DirichletReg: Dirichlet Regression for Compositional Data in R

Marco J. Maier
Wirtschaftsuniversität Wien

Abstract

. . .

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

4. Application examples

4.1. The Arctic lake (common parametrization)

```
> 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[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
```

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

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

```
> par(mfrow = c(2, 1))
> plot(AL, cex = 0.5, a2d = list(colored = FALSE, c.grid = FALSE))
> plot(rep(ArcticLake$depth, 3), as.numeric(AL), pch = 21, bg = rep(c("#E495A5",
+     "#86B875", "#7DB0DD"), each = 39), xlab = "Depth (m)", ylab = "Proportion",
+     ylim = 0:1)
```

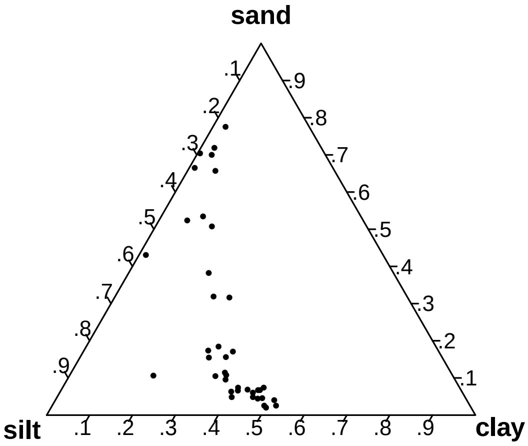


Figure 1: Arctic lake: Ternary plot and depth vs. composition.

```
Log-likelihood: 101.4 on 6 df (54+3 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
----------------------------------------
```

```
> coef(lake1)
```

```
$sand
(Intercept)       depth
 0.11662480  0.02335114

$silt
(Intercept)       depth
-0.31059591  0.05556745

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

```
> lake2 <- update(lake1, . ~ . + I(depth^2) | . + I(depth^2) | . + I(depth^2))
> 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       p-value
Model 1   -202.7393        6            -    -             -
Model 2   -217.9937        9     15.25441    3   0.001611655
```

```
> 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 p-Value
(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 p-Value
(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 p-Value
(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
---------------------------------------------------------------
Signif. codes: `***' < .001, `**' < 0.01, `*' < 0.05, `.' < 0.1

Log-likelihood: 109 on 9 df (168+2 iterations)
AIC: -200, BIC: -185.0217
Number of Observations: 39
Link: Log
Parametrization: common
```

4.2. Blood samples (alternative parametrization)

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

```
> blood1 <- DirichReg(Smp ~ Disease | 1, Bld, model = "alternative", base = 3)
> blood2 <- DirichReg(Smp ~ Disease | Disease, Bld, model = "alternative", base = 3)
> 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     p-value
Model 1   -303.8560        7            -    -           -
Model 2   -304.6147        8    0.7586655    1   0.3837465
```

```
> 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 p-Value
(Intercept)  1.11639    0.09935  11.237  <2e-16 ***
```

