dispersion whereas the dispersion in Scenario 3 is significantly greater than in Scenario 1. Finally, the arcsinh-t2 model suggests that the estimates in both Scenarios 3 and 4 have significantly less dispersion than Scenario 1. Which models should be trusted more? Returning to the log-likelihoods in the output above, we can see that the arcsinh-t2 and t2-t2 models fit the data considerably better than the logit-logistic model, with the arcsinh-t2 model best of the three. The pattern of the coefficients is similar for the arcsinh-t2 and t2-t2 models as well, whereas this pattern is not to be found in the logit-logistic model. We should therefore regard the arcsinh-t2 and t2-t2 models as more trustworthy.

```
R> options(digits = 3)
R> fit3_t2t2$coefficients$dispersion

             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.3640     0.0298 -45.757  0.00000
scenarioStm3  -0.0338     0.0438  -0.772  0.44036
scenarioStm4  -0.1140     0.0437  -2.610  0.00904

R> fit3_loglog$coefficients$dispersion

             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.8940     0.0222 -40.204  0.00000
scenarioStm3   0.0969     0.0316   3.068  0.00216
scenarioStm4   0.0199     0.0316   0.628  0.52991

R> fit3_ac$coefficients$dispersion

             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -1.9774     0.0375  -52.71  0.00000
scenarioStm3  -0.2150     0.0570   -3.77  0.00016
scenarioStm4  -0.2939     0.0571   -5.15  0.00000
```

3.5. Simulating polarized attitudes

Suppose we wish to simulate populations in which an attitude on a controversial issue is equally polarized, and we are interested in the effects of the extremity of this polarization. In this case, the attitude in question is the proportion of one's investment portfolio that should be invested in high-risk but potentially very high-return shares. A conventional approach would be to use a mixture of two distributions (e.g., two beta distributions), but a convenient and simple method is to employ one of the bimodal distributions from the family in the **cdfquantreg** package.

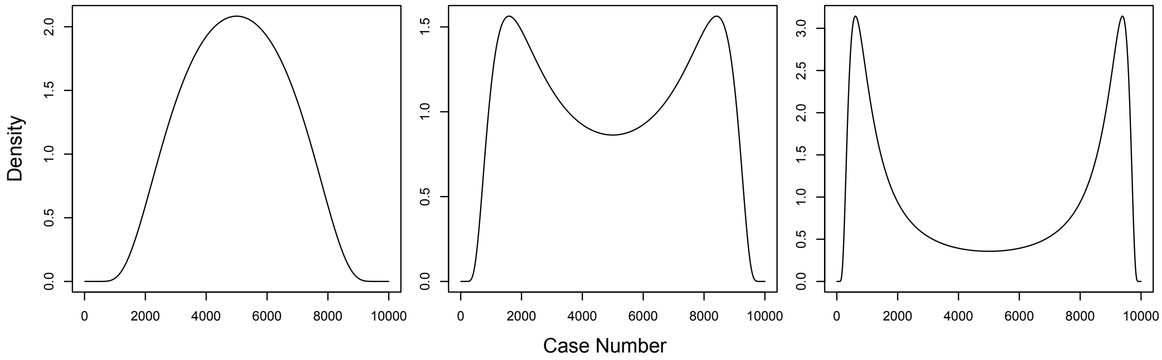


Figure 7: Illustration of simulating polarized distributions.

