DirichletReg: Dirichlet Regression for Compositional Data in R

Marco J. Maier

Research Report Series
Report 125, January 2014

Institute for Statistics and Mathematics
http://statmath.wu.ac.at/

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

DirichletReg: Dirichlet Regression for Compositional Data in R

Marco J. Maier
Wirtschaftsuniversität Wien
Institute for Statistics and Mathematics

January 18, 2014

Dirichlet regression models can be used to analyze a set of variables lying in a bounded interval that sum up to a constant (e.g., proportions, rates, compositions, etc.) exhibiting skewness and heteroscedasticity, without having to transform the data. There are two parametrization for the presented model, one using the common Dirichlet distribution's α parameters, and a reparametrization of the αs to set up a mean–and–dispersion-like model. By applying appropriate link-functions, a GLM-like framework is set up that allows for the analysis of such data in a straightforward and familiar way, because interpretation is similar to multinomial logistic regression.

This paper gives a brief theoretical foundation and describes the implementation as well as application (including worked examples) of Dirichlet regression methods implemented in the package DirichletReg (Maier, 2013) in the R language (R Core Team, 2013).

Keywords: Dirichlet regression, Dirichlet distribution, multivariate generalized linear model, rates, proportions, rates, compositional data, simplex, R

1 Introduction

In many fields as diverse as geology, psychology, medicine, economics, etc., many variables are skewed, bounded in their range of possible values, or heteroscedastic. Oftentimes it is possible to counteract such phenomena by transforming the data to make them suitable for standard statistical methods relying on the assumption of normally-distributed data, yet in many cases this is not feasible. A 'family' of data that exhibit these characteristics contain compositional data, rates, proportions, amounts, etc.

Such data arise when we have at least $C = 2$ variables y_c[1] with values in a bounded interval, for example the standard unit interval $(0, 1)$. Generally, any interval (a, b) is possible since this can be transformed to $(0, 1)$. This would, in and of itself, not pose a problem, but in addition to the bounded range, the data *must* sum up to a constant m ($m = b - a = 1$ for data in $(0, 1)$) for each observation, so $\sum_{c=1}^{C} y_c = m$ which constrains the sample space of C dependent variables to a $C - 1$ dimensional simplex. In other words, one variable y_e can always be omitted, due to $y_e = m - \sum_{c \setminus e} y_c$. For three dimensions we arrive at a regular triangle as it is displayed in the left panel of Figure 1. If we rotate that triangle so that it is orthogonal to the viewpoint, we have a 2-dimensional object with which we

[1] Other sources denote the variables as i or j, the reason for this notation is that a large number of indices will be needed in the remainder of this paper and i and j are generally perceived to stand for observations and variables (for an overview and reference of matrices, indices, etc., please consult Table 1 on page 23 in the appendix of this text).

can *losslessly* display 3-dimensional data in two dimensions, because it is a plane — this type of plot is also called a *ternary plot* and is displayed in the right panel of that Figure.

By introducing this 'sum-constraint,' we have defined *compositional data* which reflect the composition of an observation on C variables (i.e., can be regarded as relative frequencies of the respective variables in each observation). Apart from being limited in its admissible values, these data are uncorrectably skewed and heteroscedastic in the original space.

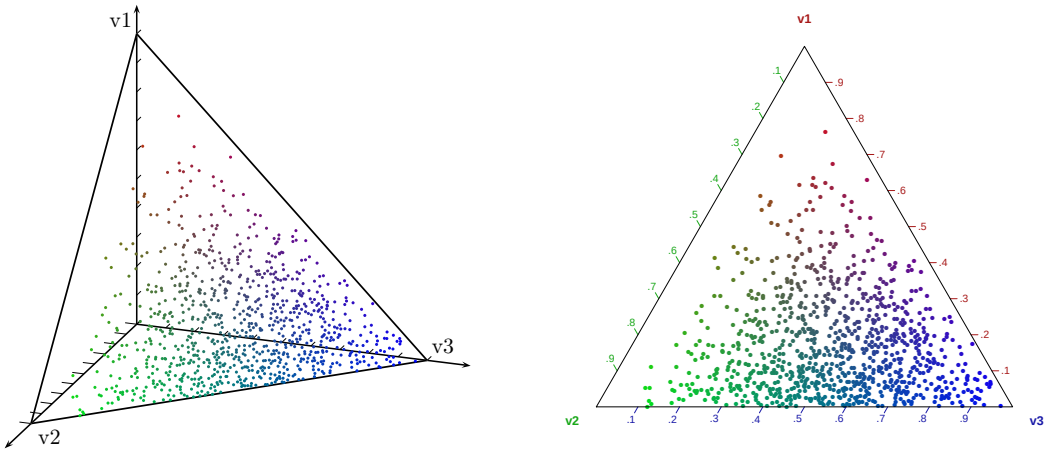


Figure 1: Data on a simplex in three and two dimensions.

Commonly, these data are transformed in various ways (e.g., isometric, centered, or additive log-ratio transformations) and then analyzed using *log-ratio analysis*, as pioneered by Aitchison (1982, 2003) where, for example, one variable is omitted and the other become the logarithmic ratios of the remaining variables divided by the omitted one ($y_c^\star = \log(y_c/y_e)$). Aitchison's methods for compositional data analysis in R are implemented in compositions by van den Boogaart et al. (2013); van den Boogaart and Tolosana-Delgado (2008) and robCompositions by Templ et al. (2013); Hron et al. (2010).

The transformed variables are then commonly analyzed using (multivariate multiple) linear regression models, which have some serious drawbacks. One of the most important difficulties for practitioners is that the resulting parameters are only interpretable in the transformed space and have no straightforward meaning. Moreover, if heteroscedasticity persists after the transformation, one has to either violate the assumption of homoscedasticity in linear models, or incorporate model terms capturing heteroscedasticity that further complicate the model (and interpretation). Barceló et al. (1996) have shown that log-ratio analysis can be seriously misleading in the presence of outliers.

The Dirichlet distribution's often evoked 'strong independence structure' has been shown to be negligible in a simulation study conducted by Brehm et al. (1998), where Dirichlet models performed as well as log-ratio analyses. Another commonly misunderstood property is the negative covariance structure among components of a *single* Dirichlet distribution — this does not apply to mixtures or covariate models, therefore a rejection of models using this distribution on these grounds is not justified (see, e.g., Hijazi and Jernigan, 2009; Hijazi, 2003). Furthermore, Dirichlet regression and beta regression (which is a special case of the former, as shown below) have been used in many papers yielding reasonable and interpretable results (e.g., Gueorguieva et al., 2008; Hijazi and Jernigan, 2009; Smithson and Verkuilen, 2006; Ferrari and Cribari-Neto, 2004; Cribari-Neto and Zeileis, 2010). For a comprehensive overview and juxtaposition of the various models, see Hijazi and Jernigan (2009); Hijazi (2003).

This package does not attempt to transform the data to correct for skewness, heteroscedasticity, as well as, bounded range. Like the package betareg (Zeileis et al., 2013a; Cribari-Neto and Zeileis, 2010) in R, this package uses strategies similar to generalized linear models (GLMs; see McCullagh and Nelder, 1989), so a suitable distribution for the data has to be chosen. The beta regression can be used to model compositional data with $C = 2$; a generalization for more than two variables can be obtained using the Dirichlet distribution. We will see that the Dirichlet distribution accounts for the — in a linear model — undesirable characteristics of compositional data and performs well in a multivariate GLM-like setting.

The presented models' parametrizations are divided in two types, heceforth called the *common* and *alternative* parametrization. One approach is based on work by Hijazi et al., who has explored Dirichlet regression models in his PhD thesis (Hijazi, 2003) and a subsequent article (Hijazi and Jernigan, 2009). Hijazi used the *common* approach where the Dirichlet distribution's α parameters are directly modeled by covariates using a log-link (this strategy is also used in papers such as Gueorguieva et al., 2008; Melo et al., 2009). The *alternative* parametrization is an extension of the approach implemented in betareg (see, e.g., Cribari-Neto and Zeileis, 2010; Ferrari and Cribari-Neto, 2004), where the Dirichlet distribution is reparametrized to allow for modeling *'means'* and *'dispersion'*. Heteroscedasticity is therefore taken in account implicitly in the common parametrization and explicitly in the alternative parametrization.

The presented methods are implemented in the R package DirichletReg. Stable versions of this package can be obtained from the Comprehensive R Archive Network (CRAN; `http://cran.r-project.org`) and development versions are hosted at R-Forge (`http://r-forge.r-project.org`).

In the remainder of this paper, section 2 focuses on theory and parametrizations, as well as model fitting, prediction, residuals, and model tests. Section 3 describes the implementation of the package along with some technical details concerning the fitting of the model (starting values, etc.). Published examples which are available in R are presented, worked and interpreted in section 4 (the complete code and output are supplied as a vignette in the package). This paper concludes with final remarks and plans for future work. The appendix consists of a short section with details on the rather complex notation needed for the models as a reference for readers, and the lengthy second derivatives of the alternative parametrization.

2 Dirichlet regression models

2.1 The Dirichlet distribution and its parametrizations

The Dirichlet distribution — as the generalization of the beta distribution — is defined in Equation 1, where α_c are the shape parameters for each variable. For this distribution, the constraints $\alpha_c > 0 \; \forall c$, $y_c \in (0, 1)$, and $\sum_{c=1}^{C} y_c = 1 \; \forall c$ have to hold.

$$\mathcal{D}(\boldsymbol{y} \,|\, \boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{c=1}^{C} y_c^{(\alpha_c - 1)} \tag{1}$$

The multinomial beta function, $\mathrm{B}(\alpha_c)$, serves as the normalizing constant and can be expressed as $\prod_{c=1}^{C} \Gamma(\alpha_c) / \Gamma(\sum_{c=1}^{C} \alpha_c)$ where $\Gamma(\cdot)$ is the gamma function defined as $\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) \, \mathrm{d}t$. If $C = 2$, Equation 1 reduces to the beta distribution. Furthermore, each variable is marginally beta-distributed with $\alpha = \alpha_i$ and $\beta = \alpha_0 - \alpha_i$; this is of great importance to find appropriate starting values for the common parametrization. As mentioned in the introduction, any kind of data in a bounded interval between a and b can be used, but they have to be transformed to suit the range of the Dirichlet distribution. If the original variables $y_c^\dagger$ are in (a, b), the transformation is $(y_c^\dagger - a)/(b - a)$.

