Francisco Cribari-Neto and Achim Zeileis

Beta Regression in R

Working Paper

Beta Regression in R

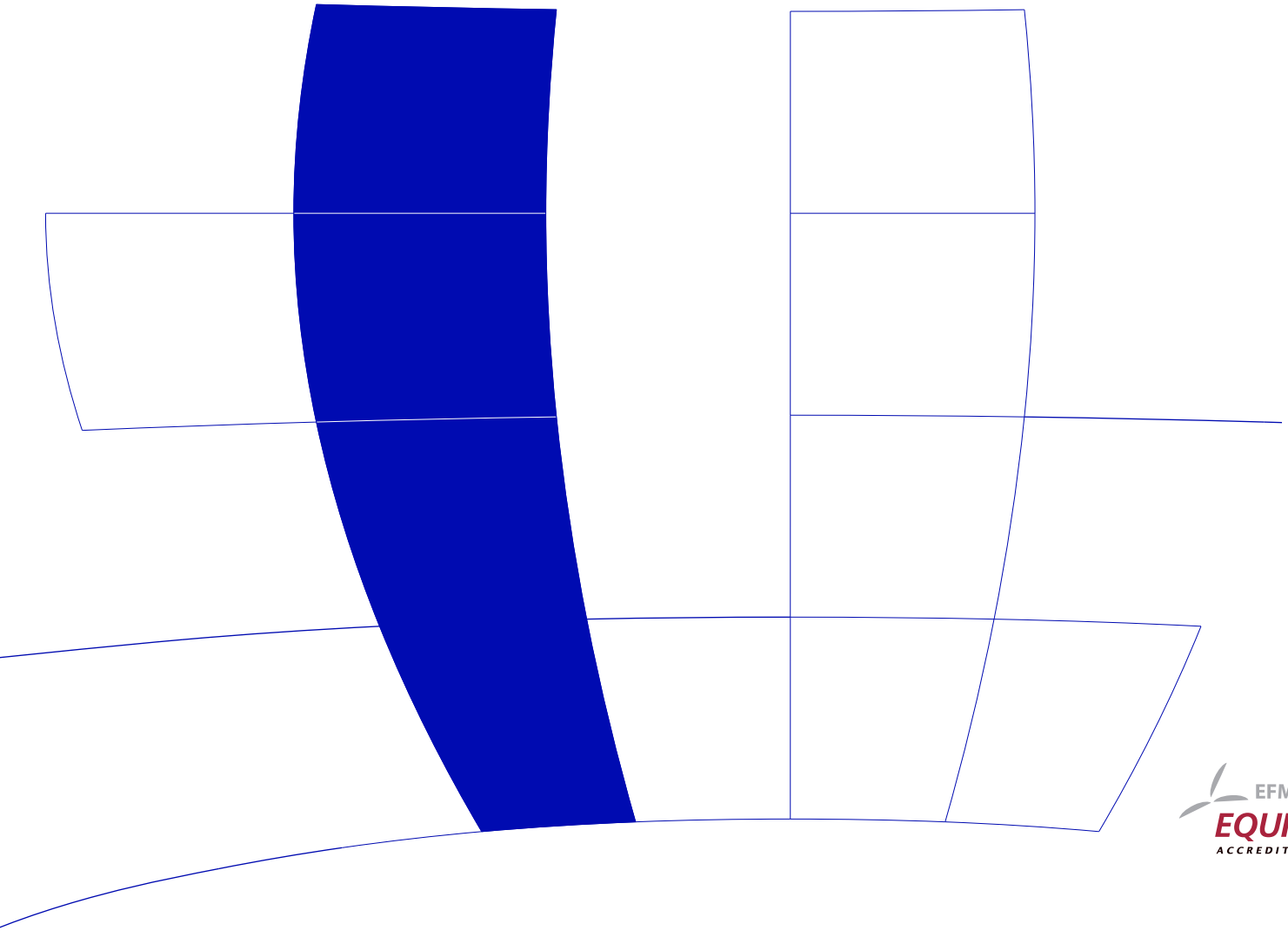
Francisco Cribari-Neto, Achim Zeileis

Department of Statistics and Mathematics
WU Wirtschaftsuniversität Wien

Research Report Series

Report 98
December 2009

http://statmath.wu.ac.at/



Beta Regression in R

Francisco Cribari-Neto
Universidade Federal de Pernambuco

Achim Zeileis
WU Wirtschaftsuniversität Wien

Abstract

The class of beta regression models is commonly used by practitioners to model variables that assume values in the standard unit interval $(0, 1)$. It is based on the assumption that the dependent variable is beta-distributed and that its mean is related to a set of regressors through a linear predictor with unknown coefficients and a link function. The model also includes a precision parameter which may be constant or depend on a (potentially different) set of regressors through a link function as well. This approach naturally incorporates features such as heteroskedasticity or skewness which are commonly observed in data taking values in the standard unit interval, such as rates or proportions. This paper describes the **betareg** package which provides the class of beta regressions in the R system for statistical computing. The underlying theory is briefly outlined, the implementation discussed and illustrated in various replication exercises.

Keywords: beta regression, rates, proportions, R.

1. Introduction

How should one perform a regression analysis in which the dependent variable (or response variable), y, assumes values in the standard unit interval $(0, 1)$? The usual practice used to be to transform the data so that the transformed response, say $\tilde{y}$, assumes values in the real line and then use it in a standard linear regression analysis. A commonly used transformation is the logit $\tilde{y} = \log(y/(1-y))$. This approach, nonetheless, has shortcomings. First, the regression parameters are interpretable in terms of the mean of $\tilde{y}$, and not in terms of the mean of y (given Jensen's inequality). Second, regressions involving data from the unit interval such as rates and proportions are typically heteroskedastic: they display more variation around the mean and less variation as we approach the lower and upper limits of the standard unit interval. Finally, the distributions of rates and proportions are typically asymmetric, and thus Gaussian-based approximations for interval estimation and hypothesis testing can be quite inaccurate in small samples. Ferrari and Cribari-Neto (2004) proposed a regression model for continuous variates that assume values in the standard unit interval, e.g., rates, proportions, or income concentration indices. Since the model is based on the assumption that the response is beta-distributed, they called their model *the beta regression model*. In their model, the regression parameters are interpretable in terms of the mean of y (the variable of interest) and the model is naturally heteroskedastic and easily accomodates asymmetries. A variant of the beta regression model that allows for nonlinearities and variable dispersion was proposed by Simas, Barreto-Souza, and Rocha (2010). In particular, in this more general model, the parameter that accounts for the precision of the data is not assumed to be constant across

observations but it is allowed to vary, which leads to the *variable dispersion beta regression model*.

In this paper, we describe the **betareg** package which can be used to perform inference in both fixed and variable dispersion beta regressions. The package is implemented in the R system for statistical computing (R Development Core Team 2009) and available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=betareg`. The initial version of the package was written by Simas and Rocha (2006) up to version 1.2 which was orphaned and archived on CRAN in mid-2009. Starting from version 2.0-0, Achim Zeileis took over maintenance after rewriting/extending the package's functionality.

The paper unfolds as follows: Section 2 outlines the theory underlying the beta regression model before Section 3 describes its implementation in R. Sections 4 and 5 provide various empirical applications: The former focuses on illustrating various aspects of beta regressions in practice while the latter provides further replications of previously published empirical research. Finally, Section 6 contains concluding remarks and directions for future research and implementation.

2. Beta regression

The class of beta regregression models, as introduced by Ferrari and Cribari-Neto (2004), is useful for modeling continuous variables y that assume values in the open standard unit interval $(0, 1)$. Note that if the variable takes on values in (a, b) (with $a < b$ known) one can model $(y - a)/(b - a)$. Furthermore, if y also assumes the extremes 0 and 1, a useful transformation in practice is $(y \cdot (n - 1) + 0.5)/n$ where n is the sample size (Smithson and Verkuilen 2006).

