Package Vignette for DirichletReg: JSS Code

Marco J. Maier
Wirtschaftsuniversität Wien

Abstract

This package vignette contains the full code from the JSS article.

This document was generated using R 2.13.2 (R Development Core Team 2011) and **DirichletReg** 0.3.0.

Keywords: DirichletReg package, Dirichlet regression.

```
> library(DirichletReg)
> head(ArcticLake)
```

```
   sand  silt  clay depth
1 0.775 0.195 0.030  10.4
2 0.719 0.249 0.032  11.7
3 0.507 0.361 0.132  12.8
4 0.522 0.409 0.066  13.0
5 0.700 0.265 0.035  15.7
6 0.665 0.322 0.013  16.3
```

```
> AL <- DR_data(ArcticLake[, 1:3])
```

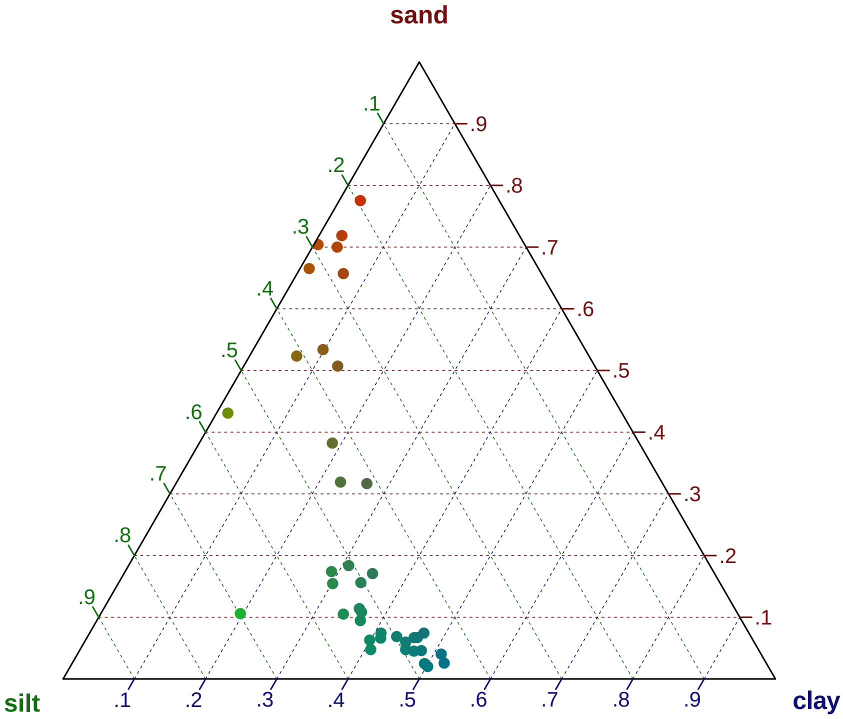
```
> AL
```

```
This object contains compositional data with 3 dimensions.
Number of observations: 39
* The data were normalized.

To access the data, use the function getdata().
```

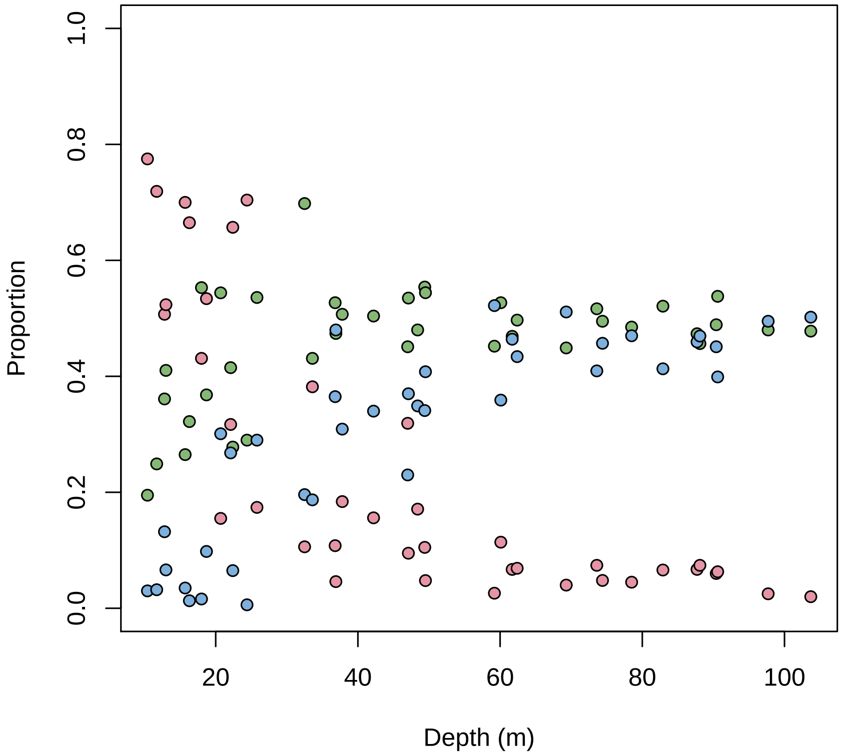
```
> head(getdata(AL), width = 15, height = 10)
```

```
       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
```

```
> plot(AL, reset_par = FALSE)
```