The logit-Cauchy distribution, for instance, has a simple quantile function and we may use it to control the degree of polarization in the simulated population. The 25th and 75th percentiles for populations with $\mu = 0$ (and therefore a median of .5) are

$$G^{-1}(.25, 0, \sigma) = \frac{\tan^{-1}(-\sigma \log(3))}{\pi} + \frac{1}{2}$$

and

$$G^{-1}(.75, 0, \sigma) = \frac{\tan^{-1}(\sigma \log(3))}{\pi} + \frac{1}{2}.$$

Solving their difference for σ gives

$$\sigma = \frac{\tan\left(\frac{\pi q}{2}\right)}{\log(3)},$$

where q is the difference between these quantiles. To simulate from three populations in which the 25th and 75th percentiles are separated by $q = .25$, .5, and .75, we assign to σ values 0.377033, 0.910239, and 2.19751, respectively. These yield a unimodal and two bimodal distributions. The second distribution has modes at approximately .15 and .85, while the third has modes at about .05 and .95. Figure 7 illustrates the shapes of the three distributions using the PDF `dq()`.

```
R> s <- seq(0.0001, 0.9999, length.out = 10000)
R> d1 <- dq(s, mu = 0, sigma = 0.377033, "logit", "cauchy")
R> d2 <- dq(s, mu = 0, sigma = 0.910239, "logit", "cauchy")
R> d3 <- dq(s, mu = 0, sigma = 2.19751, "logit", "cauchy")
R> par(mfrow=c(1, 3), mar = c(2,2,2,1), oma = c(3, 3, 0, 0))
R> plot(d1, type = "l"); plot(d2, type = "l"); plot(d3, type = "l")
R> mtext("Case Number", side = 1, outer = TRUE, line = 1)
R> mtext("density", side = 2, outer = TRUE, line = 1)
```

To illustrate how to simulate these distributions, we randomly generate 10,000 replicates from each of these distributions and ascertain that the samples accurately reproduce the relevant population characteristics.

```
R> simout <- matrix(data = c(rep(0, 12)), nrow = 3, ncol = 3)
R> rownames(simout) <- c("s1", "s2", "s3")
```

```
R> colnames(simout) <- c("p25", "p50", "p75")
R> s1 <- rq(10000, mu = 0, sigma = 0.37703, "logit", "cauchy")
R> s2 <- rq(10000, mu = 0, sigma = 0.91024, "logit", "cauchy")
R> s3 <- rq(10000, mu = 0, sigma = 2.19751, "logit", "cauchy")
R> simout[1, ] <- quantile(s1, c(.25, .5, .75))
R> simout[2, ] <- quantile(s2, c(.25, .5, .75))
R> simout[3, ] <- quantile(s3, c(.25, .5, .75))
R> simout
```

```
     p25   p50   p75
s1 0.370 0.497 0.626
s2 0.252 0.500 0.755
s3 0.125 0.499 0.878
```

We can compare the simulated quantiles with the population quantiles provided by `qq()`:

```
R> s1 <- qq(c(.25, .5, .75), mu = 0, sigma = 0.37703, "logit", "cauchy")
R> s2 <- qq(c(.25, .5, .75), mu = 0, sigma = 0.91024, "logit", "cauchy")
R> s3 <- qq(c(.25, .5, .75), mu = 0, sigma = 2.19751, "logit", "cauchy")
R> rbind(s1, s2, s3)
```

```
    [,1] [,2]  [,3]
s1 0.375  0.5 0.625
s2 0.250  0.5 0.750
s3 0.125  0.5 0.875
```

4. Conclusion

The **cdfquantreg** package and SAS macro offer researchers the ability to construct and test GLMs of random variates on the unit interval using the CDF-quantile family of distributions, which provides a viable alternative to well-known distributions such as the beta and Kumaraswamy. As demonstrated by Smithson and Shou (2017), members of the CDF-quantile family can model a variety of shapes unavailable to the beta or Kumaraswamy, and in the data-fitting examples presented in their paper members of this family out-perform those distributions. The family includes the logit-logistic distribution, and others that have not appeared before in the literature.

The package and macro also present a framework for systematically modeling quantiles of variates on the (0, 1) interval, thanks to the explicit quantile functions possessed by this family. Like beta regression, the GLMs in this family have a location and a dispersion submodel. Unlike beta regression, dispersion and location may be modeled independently of one another. Smithson and Shou (2017) show that the location parameter determines the median, and the dispersion parameter determines how far other quantiles are from the median.