$\boldsymbol{y} \sim \mathcal{D}(\boldsymbol{\alpha})$ denotes a variable that is Dirichlet-distributed with the common parametrization. For those variables, the sum of all αs — $\alpha_0 = \sum_{c=1}^{C} \alpha_c$ — can be interpreted as a 'precision' parameter (i.e., the higher this value, the more density is near the expected value) while the expected values are

defined as $\mathrm{E}[y_c] = \alpha_c/\alpha_0$, the variances are $\mathrm{VAR}[y_c] = [\alpha_c(\alpha_0 - \alpha_c)]/[\alpha_0^2(\alpha_0 + 1)]$ and the covariances are $\mathrm{COV}[y_i, y_j] = (-\alpha_i\alpha_j)/[\alpha_0^2(\alpha_0 + 1)]$.

The former derivation of expected values and the precision parameter leads directly to the *alternative parametrization* which is a generalization of Ferrari and Cribari-Neto (2004), if we define a set of parameters $\mu_c = \mathrm{E}[y_c]$ to account for the expected values of the variables y_c and $\phi = \alpha_0$ to model the dispersion (or precision in this case). This parametrization is symbolized by $\boldsymbol{y} \sim \mathcal{D}_a(\boldsymbol{\mu}, \phi)$. To convert μ and ϕ back to the Dirichlet distribution's original α parameters we define $\alpha_c = \mu_c\phi$ and $\alpha_0 = \phi$. Furthermore, we can introduce the following definitions: $\mathrm{E}[y_c] = \mu_c$, $\mathrm{VAR}[y_c] = [\mu_c(1 - \mu_c)]/(\phi + 1)$, and $\mathrm{COV}[y_i, y_j] = -\mu_i\mu_j/(\phi + 1)$. The reparametrized Dirichlet distribution therefore can be represented as in Equation 2 where $\mu_c \in (0, 1)$ and $\phi > 0$.

$$f(\boldsymbol{y} \mid \boldsymbol{\mu}, \phi) = \frac{1}{\mathrm{B}(\boldsymbol{\mu}\phi)} \prod_{c=1}^{C} y_c^{(\mu_c\phi - 1)} \tag{2}$$

Figure 2 shows the densities of three variables each and the relationship between their respective values of α, μ, and ϕ. In the upper panel of Figure 2, the distribution has an expected value of $\mu = (.6, .3, .1)$ (indicated by the red arrow) and precisions of $\phi = \alpha_0 = (1, 10, 20)$. The lower panel of Figure 2 displays a mean that is exactly in the middle of the simplex ($\mu = (1/3, 1/3, 1/3)$) with precisions of 1/4, 3, and 40. As we can observe, low precisions 'push' the densities towards the sides and corners of the simplex (i.e., more values close to 0 and 1) while higher precision centers the density around the expected value. $\boldsymbol{\alpha} = \mathbf{1}$ gives a (marginal) uniform distribution, values below 1 result in skewed U- or L-shaped forms while values above 1 result in a unimodal (marginal) distribution.

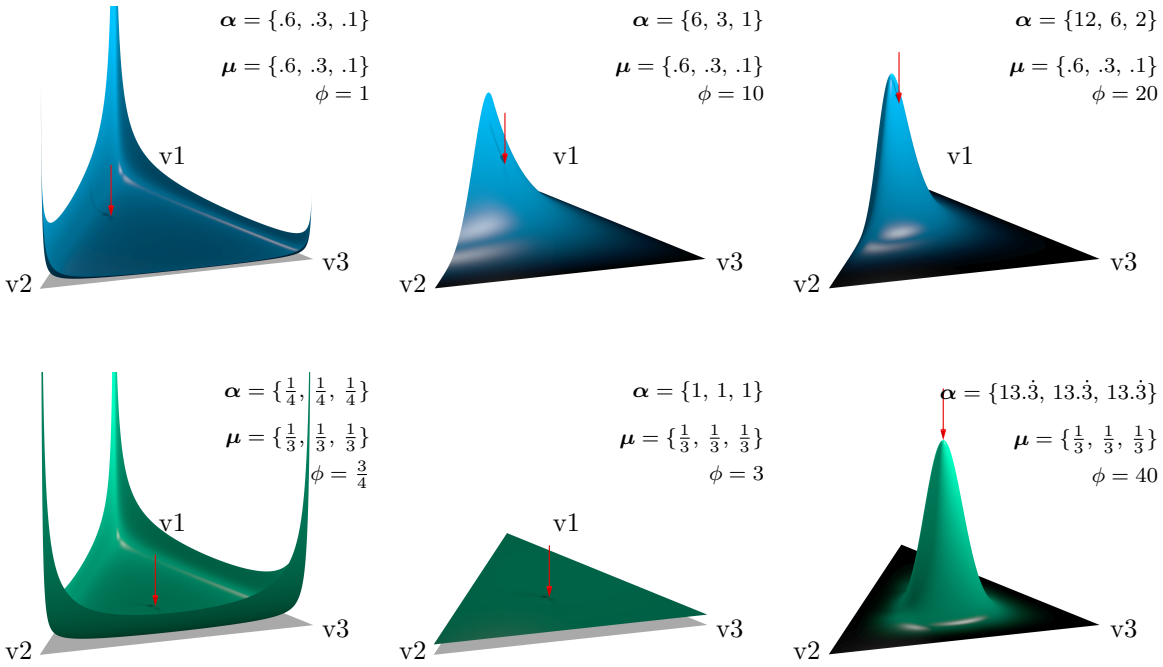


Figure 2: Various Dirichlet densities. The red arrows indicate the location of expected values. For the upper panel $\mu = \{.6, .3, .1\}$, and the lower panel $\mu = \{1/3, 1/3, 1/3\}$.

2.2 Dirichlet regression models

The two different parametrizations for the Dirichlet distribution can be used to set up appropriate linear predictors for the model along with their link-functions. This section will give the parametrization of the framework along with their respective (log-)likelihood functions. Because fitting of those

multivariate models is computationally demanding, first and second derivatives will be computed to accelerate optimization.

2.2.1 The common parametrization

The full log-likelihood of the commonly parametrized model (ℓ_c) is defined in Equation 3.

$$\ell_c(\boldsymbol{y}\,|\,\boldsymbol{\alpha}) = \log\Gamma(\sum_{c=1}^{C}\alpha_c) - \sum_{c=1}^{C}\log\Gamma(\alpha_c) + \sum_{c=1}^{C}(\alpha_c - 1)\log(y_c) \tag{3}$$

At this point we have practical implications, because data may come from the interval $[0, 1]$ instead of $(0, 1)$, which causes problems in the Equation above, because $\log(y_c = 0) = -\infty$. To remedy this, that is, if values of 0 and 1 are present in the data, we derive a generalization of the transformation proposed in Smithson and Verkuilen (2006), where N is the number of observations.

$$y^* = \frac{y(N-1) + 1/C}{N} \tag{4}$$

This 'compresses' the data symmetrically around .5 from a range of $m = 1$ to $(N-1)/N$, so extreme values are affected more than values lying close to 1/2. Additionally, we see that as $N \to \infty$ the compression vanishes, that is, larger data sets are less affected by this transformation. If we have a set of variables $y_c \sim \mathcal{D}(\alpha_c)$ for $c \in \{1, \ldots, C \geq 2\}$, the model can be set up as

$$g(\alpha_c) = \eta_c = \boldsymbol{X}^{[c]}\boldsymbol{\beta}^{[c]} \tag{5}$$

where $g(\cdot)$ is the link-function which will be the $\log(\cdot)$ for this model since $\alpha_c > 0$. Because we allow for separate predictors in each component c, the predictor matrix is represented as $\boldsymbol{X}^{[c]}$ where the superscript represents the predicted component. The potentially varying number of independent variables naturally leads to different numbers of regression coefficients in each dimension, which is accounted for by writing $\boldsymbol{\beta}^{[c]}$. The matrix of predictors for a component c, $\boldsymbol{X}^{[c]}$ has $i = N$ rows and $j = J^{[c]}$ columns while the regression coefficients $\boldsymbol{\beta}^{[c]}$ are a column vector with $j = J^{[c]}$ elements.

The first and second order derivatives have been derived and are displayed in Equations 7, 9, and 10. To shorten the terms and ease understanding we will rewrite the log-likelihood and define $\boldsymbol{A}^{[c]} = \boldsymbol{X}^{[c]}\boldsymbol{\beta}^{[c]}$. $\psi(\cdot)$ is the digamma function defined as $\psi(x) = \mathrm{d}/\mathrm{d}x \log\Gamma(x)$ and $\psi_1(\cdot)$ is the trigamma function $\psi_1(x) = \mathrm{d}^2/\mathrm{d}x^2 \log\Gamma(x)$.

$$\ell_c(y_c\,|\,\boldsymbol{X}^{[c]}, \boldsymbol{\beta}^{[c]}) = \log\Gamma(\sum_{c=1}^{C}\boldsymbol{A}^{[c]}) - \sum_{c=1}^{C}\log\Gamma(\boldsymbol{A}^{[c]}) + \sum_{c=1}^{C}(\boldsymbol{A}^{[c]} - 1)\log(\boldsymbol{Y}^{[c]}) \tag{6}$$

$$\frac{\partial \ell_c}{\partial \beta_m^{[d]}} = x_m^{[c]}\alpha^{[d]}\left[\log(y^{[d]}) - \psi(\alpha^{[d]}) + \psi(\sum_{c=1}^{C}\alpha^{[c]})\right] \tag{7}$$

A schematized version of the Hessian matrix is displayed in Equation 8 below where we have to employ two different equations to obtain values for βs on the same or different variables. In the main diagonal, Equation 9 is used while for the other cells Equation 10 is applied.

$$H = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} \frac{\partial^2 \ell_c}{\partial \beta_m^{[d=1]} \partial \beta_n^{[d=1]}} & & \\ \frac{\partial^2 \ell_c}{\partial \beta_m^{[d=2]} \partial \beta_n^{[e=1]}} & \frac{\partial^2 \ell_c}{\partial \beta_m^{[d=2]} \partial \beta_n^{[d=2]}} & \\ \frac{\partial^2 \ell_c}{\partial \beta_m^{[d=3]} \partial \beta_n^{[e=1]}} & \frac{\partial^2 \ell_c}{\partial \beta_m^{[d=3]} \partial \beta_n^{[e=2]}} & \frac{\partial^2 \ell_c}{\partial \beta_m^{[d=3]} \partial \beta_n^{[d=3]}} \end{bmatrix} \end{array} \tag{8}$$

$$\frac{\partial^2 \ell_c}{\partial \beta_m^{[d]} \partial \beta_n^{[d]}} = x_m^{[d]} x_n^{[d]} \alpha^{[d]} \left\{ \log(y_d) + \psi(\sum_{c=1}^{C} \alpha_c) - \psi(\alpha_d) + \alpha_d \left[\psi_1(\sum_{c=1}^{C} \alpha_c) - \psi_1(\alpha_d) \right] \right\} \tag{9}$$

$$\frac{\partial^2 \ell_c}{\partial \beta_m^{[d]} \partial \beta_n^{[e]}} = x_m^{[d]} x_n^{[e]} \alpha_d \alpha_e \psi_1(\sum_{c=1}^{C} \alpha_c) \tag{10}$$