```
> par(mar = c(4, 4, 4, 4) + 0.1)
> plot(rep(ArcticLake$depth, 3), as.numeric(AL), pch = 21, bg = rep(c("#E495A5",
+     "#86B875", "#7DB0DD"), each = 39), xlab = "Depth (m)", ylab = "Proportion",
+     ylim = 0:1, main = "Sediment Composition in an Arctic Lake")
> Xnew <- data.frame(depth = seq(min(ArcticLake$depth), max(ArcticLake$depth),
+     length.out = 100))
> for (i in 1:3) lines(cbind(Xnew, predict(lake2, Xnew)[, i]), col = c("#E495A5",
+     "#86B875", "#7DB0DD")[i], lwd = 2)
> legend("topleft", legend = c("Sand", "Silt", "Clay"), lwd = 2, col = c("#E495A5",
+     "#86B875", "#7DB0DD"), pt.bg = c("#E495A5", "#86B875", "#7DB0DD"), 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)
> mtext(expression(paste("Precision (", phi, ")", sep = "")), 4, line = 3)
> legend("top", legend = c(expression(hat(mu[c] == hat(alpha)[c]/hat(alpha)[0])),
+     expression(hat(phi) == hat(alpha)[0])), lty = c(1, 2), lwd = c(3, 2), bty = "n")
```

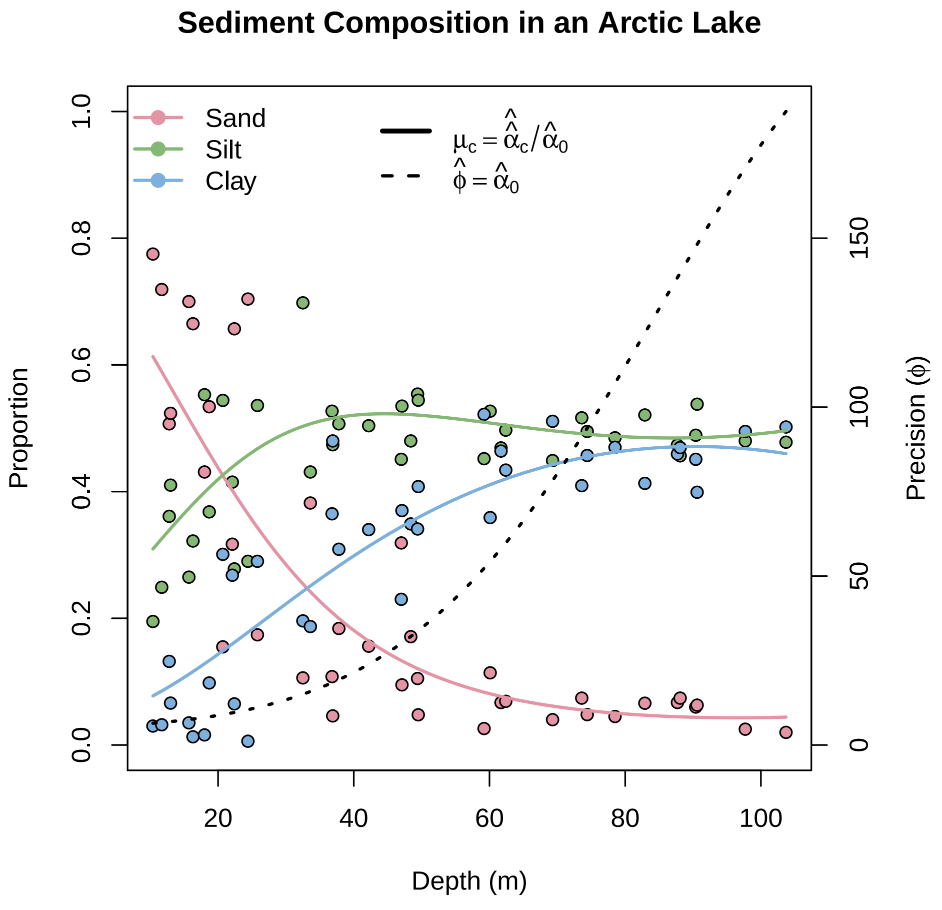


Figure 2: Arctic lake: Fitted values of the quadratic model.