The **cdfquantreg** package also enables researchers to venture beyond classical maximum likelihood inference, by providing bootstrap and Bayesian MCMC options for model estimation.

Model diagnostics, including the gradient, three types of residuals, and the dfbeta influence measures, are available for evaluating models. The package also provides pseudo-random generators for all of its distributions. Many of its functions and their usage have forms that are familiar to R users, and the documentation is extensive. The package does not depend extensively on other R packages, and (as demonstrated in the example involving the **copula** package) is easy to work with in conjunction with other packages in the R environment.

References

Aljarrah MA, Lee C, Famoye F (2014). "On Generating T-X Family of Distributions Using Quantile Functions." *Journal of Statistical Distributions and Applications*, **1**(1), 2. doi:10.1186/2195-5832-1-2.

Alzaatreh A, Lee C, Famoye F (2013). "A New Method for Generating Families of Continuous Distributions." *METRON*, **71**(1), 63–79. doi:10.1007/s40300-013-0007-y.

Barndorff-Nielsen OE, Jørgensen B (1991). "Some Parametric Models on the Simplex." *Journal of Multivariate Analysis*, **39**(1), 106–116. doi:10.1016/0047-259x(91)90008-p.

Belsley DA, Kuh E, Welsch RE (2005). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, volume 571. John Wiley & Sons. doi:10.1002/0471725153.

Budescu DV, Broomell S, Por HHH (2009). "Improving Communication of Uncertainty in the Reports of the Intergovernmental Panel on Climate Change." *Psychological Science*, **20**(3), 299–308. URL http://pss.sagepub.com/content/20/3/299.abstract.

Budescu DV, Por HH, Broomell SB, Smithson M (2014). "The Interpretation of IPCC Probabilistic Statements around the World." *Nature Climate Change*, **4**(6), 508–512. doi:10.1038/nclimate2194.

Buis ML, Cox NJ, Jenkins SP (2003). "**BETAFIT**: Stata Module to Fit a Two-Parameter Beta Distribution." Statistical Software Components, Boston College Department of Economics. URL https://ideas.repec.org/c/boc/bocode/s435303.html.

Cribari-Neto F, Zeileis A (2010). "Beta Regression in R." *Journal of Statistical Software*, **34**(2), 1–24. doi:10.18637/jss.v034.i02.

Ferrari SLP, Cribari-Neto F (2004). "Beta Regression for Modelling Rates and Proportions." *Journal of Applied Statistics*, **31**(7), 799–815. doi:10.1080/0266476042000214501.

Geraci M (2016). "**Qtools**: A Collection of Models and Tools for Quantile Inference." *The R Journal*, **8**(2), 117–138.

Geraci M, Jones MC (2015). "Improved Transformation-Based Quantile Regression." *Canadian Journal of Statistics*, **43**(1), 118–132. doi:10.1002/cjs.11240.

Gilchrist W (2000). *Statistical Modelling with Quantile Functions*. CRC Press, Boca Raton.

Grün B, Kosmidis I, Zeileis A (2012). "Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned." *Journal of Statistical Software*, **48**(11), 1–25. doi:10.18637/jss.v048.i11.

Hofert M, Kojadinovic I, Maechler M, Yan J (2017). ***copula**: Multivariate Dependence with Copulas.* R package version 0.999-18, URL https://CRAN.R-project.org/package=copula.

Johnson NL (1949). "Systems of Frequency Curves Generated by Methods of Translation." *Biometrika*, **36**(1–2), 149–176.

Koenker R (2018). ***quantreg**: Quantile Regression.* R package version 5.36, URL https://CRAN.R-project.org/package=quantreg.

Kojadinovic I, Yan J (2010). "Modeling Multivariate Distributions with Continuous Margins Using the **copula** R Package." *Journal of Statistical Software*, **34**(9), 1–20. doi:10.18637/jss.v034.i09.

Kotz S, Van Dorp JR (2004). *Beyond Beta: Other Continuous Families of Distributions with Bounded Support and Applications.* World Scientific.