2.2.2 The alternative parametrization

For this parametrization, the log-likelihood is the same as in Equation 3, because $\boldsymbol{\alpha} = \boldsymbol{\mu}\phi$ holds. We have $y_c \sim \mathcal{D}_a(\mu_c, \phi)$ for $c \in \{1, \ldots, C \geq 2\}$ and the regression is defined as

$$\begin{aligned} g_\mu(\mu_c) &= \eta_{\mu c} = \boldsymbol{X}\boldsymbol{\beta}_c \\ g_\phi(\phi) &= \eta_\phi = \boldsymbol{Z}\boldsymbol{\gamma} \end{aligned}$$

where $g_\mu(\cdot)$ and $g_\phi(\cdot)$ are the two required link functions. A natural choice for $g_\mu(\cdot)$ modeling the mean, which can take values in $(0, 1)$, is the logit while for $g_\phi(\cdot)$, predicting $\phi > 0$, we again choose the log. Because we have C categories for which the means must always sum up to 1, we employ a multinomial logit strategy as in multinomial regression, where the regression coefficients of one category b (base category) are set to zero, $\beta_b = \mathbf{0}$, whereby this variable is virtually omitted and becomes the reference. It is important to point out that the usage of a (multivariate) logit-link leads to parameter estimates that are interpretable as odds ratios if exponentiated (see Ferrari and Cribari-Neto, 2004), which is a great advantage. The individual expected values are therefore modeled in this fashion:

$$\begin{aligned} \mu_c &= \frac{\exp(\boldsymbol{X}\beta_c)}{\sum_{a=1}^{C} \exp(\boldsymbol{X}\beta_a)} \\ \mu_b &= \frac{\exp(\boldsymbol{X}\mathbf{0})}{\sum_{a=1}^{C} \exp(\boldsymbol{X}\beta_a)} = \frac{1}{\sum_{a=1}^{C} \exp(\boldsymbol{X}\beta_a)} \end{aligned} \tag{11}$$

Next follow equations for the log-likelihood (12) and the first (13 and 14) and second order derivatives (22, 23, 24, and 25). To simplify notation, we define $\boldsymbol{\alpha_c} = \boldsymbol{\mu_c}\boldsymbol{\beta_c}$, $\epsilon_c = \exp(\boldsymbol{X}_c\beta_c)$ and $\epsilon^{\Sigma} = \sum_{c=1}^{C} \epsilon_c$. We also need the term $\epsilon^{\Sigma\Sigma} = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} \epsilon_i \epsilon_j$ which gives the sum of all pairwise elements of ϵ_c (NB: this term is zero for $C = 2$).

$$\ell_a(y_c | x_j, \beta_c, z_l, \gamma) = \log\Gamma(\phi) - \sum_{c=1}^{C} \log\Gamma(\alpha_c) + \sum_{c=1}^{C} (\alpha_c - 1) \log(\boldsymbol{y_c}) \tag{12}$$

$$\frac{\partial \ell_a}{\partial \beta_{md}} = x_m \epsilon_c \phi \left[(y_d - \psi(\alpha_d)) \sum_{\substack{c=1 \\ c \neq b}}^{C} \epsilon_c + \sum_{\substack{c=1 \\ c \neq b}}^{C} \epsilon_c \psi(\alpha_c) - \log(y_{C \setminus b}) \right] \Big/ (\epsilon^{\Sigma})^2 \tag{13}$$

$$\frac{\partial \ell_a}{\partial \gamma_m} = z_m \left[\phi \cdot \psi(\phi) + \sum_{c=1}^{C} \alpha_c (\log(y_c) - \psi(\alpha_c)) \right] \tag{14}$$

The Hessian is made up of four 'blocks' in this case: the first for β parameters on the same response variable (as in cells $\{1, 1\}$ and $\{2, 2\}$), then for βs on two different responses (as in $\{2, 1\}$), for mixed β and γ parameters ($\{3, 1\}$ and $\{3, 2\}$) and finally the block containing two γs ($\{3, 3\}$).

$$H = \begin{array}{c} \\ 1 \\ 2 \\ \gamma \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & \gamma \end{array} \\ \begin{bmatrix} \frac{\partial^2 \ell_a}{\partial \beta_{md=1} \partial \beta_{nd=1}} & & \\ \frac{\partial^2 \ell_a}{\partial \beta_{md=2} \partial \beta_{ne=1}} & \frac{\partial^2 \ell_a}{\partial \beta_{md=2} \partial \beta_{nd=2}} & \\ \frac{\partial^2 \ell_a}{\partial \beta_{md=1} \partial \gamma_n} & \frac{\partial^2 \ell_a}{\partial \beta_{md=2} \partial \gamma_n} & \frac{\partial^2 \ell_a}{\partial \gamma_m \partial \gamma_n} \end{bmatrix} \end{array} \tag{15}$$

2.3 Fitted values, prediction, and residuals

The fitted values for y_c can be easily obtained given the models' parameters $\hat{\beta}$ (and $\hat{\gamma}$ for the alternative parametrization) and predictors $\boldsymbol{X}$ (and $\boldsymbol{Z}$). Likewise, predictions can be made for new values of x (and z) as we will see in the examples below.

Of the residual types proposed in Gueorguieva et al. (2008), raw, Pearson, and composite residuals are currently implemented. The *raw residuals* are defined in Eq. 16.

$$\begin{aligned} r_c^{\text{raw}} &= y_c - \mathrm{E}[y_c \mid \hat{\boldsymbol{\alpha}}] \\ &= y_c - \frac{\hat{\alpha}_c}{\hat{\alpha}_0} && \text{common parametrization} \\ &= y_c - \hat{\mu}_c && \text{alternative parametrization} \end{aligned} \tag{16}$$

Because of the skewness and heteroscedasticity, these kinds of residuals are only of limited importance. A more meaningful alternative are *Pearson residuals* which are called *standardized* and defined in Equation 17.

$$\begin{aligned} r_c^{\text{Pearson}} &= \frac{y_c - \mathrm{E}[y_c \mid \hat{\boldsymbol{\alpha}}]}{\sqrt{\mathrm{VAR}[y_c \mid \hat{\boldsymbol{\alpha}}]}} \\ &= \frac{y_c - \hat{\alpha}_c/\hat{\alpha}_0}{\sqrt{[\hat{\alpha}_c(\hat{\alpha}_0 - \hat{\alpha}_c)]/[\hat{\alpha}_0^2(\hat{\alpha}_0 + 1)]}} && \text{common parametrization} \\ &= \frac{y_c - \hat{\mu}_c}{\sqrt{\hat{\phi}}} && \text{alternative parametrization} \end{aligned} \tag{17}$$

The *composite residual* is the sum of the squared Pearson residuals, defined in Equation 18.

$$r_c^{\text{composite}} = \sum_{c=1}^{C} \left(r_c^{\text{Pearson}} \right)^2 \tag{18}$$

2.4 Confidence intervals and the Covariance Matrix of Parameters

The covariance matrix $\hat{\Sigma}(\hat{\theta})$ of the parameters can be obtained from the Hessian using $-H(\hat{\theta})^{-1}$. The standard errors (SE) in turn result from $\mathrm{SE}(\hat{\theta}) = \sqrt{\mathrm{diag}(\hat{\Sigma}(\hat{\theta}))}$.

Using those standard errors, we can construct confidence intervals (CIs). For any parameter $\hat{\theta}$, we can calculate the CI following Equation 19.

$$\mathrm{CI}_{\hat{\theta}} = \left[\hat{\theta} - z \cdot \mathrm{SE}(\hat{\theta}),\ \hat{\theta} + z \cdot \mathrm{SE}(\hat{\theta}) \right] \tag{19}$$

Here, z is the standard normal distribution's quantile that determines the level of confidence. To ease interpretation, it may be helpful to use exp(CI) in some situations; see below.

2.5 Model selection and tests

In order to select and test (nested) models (e.g., a vs. b) against each other, a likelihood-ratio-test can be applied.

$$LRT = -2\log\left(\frac{L_b}{L_a}\right) \overset{\text{as.}}{\sim} \chi^2_{n_a - n_b} \tag{20}$$

$$LRT = -2(\ell_b - \ell_a) \overset{\text{as.}}{\sim} \chi^2_{n_a - n_b} \tag{21}$$

In the above Equations 20 and 21, we test model b that is nested in a. The resulting test statistic is, under the null hypothesis, asymptotically χ^2-distributed with the degrees of freedom equal to the difference in model parameters ($n_a - n_b$).

3 Implementation in R

3.1 Interface

The models elaborated in the previous section are implemented in R using S3 classes. To fit models, the function `DirichReg()` is used which processes dependent variables, prepared by an auxilliary function `DR_data()` first, along with respective predictors.

```
DR_data(Y, trafo = sqrt(.Machine$double.eps), base = 1)
```

The argument `Y` must be either a matrix of observations with ≥ 2 columns or a vector of values in $[0, 1]$. In the first case, the variables are checked and normalized to fit in the interval $(0, 1)$ if they lie outside this range (i.e., do not sum up to 1 for each observation). If only a vector of values in $[0, 1]$ is supplied, the function assumes it to be a beta-distributed (i.e., Dirichlet-distributed with $C = 2$) variable and acts accordingly.

If extreme values of 0 and 1 are present, the data must be transformed as described in Equation 4 given above. However, due to numerical reasons, values close to either extreme can also cause problems, therefore the Argument `trafo` is principally regarded as a numeric threshold. By default, the square root of the machine epsilon ($\epsilon_M =$ `.Machine$double.eps` $\approx 2.22 \cdot 10^{-16}$; $\sqrt{\epsilon_M} \approx 1.49 \cdot 10^{-08}$) will be used, so if there are values $y < \sqrt{\epsilon_M}$ or $y > 1 - \sqrt{\epsilon_M}$ transformation will be carried out. Of course, other values can be used here, for example, `trafo = 1e-10`. To ensure comparability with other analyses, this behavior can be forced by setting the argument `trafo = TRUE`. Suppression of any transformation can be done using `trafo = FALSE` which will generate an error if values in `Y` are *exactly* 0 or 1.