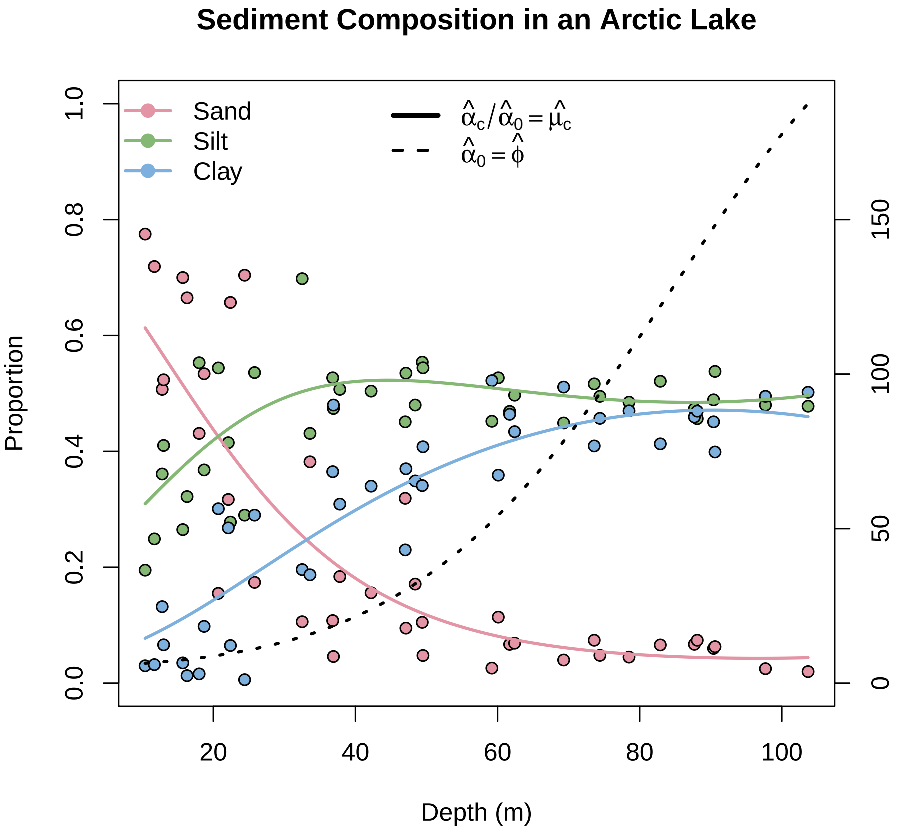
```
> plot(rep(ArcticLake$depth, 3), unlist(getdata(AL)), pch = 21,
+      bg = rep(rainbow_hcl(3), each = 39), xlab = "Depth (m)",
+      ylab = "Proportion", ylim = 0:1)
```