Kumaraswamy P (1980). "A Generalized Probability Density Function for Double-Bounded Random Processes." *Journal of Hydrology*, **46**(1–2), 79–88. doi:10.1016/0022-1694(80)90036-0.

Lemonte AJ, Bazán JL (2015). "New Class of Johnson S_B Distributions and Its Associated Regression Model for Rates and Proportions." *Biometrical Journal*, **58**(4), 727–746. doi:10.1002/bimj.201500030.

Parzen E (1979). "Nonparametric Statistical Data Modeling." *Journal of the American Statistical Association*, **74**(365), 105–121. doi:10.2307/2286734.

R Core Team (2018). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rigby RA, Stasinopoulos DM (2005). "Generalized Additive Models for Location, Scale and Shape." *Journal of the Royal Statistical Society C*, **54**(3), 507–554. doi:10.1111/j.1467-9876.2005.00510.x.

SAS Institute Inc (2013). *The SAS System, Version 9.4.* SAS Institute Inc., Cary. URL https://www.sas.com/.

Shou Y, Smithson M (2019). ***cdfquantreg**: Quantile Regression for Random Variables on the Unit Interval.* R package version 1.2.1, URL https://CRAN.R-project.org/package=cdfquantreg.

Smithson M, Budescu DV, Broomell SB, Por HHH (2012). "Never Say "Not": Impact of Negative Wording in Probability Phrases on Imprecise Probability Judgments." *International Journal of Approximate Reasoning*, **53**(8), 1262–1270. doi:10.1016/j.ijar.2012.06.019.

Smithson M, Merkle EC (2014). *Generalized Linear Models for Categorical and Continuous Limited Dependent Variables.* Chapman & Hall/CRC, Boca Raton.

Smithson M, Priest D, Shou Y, Newell B (2018). "Ambiguity and Conflict Aversion When Uncertainty is in the Outcomes." Under review.

Smithson M, Shou Y (2017). "CDF-Quantile Distributions for Modeling Random Variables on the Unit Interval." *British Journal of Mathematical and Statistical Psychology*, **70**(3), 412–438. doi:10.1111/bmsp.12091.

Smithson M, Verkuilen J (2006). "A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Dependent Variables." *Psychological Methods*, **11**(1), 54–71. doi:10.1037/1082-989x.11.1.54.

Stasinopoulos DM, Rigby RA (2008). "Generalized Additive Models for Location Scale and Shape (GAMLSS) in R." *Journal of Statistical Software*, **23**(7), 1–46. doi:10.18637/jss.v023.i07.

StataCorp (2015). *Stata Data Analysis Statistical Software: Release 14*. StataCorp LP, College Station. URL https://www.stata.com/.

Tadikamalla P, Johnson NL (1982). "Systems of Frequency Curves Generated by Transformations of Logistic Variables." *Biometrika*, **69**(2), 461–465.

Van Zwet WR (1964). "Convex Transformations: A New Approach to Skewness and Kurtosis." *Canadian Journal of Statistics*, **18**(4), 433–441. doi:10.1007/978-1-4614-1314-1_1.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Verkuilen J, Smithson M (2012). "Mixed and Mixture Regression Models for Continuous Bounded Responses Using the Beta Distribution." *Journal of Educational and Behavioral Statistics*, **37**(1), 82–113. doi:10.3102/1076998610396895.

Yan J (2007). "Enjoy the Joy of Copulas: With a Package **copula**." *Journal of Statistical Software*, **21**(4), 1–21. doi:10.18637/jss.v021.i04.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. doi:10.18637/jss.v034.i01.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10.

Zhang P, Qiu Z, Shi C (2016). "**simplex**: An R Package for Regression Analysis of Proportional Data Using the Simplex Distribution." *Journal of Statistical Software*, **71**(11), 1–21. doi:10.18637/jss.v071.i11.