The beta regression model is based on an alternative parameterization of the beta density in terms of the variate mean and a precision parameter. The beta density is usually expressed as

$$f(y; p, q) = \frac{\Gamma(p + q)}{\Gamma(p)\Gamma(q)} y^{p-1}(1 - y)^{q-1}, \quad 0 < y < 1,$$

where $p, q > 0$ and $\Gamma(\cdot)$ is the gamma function. Ferrari and Cribari-Neto (2004) proposed a different parameterization by setting $\mu = p/(p + q)$ and $\phi = p + q$:

$$f(y; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1 - \mu)\phi)} y^{\mu\phi-1}(1 - y)^{(1-\mu)\phi-1}, \quad 0 < y < 1,$$

with $0 < \mu < 1$ and $\phi > 0$. We write $y \sim \mathcal{B}(\mu, \phi)$. Here, $\mathsf{E}(y) = \mu$ and $\mathsf{VAR}(y) = \mu(1-\mu)/(1+\phi)$. The parameter ϕ is known as the precision parameter since, for fixed μ, the larger ϕ the smaller the variance of y; ϕ^{-1} is a dispersion parameter.

Let $y_1, \ldots, y_n$ be a random sample such that $y_i \sim \mathcal{B}(\mu_i, \phi)$, $i = 1, \ldots, n$. The beta regression model is defined as

$$g(\mu_i) = x_i^\top \beta_i = \eta_i,$$

where $\beta = (\beta_1, \ldots, \beta_k)^\top$ is a $k \times 1$ vector of unknown regression parameters ($k < n$), η_i is a linear predictor and $x_i = (x_{i1}, \ldots, x_{ik})^\top$ is the vector of k regressors (or independent variables or covariates). Here, $g(\cdot) : (0, 1) \mapsto \mathbb{R}$ is a link function, which is strictly increasing and twice differentiable. Some useful link functions are: logit $g(\mu) = \log(\mu/(1 - \mu))$; probit

$g(\mu) = \Phi^{-1}(\mu)$, where $\Phi(\cdot)$ is the standard normal distribution function; complementary log-log $g(\mu) = \log\{-\log(1-\mu)\}$; log-log $g(\mu) = -\log\{-\log(\mu)\}$; and Cauchy $g(\mu) = \tan\{\pi(\mu - 0.5)\}$. Note that the variance of y is a function of μ which renders the regression model based on this parameterization naturally heteroskedastic. In particular,

$$\mathsf{VAR}(y_i) = \frac{\mu_i(1-\mu_i)}{1+\phi} = \frac{g^{-1}(x_i^\top\beta)[1-g^{-1}(x_i^\top\beta)]}{1+\phi}. \tag{1}$$

The log-likelihood function is $\ell(\beta,\phi) = \sum_{i=1}^n \ell_i(\mu_i,\phi)$, where

$$\begin{aligned} \ell_i(\mu_i,\phi) &= \log\Gamma(\phi) - \log\Gamma(\mu_i\phi) - \log\Gamma((1-\mu_i)\phi) + (\mu_i\phi - 1)\log y_i & (2)\\ &\quad +\{(1-\mu_i)\phi - 1\}\log(1-y_i). & (3)\end{aligned}$$

Notice that $\mu_i = g^{-1}(x_i^\top\beta)$ is a function of β, the vector of regression parameters. Parameter estimation is performed by maximum likelihood (ML).

An extension of the beta regression model above which was employed by Smithson and Verkuilen (2006) and formally introduced (along with further extensions) by Simas *et al.* (2010) is the variable dispersion beta regression model. In this model the precision parameter is not constant for all observations but instead modelled in a similar fashion as the mean parameter. More specifically, $y_i \sim \mathcal{B}(\mu_i,\phi_i)$ independently, $i = 1,\ldots,n$, and

$$\begin{aligned} g_1(\mu_i) &= \eta_{1i} = x_i^\top\beta, & (4)\\ g_2(\phi_i) &= \eta_{2i} = z_i^\top\gamma, & (5)\end{aligned}$$

where $\beta = (\beta_1,\ldots,\beta_k)^\top$, $\gamma = (\gamma_1,\ldots,\gamma_h)^\top$, $k+h < n$, are the sets of regression coefficients in the two equations, η_{1i} and η_{2i} are the linear predictors, and x_i and z_i are regressor vectors. As before, both coefficient vectors are estimated by ML, simply replacing ϕ by ϕ_i in Equation 2.

Simas *et al.* (2010) further extend the model above by allowing nonlinear predictors in Equations 4 and 5. Also, they have obtained analytical bias corrections for the ML estimators of the parameters, thus generalizing the results of Ospina, Cribari-Neto, and Vasconcellos (2006), who derived bias corrections for fixed dispersion beta regressions. However, as these extensions are not (yet) part of the **betareg** package, we confine ourselve to these short references and do not provide detailed formulas.

Various types of residuals are available for beta regression models. The raw response residuals $y_i - \hat\mu_i$ are typically not used due to the heteroskedasticity inherent in the model (see Equation 1). Hence, a natural alternative are *Pearson residuals* which Ferrari and Cribari-Neto (2004) call *standardized ordinary residual* and define as

$$r_{\mathrm{P},i} = \frac{y_i - \hat\mu_i}{\sqrt{\widehat{\mathsf{VAR}}(y_i)}}, \tag{6}$$

where $\widehat{\mathsf{VAR}}(y_i) = \hat\mu_i(1-\hat\mu_i)/(1+\hat\phi_i)$, $\hat\mu_i = g_1^{-1}(x_i^\top\,\hat\beta)$, and $\hat\phi_i = g_2^{-1}(z_i^\top\,\hat\beta)$. Similarly, deviance residuals can be defined in the standard way via signed contributions to the excess likelihood. Further residuals were proposed by Espinheira, Ferrari, and Cribari-Neto (2008b), in particular one residual with better properties that they named *standardized weighted residual 2*:

$$r_{\mathrm{sw2},i} = \frac{y_i^* - \hat\mu_i^*}{\sqrt{\hat v_i(1-h_{ii})}}, \tag{7}$$

where $y_i^* = \log\{y_i/(1-y_i)\}$ and $\mu_i^* = \psi(\mu_i\phi) - \psi((1-\mu_i)\phi)$, $\psi(\cdot)$ denoting the digamma function. Standardization is then by $v_i = \{\psi'(\mu_i\phi) + \psi'((1-\mu_i)\phi)\}$ and h_{ii}, the ith diagonal element of the hat matrix (for details see Ferrari and Cribari-Neto 2004; Espinheira *et al.* 2008b). As before, hats denote evaluation at the ML estimates.

3. Implementation in R

To turn the conceptual model from the previous section into computational tools in R, it helps to emphasize some properties of the model: It is a standard maximum likelihood (ML) task for which there is no closed-form solution but numerical optimization is required. Furthermore, the model shares some properties (such as linear predictor, link function, dispersion parameter) with generalized linear models (GLMs McCullagh and Nelder 1989), but it is not a special case of this framework (not even for fixed dispersion). There are various models with implementations in R that have similar features – here, we specifically reuse some of the ideas employed for generalized count data regression by Zeileis, Kleiber, and Jackman (2008).

The main model-fitting function in **betareg** is `betareg()` which takes a fairly standard approach for implementing ML regression models in R: `formula` plus `data` is used for data specification, then the likelihood and corresponding gradient (or estimating function) is set up, `optim()` is called for maximizing the likelihood, and finally an object of S3 class "`betareg`" is returned for which a large set of methods to standard generics is available. The workhorse function is `betareg.fit()` which provides the core computations without `formula`-related data pre- and post-processing.