```
> AL <- ArcticLake
> AL$AL <- DR_data(ArcticLake[, 1:3])
> dd <- range(ArcticLake$depth)
> X <- data.frame(depth = seq(dd[1], dd[2], length.out = 200))
> pp <- predict(DirichReg(AL ~ depth + I(depth^2), AL), X)
> plot(AL$AL, cex = 0.1, reset_par = FALSE)
> points(DirichletReg:::coord.trafo(AL$AL[, 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 = c("#6E1D34",
+     "#004E42")[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 = c("#6E1D34",
+     "#004E42")[1], lty = "21")
```

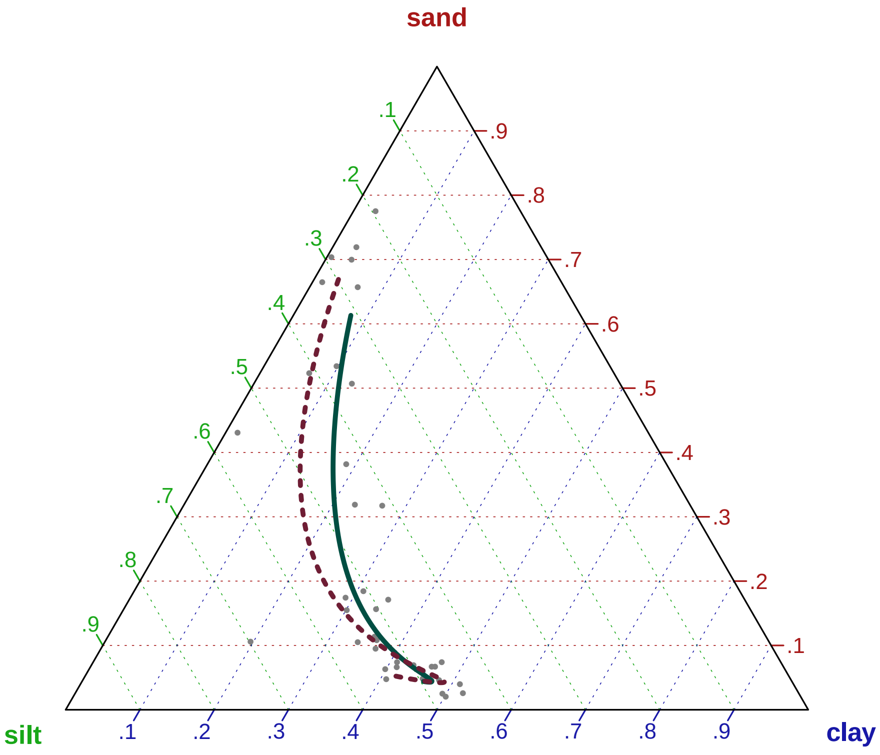


Figure 3: Arctic lake: OLS (dashed) vs. Dirichlet regression (solid) predictions.

```
> par(mfrow = c(1, 4))
> for (i in 1:4) {
+     boxplot(Bld$Smp[, i] ~ Bld$Disease, ylim = range(Bld$Smp[, 1:4]), main = paste(names(Bld)[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)
+ }
```

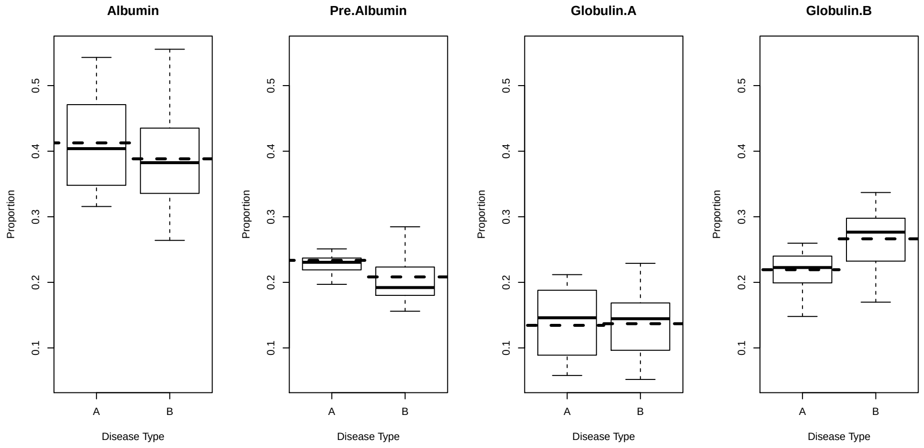


Figure 4: Blood samples: Box plots and fitted values (dashed lines indicate the fitted values for each group).