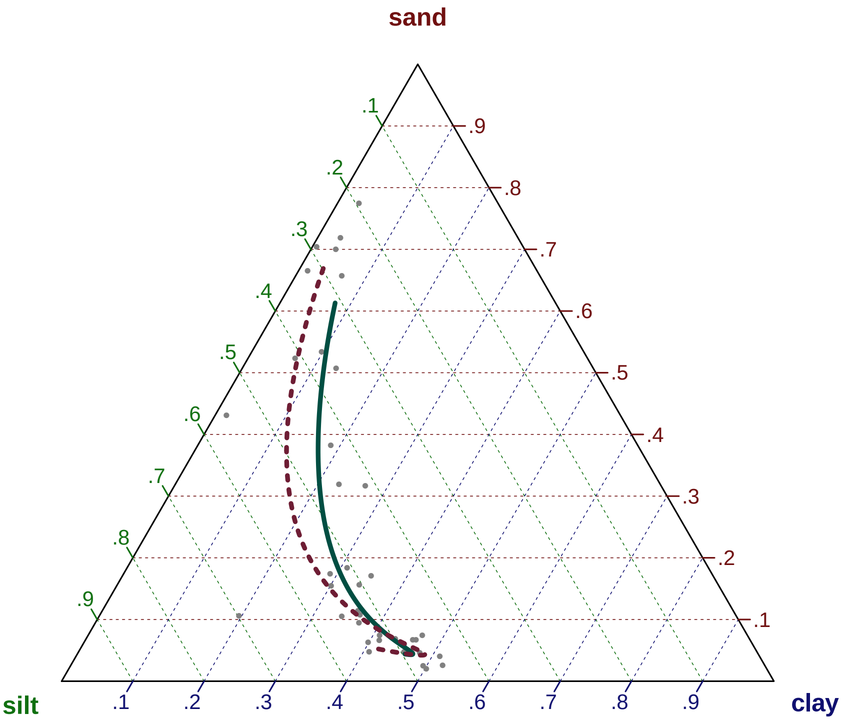
```
> lake1 <- DirichReg(AL ~ depth, ArcticLake)
> summary(lake1)

Call:
DirichReg(formula = AL ~ depth, data = ArcticLake)


Standardized Residuals:
          Min       1Q   Median      3Q     Max
sand  -1.7724  -0.8319   0.0120  1.3435  2.2862
silt  -1.0863  -0.5343  -0.1279  0.2892  1.4493
clay  -2.0868  -0.7516  -0.0012  0.4391  1.9660


--------------------------------------------------------------------------------
Beta-Coefficients for variable no. 1: sand
            Estimate Std. Error z-Value p-Value
(Intercept) 0.116625   0.409292   0.285 0.77569
depth       0.023351   0.007454   3.133 0.00173 **
--------------------------------------------------------------------------------
Beta-Coefficients for variable no. 2: silt
             Estimate Std. Error z-Value p-Value
(Intercept) -0.310596   0.344731  -0.901   0.368
depth        0.055567   0.006365   8.730  <2e-16 ***
```

```
------------------------------------------------------------------------------
Beta-Coefficients for variable no. 3: clay
             Estimate Std. Error z-Value  p-Value
(Intercept) -1.151956   0.298473   -3.86 0.000114 ***
depth        0.064302   0.005738   11.21  < 2e-16 ***
------------------------------------------------------------------------------
Signif. codes: `***' < .001, `**' < 0.01, `*' < 0.05, `.' < 0.1


Log-likelihood: 101.4 on 6 df (102+2 iterations)
Link: Log
Parametrization: common

> lake2 <- DirichReg(AL ~ depth + I(depth^2), ArcticLake)
> anova(lake1, lake2)

Analysis of Deviance Table

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

          Deviance   N. par   Difference   df       p-value
Model 1   -202.7393       6            -    -             -
Model 2   -217.9937       9     15.25441    3   0.001611655

> lake2

Call:
DirichReg(formula = AL ~ depth + I(depth^2), data = ArcticLake)
using the common parametrization

Log-likelihood: 109 on 9 df (99+2 iterations)

------------------------------------------------------------------------------
Coefficients for variable no. 1: sand
(Intercept)        depth   I(depth^2)
  1.4361967   -0.0072383    0.0001324
------------------------------------------------------------------------------
Coefficients for variable no. 2: silt
(Intercept)        depth   I(depth^2)
 -0.0259705    0.0717450   -0.0002679
------------------------------------------------------------------------------
Coefficients for variable no. 3: clay
(Intercept)        depth   I(depth^2)
 -1.7931487    0.1107906   -0.0004872
------------------------------------------------------------------------------

> plot(rep(ArcticLake$depth, 3), unlist(getdata(AL)), pch = 21,
+     bg = rep(rainbow_hcl(3), each = 39), xlab = "Depth (m)",
+     ylab = "Proportion", ylim = 0:1, main = "Sediment Composition in an Arctic Lake")
> Xnew <- data.frame(depth = seq(min(ArcticLake$depth), max(ArcticLake$depth),
+     length.out = 100))
```

```
> for (i in 1:3) lines(cbind(Xnew, predict(lake2, Xnew)[, i]),
+     col = rainbow_hcl(3)[i], lwd = 2)
> legend("topleft", legend = c("Sand", "Silt", "Clay"), lwd = 2,
+     col = rainbow_hcl(3), pt.bg = rainbow_hcl(3), pch = 21, bty = "n")
> par(new = TRUE)
> plot(cbind(Xnew, predict(lake2, Xnew, F, F, T)), lty = "24",
+     type = "l", ylim = c(0, max(predict(lake2, Xnew, F, F, T))),
+     axes = F, ann = F, lwd = 2)
> axis(4)
> legend("top", legend = c(expression(hat(alpha)[c]/hat(alpha)[0] ==
+     hat(mu[c])), expression(hat(alpha)[0] == hat(phi))), lty = c(1,
+     2), lwd = c(3, 2), bty = "n")
```