The model-fitting function `betareg()` and its associated class are designed to be as similar as possible to the standard `glm()` function (R Development Core Team 2009) for fitting GLMs. An important difference is that there are potentially two equations for mean and precision (Equations 4 and 5, respectively), and consequently two regressor matrices, two linear predictors, two sets of coefficients, etc. In this respect, the design of `betareg()` is similar to the functions described by Zeileis *et al.* (2008) for fitting zero-inflation and hurdle models which also have two model components. The arguments of `betareg()` are

```
betareg(formula, data, subset, na.action, weights, offset,
  link = "logit", link.phi = NULL, control = betareg.control(...),
  model = TRUE, y = TRUE, x = FALSE, ...)
```

where the first line contains the standard model-frame specifications (see Chambers and Hastie 1992), the second line has the arguments specific to beta regression models and the arguments in the last line control some components of the return value.

If a `formula` of type `y ~ x1 + x2` is supplied, it describes y_i and x_i for the mean equation of the beta regression (4). In this case a constant ϕ_i is assumed, i.e., $z_i = 1$ and g_2 is the identity link, corresponding to the basic beta regression model as introduced in Ferrari and Cribari-Neto (2004). However, a second set of regressors can be specified by a two-part formula of type `y ~ x1 + x2 | z1 + z2 + z3` as provided in the **Formula** package (Zeileis and Croissant 2009). This model has the same mean equation as above but the regressors z_i in the precision equation (5) are taken from the `~ z1 + z2 + z3` part. The default link function in this case is the log link $g_2(\cdot) = \log(\cdot)$. Consequently, `y ~ x1 + x2` and `y ~ x1 + x2 | 1` correspond to equivalent beta likelihoods but use different parametrizations for ϕ_i:

Function	Description
`print()`	simple printed display with coefficient estimates
`summary()`	standard regression output (coefficient estimates, standard errors, partial Wald tests); returns an object of class "`summary.betareg`" containing the relevant summary statistics (which has a `print()` method)
`coef()`	extract coefficients of model (full, mean, or precision components), a single vector of all coefficients by default
`vcov()`	associated covariance matrix (with matching names)
`predict()`	predictions (of means μ_i, linear predictors η_{1i}, precision parameter ϕ_i, or variances $\mu_i(1-\mu_i)/(1+\phi_i)$) for new data
`fitted()`	fitted means for observed data
`residuals()`	extract residuals (deviance, Pearson, response, or different weighted residuals, see Espinheira *et al.* 2008b), defaulting to standardized weighted residuals 2 from Equation 7
`estfun()`	compute empirical estimating functions (or score functions), evaluated at observed data and estimated parameters (see Zeileis 2006b)
`bread()`	extract "bread" matrix for sandwich estimators (see Zeileis 2006b)
`terms()`	extract terms of model components
`model.matrix()`	extract model matrix of model components
`model.frame()`	extract full original model frame
`logLik()`	extract fitted log-likelihood
`plot()`	diagnostic plots of residuals, predictions, leverages etc.
`hatvalues()`	hat values (diagonal of hat matrix)
`cooks.distance()`	(approximation of) Cook's distance
`gleverage()`	compute generalized leverage (Wei, Hu, and Fung 1998); based on the formula derived for fixed ϕ
`coeftest()`	partial Wald tests of coefficients
`waldtest()`	Wald tests of nested models
`linear.hypothesis()`	Wald tests of linear hypotheses
`lrtest()`	likelihood ratio tests of nested models
`AIC()`	compute information criteria (AIC, BIC, ...)

Table 1: Functions and methods for objects of class "`betareg`". The first four blocks refer to methods, the last block contains generic functions whose default methods work because of the information supplied by the methods above.

simply $\phi_i = \gamma_1$ in the former case and $\log(\phi_i) = \gamma_1$ in the latter case. The link for the ϕ_i precision equation can be changed by `link.phi` in both cases where `"identity"`, `"log"`, and `"sqrt"` are allowed as admissible values. The default for the μ_i mean equation is always the logit link but all link functions for the `binomial` family in `glm()` are allowed as well as the log-log link: `"logit"`, `"probit"`, `"cloglog"`, `"cauchit"`, `"log"`, and `"loglog"`.

ML estimation of all parameters employing analytical gradients is carried out using R's

`optim()` with control options set in `betareg.control()`. Starting values can be user-supplied, otherwise the β starting values are estimated by a regression of $g_1(y_i)$ on x_i and similarly the γ starting values are obtained from a regression of a transformed y_i on the z_i. The transformed y_i are derived in Ferrari and Cribari-Neto (2004, p. 805) where only their mean is used as the starting values (corresponding to constant ϕ_i with identity link). The covariance matrix estimate is derived analytically as in Simas *et al.* (2010). However, by setting `hessian = TRUE` the numerical Hessian matrix returned by `optim()` can also be obtained.

The returned fitted-model object of class "`betareg`" is a list similar to "`glm`" objects. Some of its elements—such as `coefficients` or `terms`—are lists with a mean and precision component, respectively. A set of standard extractor functions for fitted model objects is available for objects of class "`betareg`", including the usual `summary()` method that includes partial Wald tests for all coefficients. No `anova()` method is provided, but the general `coeftest()` and `waldtest()` from **lmtest** (Zeileis and Hothorn 2002), and `linear.hypothesis()` from **car** (Fox 2002) can be used for Wald tests while `lrtest()` from **lmtest** provides for likelihood-ratio tests of nested models. See Table 1 for a list of all available methods. Most of these are standard in base R, however, methods to a few less standard generics are also provided. Specifically, there are tools related to specification testing and computation of sandwich covariance matrices as discussed by Zeileis (2004, 2006b) as well as a method to a new generic for computing generalized leverages (Wei *et al.* 1998).

4. Beta regression in practice

To illustrate the usage of **betareg** in practice we replicate and slightly extend some of the analyses from the original papers that suggested the methodology. More specifically, we estimate and compare various flavors of beta regression models for the gasoline yield data of Prater (1956), see Figure 1, and for the household food expenditure data taken from Griffiths, Hill, and Judge (1993), see Figure 3. Further pure replication exercises are provided in Section 5.

4.1. The basic model: Estimation, inference, diagnostics

Prater's gasoline yield data

The basic beta regression model as suggested by Ferrari and Cribari-Neto (2004) is illustrated in Section 4 of their paper using two empirical examples. The first example employs the well-known gasoline yield data taken from Prater (1956). The variable of interest is `yield`, the proportion of crude oil converted to gasoline after distillation and fractionation, for which a beta regression model is rather natural. Ferrari and Cribari-Neto (2004) employ two explanatory variables: `temp`, the temperature (in degrees Fahrenheit) at which all gasoline has vaporized, and `batch`, a factor indicating ten unique batches of conditions in the experiments (depending on further variables). The data, encompassing 32 observations, is visualized in Figure 1.

