mvord: An R Package for Fitting Multivariate Ordinal Regression Models

Rainer Hirk
WU Wirtschaftsuniversität Wien

Kurt Hornik
WU Wirtschaftsuniversität Wien

Laura Vana
WU Wirtschaftsuniversität Wien

Abstract

The R package **mvord** implements composite likelihood estimation in the class of multivariate ordinal regression models with probit and logit link. A flexible modeling framework for ordinal repeated measurements on the same subject is set up, which takes into consideration the dependence among the multiple observations by employing different error structures. Heterogeneity in the error structure across the subjects can be accounted for by the package, which allows for covariate dependent error structures. In addition, regression coefficients and threshold parameters are varying across the multiple response dimensions in the default implementation. However, constraints can be defined by the user if a reduction of the parameter space is desired.

Keywords: Composite likelihood, Multivariate ordered logit, Multivariate ordered probit, R package.

1. Model Class

Multivariate ordinal regression models are based on *cumulative link models* (Tutz 2012) which are amongst the most popular models for univariate ordinal data analysis. In cumulative link models the observed ordinal outcome Y is assumed to be a coarser (categorized) version of a latent continuous variable $\widetilde{Y}$. If multiple observations on the same subject are observed, univariate cumulative link models can be extended to a multivariate framework. These repeated measurements for each subject may take place either at the same time yielding a cross-sectional multivariate ordinal regression model or at different points in time yielding a longitudinal multivariate ordinal regression model.

1.1. Model formulation

Let Y_{ij} denote the ordinal observation and $\boldsymbol{x}_{ij}$ be a p-dimensional vector of covariates for subject i and outcome j, where $i = 1, \dots, n$ and $j \in J_i$, for J_i a subset of all available outcomes J in the data set. Moreover, we denote by $q = |J|$ and $q_i = |J_i|$ the number of elements in the set J and J_i, respectively. Following the cumulative link modeling approach (Agresti 2002), the ordinal response Y_{ij} is assumed to be a coarser (categorized) version of a latent continuous variable $\widetilde{Y}_{ij}$. The observable categorical outcome Y_{ij} and the unobservable latent variable $\widetilde{Y}_{ij}$ are connected by:

$$Y_{ij} = r_{ij} \quad \Leftrightarrow \quad \theta_{j,r_{ij}-1} < \widetilde{Y}_{ij} \leq \theta_{j,r_{ij}}, \qquad r_{ij} \in \{1, \dots, K_j\}$$

where r_{ij} is a category out of K_j ordered categories and $\boldsymbol{\theta}_j$ is a vector of suitable threshold parameters for outcome j with the following restriction: $-\infty < \theta_{j,1} < \cdots < \theta_{j,K_j-1} < \infty$. Note that in this setting binary observations can be treated as ordinal observations with two categories ($K_j = 2$).

For the relationship between the latent variable $\widetilde{Y}_{ij}$ and the vector of covariates $\boldsymbol{x}_{ij}$ we assume the following linear model:

$$\widetilde{Y}_{ij} = \beta_{j0} + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}_j + \epsilon_{ij}, \tag{1}$$

where β_{j0} is an intercept term, $\boldsymbol{\beta}_j = (\beta_{j1}, \dots, \beta_{jp})^\top$ is a vector of regression coefficients, both corresponding to outcome j, and ϵ_{ij} is a mean zero error term. The number of ordered categories K_j as well as the threshold parameters $\boldsymbol{\theta}_j$ and the regression coefficients $\boldsymbol{\beta}_j$ are allowed to vary across outcome dimensions $j \in J$ to account for possible heterogeneity across the response variables. We further assume the n subjects to be independent and that the error terms are uncorrelated with the covariates.