If the alternative parametrization will be used, the base (i.e., omitted) variable, which is the first by default, can be set to be any other of the variables. The transformed variables *must* be re-integrated into the original data frame with a new, unique name (e.g., `DF$Y <- DR_data(DF[, 1:3])`; see examples below) which is then passed on the the fitting function.

After processing the model formula (using the Formula package by Zeileis and Croissant, 2013, 2010) which is similar to those used in `betareg()`, the internal function `DirichReg_fit()` fits the model using maximum-likelihood methods implemented in the package maxLik (Toomet et al., 2013; Henningsen and Toomet, 2010).

```
DirichReg(formula, data, model = c("common", "alternative"), subset,
          sub.comp, base, weights, control, verbosity = 0)
```

`formula` contains the code to set up the desired covariates. In its simplest form `formula` can be `dv ~ 1` which leads to a model including intercepts only. The parametrization of model is determined by `model` which can be `"common"` (i.e., all α parameters are modeled independently) or `"alternative"` (i.e., means and precision are modeled). Of course it is possible to use this formula interface as it is common practice in R, for example, `y ~x1 * x2 + x3 + I(x3^2)`. This formula sets up predictors that are the same for all dependent variables in the common parametrization. If one wishes to use different sets of predictors for the αs, pipes '`|`' are used in the formula to separate the blocks. `y ~ x1 + x2 | x1 * x3 | 1` sets up the blocks for three dependent variables' αs of which the first has the predictors `x1 + x2`, the second has `x1 * x3` and the third has only the intercept `1`. Of course, the number of blocks must match the number of dependent variables.

Like in betareg, the alternative parametrization is defined using two blocks of variables. The first gives the dependent variable and predictors for the means while the latter specifies the precision model to set up the according predictors. Therefore `y ~ x1 * x2 | x1 + x2` sets up a model where the main effects and the interaction of `x1` and `x2` are used to predict the mean, but the precision is modeled merely by their main effects. To use the alternative parametrization, `model = "alternative"` has to be specified in any case, since `"common"` is the default.

The argument `base` can be used to re-define the base (i.e., omitted) variable for the alternative parametrization. `sub.comp` can be used to define and analyze sub-compositions 'on the fly,' for example, in a dependent data set with 6 components `sub.comp = c(1, 5, 6)` will gather the 'irrelevant'

components 2, 3, and 4 in a new variable (becoming the new base variable) and perform the analysis only for the three selected plus the new 'residual variable.'

`data` takes the data set containing the independent *and* dependent variables, `weights` can be used to weight observations (by probability or frequency), `subset` can be used to fit the model only to a part of the data, `control` contains a list of parameters that control the optimization and `verbosity` takes non-negative integer values that determine how much information about the model fitting process is printed.

3.2 Optimization

To optimize these multivariate models which potentially contain a large number of dependent variables, the model is fitted in three stages: (1) starting values and refinement thereof, (2) Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization, and (3) Newton-Raphson optimization.

3.2.1 Starting values

The internal function `get_starting_values()` calculates starting values for both parametrizations. For the common parametrization, each dependent variable is treated as beta-distributied (this is admissible since the individual variables are marginally beta-distributed, see Sec. 2.1 above) and the distribution's α and β parameters are estimated using the `maxBFGS()` algorithm of maxLik. Starting values for the alternative parametrization are composed of regression coefficients for the logit-transformed responses and the precisions. The latter are estimated first by predicting a value of 0.5 for the matrix $\boldsymbol{Z}$ and then are refined by using `maxBFGS()` to fit the complete likelihood with all βs held fixed.

3.2.2 BFGS optimization

With the respective starting values the BFGS algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) using the analytical rather than numerical gradient, is carried out for both parametrizations with a convergence criterion of 10^{-5}. This method has shown to be fast, robust, and reliable in practice (estimates do not change in the next step), however, standard errors which are extracted from the Hessian are not reliable, so this is not the final step of the optimization.

3.2.3 Newton-Raphson optimization

With the estimates obtained before, the final estimation is done using the Newton-Raphson algorithm as implemented in maxLik in the function `maxNR()` where analytical first and second order derivatives are supplied to speed up convergence (the criterion is 10^{-10}) and to insure numerical stability. Standard errors are obtained from the Hessian ($H(\hat{\theta})$) by $\mathrm{SE}(\hat{\theta}) = \sqrt{\mathrm{diag}(\Sigma(\hat{\theta}))} = \sqrt{\mathrm{diag}(-H(\hat{\theta})^{-1})}$.

3.3 Object classes

For the manipulation of regression results, data, graphics, etc., functions and methods have been implemented in DirichletReg of which the most important are depicted in Figure 3. To prepare the data, the function `DR_data()` is used which has already been described above. The resulting object is of class `DirichletRegData` which has the methods `print()` (by default, the processed data), `summary()` (a short description of the data), and `plot()` (one-, two- and three-dimensional plots of the data).

From `DirichReg()` results an object of class `DirichletRegModel` which has the typical (G)LM methods which are commonly used in R: `print()` and `summary()` (display a short and long summary of the regression), `fitted()` and `predict()` (extract fitted values or predict values for new observations), `residuals()` (extract residuals of various types as described above), `confint()` (compute confidence intervals), `anova()` (compare models using a likelihood-ratio-test), `coef()` (extract coefficients), `logLik()`, `AIC()` and `BIC()` (extract the log-likelihood and information criteria), `vcov()` (extract

the parameter estimates' covariance-matrix), and `update()` (update an existing model by adding/removing terms of the formula).

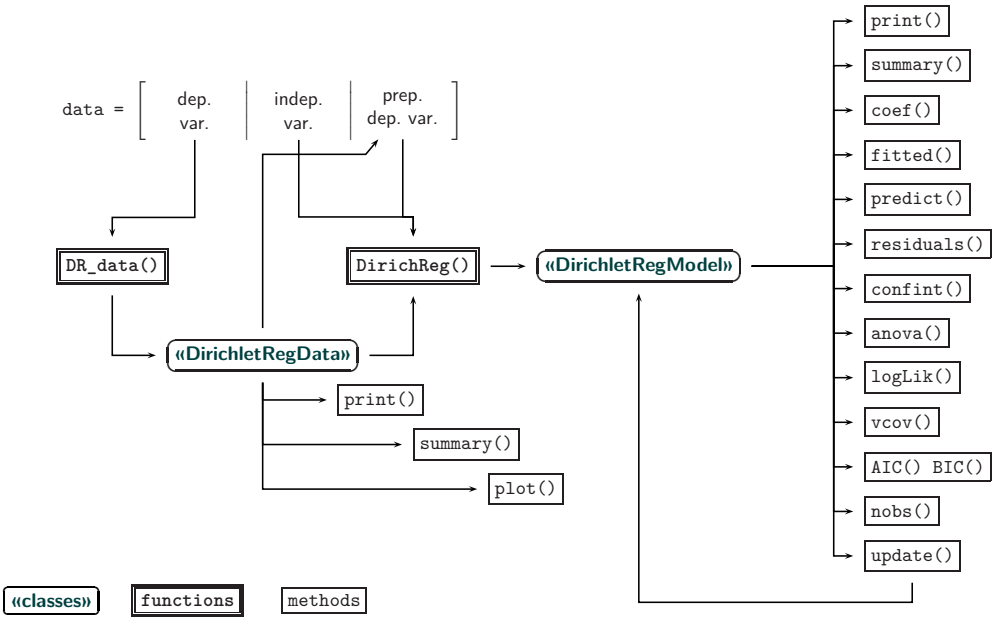


Figure 3: Workflow: Classes, functions, and methods in DirichletReg.

4 Application examples

This section gives some published examples that have been analyzed using conventional methods or beta/Dirichlet regression. It is designed to give the readers a broad overview of the package's capabilities.

4.1 The Arctic lake (common parametrization)

This dataset from Aitchison (2003) shows the sediment composition in an Arctic lake. In this example from sedimentology, samples are taken and classified into 'sand,' 'silt,' and 'clay' by their weights. The question is, "How does the sample depth affect the sediment composition?" To answer this, we first have a look at the data and then prepare them using `DR_data()`.

```
R> library("DirichletReg")
R> head(ArcticLake)
R> AL <- DR_data(ArcticLake[, 1:3])
```

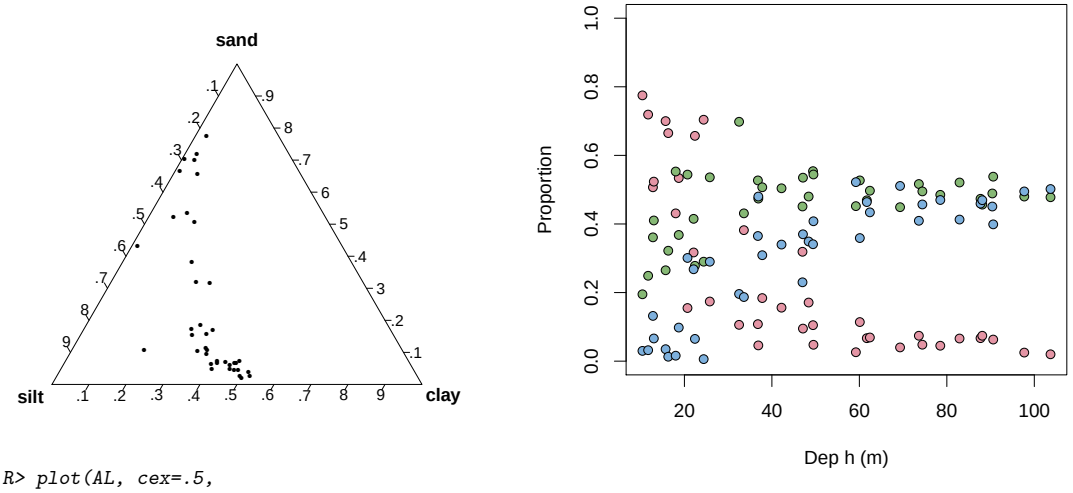
```
Warning message:
In DR_data(ArcticLake[, 1:3]) :
  not all rows sum up to 1 => normalization forced
```

`DR_data()` gives a warning that not all rows (of the dependent variables) sum up to 1. This can be seen in row 4, for example, which only sums up to .997 for the three components.

```
R> AL[1:6, ]
```

```
       sand      silt      clay
1 0.7750000 0.1950000 0.0300000
2 0.7190000 0.2490000 0.0320000
3 0.5070000 0.3610000 0.1320000
4 0.5235707 0.4102307 0.0661986
5 0.7000000 0.2650000 0.0350000
6 0.6650000 0.3220000 0.0130000
```

To get a ternary plot, we use the `plot()` method and in addition we plot the compositions against the depth they were sampled at. On the ternary plot (see Figure 4, left panel) we see that the data lie on an arc ranging from a high amount of sand to a 50/50 composition of silt and clay. In the right panel of Figure 4, we see the composition plotted against depth, which shows the way the composition changes with increasing depth.




```
R> plot(AL, cex=.5,
+       a2d=list(colored=FALSE,
+       c.grid=FALSE))
```

```
R> plot(rep(ArcticLake$depth, 3),
+       as.numeric(AL), ylim=0:1, pch=21,
+       bg=rep(rainbow_hcl(3), each=39),
+       xlab="Depth (m)", ylab="Proportion")
```

Figure 4: Arctic lake: Ternary plot (left) and a scatter plot of depth vs. composition (right).