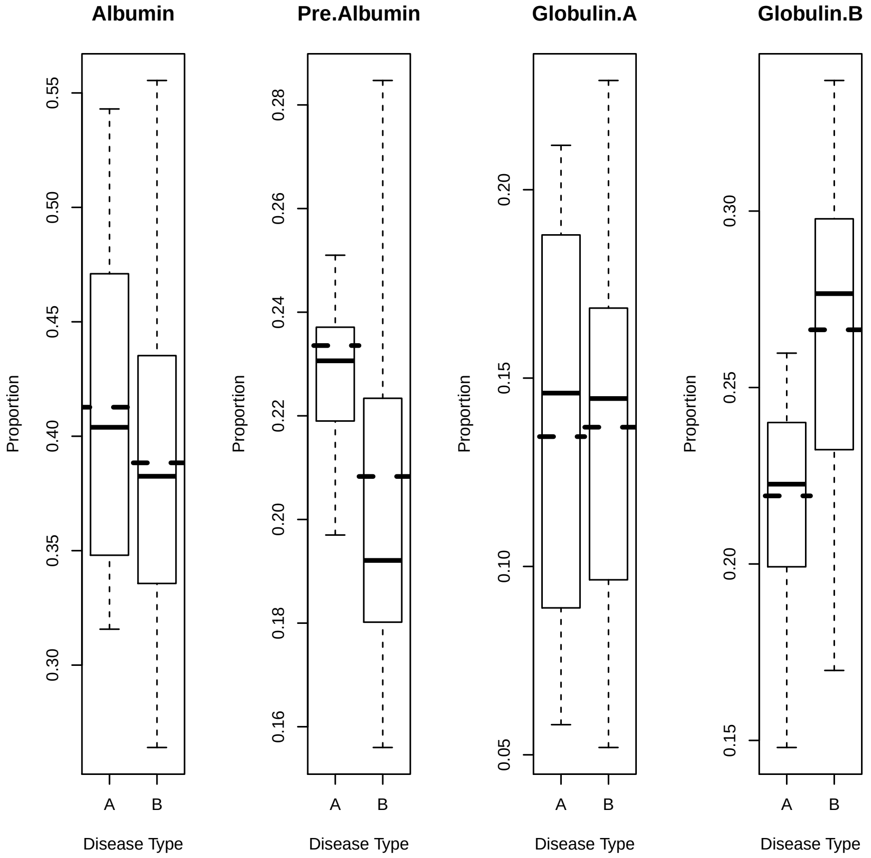
```
> AL <- DR_data(ArcticLake[, 1:3])
> dd <- range(ArcticLake$depth)
> X <- data.frame(depth = seq(dd[1], dd[2], length.out = 200))
> pp <- predict(DirichReg(AL ~ depth + I(depth^2), ArcticLake),
+     X)
> plot(AL, cex = 0.1, reset_par = FALSE)
> points(DirichletReg:::coord.trafo(AL$Y[, c(2, 3, 1)]), pch = 16,
+     cex = 0.5, col = gray(0.5))
> lines(DirichletReg:::coord.trafo(pp[, c(2, 3, 1)]), lwd = 3,
+     col = rainbow_hcl(2, l = 25)[2])
> Dols <- log(cbind(ArcticLake[, 2]/ArcticLake[, 1], ArcticLake[,
+     3]/ArcticLake[, 1]))
> ols <- lm(Dols ~ depth + I(depth^2), ArcticLake)
> p2 <- predict(ols, X)
> p2m <- exp(cbind(0, p2[, 1], p2[, 2]))/rowSums(exp(cbind(0, p2[,
+     1], p2[, 2])))
> lines(DirichletReg:::coord.trafo(p2m[, c(2, 3, 1)]), lwd = 3,
+     col = rainbow_hcl(2, l = 25)[1], lty = "21")
```



```
> B <- DR_data(BloodSamples[1:30, 1:4])
> blood1 <- DirichReg(B ~ Disease | phi ~ 1, BloodSamples)
> blood2 <- DirichReg(B ~ Disease | phi ~ Disease, BloodSamples)
> anova(blood1, blood2)

Analysis of Deviance Table

Model 1:
DirichReg(formula = B ~ Disease | phi ~ 1, data = BloodSamples)
Model 2:
DirichReg(formula = B ~ Disease | phi ~ Disease, data = BloodSamples)

           Deviance   N. par   Difference   df     p-value
Model 1   -303.8560        7            -    -           -
Model 2   -304.6147        8    0.7586655    1   0.3837465

> summary(blood1)
```

```
Call:
DirichReg(formula = B ~ Disease | phi ~ 1, data = BloodSamples)


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
- variable omitted (reference category) -
--------------------------------------------------------------------------------
Coefficients for variable no. 2: Pre.Albumin
            Estimate Std. Error z-Value  p-Value
(Intercept) -0.56737    0.08272  -6.859 6.94e-12 ***
DiseaseB    -0.05761    0.11575  -0.498    0.619
--------------------------------------------------------------------------------
Coefficients for variable no. 3: Globulin.A
            Estimate Std. Error z-Value p-Value
(Intercept) -1.11639    0.09935 -11.237  <2e-16 ***
DiseaseB     0.07002    0.13604   0.515   0.607
--------------------------------------------------------------------------------
Coefficients for variable no. 4: Globulin.B
            Estimate Std. Error z-Value  p-Value
(Intercept) -0.63011    0.08435  -7.470 8.04e-14 ***
DiseaseB     0.25192    0.11300   2.229   0.0258 *
--------------------------------------------------------------------------------

PRECISION MODEL:
--------------------------------------------------------------------------------
            Estimate Std. Error z-Value p-Value
(Intercept)   4.2227     0.1475   28.64  <2e-16 ***
--------------------------------------------------------------------------------
Signif. codes: `***' < .001, `**' < 0.01, `*' < 0.05, `.' < 0.1


Log-likelihood: 151.9 on 7 df (43+2 iterations)
Links: Logit (Means) and Log (Precision)
Parametrization: alternative
```

