mixor: An R Package for Longitudinal and Clustered Ordinal Response Modeling

Kellie J. Archer The Ohio State University
Donald Hedeker University of Illinois Chicago
Rachel Nordgren University of Chicago
Robert D. Gibbons University of Chicago

Abstract

This paper describes an R package, **mixor**, that provides a function for fitting an ordinal response model when observations are either clustered or collected longitudinally. The function, `mixor` uses either adaptive (default) or non-adaptive Gauss-Hermite quadrature to numerically integrate over the distribution of random effects and Fisher scoring to obtain the likelihood solution, as described by Hedeker and Gibbons (1996). Generic methods including `summary`, `print`, `vcov`, `plot`, `predict`, and `coef` can be applied to a `mixor` object. Examples of modeling longitudinal data, clustered data, grouped survival times, and weighted data are provided.

Keywords: ordinal response, longitudinal data, clustered data, random effects, R.

1. Introduction

Health status and outcomes, such as quality of life, functional status, and patient satisfaction, are frequently measured on an ordinal scale. In addition, most histopathological variables are ordinal, including scoring methods for liver biopsy specimens from patients with chronic hepatitis, such as the Knodell hepatic activity index, the Ishak score, and the METAVIR score. Often, outcomes collected are either clustered or collected longitudinally. For example, stage of breast cancer is derived using degree of tubule formation, nuclear pleomorphism, and mitotic count (Ivshina, George, Senko, Mow, Putti, Smeds, Lindahl, Pawitan, Hall, Nordgren, Wong, Liu, Bergh, Kuznestsov, and Miller 2006). Similarly, stage of hypopharngeal cancer is derived using three ordinally scaled measures: tumor, node, and metastasis scores (Cromer, Carles, Millon, Ganguli, Chalmel, Lemaire, Young, Dembélé, Thibault, Muller, Poch, Abesassis, and Wasylyk 2004). Multiple ordinally scaled variables give rise to clustered ordinal response data. In addition, some studies collect an ordinal response on each subject at multiple time-points. Currently only the **ordinal** package in the R programming environment provides a function for fitting cumulative link ordinal response random effects models (Christensen 2013). Although **ordinal** can fit both random intercept and random coefficient models, it currently does not implement quadrature methods for vector-valued random effects (including random coefficients models) nor can it handle nested random effects structures.

MIXOR, written in Fortran, is a stand-alone package for fitting cumulative link ordinal (and dichotomous) response models (Hedeker and Gibbons 1996). **MIXOR** supports probit, logit, and complementary log-log link functions and can fit models that include multiple random

effects. The stand-alone program requires the user to either specify all model instructions in an ASCII text file following a precisely defined format (Hedeker and Gibbons 1996) or specify the model options using a GUI interface which subsequently creates the batch-defined ASCII file. After submitting the job using the batch or interactive model, the empicial Bayes estimates, parameter estimates, and asymptotic variance-covariance matrix of the parameter estimates for the fitted model are written to different files (MIXOR.RES, MIXOR.EST, and MIXOR.VAR, respectively). To enhance the usability of the stand-alone program, we developed an R package **mixor** that interfaces to a Fortran dynamic-link library for fitting cumulative link ordinal response mixed effects models. Modeling results are returned within R providing a convenient environment for additional model fitting, testing, and plotting.

Several software packages include procedures for fitting mixed models for ordinal outcomes, however **MIXOR** contains some unique features. **MIXOR** uses full-likelihood methods to estimate the model parameters, which is also the default method in Stata `meologit`, and can be obtained in SAS `PROC GLIMMIX`. Alternatively, IBM SPSS only provides quasi-likelihood solutions. All of these major software programs only provide for proportional odds models (or the equivalent under the probit or clog-log links), and thus cannot be used to estimate partial, non-proportional odds, and scaling models described in Sections 5.2 and 5.3. Even if one is not interested in a non-proportional odds model, by being able to estimate such a model, users can perform a likelihood-ratio test of the proportional odds assumption. Such a test is not possible with these other software programs. Furthermore, estimation of survival models, as described in Section 5.2, is also unique to **MIXOR**. Finally, **MIXOR**, which is written in Fortran, is a relatively fast program which can easily accommodate multiple (correlated) random effects.

2. Ordinal response model

Herein we briefly describe the cumulative logit model for the traditional setting where data are neither clustered nor collected longitudinally, to demonstrate the natural connections to the dichotomous setting. Let Y_i represent the ordinal response for observation i that can take on one of K ordinal levels. Denote the $N \times P$ covariate matrix as $\mathbf{x}$ so that $\mathbf{x}_i$ represents a $P \times 1$ vector for observation i and $\mathbf{x}_p$ represents the $N \times 1$ vector for covariate p. For observations $i = 1, \ldots, N$, the response Y_i can be reformatted as a response matrix consisting of N rows and K columns where

$$y_{ik} = \begin{cases} 1 \text{ if observation } i \text{ is class } k \\ 0 \text{ otherwise.} \end{cases} .$$

Therefore $\mathbf{y}_k$ is an $N \times 1$ vector representing class k membership. Letting $\pi_k(\mathbf{x}_i)$ represent the probability that observation i with covariates $\mathbf{x}_i$ belongs to class k, the likelihood for an ordinal response model with K ordinal levels can be expressed as

$$L = \prod_{i=1}^{N} \prod_{k=1}^{K} \pi_k(\mathbf{x}_i)^{y_{ik}}. \tag{1}$$

The cumulative logit model models $K - 1$ logits of the form

$$P(Y_i \leq k) = \frac{\exp(\alpha_k - \mathbf{x}_i^\top \boldsymbol{\beta})}{1 + \exp(\alpha_k - \mathbf{x}_i^\top \boldsymbol{\beta})} \tag{2}$$

where α_k denotes the class-specific intercept or threshold and $\boldsymbol{\beta}$ is a $P \times 1$ vector of coefficients associated with explanatory variables $\mathbf{x}_i$ (Agresti 2010). Equation 2 is formulated to subtract the $\mathbf{x}_i^\top \boldsymbol{\beta}$ term from the thresholds as described in the seminal paper by McCullagh (1980), which provides an intuitive interpretation of the relationship between $\mathbf{x}_i^\top \boldsymbol{\beta}$ and the probability of response; larger values of $\mathbf{x}_i^\top \boldsymbol{\beta}$ correspond to higher probability of the response belonging to an ordinal class at the higher end of the scale. Other software packages fit cumulative link models using a plus sign in Equation 2 so the **mixor** package flexibly permits the user to change the model parameterization. Note that the class-specific probabilities can be calculated by subtracting successive cumulative logits,