```
DiseaseB    -0.07002    0.13604  -0.515   0.607
---------------------------------------------------------------
Coefficients for variable no. 2: Pre.Albumin
            Estimate Std. Error z-Value  p-Value
(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 p-Value
(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 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 (56+2 iterations)
AIC: -289.9, BIC: -280.0476
Number of Observations: 30
Links: Logit (Means) and Log (Precision)
Parametrization: alternative
```

```
> alpha <- predict(blood2, data.frame(Disease = factor(c("A", "B"))), F, T, F)
> L <- sapply(1:2, function(i) ddirichlet(DR_data(Bld[31:36, 1:4]), unlist(alpha[i,
+     ])))
```

```
> LP <- L/rowSums(L)
> dimnames(LP) <- list(paste("C", 1:6), c("A", "B"))
> 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
```

4.3. Reading skills data (alternative parametrization)

```
> RS <- ReadingSkills
> RS$acc <- DR_data(RS$accuracy)
> RS$dyslexia <- C(RS$dyslexia, treatment)
> rs1 <- DirichReg(acc ~ dyslexia * iq | dyslexia * iq, RS, model = "alternative")
> rs2 <- DirichReg(acc ~ dyslexia * iq | dyslexia + iq, RS, model = "alternative")
> 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     p-value
Model 1   -133.4682        8            -    -           -
Model 2   -131.8037        7     1.664453    1   0.1970031
```

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

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)
> B2 <- DR_data(BloodSamples[, c(1, 2, 4)])
> plot(B2, cex = 0.001, reset_par = FALSE)
> div.col <- c("#023FA5", "#1A44A4", "#2749A4", "#314DA4", "#3952A5", "#4056A6",
+     "#465BA8", "#4D5FA9", "#5264AA", "#5868AC", "#5D6CAE", "#6371AF", "#6875B1",
+     "#6D79B3", "#727DB5", "#7681B7", "#7B85B8", "#8089BA", "#848DBC", "#8991BE",
+     "#8D95BF", "#9199C1", "#959CC3", "#9AA0C5", "#9EA4C6", "#A2A7C8", "#A5ABCA",
+     "#A9AECB", "#ADB1CD", "#B1B5CE", "#B4B8D0", "#B8BBD1", "#BBBED2", "#BEC1D4",
+     "#C1C4D5", "#C5C7D6", "#C8CAD8", "#CBCCD9", "#CDCFDA", "#D0D1DB", "#D3D4DC",
+     "#D5D6DD", "#D7D8DE", "#D9DADF", "#DBDCE0", "#DDDEE0", "#DFDFE1", "#E0E0E1",
+     "#E1E1E2", "#E2E2E2", "#E2E2E2", "#E2E1E1", "#E2E0E0", "#E1DFDF", "#E1DDDD",
+     "#E0DBDC", "#E0D9DA", "#DFD6D8", "#DED4D6", "#DDD1D3", "#DCCFD1", "#DBCCCE",
+     "#DBC9CC", "#D9C6C9", "#D8C2C6", "#D7BFC3", "#D6BCC0", "#D5B8BD", "#D4B5BA",
+     "#D2B1B7", "#D1ADB3", "#CFA9B0", "#CEA5AC", "#CCA1A9", "#CB9DA5", "#C999A2",
+     "#C7949E", "#C6909A", "#C48C96", "#C28792", "#C0828E", "#BE7E8A", "#BC7986",
+     "#B97482", "#B76F7E", "#B56A7A", "#B36576", "#B06071", "#AE5A6D", "#AB5569",
+     "#A94F64", "#A64A60", "#A3445B", "#A03E57", "#9D3752", "#9B304E", "#982949",
+     "#952045", "#911640", "#8E063B")
> temp <- (alpha/rowSums(alpha))[, c(2, 4, 1)]
> points(DirichletReg:::coord.trafo(temp/rowSums(temp)), pch = 22, bg = div.col[c(1,
+     100)], cex = 2, lwd = 0.25)
> temp <- B2[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[31:36, c(2, 3, 1)]
> points(DirichletReg:::coord.trafo(temp/rowSums(temp)), pch = 21, bg = div.col[round(100 *
+     LP[, 2], 0)], cex = 1, 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"))
```