Ferrari and Cribari-Neto (2004) start out with a model where `yield` depends on `batch` and `temp`, employing the standard logit link. In **betareg**, this can be fitted via

```
R> data("GasolineYield", package = "betareg")
```

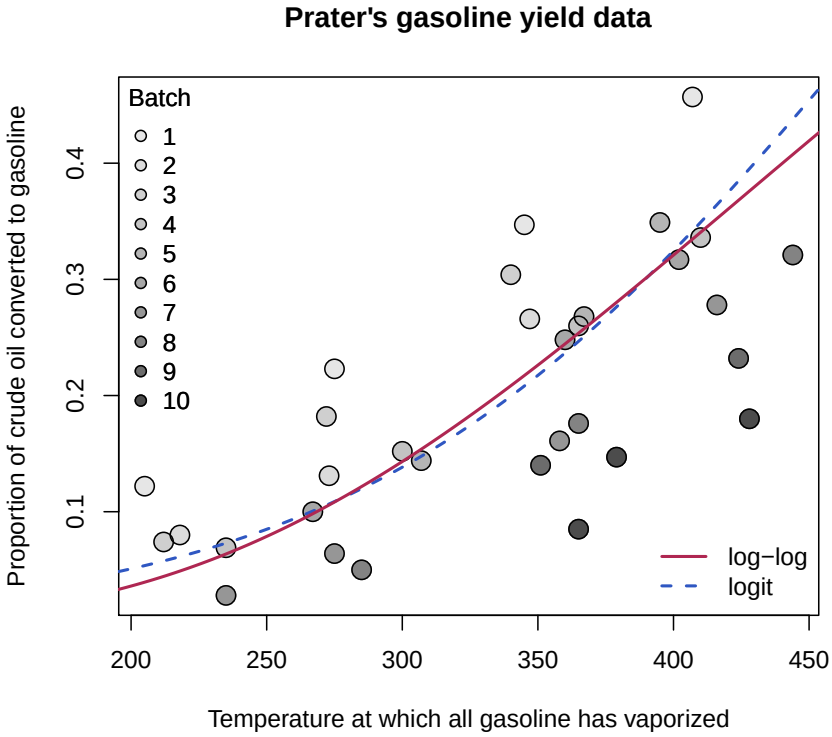


Figure 1: Gasoline yield data from Prater (1956): Proportion of crude oil converted to gasoline explained by temperature (in degrees Fahrenheit) at which all gasoline has vaporized and given batch (indicated by gray level). Fitted curves correspond to beta regressions `gy_loglog` with log-log link (solid, red) and `gy_logit` with logit link (dashed, blue). Both curves were evaluated at varying temperature with the intercept for batch 6 (i.e., roughly the average intercept).

```
R> gy_logit <- betareg(yield ~ batch + temp, data = GasolineYield)
R> summary(gy_logit)

Call:
betareg(formula = yield ~ batch + temp, data = GasolineYield)

Standardized weighted residuals 2:
    Min      1Q  Median      3Q     Max
-2.8750 -0.8149  0.1601  0.8384  2.0483

Coefficients (mean model with logit link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.1595710  0.1823247 -33.784  < 2e-16 ***
batch1       1.7277289  0.1012294  17.067  < 2e-16 ***
batch2       1.3225969  0.1179021  11.218  < 2e-16 ***
```

```
batch3       1.5723099  0.1161045  13.542  < 2e-16 ***
batch4       1.0597141  0.1023598  10.353  < 2e-16 ***
batch5       1.1337518  0.1035232  10.952  < 2e-16 ***
batch6       1.0401618  0.1060365   9.809  < 2e-16 ***
batch7       0.5436922  0.1091275   4.982 6.29e-07 ***
batch8       0.4959007  0.1089257   4.553 5.30e-06 ***
batch9       0.3857929  0.1185933   3.253  0.00114 **
temp         0.0109669  0.0004126  26.577  < 2e-16 ***

Phi coefficients (precision model with identity link):
      Estimate Std. Error z value Pr(>|z|)
(phi)    440.3      110.0   4.002 6.29e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood:  84.8 on 12 Df
Pseudo R-squared: 0.9617
Number of iterations in BFGS optimization: 52
```

which replicates their Table 1. The goodness of fit is assessed using different types of diagnostic displays shown in their Figure 2. This graphic can be reproduced (in a slightly different order) using the `plot()` method for "`betareg`" objects, see Figure 2.

```
R> set.seed(123)
R> plot(gy_logit, which = 1:4, type = "pearson")
R> plot(gy_logit, which = 5, type = "deviance", sub.caption = "")
R> plot(gy_logit, which = 1, type = "deviance", sub.caption = "")
```

As observation 4 corresponds to a large Cook's distance and large residual, Ferrari and Cribari-Neto (2004) decided to refit the model excluding this observation. While this does not change the coefficients in the mean model very much, the precision parameter ϕ increases clearly.

```
R> gy_logit4 <- update(gy_logit, subset = -4)
R> coef(gy_logit, model = "precision")

   (phi)
440.2783

R> coef(gy_logit4, model = "precision")

   (phi)
577.7907
```

Household food expenditures

Ferrari and Cribari-Neto (2004) also consider a second example: household food expenditure data for 38 households taken from Griffiths *et al.* (1993, Table 15.4). The dependent variable is

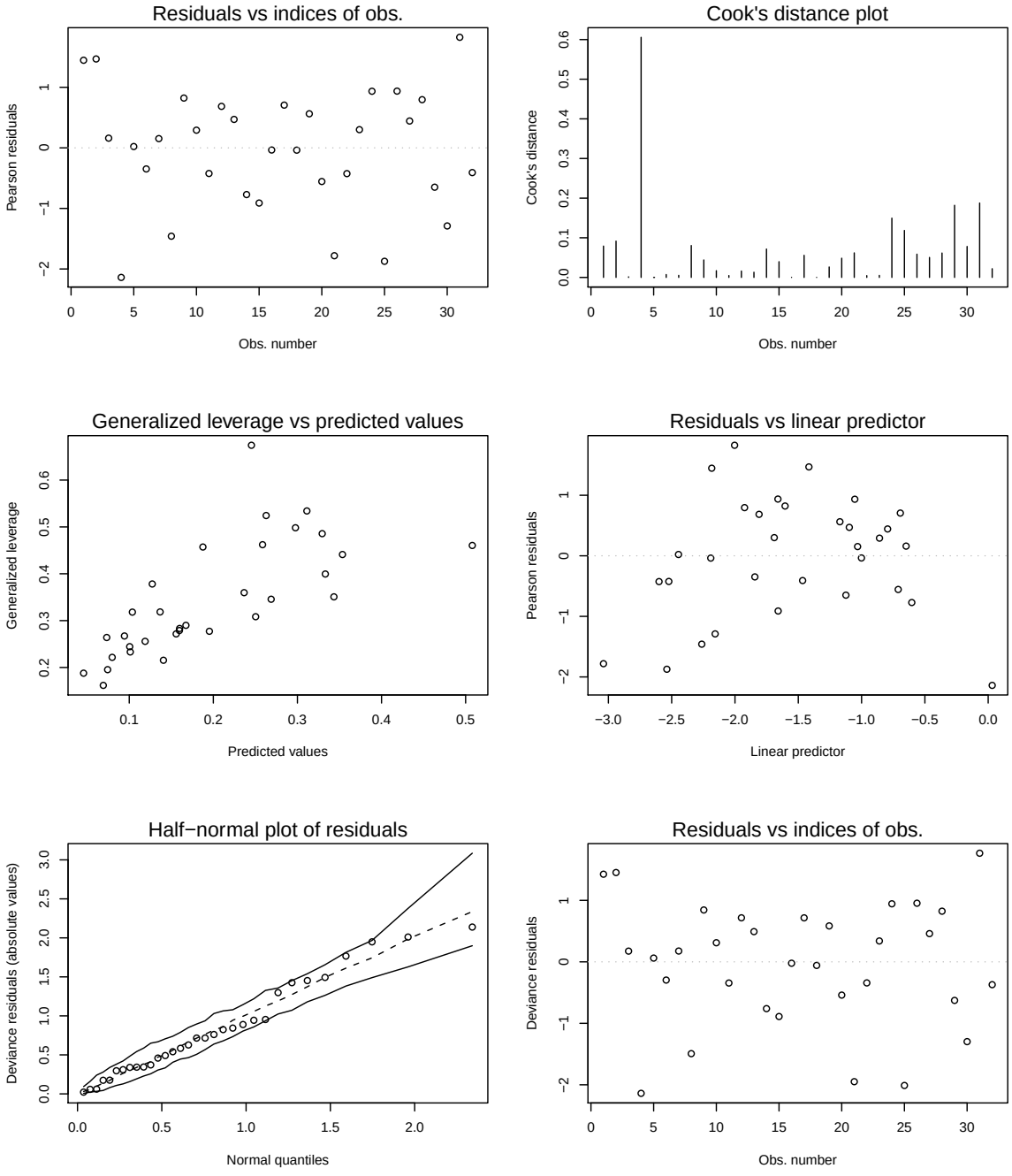


Figure 2: Diagnostic plots for beta regression model `gy_logit`.