$$P(Y_i = k) = P(Y_i \leq k) - P(Y_i \leq k-1).$$

Therefore for any class k, providing we let $-\infty = \alpha_0 < \alpha_1 < \cdots < \alpha_{K-1} < \alpha_K = \infty$, we can express the class-specific probabilities by

$$\pi_k(\mathbf{x}_i) = \frac{\exp(\alpha_k - \mathbf{x}_i^\top \boldsymbol{\beta})}{1 + \exp(\alpha_k - \mathbf{x}_i^\top \boldsymbol{\beta})} - \frac{\exp(\alpha_{k-1} - \mathbf{x}_i^\top \boldsymbol{\beta})}{1 + \exp(\alpha_{k-1} - \mathbf{x}_i^\top \boldsymbol{\beta})}.$$

The function that links the probability to the linear predictor in Equation 2 is the logit link,

$$\log\left(\frac{P(Y_i \leq k)}{1 - P(Y_i \leq k)}\right) = \alpha_k - \mathbf{x}_i^\top \boldsymbol{\beta}.$$

Other link functions that can be used to link the cumulative probabilities to the linear predictor include the probit link,

$$\Phi^{-1}(P(Y_i \leq k))$$

where Φ^{-1} is the inverse of the cumulative standard normal distribution function; and the complementary log-log link

$$\log(-\log(1 - P(Y_i = k))).$$

3. Longitudinal/clustered ordinal response models

Consider now the scenario where subjects $i = 1, \ldots, N$ (level-2 units) are each observed $j = 1, \cdots, n_i$ times (level-1 units) where the response at each j belongs to one of $k = 1, \ldots, K$ ordered categories. Here j could index either clustered or longitudinal observations for unit i. We let p_{ijk} represent the probability that a subject i at j falls into class k. Therefore, the cumulative probability at j is $P(Y_{ij} \leq k) = \sum_{l=1}^{k} p_{ijl}$. The mixed-effects logistic regression model for the $K - 1$ cumulative logits is then given by

$$\log\left(\frac{P(Y_{ij} \leq k)}{1 - P(Y_{ij} \leq k)}\right) = \alpha_k - (\mathbf{x}_{ij}^\top \boldsymbol{\beta} + \mathbf{z}_{ij}^\top \mathbf{T} \boldsymbol{\theta}_i) \tag{3}$$

where the thresholds given by $(\alpha_1, \alpha_2, \ldots, \alpha_{K-1})$ are strictly increasing, $\mathbf{x}_{ij}$ is the covariate vector, and $\boldsymbol{\beta}$ is the vector of regression parameters. The unstandardardized random effects $\boldsymbol{\nu}_i \sim \text{MVN}(\mu, \boldsymbol{\Sigma}_{\nu_i})$ which are expressed by the standardized vector of the r random effects $\boldsymbol{\theta}_i$ and where $\mathbf{T}$ is the Cholesky factorization of $\boldsymbol{\Sigma}_\nu$ and $\mathbf{z}_{ij}$ is the design vector for the r random effects. Letting the response vector be denoted by $\mathbf{y}_i = (Y_{ij1}, Y_{ij1}, \ldots, Y_{ijK})$ where $y_{ijk} = 1$

if the response for subject i at j is in category k and 0 otherwise, so that $n_i = \sum_j \sum_{k=1}^{K} y_{ijk}$. The likelihood can be expressed as

$$l(\mathbf{y}_i|\theta_i) = \prod_{j=1}^{n_i} \prod_{k=1}^{K} (P(y_{ij} \leq k) - P(y_{ij} \leq k-1))^{y_{ijk}} \tag{4}$$

Details on the maximum marginal likelihood estimation procedure which uses multidimensional quadrature to integrate over the distribution of random effects and Fisher's scoring for obtaining the solution to the likelihood have been previously described (Hedeker and Gibbons 2006, 1996).

4. Implementation

The **mixor** package was written in the R programming environment (R Core Team 2014) and requires the **survival** package (Therneau 2014) when fitting discrete survival time models. The `mixor` function allows the user to specify a model formula, identify the level-2 identifier using the `id` parameter, and additionally specify whether any variables have a random slope (`which.random.slope`). The function also supports fitting non-proportional odds models by specifying the variables for which proportional odds is not assumed (`KG`) and can fit scaling models (`KS`). The function parses the user-specified parameters which are subsequently passed to a **MIXOR** Fortran dynamic linked library (Hedeker and Gibbons 1996) and results are returned to the fitted object in the form of a list. The default link is `link = "probit"`. Other allowable links include `logit` and `cloglog`. The function uses adaptive quadrature with 11 quadrature points by default which can be changed by specifying `adaptive.quadrature = FALSE` to perform non-adaptive quadrature; the number of quadrature points to be used for each dimension of the integration can be changed by specifying a different integer value to `nAGQ` in the function call. By default the quadrature distribution is normal (`quadrature.dist = "Normal"`) but the function supports usage of a uniform distribution (`quadrature.dist = "Uniform"`). Also, by default the random effects are assumed to be correlated; if independent random effects are assumed, then `indep.re = TRUE` should be specified. Generic methods for returning coefficient estimates, printing summaries, extracting variance-covariance estimates, and obtaining predictions from the fitted model are available using `print`, `coef`, `summary`, `plot`, `vcov`, and `predict`.

5. Examples

The **mixor** package includes example datasets that are useful for demonstrating modeling a longitudinal ordinal response (`schizophrenia`), grouped survival data (`SmokeOnset`), frequency weighted data (`norcag`), and clustered ordinal response (`concen`).