The dependence among the different responses is accounted for by assuming the vector of error terms for each subject $\boldsymbol{\epsilon}_i = [\epsilon_{ij}]_{j \in J_i}$ to follow a multivariate distribution. The multivariate distribution functions we consider are the multivariate normal distribution $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i)$, which corresponds to the probit link, and the multivariate logistic distribution $\boldsymbol{\epsilon}_i \sim \mathcal{L}(\mathbf{0}, \boldsymbol{\Sigma}_i)$ yielding the logit link, where the covariance matrix $\boldsymbol{\Sigma}_i$ captures the correlation between the vector of responses for subject i. For the logit link we approximate the multivariate logistic distribution by a multivariate t-distribution with fixed degrees of freedom (following the approach of O'Brien and Dunson 2004). More details can be found in Hirk, Hornik, and Vana (2017).

1.2. Identifiability Issues

As the absolute scale and the absolute location are not identifiable in ordinal models further restrictions on the parameter set need to be imposed. Assuming $\boldsymbol{\Sigma}_i$ is a covariance matrix with diagonal elements $[\sigma_{ij}^2]_{j \in J_i}$, only the quantities $\boldsymbol{\beta}_j / \sigma_{ij}$ and $(\theta_{j,r_{ij}} - \beta_{j0}) / \sigma_{ij}$ are identifiable in the model in Equation 1. The scale can be fixed either by restricting the full variance-covariance matrix $\boldsymbol{\Sigma}_i$ to be a correlation matrix $\boldsymbol{R}_i$, by fixing two threshold parameters, or the intercept and a threshold parameter. In order to fix the location either the intercept β_{j0} or one threshold parameter has to be set to some value. Hence, in order to obtain an identifiable model the parameter set is typically constrained in one of the following ways:

- Fixing the intercept β_{j0} (e.g., to zero), using flexible thresholds $\boldsymbol{\theta}_j$ and fixing σ_{ij} (e.g., to unity) $\forall j \in J_i$, $\forall i \in \{1, \dots, n\}$;
- Leaving the intercept β_{j0} unrestricted, fixing one threshold parameter (e.g., $\theta_{j,1} = 0$) and fixing σ_{ij} (e.g., to unity) $\forall j \in J_i$, $\forall i \in \{1, \dots, n\}$;
- Fixing the intercept β_{j0} (e.g., to zero), fixing one threshold parameter (e.g., $\theta_{j,1} = 0$) and leaving σ_{ij} unrestricted $\forall j \in J_i$, $\forall i \in \{1, \dots, n\}$;

- Leaving the intercept β_{j0} unrestricted, fixing two threshold parameters (e.g., $\theta_{j,1} = 0$ and $\theta_{j,2} = 1$) and leaving σ_{ij} unrestricted $\forall j \in J_i$, $\forall i \in \{1, \dots, n\}$ (note that this parameterization cannot be applied to the binary case).

Note that the first two options are the most commonly used in the literature. All of these alternative parameterizations of the models are supported by the **mvord** package, allowing the user to choose the most convienient one for each specific application. Table 2.5.2 gives an overview on all identifiable model parameterizations.

1.3. Error Structures

We mainly distinguish between two different model types with different parameterizations, one with standardized error variances (*correlation error structure*) and one with unrestricted error variances (*covariance error structure*). For both model types we allow for a factor dependent error structure and in case of a correlation error structure we additionally allow for a covariate dependent equicorrelation and $AR(1)$ error structures. For the sake of notation we assume in the following that the number of repeated measurements is equal for all subjects and denoted by q. In the case of $q_i \neq q$, the matrices presented below will be subsetted by picking the rows and columns corresponding to $j \in J_i$.

Correlation error structure

- **General correlation structure**
  In this parameterization we fix the scale by restricting the full variance-covariance to be a correlation matrix and obtain the following error distribution:

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \dots, \epsilon_{iq})^\top \sim F_q \left(\mathbf{0}, \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1q} \\ \rho_{12} & 1 & \cdots & \rho_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1q} & \rho_{2q} & \cdots & 1 \end{pmatrix} \right). \qquad (2)$$

  As absolute location is not identifiable in this model one of the following constraints need to be imposed for all $j \in J$:

  - the intercept β_{j0} is fixed to some constant c (e.g., the default value is zero), or
  - the first threshold $\theta_{j,1}$ is fixed to some value.