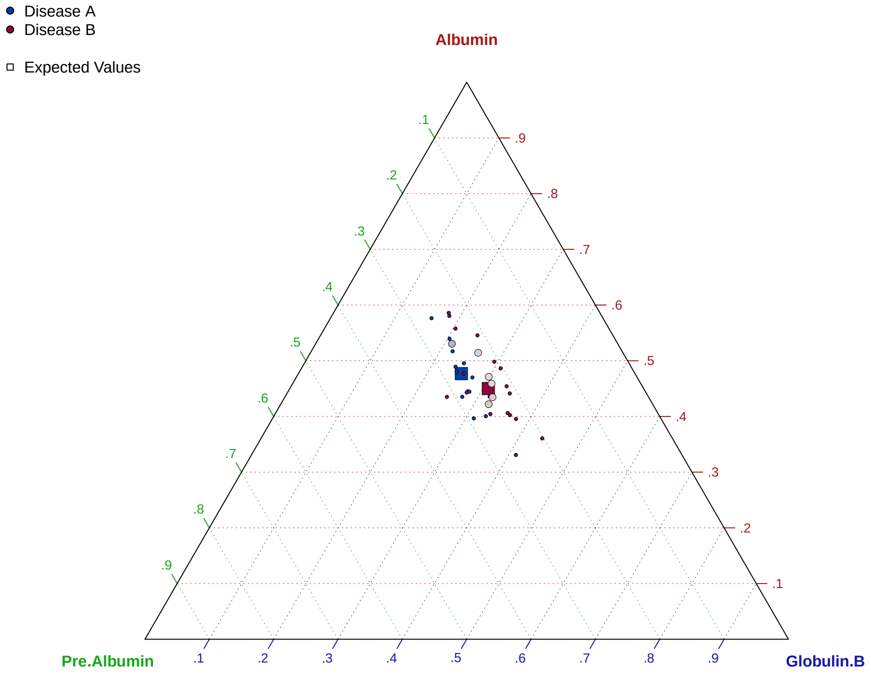


Figure 5: Blood samples: Observed values and predictions

```
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  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 (37+2 iterations)
AIC: -117.8, BIC: -105.3144
Number of Observations: 44
Links: Logit (Means) and Log (Precision)
Parametrization: alternative
```

```
> 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
```

```
> confint(rs2, exp = TRUE)
```

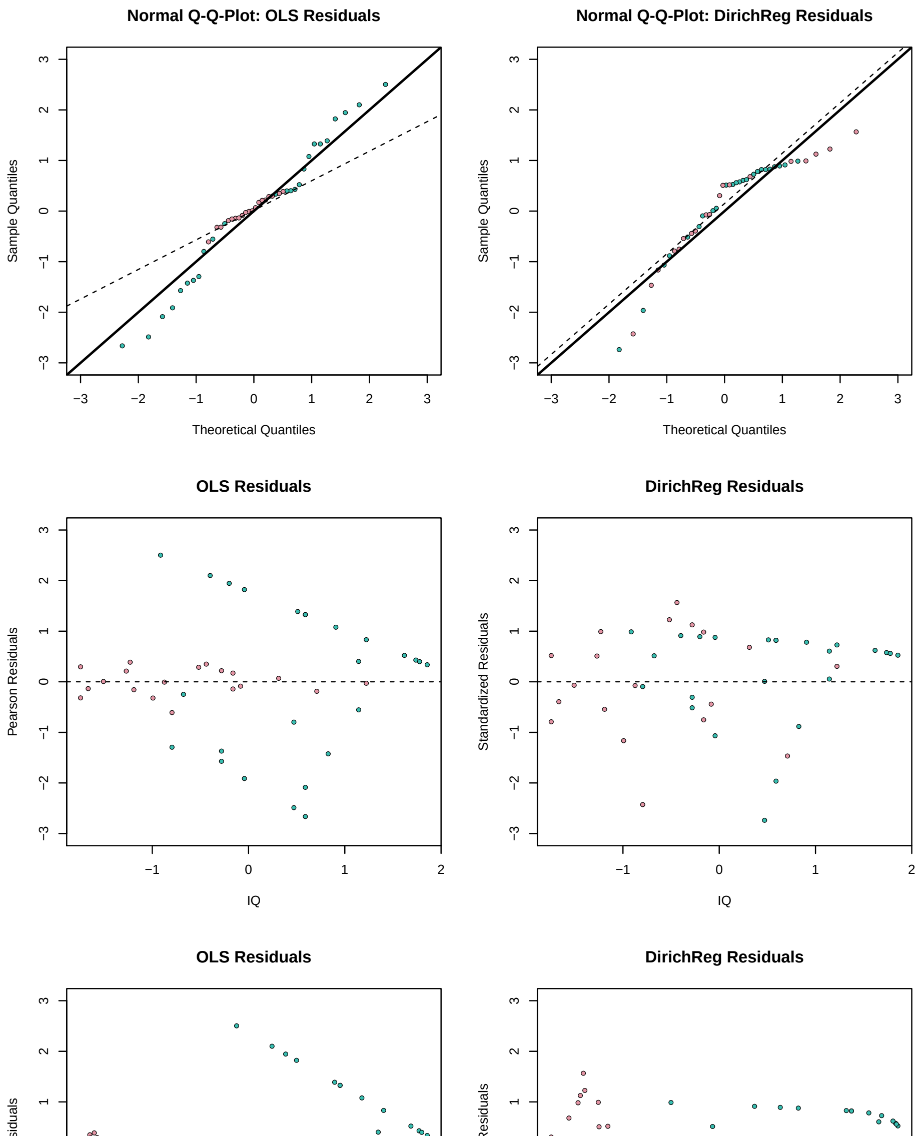
```
95% Confidence Intervals (exponentiated)

- 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
```