5.1. Longitudinal data

These data are from the National Institute of Mental Health Schizophrenia Collaborative Study and are stored in the `data.frame schizophrenia` (Gibbons and Hedeker 1994). Patients were randomized to receive one of four medications, either placebo or one of three different anti-psychotic drugs (chlorpromazine, fluphenazine, or thioridazine). The protocol

indicated subjects were to be evaluated at weeks 0, 1, 3, 6 to assess severity of illness; additionally some measurements were made at weeks 2, 4, and 5. The primary outcome was item 79 on the Inpatient Multidimensional Psychiatric Scale which indicates severity of illness having the following interpretation: 1 = normal, not ill at all; 2 = borderline mentally ill; 3 = mildly ill; 4 = moderately ill; 5 = markedly ill; 6 = severly ill; and 7 = among the most extremely ill. Because the number of subjects in the response categories was highly imbalanced, the response was previously analyzed as a dichotomous variable (≤ 3 vs ≥ 4) (Gibbons and Hedeker 1994). To retain more information about the response but to ensure each response category has a relatively large number of respondents, here we analyze `imps79o`, which is an ordinally scaled version of the original variable `imps79`. The four category ordinal version (`imps79o`) grouped the responses as follows:

`imps79`	`imps79o`
1 & 2	1 (not ill or borderline)
3 & 4	2 (mildly or moderately)
5	3 (markedly)
6 & 7	4 (severely or most extremely ill)

Predictor variables of interest are `TxDrug` a dummy coded variable indicating treatment with drug (`1`) or placebo (`0`), the square root of the Week variable (`SqrtWeek`), and their interaction (`TxSWeek`).

Random intercept model

A random intercepts logit link model can be fit as follows:

```
R> library("mixor")
R> data("schizophrenia")
R> SCHIZO1.fit <- mixor(imps79o ~ TxDrug + SqrtWeek + TxSWeek,
+    data = schizophrenia, id = id, link = "logit")
```

Note that the user supplies the model formula in the traditional way, specifies the `data.frame` name using `data`, the level-2 variable using `id`, and the link function using `link`. Methods such as `summary` and `print` can be applied to `mixor` model objects.

```
R> summary(SCHIZO1.fit)

Call:
mixor(formula = imps79o ~ TxDrug + SqrtWeek + TxSWeek, data = schizophrenia,
    id = id, link = "logit")

Deviance =          3402.758
Log-likelihood =  -1701.379
RIDGEMAX =          0.2
AIC =              3416.758
SBC =              3445.318
```

```
                  Estimate Std. Error z value   P(>|z|)
(Intercept)        5.85924    0.34288 17.0881 < 2.2e-16 ***
TxDrug            -0.05843    0.31086 -0.1880    0.8509
SqrtWeek          -0.76577    0.11975 -6.3950 1.606e-10 ***
TxSWeek           -1.20615    0.13314 -9.0595 < 2.2e-16 ***
Random.(Intercept) 3.77378    0.49543  7.6172 2.598e-14 ***
Threshold2         3.03282    0.13238 22.9103 < 2.2e-16 ***
Threshold3         5.15077    0.17925 28.7345 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

While `print` simply prints a brief summary of the coefficients to the console, `coef` extracts the estimated parameters and returns them as a vector and `vcov` extracts and returns the asymptotic variance-covariance matrix of the parameter estimates. Note that the variance of the random effect is returned rather than the Cholesky, which is returned in the original stand-alone Fortran implementation of **MIXOR**.

```
R> print(SCHIZO1.fit)

Call:
mixor(formula = imps79o ~ TxDrug + SqrtWeek + TxSWeek, data = schizophrenia,
    id = id, link = "logit")

Coefficients:
       (Intercept)              TxDrug            SqrtWeek
           5.85924            -0.05843            -0.76577
           TxSWeek  Random.(Intercept)          Threshold2
          -1.20615             3.77378             3.03282
        Threshold3
           5.15077

R> coef(SCHIZO1.fit)

       (Intercept)              TxDrug            SqrtWeek
        5.85924463        -0.05843032        -0.76577287
           TxSWeek Random.(Intercept)         Threshold2
       -1.20615055         3.77377698         3.03282074
        Threshold3
        5.15076669

R> vcov(SCHIZO1.fit)

                  (Intercept)        TxDrug     SqrtWeek
(Intercept)        0.11757010 -0.0748269661 -0.025627080
TxDrug            -0.07482697  0.0966361929  0.019424570
SqrtWeek          -0.02562708  0.0194245699  0.014339110
```

```
TxSWeek             0.01101348 -0.0231741746 -0.012056227
Random.(Intercept)  0.05476386 -0.0148306049 -0.008162231
Threshold2          0.02417834  0.0003265383 -0.002781150
Threshold3          0.03843122 -0.0029222399 -0.005920412
                        TxSWeek Random.(Intercept)    Threshold2
(Intercept)         0.011013480        0.054763860  0.0241783364
TxDrug             -0.023174175       -0.014830605  0.0003265383
SqrtWeek           -0.012056227       -0.008162231 -0.0027811500
TxSWeek             0.017725465       -0.011636936 -0.0045471359
Random.(Intercept) -0.011636936        0.245449190  0.0230803673
Threshold2         -0.004547136        0.023080367  0.0175239795
Threshold3         -0.006323587        0.047288216  0.0198707540
                     Threshold3
(Intercept)         0.038431220
TxDrug             -0.002922240
SqrtWeek           -0.005920412
TxSWeek            -0.006323587
Random.(Intercept)  0.047288216
Threshold2          0.019870754
Threshold3          0.032131934
```

By default the `mixor` function returns the `Intercept` and the `Threshold2` and `Threshold3` values which represent the ordinal departures from the intercept. If the $K-1$ cutpoints are desired, they can be obtained using the `Contrasts` function.

```
R> cm <- matrix(c(-1, 0, 0, 0, 0, 0, 0,
+    -1, 0, 0, 0, 0, 1, 0,
+    -1, 0, 0, 0, 0, 0, 1), ncol = 3)
R> Contrasts(SCHIZO1.fit, contrast.matrix = cm)
```

```
                    1  2  3
(Intercept)        -1 -1 -1
TxDrug              0  0  0
SqrtWeek            0  0  0
TxSWeek             0  0  0
Random.(Intercept)  0  0  0
Threshold2          0  1  0
Threshold3          0  0  1
```

```
  Estimate Std. Error  z value   P(>|z|)
1 -5.85924    0.34288 -17.0881 < 2.2e-16 ***
2 -2.82642    0.29451  -9.5970 < 2.2e-16 ***
3 -0.70848    0.26989  -2.6251  0.008663 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `plot` function produces a histogram and normal quantile-quantile plot of the empirical Bayes means for each random term. The `predict` function returns an object that includes

```
R> plot(SCHIZO1.fit)
```