`food/income`, the proportion of household `income` spent on `food`. Two explanatory variables are available: the previously mentioned household `income` and the number of `persons` living in the household. All three variables are visualized in Figure 3.

To start their analysis, Ferrari and Cribari-Neto (2004) consider a simple linear regression

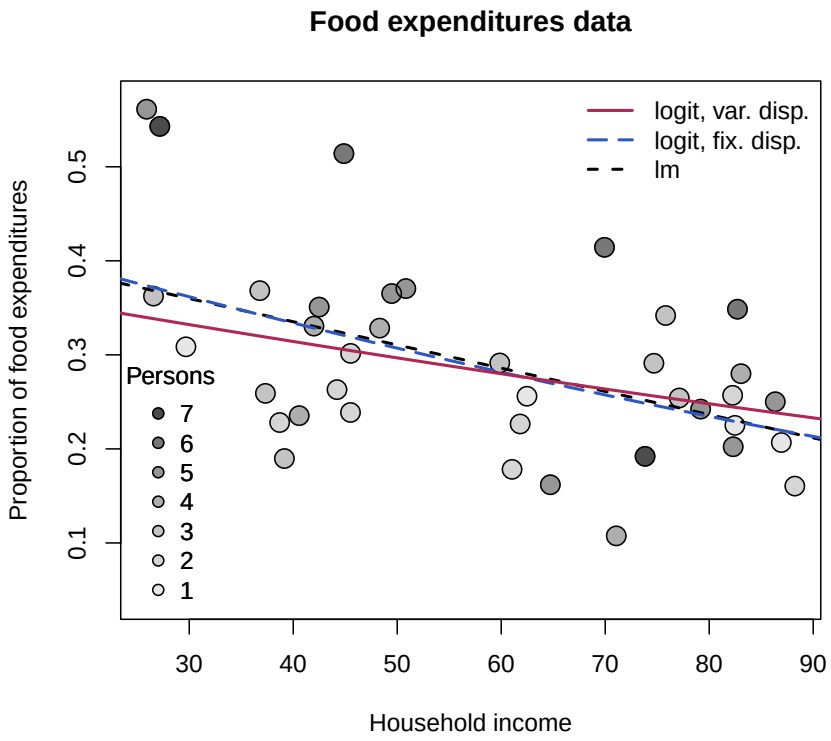


Figure 3: Household food expenditure data from Griffiths *et al.* (1993): Proportion of household income spent on food explained by household income and number of persons in household (indicated by gray level). Fitted curves correspond to beta regressions `fe_beta` with fixed dispersion (long-dashed, blue), `fe_beta2` with variable dispersion (solid, red), and the linear regression `fe_lin` (dashed, black). All curves were evaluated at varying income with the intercept for mean number of persons (= 3.58).

model fitted by ordinary least squares (OLS)

```
R> data("FoodExpenditure", package = "betareg")
R> fe_lm <- lm(I(food/income) ~ income + persons, data = FoodExpenditure)
```

To show that this model exhibits heteroskedasticity, they employ the studentized Breusch and Pagan (1979) test of Koenker (1981) which is available in R in the **lmtest** package (Zeileis and Hothorn 2002).

```
R> library("lmtest")
R> bptest(fe_lm)

        studentized Breusch-Pagan test

data:  fe_lm
BP = 5.9348, df = 2, p-value = 0.05144
```

One alternative would be to consider a logit-transformed response in a traditional OLS regression but this would make the residuals asymmetric. However, both issues – heteroskedasticity and skewness – can be alleviated when a beta regression model with a logit link for the mean is used.

```
R> fe_beta <- betareg(I(food/income) ~ income + persons,
+    data = FoodExpenditure)
R> summary(fe_beta)

Call:
betareg(formula = I(food/income) ~ income + persons, data = FoodExpenditure)

Standardized weighted residuals 2:
    Min      1Q  Median      3Q     Max
-2.7818 -0.4445  0.2024  0.6852  1.8755

Coefficients (mean model with logit link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.622547   0.223853  -2.781 0.005418 **
income      -0.012299   0.003036  -4.052 5.09e-05 ***
persons      0.118462   0.035341   3.352 0.000802 ***

Phi coefficients (precision model with identity link):
      Estimate Std. Error z value Pr(>|z|)
(phi)    35.61       8.08   4.407 1.05e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood: 45.33 on 4 Df
Pseudo R-squared: 0.3878
Number of iterations in BFGS optimization: 28
```

This replicates Table 2 from Ferrari and Cribari-Neto (2004). The predicted means of the linear and the beta regression model, respectively, are very similar: the proportion of household income spent on food decreases with the overall income level but increases in the number of persons in the household (see also Figure 3).

Below, further extended models will be considered for these data sets and hence all model comparisons are deferred.

4.2. Variable dispersion model

Prater's gasoline yield data

Although the beta model already incorporates naturally a certain pattern in the variances of the response (see Equation 1), it might be necessary to incorporate further regressors to account for heteroskedasticity as in Equation 5 (Simas *et al.* 2010). For illustration of this approach, the example from Section 3 of the online supplements to Simas *et al.* (2010) is

considered. This investigates Prater's gasoline yield data based on the same mean equation as above, but now with temperature `temp` as an additional regressor for the precision parameter ϕ_i:

```
R> gy_logit2 <- betareg(yield ~ batch + temp | temp, data = GasolineYield)
```

for which `summary(gy_logit2)` yields the MLE column in Table 19 of Simas *et al.* (2010). To save space, only the parameters pertaining to ϕ_i are reported here

```
            Estimate Std. Error z value  Pr(>|z|)
(Intercept) 1.3640889  1.2257812  1.1128    0.2658
temp        0.0145703  0.0036183  4.0269 5.653e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

which signal a significant improvement by including the `temp` regressor. Instead of using this Wald test, the models can also be compared by means of a likelihood-ratio test (see their Table 18) that confirms the results:

```
R> lrtest(gy_logit, gy_logit2)

Likelihood ratio test

Model 1: yield ~ batch + temp
Model 2: yield ~ batch + temp | temp
  #Df LogLik Df Chisq Pr(>Chisq)
1  12 84.798
2  13 86.977  1 4.359    0.03681 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that this can also be interpreted as testing the null hypothesis of equidispersion against a specific alternative of variable dispersion.

Household food expenditures

For the household food expenditure data, the Breusch-Pagan test carried out above illustrated that there is heteroskedasticity that can be captured by the regressors `income` and `persons`. Closer investigation reveals that this is mostly due to the number of persons in the household, also brought out graphically by some of the outliers with high values in this variable in Figure 3. Hence, it seems natural to consider the model employed above with `persons` as an additional regressor in the precision equation.

```
R> fe_beta2 <- betareg(I(food/income) ~ income + persons | persons,
+    data = FoodExpenditure)
```

This leads to significant improvements in terms of the likelihood and the associated BIC.

```
R> lrtest(fe_beta, fe_beta2)
```

```
Likelihood ratio test

Model 1: I(food/income) ~ income + persons
Model 2: I(food/income) ~ income + persons | persons
  #Df LogLik Df  Chisq Pr(>Chisq)
1   4 45.334
2   5 49.185  1 7.7029   0.005513 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
R> AIC(fe_beta, fe_beta2, k = log(nrow(FoodExpenditure)))

         df       AIC
fe_beta   4 -76.11667
fe_beta2  5 -80.18198
```

Thus, there is evidence for variable dispersion and model `fe_beta2` seems to be preferable. As visualized in Figure 3, it describes a similar relationship between response and explanatory variables although with a somewhat shrinked `income` slope.