A. The SAS macro: %cdfquantreg

CDF-quantile regression also is implemented in the SAS macro `%cdfquantreg`, using the `NLMIXED` (nonlinear mixed models) procedure to estimate model parameters. A variety of optimization techniques such as BFGS can be chosen, while the default optimization used in this macro is the trust region optimization. Successful convergence yields parameter estimates along with their approximate standard errors based on the Hessian matrix.

The macro employs the following input statement:

```
%cdfquantreg(DATA, DV, FD, SD, LMIV, DMIV, INIT);
```

where:

- `DATA` is the name of a data set that includes the dependent variable and independent variables. The dependent variable should be within (0, 1) interval.
- `DV` is the name of the dependent variable.
- `FD, SD` are the abbreviations (without quotes) for the parent and child distributions. The family and usage are similar to those in R as outlined in Section 2.1.
- `LMIV` are the names of the independent variables in the location submodel. Dummy-coded variables should be used to represent any categorical independent variables.
- `DMIV` are the names of the independent variables in the dispersion submodel.
- `INIT` are user-specified starting values for parameters, including intercepts. Different starting values are separated by '`|`', and in the same order as the names in `LMIV` and `DMIV`.

An example using the t2-t2 distribution to fit a null model is as follows:

```
%cdfquantreg(data, y, "t2", "t2");
```

The macro returns the usual `NLMIXED` model outputs including iteration history, convergence results, fit statistics, parameter estimates, Hessian matrix, and covariance matrix of the parameter estimates. The fit statistics consist of the common metrics including -2 log-likelihood, AIC, AICC and BIC. A new data-set called "dataout" is generated and stored in the working directory. In addition to the original data-set, "dataout" includes the fitted μ_i, σ_i, and y_i values, raw residuals, and Pearson residuals. The user can utilize these for further model diagnoses.

A.1. Ambiguity and conflict study with SAS

We replicate the example analyses from the ambiguity and conflict study by using the SAS macro. We first run the t2-t2 models to examine the model fit for an intercept-only model, a model with location predictors, and a model with both location and dispersion predictors.

```
%LET dir = ''; /* Specify the location of the macro files */
FILENAME tempdir &dir;
%INCLUDE tempdir(data); /* Load the ambiguity study data */
%INCLUDE tempdir(SAS_MACRO); /* Load the SAS cdfquantreg program */

/* Null Model */
%cdfquantreg(ambdata, prob, 't2', 't2');
/* Location model only */
%cdfquantreg(ambdata, prob, 't2', 't2', lmiv = scenarioStm3 scenarioStm4);
/* Dispersion + location model */
%cdfquantreg(ambdata, prob, 't2', 't2', lmiv = scenarioStm3 scenarioStm4,
  dmiv = scenarioStm3 scenarioStm4);
```

The output window displays both fit statistics and parameter estimates for the three models. We only shows the output for the last model as a demonstration below. SAS returns similar -2 log-likelihood values to those produced by the **cdfquantreg** package: $-5226, -5226$ and -5233 for the three models, respectively. The parameter estimates and their standard errors likewise are very similar to those produced by **cdfquantreg**.

```
/*Output for the full model*/
                                          Fit Statistics
                            -2 Log Likelihood                 -5233
                            AIC (smaller is better)           -5221
                            AICC (smaller is better)          -5221
                            BIC (smaller is better)           -5183

Parameter    Standard
 Estimate     Error     DF   t Value Pr > |t| Alpha   Lower     Upper   Gradient
b0 -0.01446   0.008057 4710 -1.79   0.0727    0.05  -0.03026  0.00133 -1.39E-7
b1 -0.00663   0.01102  4710 -0.60   0.5474    0.05  -0.02824  0.01498 -2.44E-9
b2 -0.00278   0.01064  4710 -0.26   0.7935    0.05  -0.02364  0.01807 -1.36E-7
d0 -1.3640    0.02981  4710 -45.76 <.0001     0.05  -1.4225  -1.3056  -2.78E-8
d1 -0.03381   0.04382  4710 -0.77   0.4404    0.05  -0.1197   0.05209 -832E-13
d2 -0.1140    0.04369  4710 -2.61   0.0091    0.05  -0.1997  -0.02840 -2.79E-8
```