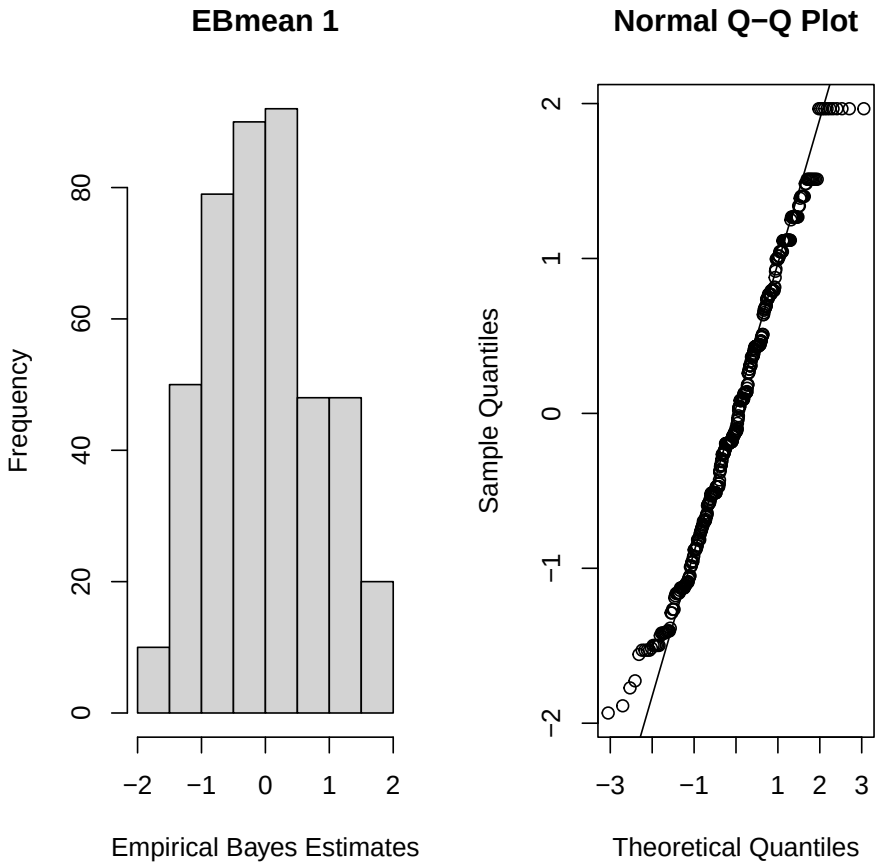

Figure 1: Histogram and normal quantile-quantile plot of the empirical Bayes means from a random intercepts cumulative logit model fit using `mixor` to the `schizophrenia` data.

a matrix of the class-specific probabilities estimated from the model (`predicted`) as well as the predicted class (`class`). This function includes an optional `newdata` parameter to be used for obtaining predictions on an independent dataset, and returns predictions when the random effects are zero (e.g., for an average subject). All variables used in the `mixor` model, the fixed and the random effects models, as well as the grouping factors, must be present in the new data frame. When `newdata` is not supplied, the random effects estimates are used in obtaining model predictions.

```
R> pihat <- predict(SCHIZO1.fit)
R> names(pihat)

[1] "predicted" "class"
```

```
R> table(pihat$class, schizophrenia$imps79o)

     1   2   3   4
  1 31   4   0   0
  2 151 305 111  29
  3   8 139 187 111
  4   0  26 114 387

R> head(pihat$predicted)

               1          2          3          4
1103 0.003487685 0.06423542 0.30881779 0.62345911
1103 0.024528152 0.31839851 0.46977585 0.18729749
1103 0.096262975 0.59229066 0.25984917 0.05159719
1103 0.304746385 0.59622137 0.08598403 0.01304821
1104 0.004742665 0.08526172 0.36124215 0.54875346
1104 0.033102659 0.38230413 0.43983111 0.14476210
```

Random intercept and slope

It may be of interest to account for subject heterogeneity through both the intercept and by time. A model that includes a random intercept and slope can be fit by additionally specifying the index corresponding to the variable(s) on the right-hand side (RHS) of the equation that should have a random coefficient(s) using the `which.random.slope` parameter. Note that multiple variables can be specified by the `which.random.slope` parameter. For example, `which.random.slope=c(1,3)` indicates that the first and third variables listed on the RHS of the model formula should be random coefficients. In this example, `SqrtWeek` is the second variable listed in the RHS of the model formula and allowing it to have a random coefficient is specified by `which.random.slope=2`. The variance-covariance matrix of the random effects is returned (`(Intercept) (Intercept)`, `(Intercept) SqrtWeek`, `SqrtWeek SqrtWeek`) rather than the Cholesky, which is returned in the original stand-alone Fortran implementation of **MIXOR**.

```
R> SCHIZO2.fit <- mixor(imps79o ~ TxDrug + SqrtWeek + TxSWeek,
+    data = schizophrenia, id = id, which.random.slope = 2, link = "logit")
R> summary(SCHIZO2.fit)

Call:
mixor(formula = imps79o ~ TxDrug + SqrtWeek + TxSWeek, data = schizophrenia,
    id = id, which.random.slope = 2, link = "logit")

Deviance =         3325.486
Log-likelihood =  -1662.743
RIDGEMAX =         0
AIC =             3343.486
SBC =             3380.205
```

```
                         Estimate Std. Error z value   P(>|z|)
(Intercept)              7.318831   0.480778 15.2229 < 2.2e-16 ***
SqrtWeek                -0.882261   0.234568 -3.7612 0.0001691 ***
TxDrug                   0.057917   0.399102  0.1451 0.8846178
TxSWeek                 -1.694861   0.268131 -6.3210 2.598e-10 ***
(Intercept) (Intercept)  6.997646   1.369273  5.1105 3.213e-07 ***
(Intercept) SqrtWeek    -1.508514   0.536023 -2.8143 0.0048888 **
SqrtWeek SqrtWeek        2.008916   0.453587  4.4290 9.469e-06 ***
Threshold2               3.901260   0.213257 18.2937 < 2.2e-16 ***
Threshold3               6.507172   0.289991 22.4392 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By default, the model is fit assuming the random effect terms are correlated. The following components can be extracted from the fitted object as needed.

```
R> names(SCHIZO2.fit)
```

```
 [1] "call"                "Deviance"            "Quadrature.points"
 [4] "Model"               "varcov"              "EBmean"
 [7] "EBvar"               "RIDGEMAX"            "RLOGL"
[10] "SE"                  "AIC"                 "SBC"
[13] "AICD"                "SBCD"                "MU"
[16] "ALPHA"               "SIGMA"               "GAM"
[19] "TAU"                 "IADD"                "Y"
[22] "X"                   "W"                   "MAXJ"
[25] "random.effect.mean" "KS"                  "KG"
[28] "id"                  "which.random.slope" "ICEN"
[31] "link"                "terms"
```

However, specific functions have been developed for extracting the log-likelihood, deviance, AIC, and BIC from the random intercept and the random coefficient model.