To model these relationships, we will fit a Dirichlet regression model using the common parametrization, as in Hijazi (2003) and Hijazi and Jernigan (2009).

```
R> lake1 <- DirichReg(AL ~ depth, ArcticLake)
R> lake1

Call:
DirichReg(formula = AL ~ depth, data = ArcticLake)
using the common parametrization

Log-likelihood: 101.4 on 6 df (54 BFGS + 3 NR Iterations)

----------------------------------------
Coefficients for variable no. 1: sand
(Intercept)        depth
    0.11662      0.02335
----------------------------------------
Coefficients for variable no. 2: silt
(Intercept)        depth
   -0.31060      0.05557
----------------------------------------
Coefficients for variable no. 3: clay
(Intercept)        depth
    -1.1520       0.0643
----------------------------------------

R> coef(lake1)

$sand
(Intercept)       depth
 0.11662480  0.02335114
```

```
$silt
(Intercept)       depth
-0.31059591  0.05556745

$clay
(Intercept)       depth
-1.15195642  0.06430175
```

The `print()` method shows the function call along with the parametrization, log-likelihood, number of parameters, number of iterations, and the resulting regression coefficients. Below this, the coefficients are extracted using the `coef()` method.

From the coefficients we can see that, with increasing depth, all αs increase, but sand the least, silt more, and clay the most, which means that sand will virtually disappear with increasing soil depth, because the expected values are the individual αs divided by their sums. The increasing values of the αs also mean that the precision (α_0) gets higher as depth increases. We fit a second model including quadratic terms for `depth` using `update()` on our previous model.

```
R> lake2 <- update(lake1, . ~ . + I(depth^2) | . + I(depth^2) | . + I(depth^2))
R> anova(lake1, lake2)

Analysis of Deviance Table

Model 1: DirichReg(formula = AL ~ depth, data = ArcticLake)
Model 2: DirichReg(formula = AL ~ depth + I(depth^2) | depth + I(depth^2) | depth +
  I(depth^2), data = ArcticLake)

        Deviance N. par Difference df Pr(>Chi)
Model 1  -202.74      6
Model 2  -217.99      9     15.254  3 0.001612 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

To test the new model including a quadratic term against `lake1`, we can the `anova()` method. Obviously, `lake2` including quadratic terms for `depth` predicts the composition significantly better ($p = .002$), therefore we select this model and investigate it in detail using `summary()`.

```
R> summary(lake2)

Call:
DirichReg(formula = AL ~ depth + I(depth^2) | depth + I(depth^2) | depth + I(depth^2),
data = ArcticLake)

Standardized Residuals:
          Min       1Q   Median       3Q     Max
sand  -1.7647  -0.7080  -0.1786  0.9598  3.0460
silt  -1.1379  -0.5330  -0.1546  0.2788  1.5604
clay  -1.7661  -0.6583  -0.0454  0.6584  2.0152

--------------------------------------------------------------------
Beta-Coefficients for variable no. 1: sand
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.4361967  0.8026814   1.789   0.0736 .
depth       -0.0072383  0.0329433  -0.220   0.8261
I(depth^2)   0.0001324  0.0002761   0.480   0.6315
--------------------------------------------------------------------
Beta-Coefficients for variable no. 2: silt
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.0259705  0.7598827  -0.034   0.9727
depth        0.0717450  0.0343089   2.091   0.0365 *
I(depth^2)  -0.0002679  0.0003088  -0.867   0.3857
```

```
---------------------------------------------------------------------
Beta-Coefficients for variable no. 3: clay
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.7931487  0.7362293  -2.436  0.01487 *
depth        0.1107906  0.0357705   3.097  0.00195 **
I(depth^2)  -0.0004872  0.0003308  -1.473  0.14079
---------------------------------------------------------------------
Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood: 109 on 9 df (168 BFGS + 2 NR Iterations)
AIC: -200, BIC: -185
Number of Observations: 39
Link: Log
Parametrization: common
```

At the top of the output is the function call and a summary of the standardized residuals for each component, including minimum, maximum and the second through third quartiles. Below the estimates (now including standard errors, z, and p values), `summary()` shows the log-likelihood along with further informations as AIC/BIC, the number of observations, the number of BFGS and Newton-Raphson iterations, and the employed parametrization including used link-functions.

The resulting object `lake2` will be used to predict the fitted values and investigate their trajectories. For this we can use the `fitted()` method, but this will only return values where we have observed data, so to get smooth curves `predict()` was employed.

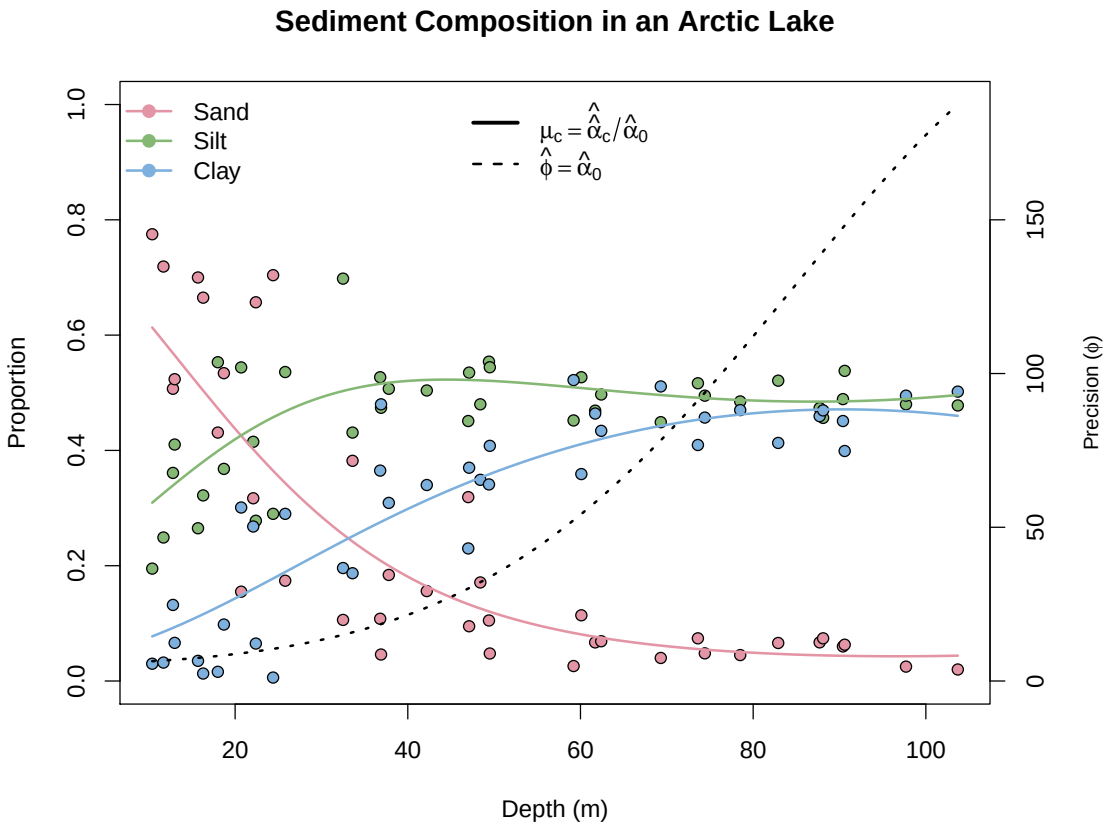


Figure 5: Arctic lake: Fitted values of the quadratic model.

As we have already inferred from the model's parameter estimates, we can see in Figure 5, that the αs increase leads to an increase of precision as well. Obviously, the data are closer to the expected values as depth increases. To show the difference between this approach and a log-ratio OLS approach,

see Figure 6. Apart from the fact that the OLS-curve seems to be biased to the left, the 'hook' at the bottom is a noteworthy aberration.

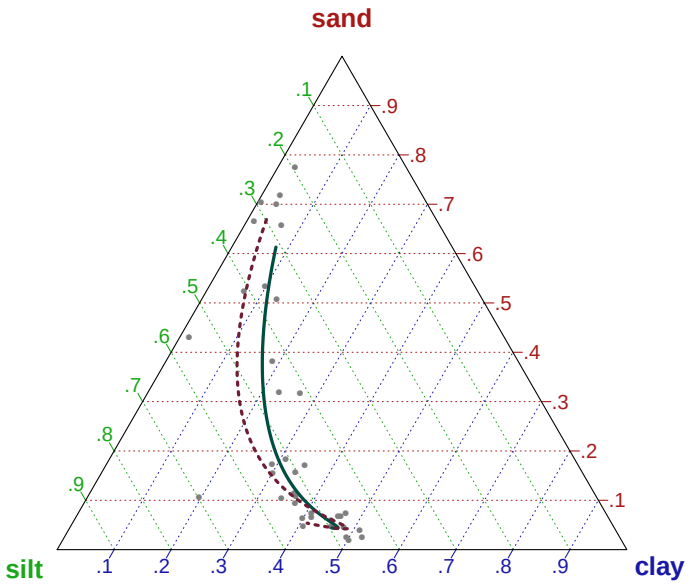


Figure 6: Arctic lake: OLS (dashed) vs. Dirichlet regression (solid) predictions.

4.2 Blood samples (alternative parametrization)

This is a medical example from Aitchison (2003) dealing with the differential diagnosis for two diseases 'A' and 'B' which are reflected in the composition of four serum protein components ('Albumin', 'Pre-Albumin', 'Globulin A', 'Globulin B'). We have 30 blood samples of diagnosed patients, that is, we know which kind of disease they are suffering from and six more samples of patients which are undiagnosed. For this example, we want to know (1) "How do the blood samples' compositions differ between the two disease types?" and (2) "Can we use the resulting model to classify the undiagnosed patients?"

Again, we prepare the data, including all patients although only the first 30 will be included in the model because they are diagnosed (we are warned that some observations do not exactly sum up to 1 and are normalized).