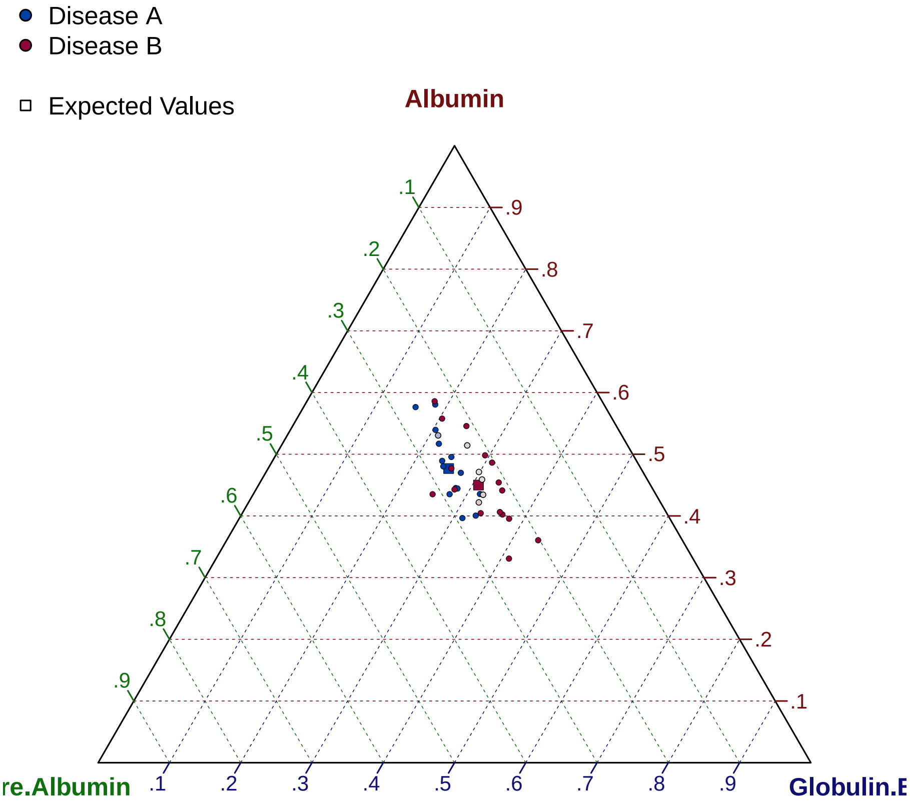
```
> par(mfrow = c(1, 4))
> for (i in 1:4) {
+     boxplot(B$Y[, i] ~ BloodSamples$Disease[1:30], main = paste(names(BloodSamples)[i]),
+         xlab = "Disease Type", ylab = "Proportion")
+     segments(c(-5, 1.5), unique(fitted(blood2)[, i]), c(1.5,
+         5), unique(fitted(blood2)[, i]), lwd = 3, lty = 2)
+ }
```



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

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

```
> B2 <- DR_data(BloodSamples[, c(1, 2, 4)])
> plot(B2, cex = 0.001, reset_par = FALSE)
> div.col <- diverge_hcl(100)
> temp <- (alpha/rowSums(alpha))[, c(2, 4, 1)]
> points(DirichletReg:::coord.trafo(temp/rowSums(temp)), pch = 22,
+     bg = div.col[c(1, 100)], cex = 1, lwd = 0.25)
> temp <- B2$Y[1:30, c(2, 3, 1)]
> points(DirichletReg:::coord.trafo(temp/rowSums(temp)), pch = 21,
+     bg = (div.col[c(1, 100)])[BloodSamples$Disease[1:30]], cex = 0.5,
+     lwd = 0.25)
> temp <- B2$Y[31:36, c(2, 3, 1)]
> points(DirichletReg:::coord.trafo(temp/rowSums(temp)), pch = 21,
+     bg = div.col[round(100 * LP[, 2], 0)], cex = 0.5, lwd = 0.5)
> legend("topleft", bty = "n", legend = c("Disease A", "Disease B",
+     NA, "Expected Values"), pch = c(21, 21, NA, 22), pt.bg = c(div.col[c(1,
+     100)], NA, "white"))
```