```
R> logLik(SCHIZO1.fit)
```

```
[1] -1701.379
```

```
R> deviance(SCHIZO1.fit)
```

```
[1] 3402.758
```

```
R> AIC(SCHIZO1.fit)
```

```
[1] 3416.758
```

```
R> BIC(SCHIZO1.fit)
```

```
[1] 3445.318
```

To fit a model assuming independent random effects, the `indep.re` parameter should be `TRUE`.

```
R> SCHIZO3.fit <- mixor(imps79o ~ TxDrug + SqrtWeek + TxSWeek,
+    data = schizophrenia, id = id, which.random.slope = 2, indep.re = TRUE,
+    link = "logit")
R> summary(SCHIZO3.fit)

Call:
mixor(formula = imps79o ~ TxDrug + SqrtWeek + TxSWeek, data = schizophrenia,
    id = id, which.random.slope = 2, link = "logit", indep.re = TRUE)

Deviance =          3338.65
Log-likelihood =  -1669.325
RIDGEMAX =          0
AIC =              3354.65
SBC =              3387.289

                   Estimate Std. Error z value   P(>|z|)
(Intercept)         6.79432    0.40000 16.9858 < 2.2e-16 ***
SqrtWeek           -0.69872    0.19527 -3.5783 0.0003459 ***
TxDrug              0.08664    0.31505  0.2750 0.7833147
TxSWeek            -1.66334    0.22432 -7.4151 1.215e-13 ***
Random.(Intercept)  4.10323    0.73666  5.5700 2.547e-08 ***
Random.SqrtWeek     1.24592    0.27490  4.5323 5.834e-06 ***
Threshold2          3.77717    0.20157 18.7386 < 2.2e-16 ***
Threshold3          6.19845    0.26300 23.5680 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.2. Grouped-time survival data

The `SmokeOnset data.frame` are data from the Television School and Family Smoking Prevention and Cessation Project, a study designed to increase knowledge of the effects of tobacco use in school-age children (Flay, Miller, Hedeker, Siddiqui, Brannon, Johnson, Hansen, Sussman, and Dent 1995). In this study students are nested within class and classes are nested within schools, so either `class` or `school` can be used as the level-2 variable. The primary outcome is time to smoking experimentation (`smkonset`) which was recorded as `1` = post-intervention, `2` = 1-year follow-up, and `3` = 2-year follow-up. Because not all students tried smoking by the end of the study, the variable `event` indicates whether the student smoked (`1`) or did not smoke (`0`). Therefore, the possible outcomes and their descriptions are provided in Table 1. This dataset represents grouped-time survival data which can be modeled via the `mixor` function using `link="cloglog"` to yield a proportional hazards survival model.

The first example represents `class` as the level-2 variable. The `IADD` parameter indicates whether the $\mathbf{x}^\top\boldsymbol{\beta}$ term is added (`IADD=1`) or subtracted (`IADD=0`, default) from the thresholds

Table 1: Interpretation of `smkonset` and `event` in the `SmokeOnset` dataset.

smkonset	event	description
1	1	smoked at post-intervention
1	0	did not smoke at post-intervention, was censored thereafter
2	1	did not smoke at post-intervention, but did smoke at 1-year follow-up
2	0	did not smoke at post-intervention or 1-year follow-up, was censored thereafter (i.e., was not measured at year 2)
3	1	did not smoke at post-intervention or 1-year, but did smoke at 2-year follow-up
3	0	did not smoke at post-intervention, 1-year, or 2-year follow-up

in Equation 2. In survival models, it is customary that positive regression coefficients represent increased hazard so `IADD=1` is specified here.

```
R> data("SmokeOnset")
R> require("survival")
R> Surv.mixord <- mixor( Surv(smkonset, event) ~ SexMale + cc + tv,
+    data = SmokeOnset, id = class, link = "cloglog", nAGQ = 20,
+    IADD = 1)
R> summary(Surv.mixord)

Call:
mixor(formula = Surv(smkonset, event) ~ SexMale + cc + tv, data = SmokeOnset,
    id = class, nAGQ = 20, link = "cloglog", IADD = 1)

Deviance =          3185.551
Log-likelihood =  -1592.776
RIDGEMAX =          0
AIC =              3199.551
SBC =              3219.836

                    Estimate Std. Error  z value P(>|z|)
(Intercept)        -1.668452   0.101856 -16.3806  <2e-16 ***
SexMale             0.060923   0.083776   0.7272  0.4671
cc                  0.052429   0.090537   0.5791  0.5625
tv                  0.012725   0.092201   0.1380  0.8902
Random.(Intercept)  0.035945   0.035590   1.0100  0.3125
Threshold2          0.717160   0.046715  15.3517  <2e-16 ***
Threshold3          1.234106   0.054928  22.4676  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that while there are only three possible outcomes for `smkonset`, due to censoring in the data there are three thresholds (`Intercept`, `Threshold2`, and `Threshold3`) because the last

threshold is like an additional category beyond `smkonset=3`. Alternatively, we can perform a students in schools analysis.

```
R> School.mixord <- mixor(Surv(smkonset, event) ~ SexMale + cc + tv,
+    data = SmokeOnset, id = school, link = "cloglog", nAGQ = 20, IADD = 1)
R> summary(School.mixord)

Call:
mixor(formula = Surv(smkonset, event) ~ SexMale + cc + tv, data = SmokeOnset,
    id = school, nAGQ = 20, link = "cloglog", IADD = 1)

Deviance =          3187.388
Log-likelihood =  -1593.694
RIDGEMAX =          0.6
AIC =              3201.388
SBC =              3210.713