```
R> Bld <- BloodSamples
R> Bld$Smp <- DR_data(Bld[, 1:4])
```

The fitted model can be seen as the one-way ANOVA-counterpart in Dirichlet regression models, because we have our dependent variables and a categorical predictor (i.e., the type of disease). Because we are in an ANOVA-like setting where we want to make inferences about the expected values, we fit two models (one with constant and one with varying precision) using the alternative parametrization (note that the base was shifted to be component 3, `Globulin.A`) and compare them:

```
R> blood1 <- DirichReg(Smp ~ Disease | 1, Bld, model = "alternative", base = 3)
R> blood2 <- DirichReg(Smp ~ Disease | Disease, Bld, model = "alternative",
+      base = 3)
R> anova(blood1, blood2)

Analysis of Deviance Table

Model 1: DirichReg(formula = Smp ~ Disease | 1, data = Bld, model = "alternative", base =
  3)
```

```
Model 2: DirichReg(formula = Smp ~ Disease | Disease, data = Bld, model = "alternative",
  base = 3)

       Deviance N. par Difference df Pr(>Chi)
Model 1  -303.86      7
Model 2  -304.61      8     0.7587  1   0.3837
```

The precision seems to be the same for both groups ($p = .384$), so we take the simpler model and investigate the parameters:

```
R> summary(blood1)

Call:
DirichReg(formula = Smp ~ Disease | 1, data = Bld, model = "alternative", base = 3)

Standardized Residuals:
                Min       1Q   Median      3Q     Max
Albumin      -2.1310  -0.9307  -0.1234  0.8149  2.8429
Pre.Albumin  -1.0687  -0.4054  -0.0789  0.1947  1.5691
Globulin.A   -2.0503  -1.0392   0.1938  0.7927  2.2393
Globulin.B   -1.8176  -0.5347   0.1488  0.5115  1.3284

MEAN MODELS:
-------------------------------------------------------------------
Coefficients for variable no. 1: Albumin
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.11639    0.09935  11.237   <2e-16 ***
DiseaseB    -0.07002    0.13604  -0.515    0.607
-------------------------------------------------------------------
Coefficients for variable no. 2: Pre.Albumin
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.5490     0.1082   5.076 3.86e-07 ***
DiseaseB     -0.1276     0.1493  -0.855    0.393
-------------------------------------------------------------------
Coefficients for variable no. 3: Globulin.A
- variable omitted (reference category) -
-------------------------------------------------------------------
Coefficients for variable no. 4: Globulin.B
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.4863     0.1094   4.445  8.8e-06 ***
DiseaseB      0.1819     0.1472   1.236    0.216
-------------------------------------------------------------------

PRECISION MODEL:
-------------------------------------------------------------------
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.2227     0.1475   28.64   <2e-16 ***
-------------------------------------------------------------------
Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood: 151.9 on 7 df (56 BFGS + 2 NR Iterations)
AIC: -289.9, BIC: -280
Number of Observations: 30
Links: Logit (Means) and Log (Precision)
Parametrization: alternative
```

Figure 7 displays boxplots of the values on each dimension along with their fitted values as dashed lines. 'Globulin A' does not seem to be relevant, so we exclude it (marginal Dirichlet distribution) for plotting and use ternary plots below (see, Fig. 8).

To make predictions for the undiagnosed patients we can, for example, use a likelihood-based approach where we first predict the αs for each disease — using the model's estimates — and then calculate the densities/likelihoods of the new observations given the parameters of either group using

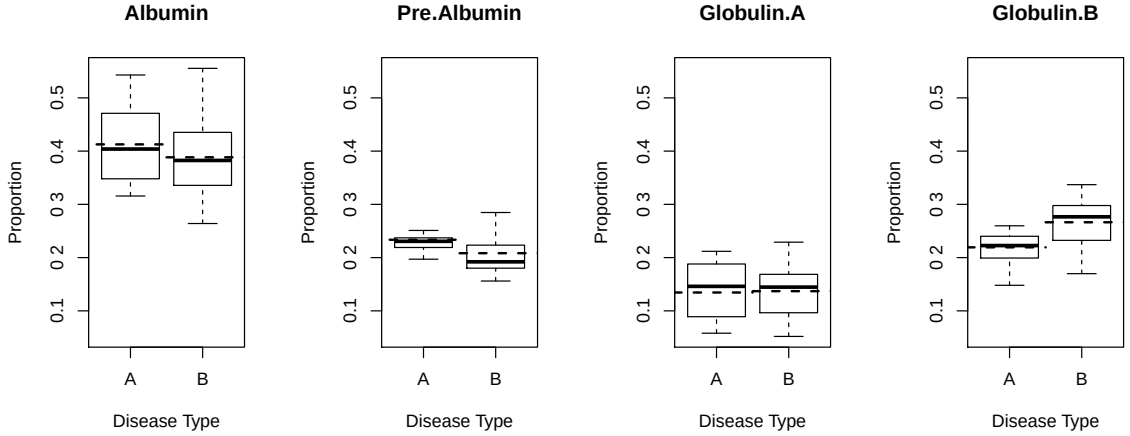


Figure 7: Blood samples: Box plots and fitted values (dashed lines indicate the fitted values for each group).

the function `ddirichlet()`. Those are then compared per observation and the relative frequencies hint at the class probabilities for each observation.

```
R> alpha <- predict(blood2, data.frame(Disease = factor(c("A", "B"))),
+     F, T, F)
R> L <- sapply(1:2, function(i) ddirichlet(DR_data(Bld[31:36, 1:4]), unlist(alpha[i,
+     ])))
R> LP <- L/rowSums(L)
R> dimnames(LP) <- list(paste("C", 1:6), c("A", "B"))
R> print(data.frame(round(LP * 100, 1), pred. = as.factor(ifelse(LP[, 1] >
+     LP[, 2], "==> A", "==> B"))), print.gap = 2)
```

```
        A     B  pred.
C 1  59.4  40.6  ==> A
C 2  43.2  56.8  ==> B
C 3  38.4  61.6  ==> B
C 4  43.8  56.2  ==> B
C 5  36.6  63.4  ==> B
C 6  70.2  29.8  ==> A
```

If we had to make a prediction about the disease types, `C 1` and `C 6` would be of type A while the other four persons pertain to type B. In Figure 8, we have a graphical display of these findings where we can see that two observations lie close to the expected value of A while the others are scattered around B.

4.3 Reading skills data (alternative parametrization)

The final example will be a replication of the Reading Skills analysis done in Smithson and Verkuilen (2006) and Cribari-Neto and Zeileis (2010). We have data about the z-transformed IQ score, a measure of reading accuracy and whether the child was diagnosed with dyslexia or not (control group). We want to regress reading accuracy on IQ and group membership. In a first step, two models are estimated: (1) main effects and their interaction in mean and precision model (i.e., the 'full' model) and (2) same predictors for the means, but only main effects in the precision model. A likelihood ratio test shows that the reduced model can be used ($p = .197$), so we use this for subsequent investigations.

```
R> RS <- ReadingSkills
R> RS$acc <- DR_data(RS$accuracy)
R> RS$dyslexia <- C(RS$dyslexia, treatment)
R> rs1 <- DirichReg(acc ~ dyslexia * iq | dyslexia * iq, RS, model = "alternative")
```

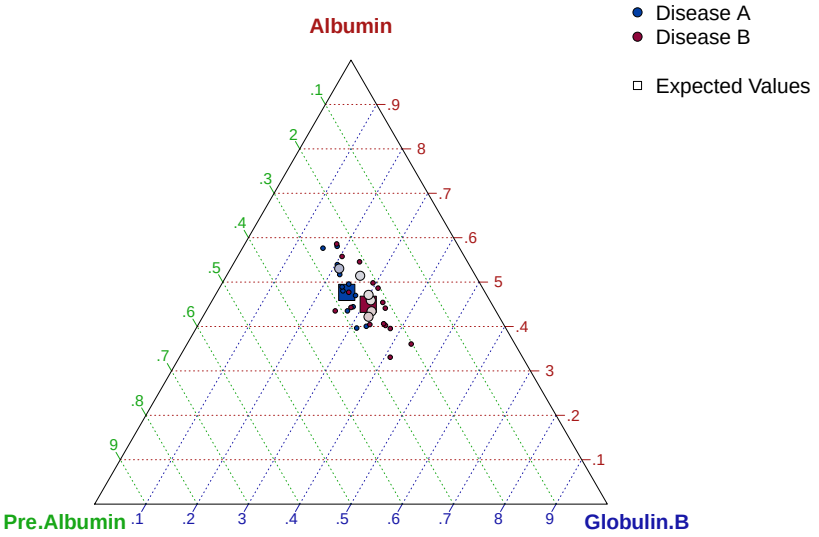


Figure 8: Blood samples: Observed values and predictions.

```
R> rs2 <- DirichReg(acc ~ dyslexia * iq | dyslexia + iq, RS, model = "alternative")
R> anova(rs1, rs2)

Analysis of Deviance Table

Model 1: DirichReg(formula = acc ~ dyslexia * iq | dyslexia * iq, data = RS, model =
  "alternative")
Model 2: DirichReg(formula = acc ~ dyslexia * iq | dyslexia + iq, data = RS, model =
  "alternative")

        Deviance N. par Difference df Pr(>Chi)
Model 1  -133.47      8
Model 2  -131.80      7     1.6645  1    0.197
```

To demonstrate the bias OLS-regression applied to log-ratio transformed data can produce, we estimate the model and plot it in Figure 9. The effects of heteroscedasticity on residuals can be observed in Figure 10 where we see clear patterns for OLS regression while Dirichlet regression does not exhibit any noticeable shapes.