We also run the three models by using the logit-logistic distribution. Again, SAS returns similar -2 log likelihood values, parameter estimates and standard errors to those produced by the cdfquantreg package.

```
%cdfquantreg(ambdata, prob, 'logit', 'logistic');
%cdfquantreg(ambdata, prob, 'logit', 'logistic',
  lmiv = scenarioStm3 scenarioStm4);
%cdfquantreg(ambdata, prob, 'logit', 'logistic',
  lmiv = scenarioStm3 scenarioStm4, dmiv = scenarioStm3 scenarioStm4);
```

```
/*Output for the full model*/
                                          Fit Statistics
```

```
                            -2 Log Likelihood              -3251
                            AIC (smaller is better)        -3239
                            AICC (smaller is better)       -3239
                            BIC (smaller is better)        -3200

Parameter    Standard
Estimate      Error    DF  t Value Pr > |t| Alpha   Lower    Upper   Gradient
b0 -0.01198  0.01689 4710  -0.71   0.4782  0.05  -0.04508  0.02113  2.862E-7
b1 -0.04324  0.02499 4710  -1.73   0.0837  0.05  -0.09224  0.00576  -1.75E-9
b2 -0.02270  0.02400 4710  -0.95   0.3441  0.05  -0.06975  0.02434  -828E-12
d0 -0.8940   0.02224 4710 -40.20   <.0001  0.05  -0.9376  -0.8504   -1.83E-7
d1  0.09691  0.03159 4710   3.07   0.0022  0.05   0.03498  0.1588   7.48E-10
d2  0.01986  0.03161 4710   0.63   0.5299  0.05  -0.04212  0.08183  1.419E-9
```

Finally, we replicate the arcsinh-t2 distribution run, and verify that SAS produces similar parameter estimates and log-likelihood values to those from **cdfquantreg**.

```
%cdfquantreg(ambdata, prob, 'arcsinh', 't2');
%cdfquantreg(ambdata, prob, 'arcsinh', 't2',
  lmiv = scenarioStm3 scenarioStm4);
%cdfquantreg(ambdata, prob, 'arcsinh', 't2',
  lmiv = scenarioStm3 scenarioStm4, dmiv = scenarioStm3 scenarioStm4);
```

```
/*Output for the full model*/
                                         Fit Statistics
                            -2 Log Likelihood              -5800
                            AIC (smaller is better)        -5796
                            AICC (smaller is better)       -5796
                            BIC (smaller is better)        -5783

Parameter    Standard
Estimate      Error    DF  t Value Pr > |t| Alpha   Lower     Upper   Gradient
b0 -0.01697  0.00628  4710  -2.70  0.0069  0.05  -0.02928  -0.00466  1.093E-9
b1  0.00948  0.00768  4710   1.23  0.2171  0.05  -0.00558   0.02454  -311E-14
b2  0.00875  0.00749  4710   1.17  0.2428  0.05  -0.00593   0.02342  1.099E-9
d0 -1.9774   0.03751  4710 -52.71  <.0001  0.05  -2.0510   -1.9039   -1.28E-9
d1 -0.2150   0.05705  4710  -3.77  0.0002  0.05  -0.3269   -0.1032   1.03E-11
d2 -0.2939   0.05709  4710  -5.15  <.0001  0.05  -0.4058   -0.1819   -1.27E-9
```

Affiliation:

Yiyun Shou
Research School of Psychology
The Australian National University
Canberra, Australian Capital Territory, Australia
E-mail: Yiyun.Shou@anu.edu.au

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
January 2019, Volume 88, Issue 1 *Submitted:* 2016-12-11
doi:10.18637/jss.v088.i01 *Accepted:* 2017-12-03