4.3. Selection of different link functions

Prater's gasoline yield data

As in binomial GLMs, selection of an appropriate link function can greatly improve the model fit (McCullagh and Nelder 1989), especially if extreme proportions (close to 0 or 1) have been observed in the data. To illustrate this problem in beta regressions, we replicate parts of the analysis in Section 5 of Cribari-Neto and Lima (2007). This reconsiders Prater's gasoline yield data but employs a log-log link instead of the previously used (default) logit link

```
R> gy_loglog <- betareg(yield ~ batch + temp, data = GasolineYield,
+    link = "loglog")
```

which clearly improves pseudo R^2 of the model:

```
R> summary(gy_logit)$pseudo.r.squared

[1] 0.9617312

R> summary(gy_loglog)$pseudo.r.squared

[1] 0.9852334
```

Similarly, the AIC[1] (and BIC) of the fitted model is not only superior to the logit model with fixed dispersion `gy_logit` but also to the logit model with variable dispersion `gy_logit2` considered in the previous section.

[1]Note that Cribari-Neto and Lima (2007) did not account for estimation of ϕ in their degrees of freedom. Hence, their reported AICs differ by 2.

```
R> AIC(gy_logit, gy_logit2, gy_loglog)

           df       AIC
gy_logit   12 -145.5951
gy_logit2  13 -147.9541
gy_loglog  12 -168.3101
```

Moreover, if `temp` were included as a regressor in the precision equation of `gy_loglog`, it would no longer yield significant improvements. Thus, improvement of the model fit in the mean equation by adoption of the log-log link have waived the need for a variable precision equation.

To underline the appropriateness of the log-log specification, Cribari-Neto and Lima (2007) consider a sequence of diagnostic tests inspired by the RESET (regression specification error test; Ramsey 1969) in linear regression models. To check for misspecifications, they consider powers of fitted means or linear predictors to be included as auxiliary regressors in the mean equation. In well-specified models, these should not yield significant improvements. For the gasoline yield model, this can only be obtained for the log-log link while all other link functions result in significant results indicating misspecification. Below, this is exemplified for a likelihood-ratio test of squared linear predictors. Analogous results can be obtained for `type = "response"` or higher powers.

```
R> lrtest(gy_logit, . ~ . + I(predict(gy_logit, type = "link")^2))

Likelihood ratio test

Model 1: yield ~ batch + temp
Model 2: yield ~ batch + temp + I(predict(gy_logit, type = "link")^2)
  #Df LogLik Df  Chisq Pr(>Chisq)
1  12 84.798
2  13 96.001  1 22.407  2.205e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> lrtest(gy_loglog, . ~ . + I(predict(gy_loglog, type = "link")^2))

Likelihood ratio test

Model 1: yield ~ batch + temp
Model 2: yield ~ batch + temp + I(predict(gy_loglog, type = "link")^2)
  #Df LogLik Df  Chisq Pr(>Chisq)
1  12 96.155
2  13 96.989  1 1.6671     0.1966
```

The improvement of the model fit can also be brought out graphically in a display of predicted vs. observed values (see Figure 4).

```
R> plot(gy_logit, which = 6)
R> plot(gy_loglog, which = 6)
```

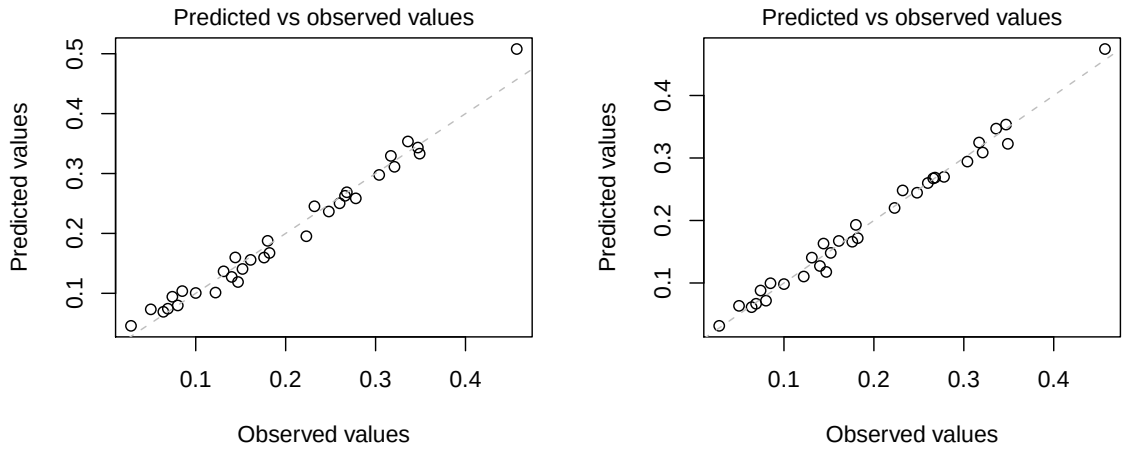


Figure 4: Diagnostic plots: Predicted vs. observed values for beta regression model `gy_logit` with logit link (left) and `gy_loglog` with log-log link (right).

This shows that especially for the extreme observations, the log-log link leads to better predictions.

In principle, the link function g_2 in the precision equation could also influence the model fit. However, as the best-fitting model `gy_loglog` has a constant ϕ, all links g_2 lead to equivalent estimates of ϕ and thus to equivalent fitted log-likelihoods. However, the link function can have consequences in terms of the inference about ϕ and in terms of convergence of the optimization. Typically, a log-link leads to somewhat improved quadratic approximations of the likelihood and less iterations in the optimization. For example, refitting `gy_loglog` with $g_2(\cdot) = \log(\cdot)$ converges more quickly:

```
R> gy_loglog2 <- update(gy_loglog, link.phi = "log")
R> summary(gy_loglog2)$iterations

[1] 21
```

with a lower number of iterations than for `gy_loglog` which had 51 iterations.

Household food expenditures

One could conduct a similar analysis as above for the household food expenditure data. However, as the response takes less extreme observations than for the gasoline yield data, the choice of link function is less important. In fact, refitting the model with various link functions shows no large differences in the resulting log-likelihoods.

```
R> sapply(c("logit", "probit", "cloglog", "cauchit", "loglog"), function(x)
+    logLik(betareg(I(food/income) ~ income + persons | persons,
+    link = x, data = FoodExpenditure)))
```

```
   logit   probit  cloglog  cauchit   loglog
49.18495 49.08044 49.35888 50.01105 48.86718
```

Only the Cauchy link performs slightly better than the logit link and might hence deserve further investigation.

5. Further replication exercises

In this section, further empirical illustrations of beta regressions are provided. While the emphasis in the previous section was to present how the various features of **betareg** can be used in pracice, we focus more narrowly on replication of previously published research articles below.

5.1. Dyslexia and IQ predicting reading accuracy

We consider an application that analyzes reading accuracy data for nondyslexic and dyslexic Australian children (Smithson and Verkuilen 2006). The variable of interest is `accuracy` providing the scores on a test of reading accuracy taken by 44 children, which is predicted by the two regressors `dyslexia` (a factor with sum contrasts separating a dyslexic and a control group) and nonverbal intelligent quotient (`iq`, converted to z scores), see Figure 5 for a visualization. The sample includes 19 dyslexics and 25 controls who were recruited from primary schools in the Australian Capital Territory. The children's ages ranged from eight years five months to twelve years three months; mean reading accuracy was 0.606 for dyslexic readers and 0.900 for controls.