                     Estimate Std. Error  z value P(>|z|)
(Intercept)       -1.6560991  0.1073185 -15.4316  <2e-16 ***
SexMale            0.0572625  0.1236525   0.4631  0.6433
cc                 0.0446670  0.1041799   0.4287  0.6681
tv                 0.0213289  0.0937251   0.2276  0.8200
Random.(Intercept) 0.0026546  0.0165554   0.1603  0.8726
Threshold2         0.7133332  0.0486621  14.6589  <2e-16 ***
Threshold3         1.2250775  0.0523045  23.4220  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The last example demonstrates a students in classrooms analysis with varying sex effect across time intervals. Thus, this model does not assume proportional hazards for the `SexMale` variable. Recall that when assuming proportional hazards, there is no k class subscript on the parameter estimates but rather the estimated coefficient for a given variable reflects the explanatory relationship between that covariate and the cumulative link. Partial proportional odds models are less restrictive and are fit by including $K-1$ parameter estimates for those variables for which the proportional odds (PO) is not assumed (Peterson and Harrell 1990). This method is applicable for fitting partial proportional hazards models. In `mixor`, `KG=N` (where `N` is an integer) indicates not to assume proportional hazards (or odds for PO models) for the first `N` variables on the RHS of the equation. When using `KG`, the order of the variables on the RHS is important because the integer value passed to `KG` represents the number of variables, starting with the first, for non-proportional hazards (or odds for PO models) estimation. In this example we specify `KG=1` as we do not want to assume proportional hazards for `SexMale`.

```
R> students.mixord <- mixor( Surv(smkonset, event) ~ SexMale + cc + tv,
+    data = SmokeOnset, id = class, link = "cloglog", KG = 1, nAGQ = 20,
+    IADD = 1)
R> summary(students.mixord)
```

```
Call:
mixor(formula = Surv(smkonset, event) ~ SexMale + cc + tv, data = SmokeOnset,
    id = class, nAGQ = 20, link = "cloglog", KG = 1, IADD = 1)

Deviance =         3177.6
Log-likelihood =  -1588.8
RIDGEMAX =         0
AIC =             3195.6
SBC =             3221.68

                    Estimate Std. Error  z value   P(>|z|)
(Intercept)        -1.796033   0.116530 -15.4127 < 2.2e-16 ***
SexMale             0.308367   0.124633   2.4742  0.013353 *
cc                  0.054213   0.094478   0.5738  0.566090
tv                  0.012302   0.093452   0.1316  0.895273
Random.(Intercept)  0.035162   0.035615   0.9873  0.323513
Threshold2          0.835273   0.081198  10.2869 < 2.2e-16 ***
Threshold3          1.397184   0.089380  15.6319 < 2.2e-16 ***
Threshold2SexMale  -0.229161   0.105351  -2.1752  0.029615 *
Threshold3SexMale  -0.321411   0.124149  -2.5889  0.009628 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the above, the estimate for `SexMale` represents the gender effect on the first cutpoint (i.e., at post-intervention), and `Threshold2SexMale` and `Threshold3SexMale` represent how the effect is different at these additional cutpoints, relative to the first cutpoint. A likelihood ratio test of the proportional hazards assumption can be made by comparing the deviances of the two models which is significant ($p = 0.019$). Note that this test statistic and p-value can be obtained using

```
R> LRtest <- deviance(Surv.mixord) - deviance(students.mixord)
R> LRtest

[1] 7.951294

R> pchisq(LRtest, 2, lower.tail=FALSE)

[1] 0.01876716
```

Thus the assumption of proportional hazards is rejected. Again, if the estimates of the ordinal cutpoints and the `SexMale` effects at the latter two cutpoints are desired, the `Contrasts` function can be applied after structuring a suitable contrast matrix.

```
R> cm <- matrix(c(1, 1, 0, 0,
+    0, 0, 1, 1,
+    0, 0, 0, 0,
+    0, 0, 0, 0,
```

```
+    0, 0, 0, 0,
+    1, 0, 0, 0,
+    0, 1, 0, 0,
+    0, 0, 1, 0,
+    0, 0, 0, 1), byrow = TRUE, ncol = 4)
R> Contrasts(students.mixord, contrast.matrix = cm)

                  1 2 3 4
(Intercept)       1 1 0 0
SexMale           0 0 1 1
cc                0 0 0 0
tv                0 0 0 0
Random.(Intercept) 0 0 0 0
Threshold2        1 0 0 0
Threshold3        0 1 0 0
Threshold2SexMale  0 0 1 0
Threshold3SexMale  0 0 0 1


   Estimate Std. Error  z value   P(>|z|)
1 -0.960760   0.089917 -10.6849 < 2.2e-16 ***
2 -0.398849   0.087281  -4.5697 4.884e-06 ***
3  0.079206   0.097440   0.8129    0.4163
4 -0.013044   0.093043  -0.1402    0.8885
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results above show that the `SexMale` effects at the latter two cutpoints (i.e., at 1 and 2 year follow-ups) are close to zero and non-significant.

5.3. Frequency weighted data

The `mixor` function also accomodates weighted data where the data are stored as the number of level-2 observations observed (frequency weight) for a unique response pattern and covariate vector. In this example, the `norcag` data pertain to attitudes towards sex as measured in the 1989 General Social Survey (Agresti and Lang 1993). Each subject provided ordinal responses on three items concerning their opinion on early teens (age 14-16) having sex before marriage (Item 1), a man and a woman having sex before marriage (Item 2), and a married person having sex with someone other than their spouse (Item 3). However, the data are provided as frequencies (`freq`) by unique response pattern (`ID`) where the differences in item responses were stored as `Item2vs1` (attitude towards premarital vs teenage sex) and `Item3vs1` (attitude towards extramarital vs teenage sex). To fit a random intercepts model assuming proportional odds for differences in item responses we specify our model as before except we need to specify `weights=freq`.

```
R> data("norcag")
R> Fitted.norcag <- mixor(SexItems ~ Item2vs1 + Item3vs1,
```

```
+    data = norcag, id = ID, weights = freq, link = "logit", nAGQ = 20)
R> summary(Fitted.norcag)

Call:
mixor(formula = SexItems ~ Item2vs1 + Item3vs1, data = norcag,
    id = ID, weights = freq, nAGQ = 20, link = "logit")

Deviance =          2436.854
Log-likelihood =   -1218.427
RIDGEMAX =          0.1
AIC =              2448.854
SBC =              2473.834

                    Estimate Std. Error  z value   P(>|z|)
(Intercept)        -2.081101   0.202849 -10.2594 < 2.2e-16 ***
Item2vs1            3.807599   0.268844  14.1628 < 2.2e-16 ***
Item3vs1           -0.570918   0.197825  -2.8860  0.003902 **
Random.(Intercept)  5.139592   0.959090   5.3588 8.377e-08 ***
Threshold2          1.134229   0.098389  11.5280 < 2.2e-16 ***
Threshold3          2.783234   0.186590  14.9163 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To fit this model without the proportional odds assumption, the `KG` parameter is used. Here `KG=2` indicates not to assume proportional odds for the first 2 variables on the RHS of the equation. Again recall that when using `KG`, the order of the variables on the RHS is important because the integer value passed to `KG` represents the number of variables, starting with the first, for non-proportional odds estimation.