- **Factor dependent correlation structure**
  In order to account for heterogeneity in the error terms, a first model extension allows for factor-varying correlation structures. To be more precise, we allow for different correlation matrices in the errors for each subject i, depending on some factor $f(i)$ which is assumed to be constant across repeated measurements j. The factor dependent error structure for a correlation structure has the following form:

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \dots, \epsilon_{iq})^\top \sim F_q(\mathbf{0}, \boldsymbol{R}_{f(i)}).$$

- **Covariate dependent equicorrelation structure**
  We improve the complexity of the model by allowing a covariate dependent equicorrelation structure. In this setting, we assume that correlations are equal across all pairs, but

differ across subjects i. The correlation parameter ρ_i of each subject i is assumed to depend on a vector of covariates $\boldsymbol{s}_i$. Fisher's z-transformation allows us to reparameterize the linear term $\alpha_0 + \boldsymbol{s}_i^\top \boldsymbol{\alpha}$ in terms of a correlation parameter for each subject:

$$\frac{1}{2} \log \left(\frac{1 + \rho_i}{1 - \rho_i} \right) = \alpha_0 + \boldsymbol{s}_i^\top \boldsymbol{\alpha}.$$

Solving for ρ_i gives us the following re-transformation:

$$\rho_i = \frac{e^{2(\alpha_0 + \boldsymbol{s}_i \boldsymbol{\alpha})} - 1}{e^{2(\alpha_0 + \boldsymbol{s}_i \boldsymbol{\alpha})} + 1}.$$

As a consequence, this transformation allows for subject-varying correlations which depend on subject-specific covariates that have to be constant across repeated measurements j. We obtain an equicorrelation structure that is able to account for heterogeneity in the errors:

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \dots, \epsilon_{iq})^\top \sim F_q \left(\mathbf{0}, \begin{pmatrix} 1 & \rho_i & \cdots & \rho_i \\ \rho_i & 1 & \cdots & \rho_i \\ \vdots & \vdots & \ddots & \vdots \\ \rho_i & \rho_i & \cdots & 1 \end{pmatrix} \right).$$

- $AR(1)$ **correlation structure**
  For given consecutive equi-spaced time points $t_1, \dots, t_T$ we assume an autoregressive error structure of order one with $corr(\epsilon_{it_k}, \epsilon_{it_l}) = \rho^{|t_l - t_k|}$ for each subject i. In this case the correlation structure has the following form:

$$\boldsymbol{\epsilon}_i = (\epsilon_{it_1}, \epsilon_{it_2}, \dots, \epsilon_{it_T})^\top \sim F_T \left(\mathbf{0}, \begin{pmatrix} 1 & \rho^{|t_2 - t_1|} & \cdots & \rho^{|t_T - t_1|} \\ \rho^{|t_2 - t_1|} & 1 & \cdots & \rho^{|t_T - t_2|} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{|t_T - t_1|} & \rho^{|t_T - t_2|} & \cdots & 1 \end{pmatrix} \right).$$

  This $AR(1)$ correlation structure can be extended to a covariate dependent setting in analogy to the equicorrelation structure.

Covariance error structure

- **General covariance structure**
  In a further parameterization we leave the variance-covariance matrix unrestricted and obtain the following error distribution:

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \dots, \epsilon_{iq})^\top \sim F_q \left(\mathbf{0}, \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1q}\sigma_1\sigma_q \\ \rho_1\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2q}\sigma_2\sigma_q \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1q}\sigma_1\sigma_q & \rho_{2q}\sigma_2\sigma_q & \cdots & \sigma_q^2 \end{pmatrix} \right). \tag{3}$$

  In this model we again need further restrictions on the parameter set in order to obtain an identifiable scale and location. We either fix

- the first two thresholds $\theta_{j,1}$ and $\theta_{j,2}$ to some value (e.g., in the default case we set $\theta_{j,1} = 0$ and $\theta_{j,2} = 1$),
- the first threshold $\theta_{j,1}$ and the last threshold θ_{j,K_j-1} to some value, or
- the intercept β_{j0} and the first threshold $\theta_{j,1}$ to some value

for all repeated measurements $j \in J$.