Smithson and Verkuilen (2006) set out to investigate whether `dyslexia` contributes to the explanation of `accuracy` even when corrected for `iq` score (which is on average lower for dyslexics). Hence, they consider separate regressions for the two groups fitted by the interaction of both regressors. To show that OLS regression is no suitable tool in this situation, they first fit a linear regression of the logit-transformed response:

```
R> data("ReadingSkills", package = "betareg")
R> rs_ols <- lm(qlogis(accuracy) ~ dyslexia * iq, data = ReadingSkills)
R> coeftest(rs_ols)

t test of coefficients:

            Estimate Std. Error t value  Pr(>|t|)
(Intercept)  1.60107    0.22586  7.0888 1.411e-08 ***
dyslexia    -1.20563    0.22586 -5.3380 4.011e-06 ***
iq           0.35945    0.22548  1.5941   0.11878
dyslexia:iq -0.42286    0.22548 -1.8754   0.06805 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction effect does not appear to be significant, however this is a result of the poor fit of the linear regression as will be shown below. Figure 5 clearly shows that the data

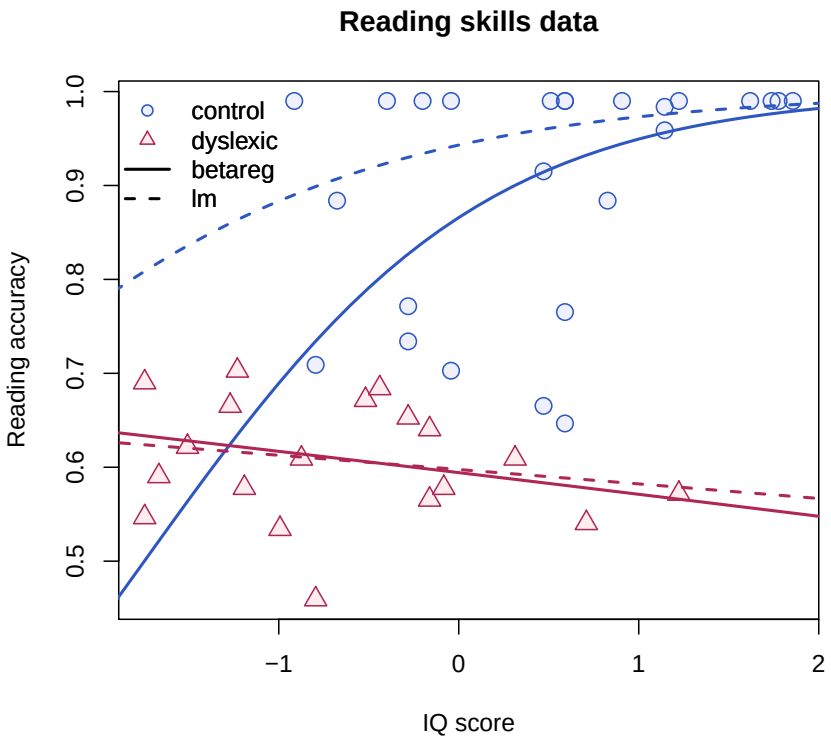


Figure 5: Reading skills data from Smithson and Verkuilen (2006): Linearly transformed reading accuracy by IQ score and dyslexia status (control, blue vs. dyslexic, red). Fitted curves correspond to beta regression `rs_beta` (solid) and OLS regression with logit-transformed dependent variable `rs_ols` (dashed).

are asymmetric and heteroskedastic (especially in the control group). Hence, Smithson and Verkuilen (2006) fit a beta regression model, again with separate means for both groups, but they also allow the dispersion to depend on the main effects of both variables.

```
R> rs_beta <- betareg(accuracy ~ dyslexia * iq | dyslexia + iq,
+    data = ReadingSkills, hessian = TRUE)
R> coeftest(rs_beta)

z test of coefficients:

                  Estimate Std. Error z value  Pr(>|z|)
(Intercept)        1.12323    0.15089  7.4441 9.758e-14 ***
dyslexia          -0.74165    0.15145 -4.8969 9.736e-07 ***
iq                 0.48637    0.16708  2.9109 0.0036034 **
dyslexia:iq       -0.58126    0.17258 -3.3681 0.0007568 ***
(phi)_(Intercept)  3.30443    0.22650 14.5890 < 2.2e-16 ***
(phi)_dyslexia     1.74656    0.29398  5.9410 2.832e-09 ***
```

```
(phi)_iq           1.22907    0.45957  2.6744 0.0074862 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This shows that precision increases with `iq` and is lower for controls while in the mean equation there is a significant interaction between `iq` and `dyslexia`. As Figure 5 illustrates, the beta regression fit does not differ much from the OLS fit for the dyslexics group (with responses close to 0.5) but fits much better in the control group (with responses close to 1).

The estimates above replicate those in Table 5 of Smithson and Verkuilen (2006), except for the signs of the coefficients of the dispersion submodel which they defined in the opposite way. Note that their results have been obtained with numeric rather than analytic standard errors hence `hessian = TRUE` is set above for replication. The results are also confirmed by Espinheira, Ferrari, and Cribari-Neto (2008a), who have also concluded that the dispersion is variable. As pointed out in Section 4.2, to formally test equidispersion against variable dispersion `lrtest(rs_beta, . ~ . | 1)` (or the analogous `waldtest()`) can be used.

Smithson and Verkuilen (2006) also consider two other psychometric applications of beta regressions the data for which are also provided in the **betareg** package: see `?MockJurors` and `?StressAnxiety`. Furthermore, `demo("SmithsonVerkuilen2006", package = "betareg")` is a complete replication script with comments.

5.2. Structural change testing in beta regressions

As already illustrated in Section 4, "`betareg`" objects can be plugged into various inference functions from other packages because they provide suitable methods to standard generic functions (see Table 1). Hence `lrtest()` could be used for performing likelihood-ratio testing inference and similarly `coeftest()`, `waldtest()` from **lmtest** (Zeileis and Hothorn 2002) and `linear.hypothesis()` from **car** (Fox 2002) can be employed for carrying out different flavors of Wald tests.

In this section, we illustrate yet another generic inference approach implemented in the **strucchange** package for structural change testing. While originally written for linear regression models (Zeileis, Leisch, Hornik, and Kleiber 2002), **strucchange** was extended by Zeileis (2006a) to compute generalized fluctuation tests for structural change in models that are based on suitable estimating functions. If these estimating functions can be extracted by an `estfun()` method, models can simply be plugged into the `gefp()` function for computing generalized empirical fluctuation processes. To illustrate this, we replicate the example from Section 5.3 in Zeileis (2006a).