```
R> a <- RS$accuracy
R> logRa_a <- log(a/(1 - a))
R> rlr <- lm(logRa_a ~ dyslexia * iq, RS)
R> summary(rlr)

Call:
lm(formula = logRa_a ~ dyslexia * iq, data = RS)

Residuals:
     Min       1Q   Median       3Q      Max
-2.66405 -0.37966  0.03687  0.40887  2.50345

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.8067     0.2822   9.944 2.27e-12 ***
dyslexiayes     -2.4113     0.4517  -5.338 4.01e-06 ***
iq               0.7823     0.2992   2.615   0.0125 *
dyslexiayes:iq  -0.8457     0.4510  -1.875   0.0681 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

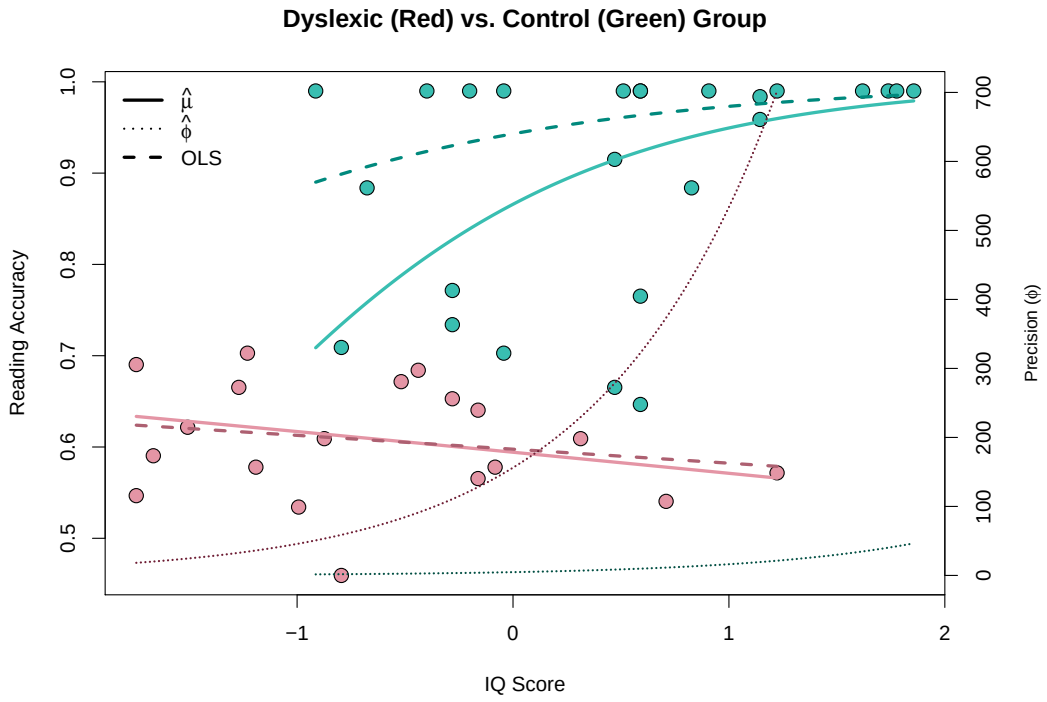


Figure 9: Reading skills: Predicted values of Dirichlet regression and OLS regression.

```
Residual standard error: 1.2 on 40 degrees of freedom
Multiple R-squared:  0.6151,        Adjusted R-squared:  0.5862
F-statistic: 21.31 on 3 and 40 DF,  p-value: 2.083e-08
```

According to the OLS regression, there is no significant evidence of an interaction between IQ and dyslexia regarding reading accuracy in children (`dyslexiayes:iq`, $p = .068$). Dirichlet regression however tells us a different story, as displayed below the interaction term is highly significant ($p < .001$).

```
R> summary(rs2)

Call:
DirichReg(formula = acc ~ dyslexia * iq | dyslexia + iq, data = RS, model =
"alternative")

Standardized Residuals:
                 Min       1Q   Median       3Q      Max
1 - accuracy  -1.5661  -0.8204  -0.5112   0.5211   3.4334
accuracy      -3.4334  -0.5211   0.5112   0.8204   1.5661

MEAN MODELS:
-------------------------------------------------------------------
Coefficients for variable no. 1: 1 - accuracy
- variable omitted (reference category) -
-------------------------------------------------------------------
Coefficients for variable no. 2: accuracy
               Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.8649     0.2991   6.235 4.52e-10 ***
dyslexiayes     -1.4833     0.3029  -4.897 9.74e-07 ***
iq               1.0676     0.3359   3.178 0.001482 **
dyslexiayes:iq  -1.1625     0.3452  -3.368 0.000757 ***
-------------------------------------------------------------------
```

```
PRECISION MODEL:
------------------------------------------------------------------
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.5579     0.3336   4.670 3.01e-06 ***
dyslexiayes   3.4931     0.5880   5.941 2.83e-09 ***
iq            1.2291     0.4596   2.674  0.00749 **
------------------------------------------------------------------
Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-likelihood: 65.9 on 7 df (37 BFGS + 2 NR Iterations)
AIC: -117.8, BIC: -105.3
Number of Observations: 44
Links: Logit (Means) and Log (Precision)
Parametrization: alternative
```

Because the mean/dispersion-model is similar to logistic regression, we can interpret the $\hat{\beta}$-parameters as odds-ratios if we take them $\exp(\hat{\beta})$ (see Sec. 2.2.2). The method `confint()` has an argument `exp` that toggles between the original and exponentiated estimates and confidence intervals — here we could say that people suffering from dyslexia have $\exp(-1.48) = 0.23\times$ the chance of getting the full accuracy score compared to non-dyslexics and each standard deviation of IQ increases the chance of a 'perfect score' by almost three times. However we have an interaction term that shows that the influence of IQ on accuracy is only .3 for dyslexics as compared to non-dyslexics. The precision parameters reveal that higher intelligence and being diagnosed with dyslexia lead to higher precision in the measured accuracy scores.

```
R> confint(rs2)

95% Confidence Intervals (original form)

- Beta-Parameters:
Variable: 1 - accuracy
  variable omitted

Variable: accuracy
                  2.5%   Est.   97.5%
(Intercept)      1.279   1.86   2.451
dyslexiayes     -2.077  -1.48  -0.890
iq               0.409   1.07   1.726
dyslexiayes:iq  -1.839  -1.16  -0.486

- Gamma-Parameters
             2.5%  Est.  97.5%
(Intercept)  0.904  1.56   2.21
dyslexiayes  2.341  3.49   4.65
iq           0.328  1.23   2.13

R> confint(rs2, exp = TRUE)

95% Confidence Intervals (exponentiated)

- Beta-Parameters:
Variable: 1 - accuracy
  variable omitted

Variable: accuracy
                 2.5%  exp(Est.)   97.5%
(Intercept)     3.592      6.455  11.601
dyslexiayes     0.125      0.227   0.411
iq              1.506      2.908   5.618
dyslexiayes:iq  0.159      0.313   0.615
```

```
- Gamma-Parameters
              2.5%  exp(Est.)   97.5%
(Intercept)   2.47       4.75    9.13
dyslexiayes  10.39      32.89  104.12
iq            1.39       3.42    8.41
```

5 Discussion

This article presented a detailed overview of Dirichlet regression models and their implementation in the R package DirichletReg. In the course of this paper, the formulation, usage, and interpretability of the Dirichlet regression as a GLM-like framework has been elaborated and two parametrizations (*common* and *alternative*) have been introduced. Three examples were presented that show how Dirichlet regression can be used in practice. As opposed to packages like compositions, this package does not rely on transformations of the data and the models account for many undesirable features such as heteroscedasticity. In contrast to betareg, which is much more elaborate, the strength of this package is that it can analyze an arbitrary number of variables where the former is confined to one beta-distributed variable.

Future work may include algorithms for automated model selection (e.g., graphical modeling and similar strategies). The common parametrization looks especially promising in this respect. Some data sets also include so many variables that dimension reduction techniques as a 'Dirichlet-PCA' would be desirable. The betareg package has many interesting features which would also make sense in the context of this package, as implementing methods using lmtest (Hothorn et al., 2013), strucchange (Zeileis et al., 2013b), etc. Lastly, mixtures of Dirichlet distributions and — in the long run — Dirichlet mixture regression models as in flexmix (Grün et al., 2013) could be implemented.

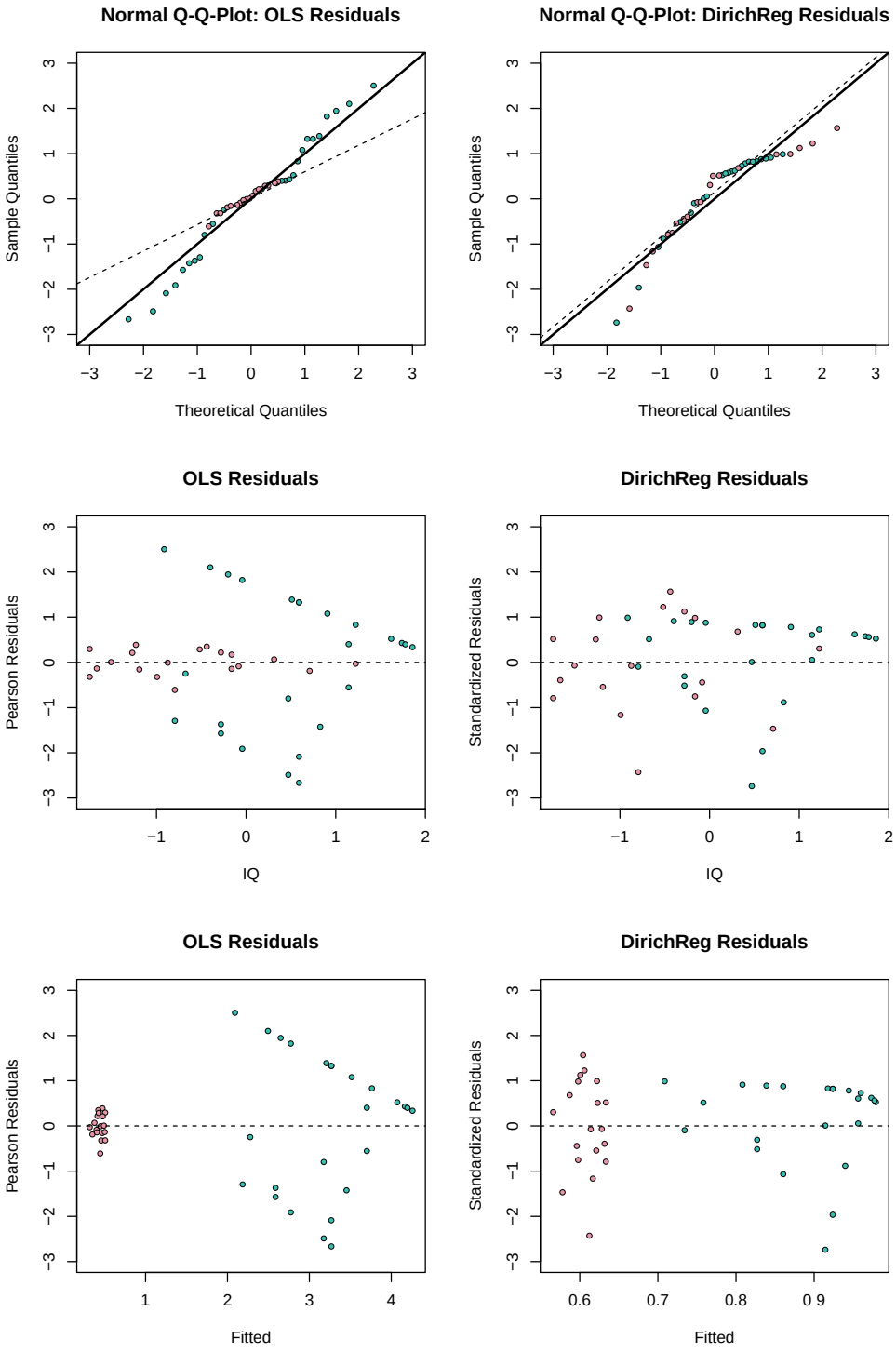


Figure 10: Reading skills: Various residual plots of OLS and Dirichlet regression models. The left columns shows OLS, the right column Dirichlet models. In the first row, QQ-plots show deviation from a normal distribution of residuals. The second row plots residuals against the covariate IQ. The third row plots residuals against fitted values.