Affiliation:

Marco J. Maier
Institute for Statistics and Mathematics
Wirtschaftsuniversität Wien
Vienna University of Economics and Business
Augasse 2–6
1090 Vienna, Austria
Telephone: +43/1/31336–4335
E-mail: Marco.Maier@wu.ac.at
URL: http://statmath.wu.ac.at/~maier/

```
> gcol <- c("#E495A5", "#39BEB1")[3 - as.numeric(RS$dyslexia)]
> tmt <- c(-3, 3)
> par(mfrow = c(3, 2))
> qqnorm(residuals(rlr, "pearson"), ylim = tmt, xlim = tmt, pch = 21, bg = gcol,
+     main = "Normal Q■Q■Plot: OLS Residuals", cex = 0.75, lwd = 0.5)
> abline(0, 1, lwd = 2)
> qqline(residuals(rlr, "pearson"), lty = 2)
> qqnorm(residuals(rs2, "standardized")[, 2], ylim = tmt, xlim = tmt, pch = 21,
+     bg = gcol, main = "Normal Q■Q■Plot: DirichReg Residuals", cex = 0.75, lwd = 0.5)
> abline(0, 1, lwd = 2)
> qqline(residuals(rs2, "standardized")[, 2], lty = 2)
> plot(ReadingSkills$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)
> 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)
> 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)
> 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)
```



```
> g.ind <- as.numeric(RS$dyslexia)
> g1 <- g.ind == 1
> g2 <- g.ind != 1
> par(mar = c(4, 4, 4, 4) + 0.1)
> plot(accuracy ~ iq, RS, pch = 21, bg = c("#E495A5", "#39BEB1")[3 - g.ind], cex = 1.5,
+     main = "Dyslexic (Red) vs. Control (Green) Group", xlab = "IQ Score", ylab = "Reading Accuracy",
+     xlim = range(ReadingSkills$iq))
> x1 <- seq(min(RS$iq[g1]), max(RS$iq[g1]), length.out = 200)
> x2 <- seq(min(RS$iq[g2]), max(RS$iq[g2]), length.out = 200)
> n <- length(x1)
> X <- data.frame(dyslexia = factor(rep(0:1, each = n), levels = 0:1, labels = c("no",
+     "yes")), iq = c(x1, x2))
> pv <- predict(rs2, X, TRUE, TRUE, TRUE)
> lines(x1, pv$mu[1:n, 2], col = c("#E495A5", "#39BEB1")[2], lwd = 3)
> lines(x2, pv$mu[(n + 1):(2 * n), 2], col = c("#E495A5", "#39BEB1")[1], lwd = 3)
> ols <- 1/(1 + exp(-predict(rlr, X)))
> lines(x1, ols[1:n], col = c("#AD6071", "#00897D")[2], lwd = 3, lty = 2)
> lines(x2, ols[(n + 1):(2 * n)], col = c("#AD6071", "#00897D")[1], lwd = 3, lty = 2)
> par(new = TRUE)
> plot(x1, pv$phi[1:n], col = c("#6E1D34", "#004E42")[2], lty = "11", type = "l",
+     ylim = c(0, max(pv$phi)), axes = F, ann = F, lwd = 2, xlim = range(RS$iq))
> lines(x2, pv$phi[(n + 1):(2 * n)], col = c("#6E1D34", "#004E42")[1], lty = "11",
+     type = "l", lwd = 2)
> axis(4)
> mtext(expression(paste("Precision (", phi, ")", sep = "")), 4, line = 3)
> legend("topleft", legend = c(expression(hat(mu)), expression(hat(phi)), "OLS"),
+     lty = c(1, 3, 2), lwd = c(3, 2, 3), bty = "n")
```