Two artificial data sets are considered: a series `y1` with a change in the mean μ, and a series `y2` with a change in the precision ϕ. Both simulated series start with the parameters $\mu = 0.3$ and $\phi = 4$ and for the first series μ changes to 0.5 after 75% of the observations while ϕ remains constant whereas for the second series ϕ changes to 8 after 50% of the observations and μ remains constant.

```
R> set.seed(123)
R> y1 <- c(rbeta(150, 0.3 * 4, 0.7 * 4), rbeta(50, 0.5 * 4, 0.5 * 4))
R> y2 <- c(rbeta(100, 0.3 * 4, 0.7 * 4), rbeta(100, 0.3 * 8, 0.7 * 8))
```

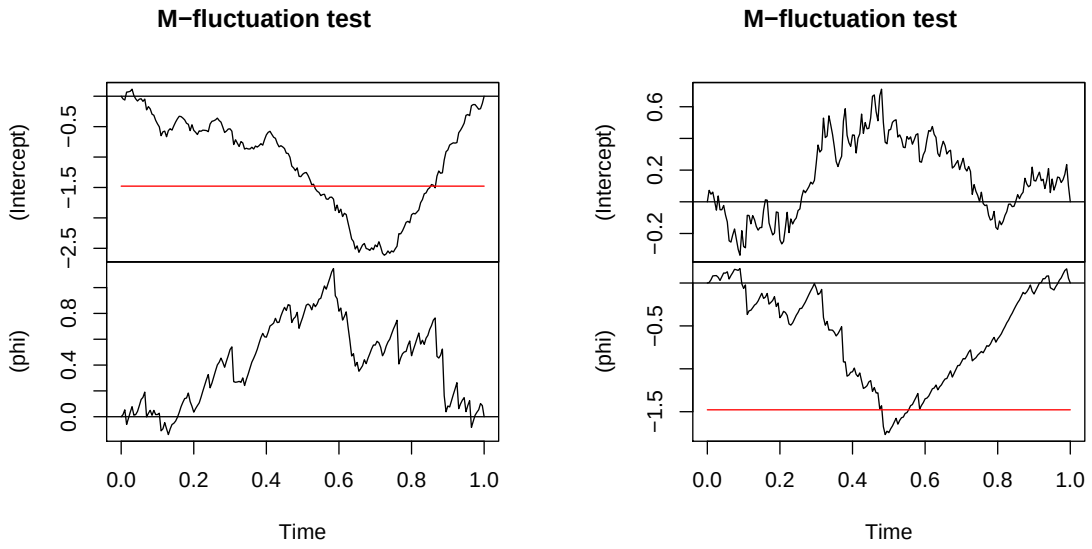


Figure 6: Structural change tests for artificial data `y1` with change in μ (left) and `y2` with change in ϕ (right).

To capture instabilities in the parameters over "time" (i.e., the ordering of the observations), the generalized empirical fluctuation processes can be derived via

```
R> library("strucchange")
R> y1_gefp <- gefp(y1 ~ 1, fit = betareg)
R> y2_gefp <- gefp(y2 ~ 1, fit = betareg)
```

and visualized by

```
R> plot(y1_gefp, aggregate = FALSE)
R> plot(y2_gefp, aggregate = FALSE)
```

The resulting Figure 6 (replicating Figure 4 from Zeileis 2006a) shows two 2-dimensional fluctuation processes: one for `y1` (left) and one for `y2` (right). Both fluctuation processes behave as expected: There is no excessive fluctuation of the process pertaining to the parameter that remained constant while there is a significant instability in the parameter that changed signalled by a boundary crossing and a peak at about the time of the change in the corresponding parameter.

6. Summary

This paper addressed the R implementation of the class of beta regression models available in the **betareg** package. We have presented the fixed and variable dispersion beta regression models, described how one can model rates and proportions using **betareg** and presented

several empirical examples reproducing previously published results. Future research and implementation shall focus on the situation where the data contain zeros and/or ones (see e.g., Kieschnick and McCullough 2003). An additional line of research and implementation is that of dynamic beta regression models, such as the class of βARMA models proposed by Rocha and Cribari-Neto (2010).

Acknowledgments

FCN gratefully acknowledges financial support from CNPq/Brazil. Both authors are grateful to A.B. Simas and A.V. Rocha for their work on the previous versions of the **betareg** package (up to version 1.2).

References

Breusch TS, Pagan AR (1979). "A Simple Test for Heteroscedasticity and Random Coefficient Variation." *Econometrica*, **47**, 1287–1294.

Chambers JM, Hastie TJ (eds.) (1992). *Statistical Models in S*. Chapman & Hall, London.

Cribari-Neto F, Lima LB (2007). "A Misspecification Test for Beta Regressions." *Technical report*.

Espinheira PL, Ferrari SLP, Cribari-Neto F (2008a). "Influence Diagnostics in Beta Regression." *Computational Statistics & Data Analysis*, **52**(9), 4417–4431.

Espinheira PL, Ferrari SLP, Cribari-Neto F (2008b). "On Beta Regression Residuals." *Journal of Applied Statistics*, **35**(4), 407–419.

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

Fox J (2002). *An R and S-PLUS Companion to Applied Regression*. Sage Publications, Thousand Oaks, CA.

Griffiths WE, Hill RC, Judge GG (1993). *Learning and Practicing Econometrics*. John Wiley & Sons, New York.

Kieschnick R, McCullough BD (2003). "Regression Analysis of Variates Observed on $(0, 1)$: Percentages, Proportions and Fractions." *Statistical Modelling*, **3**(3), 193–213.

Koenker R (1981). "A Note on Studentizing a Test for Heteroscedasticity." *Journal of Econometrics*, **17**, 107–112.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman & Hall, London.

Ospina R, Cribari-Neto F, Vasconcellos KLP (2006). "Improved Point and Interval Estimation for a Beta Regression Model." *Computational Statistics & Data Analysis*, **51**(2), 960–981.

Prater NH (1956). "Estimate Gasoline Yields from Crudes." *Petroleum Refiner*, **35**(5), 236–238.

Ramsey JB (1969). "Tests for Specification Error in Classical Linear Least Squares Regression Analysis." *Journal of the Royal Statistical Society B*, **31**, 350–371.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rocha AV, Cribari-Neto F (2010). "Beta Autoregressive Moving Average Models." *Test*. Forthcoming.

Simas AB, Barreto-Souza W, Rocha AV (2010). "Improved Estimators for a General Class of Beta Regression Models." *Computational Statistics & Data Analysis*, **54**(2), 348–366.

Simas AB, Rocha AV (2006). ***betareg**: Beta Regression*. R package version 1.2, URL http://CRAN.R-project.org/src/contrib/Archive/betareg/.

Smithson M, Verkuilen J (2006). "A Better Lemon Squeezer? Maximum-Likelihood Regression with Beta-Distributed Depedent Variables." *Psychological Methods*, **11**(1), 54–71.

Wei BC, Hu YQ, Fung WK (1998). "Generalized Leverage and Its Applications." *Scandinavian Journal of Statistics*, **25**(1), 25–37.

Zeileis A (2004). "Econometric Computing with HC and HAC Covariance Matrix Estimators." *Journal of Statistical Software*, **11**(10), 1–17. URL http://www.jstatsoft.org/v11/i10/.

Zeileis A (2006a). "Implementing a Class of Structural Change Tests: An Econometric Computing Approach." *Computational Statistics & Data Analysis*, **50**(11), 2987–3008.

Zeileis A (2006b). "Object-Oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. URL http://www.jstatsoft.org/v16/i09/.

Zeileis A, Croissant Y (2009). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Report 92*, Department of Statistics and Mathematics, WU Wirtschaftsuniversität Wien, Research Report Series. URL http://epub.wu.ac.at/.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10. URL http://CRAN.R-project.org/doc/Rnews/.

Zeileis A, Kleiber C, Jackman S (2008). "Regression Models for Count Data in R." *Journal of Statistical Software*, **27**(8), 1–25. URL http://www.jstatsoft.org/v27/i08/.

Zeileis A, Leisch F, Hornik K, Kleiber C (2002). "**strucchange**: An R Package for Testing for Structural Change in Linear Regression Models." *Journal of Statistical Software*, **7**(2), 1–38. URL http://www.jstatsoft.org/v07/i02/.

Affiliation:

Francisco Cribari-Neto
Departamento de Estatística, CCEN
Universidade Federal de Pernambuco
Cidade Universitária
Recife/PE 50740-540, Brazil
E-mail: cribari@ufpe.br
URL: http://www.de.ufpe.br/~cribari/

Achim Zeileis
Department of Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Augasse 2–6
1090 Wien, Austria
E-mail: Achim.Zeileis@R-project.org
URL: http://statmath.wu-wien.ac.at/~zeileis/