```
R> Fitted.norcag.np <- mixor(SexItems ~ Item2vs1 + Item3vs1,
+    data = norcag, id = ID, weights = freq, link = "logit", nAGQ = 10,
+    KG = 2)
R> summary(Fitted.norcag.np)

Call:
mixor(formula = SexItems ~ Item2vs1 + Item3vs1, data = norcag,
    id = ID, weights = freq, nAGQ = 10, link = "logit", KG = 2)

Deviance =          2401.609
Log-likelihood =   -1200.805
RIDGEMAX =          0.1
AIC =              2421.609
SBC =              2463.242

                    Estimate Std. Error z value   P(>|z|)
(Intercept)        -1.81905    0.19270 -9.4398 < 2.2e-16 ***
Item2vs1            3.15542    0.26733 11.8035 < 2.2e-16 ***
```

```
Item3vs1           -0.59966    0.19905 -3.0125  0.002591 **
Random.(Intercept)  4.40011    0.80989  5.4329 5.543e-08 ***
Threshold2          1.59227    0.18510  8.6020 < 2.2e-16 ***
Threshold3          3.21192    0.33989  9.4499 < 2.2e-16 ***
Threshold2Item2vs1  0.96062    0.20618  4.6591 3.176e-06 ***
Threshold3Item2vs1  1.06468    0.36491  2.9176  0.003527 **
Threshold2Item3vs1  0.26318    0.26612  0.9889  0.322690
Threshold3Item3vs1 -0.61730    0.65391 -0.9440  0.345168
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In the results above, the estimates for `Item2vs1` and `Item3vs1` are the effects on the first cutpoint or cumulative logit, and the item by threshold interactions indicate how the item effects vary across the latter two thresholds. As can be seen, these interactions are significant for Item 2, but not for Item 3. This indicates that the proportional odds assumption is violated for Item 2, but is reasonable for Item 3.

The `mixor` function also allows the user to specify covariates that influence the scale through the `KS` parameter. For the $K-1$ logits the model is

$$\log\left(\frac{P(Y_{ij}\leq k)}{1-P(Y_{ij}\leq k)}\right)=\frac{\alpha_k-(\mathbf{x}_{ij}^{\top}\boldsymbol{\beta}+\mathbf{z}_{ij}^{\top}\mathbf{T}\boldsymbol{\theta}_i)}{\exp(\boldsymbol{w}_{ij}^{\top}\boldsymbol{\tau})} \tag{5}$$

where $\boldsymbol{w}_{ij}$ is the design matrix for the covariates that influence scale and $\boldsymbol{\tau}$ are their effects (Hedeker, Berbaum, and Mermelstein 2006). The `KS` parameter is specified in the same way that `KG` is; the order of the variables on the RHS is important because the integer value passed to `KS` represents the number of variables, starting with the first, for scaling.

```
R> Fitted.norcag.scale <- mixor(SexItems ~ Item2vs1 + Item3vs1,
+    data = norcag, id = ID, weights = freq, link = "logit", nAGQ = 10,
+    KS = 2)
R> summary(Fitted.norcag.scale)

Call:
mixor(formula = SexItems ~ Item2vs1 + Item3vs1, data = norcag,
    id = ID, weights = freq, nAGQ = 10, link = "logit", KS = 2)

Deviance =          2418.607
Log-likelihood =   -1209.303
RIDGEMAX =          0.2
AIC =              2434.607
SBC =              2467.913

                   Estimate Std. Error z value   P(>|z|)
(Intercept)        -2.22116    0.33600 -6.6107 3.826e-11 ***
Item2vs1            4.50668    0.69147  6.5175 7.149e-11 ***
Item3vs1           -0.62054    0.36986 -1.6778  0.093394 .
Scale.Item2vs1      0.63125    0.19665  3.2100  0.001328 **
```

```
Scale.Item3vs1     -0.01540    0.31656 -0.0486  0.961200
Random.(Intercept)  6.78761    2.47036  2.7476  0.006003 **
Threshold2          1.39174    0.23492  5.9242 3.138e-09 ***
Threshold3          3.61363    0.57930  6.2379 4.435e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As the above results indicate, scaling for Item 2 is more pronounced than for Item 1. This suggests that responses to Item 2 are more dispersed across the 4 categories than for Item 1. Alternatively, the estimate of scaling for `Scale.Item3vs1` is near-zeo and non-significant, indicating that Items 1 and 3 are similarly dispersed across the 4 response categories.

5.4. Clustered data: varying ICC model

An example of naturally clustered data is twins clustered within twin pair. The outcome in the `concen` data reflects trouble concentrating (`TConcen`) which was recorded for both monozygotic and dizygotic twins (Ramakrishnan, Goldberg, Henderson, Eisen, True, Lyons, and Tsuang 1992). Each twin pair is uniquely identified by `ID` and type of twin is recorded using two dummy variables: `Mz`, an indicator variable representing MZ twins (`1` = MZ, `0` = DZ) and `Dz`, an indicator variable representing DZ twins (`1` = DZ, `0` = MZ). These data are also frequency weighted such that `freq` represents the frequency of the pattern. Prior to fitting the model, the data must be sorted by ID.