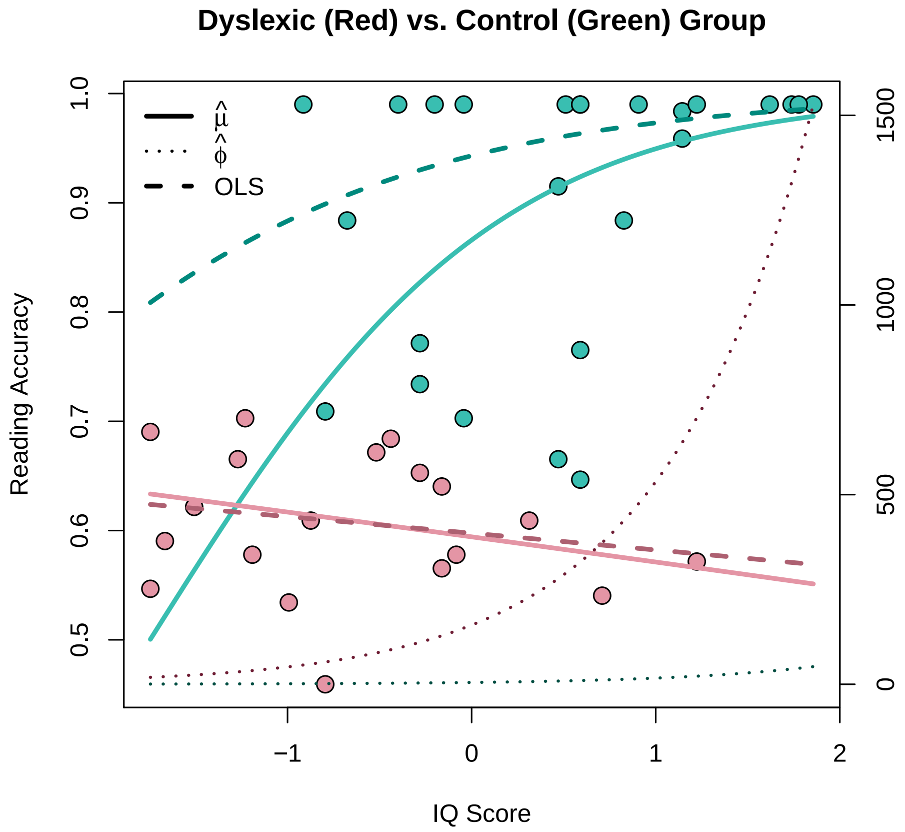
```
> data("ReadingSkills", package = "betareg")
> acc <- DR_data(ReadingSkills$accuracy)
> ReadingSkills$dyslexia <- C(ReadingSkills$dyslexia, treatment)
> rs1 <- DirichReg(acc ~ dyslexia * iq | phi ~ dyslexia * iq, ReadingSkills)
> rs2 <- DirichReg(acc ~ dyslexia * iq | phi ~ dyslexia + iq, ReadingSkills)
> anova(rs1, rs2)
```

```
Analysis of Deviance Table

Model 1:
DirichReg(formula = acc ~ dyslexia * iq | phi ~ dyslexia * iq,
    data = ReadingSkills)
Model 2:
DirichReg(formula = acc ~ dyslexia * iq | phi ~ dyslexia + iq,
    data = ReadingSkills)
```

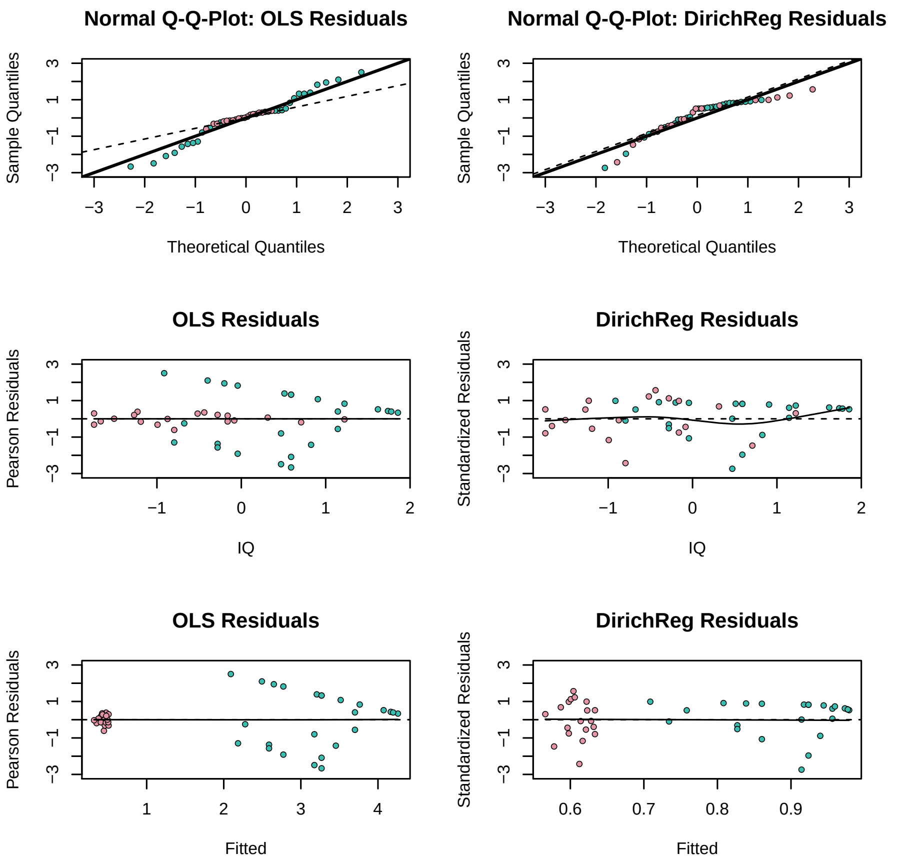
```
          Deviance   N. par   Difference   df     p-value
Model 1   -133.4682       8            -    -           -
Model 2   -131.8037       7     1.664453    1   0.1970031

> a <- ReadingSkills$accuracy
> logit_a <- log(a/(1 - a))
> rlr <- lm(logit_a ~ dyslexia * iq, ReadingSkills)
> summary(rlr)

Call:
lm(formula = logit_a ~ dyslexia * iq, data = ReadingSkills)

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

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

> summary(rs2)

Call:
DirichReg(formula = acc ~ dyslexia * iq | phi ~ dyslexia + iq,
data = ReadingSkills)


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



MEAN MODELS:
------------------------------------------------------------------------------
Coefficients for variable no. 1: 1 - data
- variable omitted (reference category) -
------------------------------------------------------------------------------
Coefficients for variable no. 2: data
               Estimate Std. Error z-Value  p-Value    