Acknowledgments

First and foremost, I want to express my gratitude to the late Reinhold Hatzinger who mentored me over the course of developing this package, presenting it at conferences, and finally writing this paper. His guidance and advice were a vital part for my progress and I want to dedicate this paper to him.

I am furthermore indebted to Wolfgang Wiedermann, Ingrid Koller, and Pui Yu Ling for their comments and suggestions as well as Rafiq Hijazi for providing his work on the analysis of compositional data. Furthermore I want to express my thanks to the doctorates of Reinhold Hatzinger at the Wirtschaftsuniversität Wien, the participants of the International Workshop on Psychometric Computing (PsychoCo) 2011 in Tübingen and anonymous reviewers for providing support and helpful comments.

Appendix: Notation and derivatives

$$\frac{\partial^2 \ell_a}{\partial \beta_{md}\,\partial \beta_{nd}} = -x_m x_d \phi \epsilon_d \left\{ \left[\sum_{\substack{c=1 \\ c\neq d}}^{C} \epsilon_c \log(y_c) - \log(y_d) \sum_{\substack{c=1 \\ c\neq d}}^{C} \epsilon_c \right] \left[2\epsilon_d^{\Sigma\Sigma} + \sum_{\substack{c=1 \\ c\neq d}}^{C} \epsilon_c^2 - \epsilon_d^2 \right] \right.$$
$$\left. \left[\sum_{\substack{c=1 \\ c\neq d}}^{C} \left(\epsilon_c \epsilon_d \phi \psi_1(\alpha_c) - \psi(\alpha_c)(2\epsilon_d^{\Sigma\Sigma} + \sum_{\substack{e=1 \\ e\neq d}}^{C} \epsilon_e^2 - \epsilon_d^2) \right) \right] \right\} \Big/ (\epsilon^{\Sigma})^4 \quad (22)$$

$$\frac{\partial^2 \ell_a}{\partial \beta_{md}\,\partial \beta_{ne}} = x_m x_n \phi \epsilon_d \epsilon_e \left\{ \sum_{i=\{d,e\}} \left[\epsilon_i \phi \psi_1(\alpha_i) \sum_{C\setminus i} \epsilon_c \right] \right.$$
$$+ \sum_{i=\{d,e\}} \psi_1(\alpha_i) \left[2\epsilon_i^{\Sigma\Sigma} + \sum_{C\setminus i} \epsilon_c^2 - \epsilon_i^2 \right]$$
$$\left. + \epsilon^{\Sigma} \sum_{i=C\setminus\{d,e\}} [-\mu_i \epsilon_i \phi \psi_1(\alpha_i) - 2\epsilon_i \psi(\alpha_i) + 2\log(y_i)\epsilon_i] \right\} \Big/ (\epsilon^{\Sigma})^4 \quad (23)$$

$$\frac{\partial^2 \ell_a}{\partial \beta_{md}\,\partial \gamma_n} = x_m z_n \phi \epsilon_d \left\{ \epsilon^{\Sigma} \sum_{\substack{i=1 \\ i\neq d}}^{C} [\epsilon\,(\psi(\alpha_i) - \log(y_i))] + \log(y_d) \right.$$
$$\left. - \psi(\alpha_d) \left[\sum_{\substack{i=1 \\ i\neq d}}^{C} \epsilon_i^2 + \epsilon_d \sum_{\substack{i=1 \\ i\neq d}}^{C} \epsilon_i + 2\epsilon_{\neq d}^{\Sigma\Sigma} \right] + \phi \left[\sum_{\substack{i=1 \\ i\neq d}}^{C} \epsilon_i^2 \psi_1(\alpha_i) - \epsilon_d \psi(\alpha_i) \sum_{\substack{j=1 \\ j\neq d}}^{C} \epsilon_j \right] \right\} \Big/ (\epsilon^{\Sigma})^3 \quad (24)$$

$$\frac{\partial^2 \ell_a}{\partial \gamma_m\,\partial \gamma_n} = z_m z_n \phi \left\{ \sum_{i=1}^{C} \epsilon_i \left[\log(y_i) - \psi(\alpha_i) - \alpha_i \psi_1(\alpha_i) \right] + \epsilon^{\Sigma} \left[\psi(\phi) + \psi_1(\phi) \right] \right\} \Big/ \epsilon^{\Sigma} \quad (25)$$

Indices		
observation	i	$\in 1, \ldots, N$
component/dep. var.	c	$\in 1, \ldots, C$
predictor	$j^{[c]}$	$\in 1, \ldots, J^{[c]}$
means predictor	k	$\in 1, \ldots, K$
variance predictor	l	$\in 1, \ldots, L$

Matrices/Vectors			Dim.
dependent variables	$\boldsymbol{Y}$	y_{ic}	$N \times C$
independent variables	$\boldsymbol{X}^{[c]}$	$x_{ij}^{[c]}$	$N \times J^{[c]}$
independent variables (means)	$\boldsymbol{X}$	x_{ik}	$N \times K$
independent variables (variance)	$\boldsymbol{Z}$	z_{il}	$N \times L$
alpha-parameters	$\boldsymbol{A}$	α_{ic}	$N \times C$
regression coefficients (common)	$\boldsymbol{\beta}^{[c]}$	$\beta_j^{[c]}$	$J^{[c]} \times 1$
regression coefficients: means	$\boldsymbol{\beta}$	β_k	$K \times 1$
regression coefficients: variance	$\boldsymbol{\gamma}$	γ_l	$L \times 1$

Table 1: Notation.

References

Aitchison, J. (1982). The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B*, 44(2):139–177.

Aitchison, J. (2003). *The Statistical Analysis of Compositional Data*. The Blackburn Press, Caldwell, NJ.

Barceló, C., Pawlowsky, V., and Grunsky, E. (1996). Some aspects of transformations of compositional data and the identification of outliers. *Mathematical Geology*, 28(4):501–518.

Brehm, J., Gates, S., and Gomez, B. (1998). A monte carlo comparison of methods for compositional data analysis. Working Paper 295, Washington University in St. Louis.

Broyden, C. (1970). The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90.

Cribari-Neto, F. and Zeileis, A. (2010). Beta regression in R. *Journal of Statistical Software*, 34(2):1–24.

Ferrari, S. P. L. and Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, 31(7):799–815.

Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer Journal*, 13:317–322.

Goldfarb, D. (1970). A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26.

Grün, B., Leisch, F., and Sarkar, D. (2013). *flexmix: Flexible Mixture Modeling*. R package version 2.3-11.

Gueorguieva, R., Rosenheck, R., and Zelterman, D. (2008). Dirichlet component regression and its applications to psychiatric data. *Computational Statistics and Data Analysis*, 52:5344–5355.

Henningsen, A. and Toomet, O. (2010). maxLik: A package for maximum likelihood estimation in R. *Computational Statistics*, 26(3):1–16.

Hijazi, R. H. (2003). *Analysis of Compositional Data Using Dirichlet Covariate Models*. PhD thesis, American University, Washington, D.C.

Hijazi, R. H. and Jernigan, R. W. (2009). Modeling compositional data using dirichlet regression models. *Journal of Applied Probability & Statistics*, 4(1):77–91.

Hothorn, T., Zeileis, A., Farebrother, R. W., Cummins, C., Millo, G., and Mitchell, D. (2013). *lmtest: Testing Linear Regression Models*. R package version 0.9-32.

Hron, K., Templ, M., and Filzmoser, P. (2010). Imputation of missing values for compositional data using classical and robust methods. *Computational Statistics and Data Analysis*, 54(12):3095–3107.

Maier, M. J. (2013). *DirichletReg: Dirichlet Regression in R*. R package version 0.4-1.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Chapman & Hall, London.

Melo, T. F. N., Vasconcellos, K. L. P., and Lemonte, A. J. (2009). Some restriction tests in a new class of regression models for proportions. *Computational Statistics and Data Analysis*, 53(12):3972–3979.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Shanno, D. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656.

Smithson, M. and Verkuilen, J. (2006). A better lemon squeezer? maximum-likelihood regression with beta-distributed dependent variables. *Psychological Methods*, 11(1):54–71.

Templ, M., Hron, K., and Filzmoser, P. (2013). *robCompositions: Robust Estimation for Compositional Data*. R package version 1.6.4.

Toomet, O., Henningsen, A., with contributions from, Graves, S., and Croissant, Y. (2013). *maxLik: Maximum Likelihood Estimation*. R package version 1.2-0.

van den Boogaart, K. G., Tolosana, R., and Bren, M. (2013). *compositions: Compositional Data Analysis*. R package version 1.30-1.

van den Boogaart, K. G. and Tolosana-Delgado, R. (2008). compositions: A unified R package to analyze compositional data. *Computers & Geosciences*, 34(4):320–338.

Zeileis, A., Cribari-Neto, F., Grün, B., Kosmidis, I., Simas, A. B., earlier version by, and Rocha, A. V. (2013a). *betareg: Beta Regression*. R package version 3.0-4.

Zeileis, A. and Croissant, Y. (2010). Extended model formulas in r: Multiple parts and multiple responses. *Journal of Statistical Software*, 34(1):1–13.

Zeileis, A. and Croissant, Y. (2013). *Formula: Extended Model Formulas*. R package version 1.1-1.

Zeileis, A., Leisch, F., Hornik, K., Kleiber, C., and Hansen, B. (2013b). *strucchange: Testing, Monitoring, and Dating Structural Changes*. R package version 1.5-0.

Author Information

Marco J. Maier
Wirtschaftsuniversität Wien
Vienna University of Economics and Business
Institute for Statistics and Mathematics
Building D4, Level 4
Welthandelsplatz 1
1020 Vienna, Austria
Telephone: +43/1/31336–4335
E-mail: Marco.Maier@wu.ac.at
URL: http://statmath.wu.ac.at/~maier/