```
R> data("concen")
R> concen<-concen[order(concen$ID),]
```

A common ICC probit model can be fit using

```
R> Common.ICC <- mixor(TConcen ~ Mz, data = concen, id = ID,
+    weights = freq, link = "probit", nAGQ = 10)
R> summary(Common.ICC)

Call:
mixor(formula = TConcen ~ Mz, data = concen, id = ID, weights = freq,
    nAGQ = 10, link = "probit")

Deviance =          8549.288
Log-likelihood =   -4274.644
RIDGEMAX =          0.3
AIC =              8555.288
SBC =              8574.029

                   Estimate Std. Error z value   P(>|z|)
(Intercept)        0.741619   0.031957 23.2069 < 2.2e-16 ***
Mz                 0.062575   0.039482  1.5849     0.113
Random.(Intercept) 0.364533   0.051840  7.0319 2.037e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A varying ICC probit model can be fit using

```
R> Varying.ICC <- mixor(TConcen ~ Mz + Dz, data = concen, id = ID,
+    weights = freq, which.random.slope = 1:2, exclude.fixed.effect = 2,
+    link = "probit", nAGQ = 20, random.effect.mean = FALSE,
+    UNID = 1)
R> summary(Varying.ICC)

Call:
mixor(formula = TConcen ~ Mz + Dz, data = concen, id = ID, which.random.slope = 1:2,
    weights = freq, exclude.fixed.effect = 2, nAGQ = 20, link = "probit",
    random.effect.mean = FALSE, UNID = 1)

Deviance =          8543.072
Log-likelihood =  -4271.536
RIDGEMAX =          0
AIC =              8551.072
SBC =              8576.06

            Estimate Std. Error z value   P(>|z|)
(Intercept) 0.707431   0.032338 21.8762 < 2.2e-16 ***
Mz          0.131456   0.047876  2.7458 0.0060370 **
Random.Mz   0.491750   0.081397  6.0414 1.528e-09 ***
Random.Dz   0.234640   0.064691  3.6271 0.0002866 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that `UNID` is an indicator variable where `UNID=0` (default) reflects that the random effects are multi-dimensional and `UNID=1` reflects that the random effects are variables related to a uni-dimensional random effect (e.g., item indicators of a latent variable). Here, a likelihood-ratio test of the common ICC assumption can be made by comparing the deviances of the two models: $\chi^2_1 = 8549.3 - 8543.1 = 6.2$ which is significant ($p = .013$). Note that this test statistic and p-value can be obtained using

```
R> LRtest <- deviance(Common.ICC) - deviance(Varying.ICC)
R> LRtest

[1] 6.215814

R> pchisq(LRtest, 1, lower.tail=FALSE)

[1] 0.01266141
```

Thus, the assumption of common ICC is rejected. Using the estimates from the varying ICC model, we obtain ICCs of $0.4917/(1+0.4917) = 0.33$ for MZ twins, and $0.2346/(1+0.2346) = 0.19$ for DZ twins. Since the probit link was selected, these ICCs are equivalent to tetrachoric correlation estimates.

Summary

Herein we have described the **mixor** package which works in conjunction with the **MIXOR** Fortran stand-alone program in the R programming environment. The package provides a function for fitting cumulative link mixed-effects ordinal response models using either a probit, logit, or complementary log-log link function.

Acknowledgments

Research reported in this publication was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R01LM011169. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Agresti A (2010). *Analysis of Ordinal Categorical Data*. John Wiley & Sons, Hoboken, NJ.

Agresti A, Lang JB (1993). "A Proportional Odds Model with Subject-Specific Effects for Repeated Ordered Categorical Responses." *Biometrika*, **80**, 527–534.

Christensen RHB (2013). "**ordinal**: Regression Models for Ordinal Data." R package version 2013.9-30, URL https://CRAN.R-project.org/package=ordinal.

Cromer A, Carles A, Millon R, Ganguli G, Chalmel F, Lemaire F, Young J, Dembélé D, Thibault C, Muller D, Poch O, Abesassis J, Wasylyk B (2004). "Identification of Genes Associated with Tumorigenesis and Metastatic Potential of Hypopharyngeal Cancer by Microarray Analysis." *Oncogene*, **23**, 2484–2498.

Flay BR, Miller TQ, Hedeker D, Siddiqui O, Brannon BR, Johnson CA, Hansen WB, Sussman S, Dent C (1995). "The Television, School and Family Smoking Prevention and Cessation Project: VIII. Student Outcomes and Mediating Variables." *Preventive Medicine*, **24**, 29–40.

Gibbons RD, Hedeker D (1994). "Application of Random-Effects Probit Regression Models." *Journal of Consulting and Clinical Psychology*, **62**, 285–296.

Hedeker D, Berbaum M, Mermelstein RJ (2006). "Location-Scale Models for Multilevel Ordinal Data: Between- and Within-Subjects Variance Modeling." *Journal of Probability and Statistical Science*, **4**, 1–20.

Hedeker D, Gibbons RD (1996). "**MIXOR**: a Computer Program for Mixed-Effects Ordinal Regression Analysis." *Computer Methods and Programs in Biomedicine*, **49**, 157–176.

Hedeker D, Gibbons RD (2006). *Longitudinal Data Analysis*. Wiley-Interscience, Hoboken, NJ.

Ivshina AV, George J, Senko O, Mow B, Putti TC, Smeds J, Lindahl T, Pawitan Y, Hall P, Nordgren H, Wong JEL, Liu ET, Bergh J, Kuznestsov VA, Miller L (2006). "Genetic Reclassification of Histologic Grade Delineates New Clinical Subtypes of Breast Cancer." *Cancer Research*, **66**, 10292–10301.

McCullagh P (1980). "Regression Models for Ordinal Data." *Journal of the Royal Statistical Society B*, **42**, 109–142.

Peterson B, Harrell FE (1990). "Partial Proportional Odds Models for Ordinal Response Variables." *Applied Statistics*, **39**, 205–217.

Ramakrishnan V, Goldberg J, Henderson WG, Eisen SA, True W, Lyons MJ, Tsuang MT (1992). "Elementary Methods for the Analysis of Dichotomous Outcomes in Unselected Samples of Twins." *Genetic Epidemiology*, **9**, 273–287.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://CRAN.R-project.org.

Therneau T (2014). "A Package for Survival Analysis in S." R package version 2.37-7, URL https://CRAN.R-project.org/package=survival.

Affiliation:

Kellie J. Archer
Division of Biostatistics
The Ohio State University
1841 Neil Ave
Columbus, OH 43210
E-mail: archer.43@use.edu
URL: https://cph.osu.edu/people/karcher

Donald Hedeker
Department of Public Health Sciences
The University of Chicago Biological Sciences
5841 S. Maryland Avenue
Room W254, MC2000
Chicago, IL 60637
E-mail: hedeker@uchicago.edu
URL: https://hedeker-sites.uchicago.edu

Rachel Nordgren
Division of Epidemiology and Biostatistics
School of Public Health
University of Illinois at Chicago
1603 West Taylor Street
Chicago, IL 60612-4394
E-mail: rknordgren@gmail.com

Robert D. Gibbons
Professor of Medicine and Health Studies
Director, Center for Health Statistics
University of Chicago
5841 S. Maryland Avenue
MC 2007 office W260
Chicago IL 60637
E-mail: rdg@uchicago.edu