(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  p-Value
(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 **
-------------------------------------------------------------------------------
Signif. codes: `***' < .001, `**' < 0.01, `*' < 0.05, `.' < 0.1


Log-likelihood: 65.9 on 7 df (39+1 iterations)
Links: Logit (Means) and Log (Precision)
Parametrization: alternative
```

```
> g.ind <- as.numeric(ReadingSkills$dyslexia)
> plot(accuracy ~ iq, ReadingSkills, pch = 21, bg = rainbow_hcl(2)[3 -
+     g.ind], cex = 1.5, main = "Dyslexic (Red) vs. Control (Green) Group",
+     xlab = "IQ Score", ylab = "Reading Accuracy")
> x <- seq(min(ReadingSkills$iq), max(ReadingSkills$iq), length.out = 200)
> n <- length(x)
> X <- data.frame(dyslexia = rep(c("yes", "no"), each = n), iq = c(x,
+     x))
> pv <- predict(rs2, X, TRUE, TRUE, TRUE)
> lines(x, pv$mu[-(1:n), 2], col = rainbow_hcl(2)[2], lwd = 3)
> lines(x, pv$mu[1:n, 2], col = rainbow_hcl(2)[1], lwd = 3)
> olsN <- 1/(1 + exp(-predict(rlr, X[-(1:n), ])))
> olsD <- 1/(1 + exp(-predict(rlr, X[1:n, ])))
> lines(x, olsD, col = rainbow_hcl(2, l = 50)[1], lwd = 3, lty = 2)
> lines(x, olsN, col = rainbow_hcl(2, l = 50)[2], lwd = 3, lty = 2)
> par(new = TRUE)
> plot(x, pv$phi[-(1:n)], col = rainbow_hcl(2, l = 25)[2], lty = 3,
+     type = "l", ylim = c(0, max(pv$phi)), axes = F, ann = F,
+     lwd = 2)
> lines(x, pv$phi[1:n], col = rainbow_hcl(2, l = 25)[1], lty = 3,
+     type = "l", lwd = 2)
> axis(4)
> legend("topleft", legend = c(expression(hat(mu)), expression(hat(phi)),
+     "OLS"), lty = c(1, 3, 2), lwd = c(3, 2, 3), bty = "n")
```