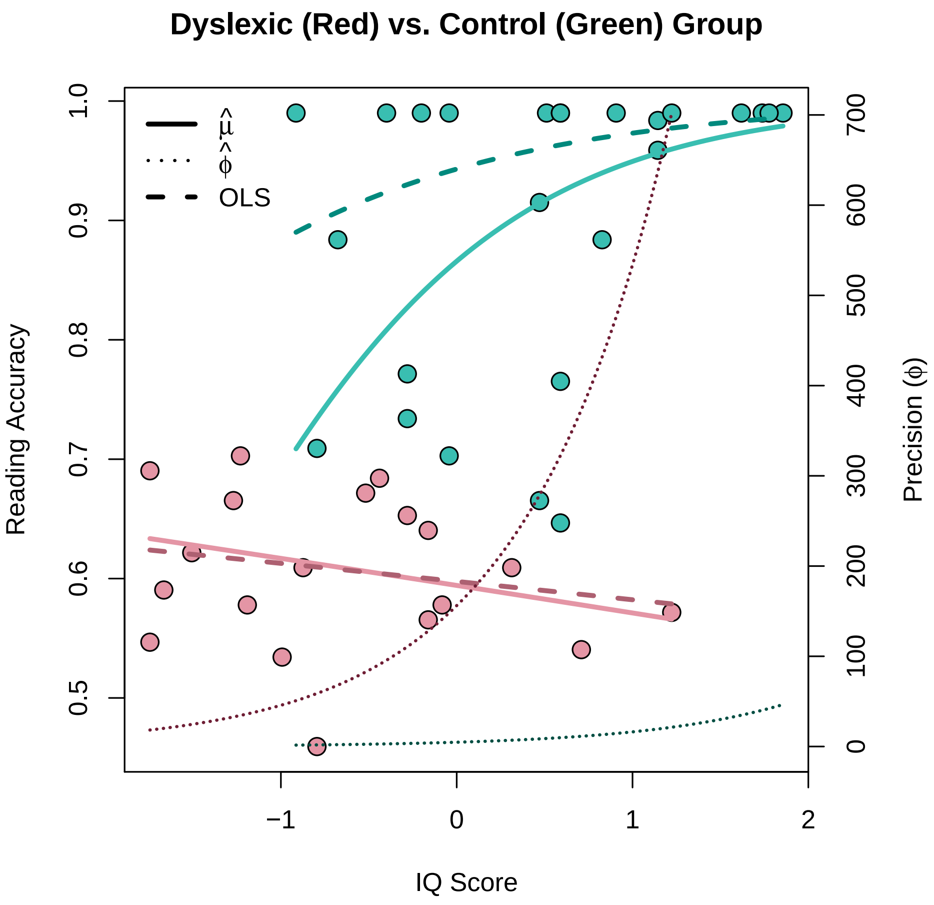


Figure 7: Reading skills: Predicted values of Dirichlet regression and OLS regression.