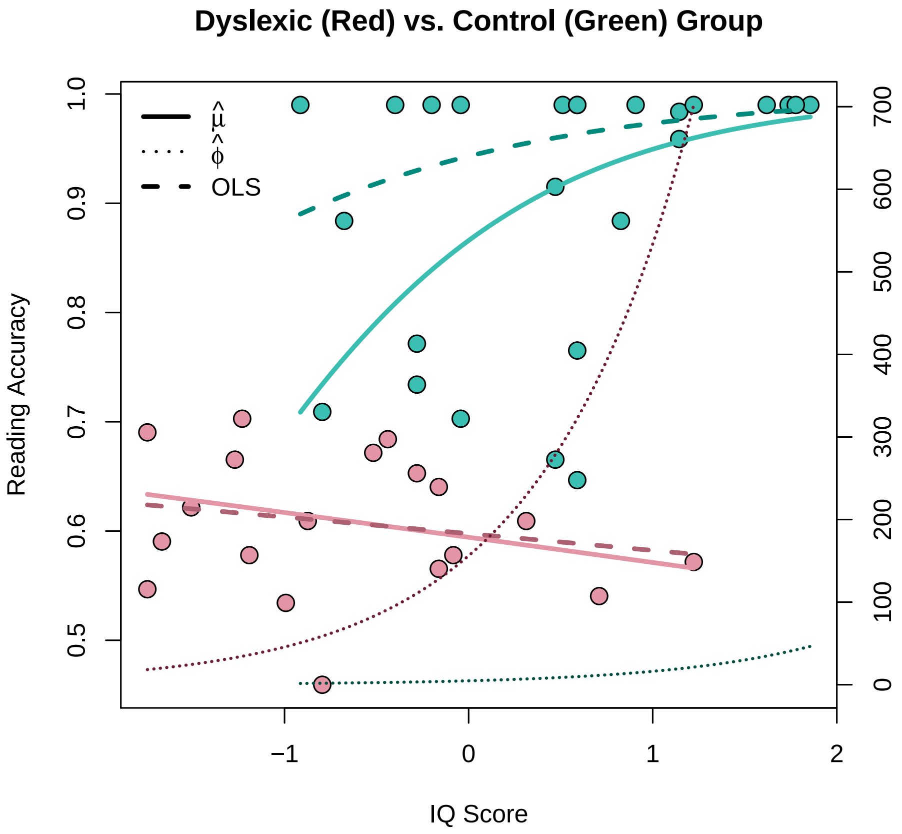


```
> gcol <- rainbow_hcl(2)[3 - as.numeric(ReadingSkills$dyslexia)]
> tmt <- c(-3, 3)
> par(mfrow = c(3, 2))
> qqnorm(residuals(rlr, "pearson"), ylim = tmt, xlim = tmt, pch = 21,
+     bg = gcol, main = "Normal QQPlot: OLS Residuals", cex = 0.75,
+     lwd = 0.5)
> abline(0, 1, lwd = 2)
> qqline(residuals(rlr, "pearson"), lty = 2)
> qqnorm(residuals(rs2, "standardized")[, 2], ylim = tmt, xlim = tmt,
+     pch = 21, bg = gcol, main = "Normal QQPlot: DirichReg Residuals",
+     cex = 0.75, lwd = 0.5)
> abline(0, 1, lwd = 2)
> qqline(residuals(rs2, "standardized")[, 2], lty = 2)
> plot(ReadingSkills$iq, residuals(rlr, "pearson"), pch = 21, bg = gcol,
+     ylim = c(-3, 3), main = "OLS Residuals", xlab = "IQ", ylab = "Pearson Residuals",
+     cex = 0.75, lwd = 0.5)
> abline(h = 0, lty = 2)
> lines(smooth.spline(ReadingSkills$iq, residuals(rlr, "pearson")))
> plot(ReadingSkills$iq, residuals(rs2, "standardized")[, 2], pch = 21,
+     bg = gcol, ylim = c(-3, 3), main = "DirichReg Residuals",
+     xlab = "IQ", ylab = "Standardized Residuals", cex = 0.75,
+     lwd = 0.5)
> abline(h = 0, lty = 2)
> lines(smooth.spline(ReadingSkills$iq, residuals(rs2, "standardized")[,
+     2]))
> plot(fitted(rlr), residuals(rlr, "pearson"), pch = 21, bg = gcol,
+     ylim = c(-3, 3), main = "OLS Residuals", xlab = "Fitted",
+     ylab = "Pearson Residuals", cex = 0.75, lwd = 0.5)
> abline(h = 0, lty = 2)
> lines(smooth.spline(fitted(rlr), residuals(rlr, "pearson")))
> plot(fitted(rs2)[, 2], residuals(rs2, "standardized")[, 2], pch = 21,
+     bg = gcol, ylim = c(-3, 3), main = "DirichReg Residuals",
+     xlab = "Fitted", ylab = "Standardized Residuals", cex = 0.75,
+     lwd = 0.5)
> abline(h = 0, lty = 2)
> lines(smooth.spline(fitted(rs2)[, 2], residuals(rs2, "standardized")[,
+     2]))
```



```
> g.ind <- as.numeric(ReadingSkills$dyslexia)
> g1 <- g.ind == 1
> g2 <- g.ind != 1
> plot(accuracy ~ iq, ReadingSkills, pch = 21, bg = rainbow_hcl(2)[3 -
+     g.ind], cex = 1.5, main = "Dyslexic (Red) vs. Control (Green) Group",
+     xlab = "IQ Score", ylab = "Reading Accuracy", xlim = range(ReadingSkills$iq))
> x1 <- seq(min(ReadingSkills$iq[g1]), max(ReadingSkills$iq[g1]),
+     length.out = 200)
> x2 <- seq(min(ReadingSkills$iq[g2]), max(ReadingSkills$iq[g2]),
+     length.out = 200)
```

```
> n <- length(x1)
> X1 <- data.frame(dyslexia = factor(rep(0, n), levels = 0:1, labels = c("no",
+     "yes")), iq = x1)
> X2 <- data.frame(dyslexia = factor(rep(1, n), levels = 0:1, labels = c("no",
+     "yes")), iq = x2)
> pv1 <- predict(rs2, X1, TRUE, TRUE, TRUE)
> pv2 <- predict(rs2, X2, TRUE, TRUE, TRUE)
> lines(x1, pv1$mu[, 2], col = rainbow_hcl(2)[2], lwd = 3)
> lines(x2, pv2$mu[, 2], col = rainbow_hcl(2)[1], lwd = 3)
> olsN <- 1/(1 + exp(-predict(rlr, X1)))
> olsD <- 1/(1 + exp(-predict(rlr, X2)))
> lines(x2, olsD, col = rainbow_hcl(2, l = 50)[1], lwd = 3, lty = 2)
> lines(x1, olsN, col = rainbow_hcl(2, l = 50)[2], lwd = 3, lty = 2)
> par(new = TRUE)
> plot(x1, pv1$phi, col = rainbow_hcl(2, l = 25)[2], lty = "11",
+     type = "l", ylim = c(0, max(pv2$phi)), axes = F, ann = F,
+     lwd = 2, xlim = range(ReadingSkills$iq))
> lines(x2, pv2$phi, col = rainbow_hcl(2, l = 25)[1], lty = "11",
+     type = "l", lwd = 2)
> axis(4)
> legend("topleft", legend = c(expression(hat(mu)), expression(hat(phi)),
+     "OLS"), lty = c(1, 3, 2), lwd = c(3, 2, 3), bty = "n")
```



References

R Development Core Team (2011). *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/`.