- **Factor dependent covariance structure**
  In order to account for some heterogeneity in the error terms, we allow for different covariance matrices in the errors for each subject i, depending on some factor $f(i)$. In this case the factor dependent covariance structure has the following form:
$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \dots, \epsilon_{iq})^\top \sim F_q(\mathbf{0}, \boldsymbol{\Sigma}_{f(i)}).$$

1.4. Composite Likelihood Estimation

In order to estimate the model parameters we use a composite likelihood approach, where the full likelihood is approximated by a pseudo-likelihood which is constructed from lower dimensional marginal distributions, more specifically by "aggregating" the likelihoods corresponding to pairs of observations (Varin, Reid, and Firth 2011).

For a given parameter vector $\boldsymbol{\Gamma}$, which contains the threshold parameters, the regression coefficients and the correlation (and variance) parameters, the likelihood is given by:

$$\mathscr{L}(\boldsymbol{\Gamma}|[X_i]_{i=1:n}, Y) = \prod_{i=1}^{n} \mathsf{P}(\cap_{j\in J_i} Y_{ij} = r_{ij}|\boldsymbol{\Gamma}, X_i)^{w_i} = \prod_{i=1}^{n} \left(\int_{D_i} f_{q_i}(\widetilde{\mathbf{Y}}_i|\boldsymbol{\Gamma}, X_i) d^{q_i}\widetilde{\mathbf{Y}}_i\right)^{w_i},$$

where X_i is a $q_i \times p$ matrix of covariates, $D_i = \prod_{j\in J_i}(\theta_{j,r_{ij}-1}, \theta_{j,r_{ij}}]$ is a Cartesian product, w_i are are subject specific non-negative weights, which are set to one in the default case, and f_{q_i} is the q_i-dimensional density of the error terms $\boldsymbol{\epsilon}_i$.

We approximate the full likelihood by a pseudolikelihood which is constructed from bivariate marginal distributions. If the number of observed outcomes for subject i is less than two ($q_i < 2$), then the univariate marginal distribution enters the likelihood. For the sake of notation we introduce an $n \times q$ binary index matrix Z, where each element z_{ij} takes a value of 1 if $j \in J_i$ and 0 otherwise. The pairwise log-likelihood function is obtained by:

$$p\ell(\boldsymbol{\Gamma}|Y) = \sum_{i=1}^{n} w_i \Bigg[\sum_{k=1}^{q-1}\sum_{l=k+1}^{q} \mathbb{1}_{\{z_{ik}z_{il}=1\}} \log\left(\mathsf{P}(Y_{ik} = r_{ik}, Y_{il} = r_{il}|\boldsymbol{\Gamma})\right) + \mathbb{1}_{\{q_i=1\}} \sum_{k=1}^{q} \mathbb{1}_{\{z_{ik}=1\}} \log\left(\mathsf{P}(Y_{ik} = r_{ik}, |\boldsymbol{\Gamma})\right)\Bigg]. \quad (4)$$

Denoting by $U_{ij} = (\theta_{j,r_{ij}} - \beta_{j0} - \boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_j)/\sigma_{ij}$ the upper and by $L_{ij} = (\theta_{j,r_{ij}-1} - \beta_{j0} - \boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_j)/\sigma_{ij}$ the lower integration bounds, the uni- and bivariate probabilities are given by:

$$\mathsf{P}(Y_{ik} = r_{ik}, Y_{il} = r_{il}|\cdot) = \int_{L_{ik}}^{U_{ik}}\int_{L_{il}}^{U_{il}} f_2(v_{ik}, v_{il}|\cdot) dv_{ik} dv_{il},$$
$$\mathsf{P}(Y_{ik} = r_{ik}|\cdot) = \int_{L_{ik}}^{U_{ik}} f_1(v_{ik}) dv_{ik}.$$

The maximum pairwise likelihood estimates $\hat{\mathbf{\Gamma}}_{\mathrm{PL}}$ are obtained by direct maximization of the composite likelihood given in Equation 4. The parameters to be estimated are reparametrized (where needed) such that unconstrained optimization can be performed. First, we reparametrize the threshold parameters in order to achieve monotonicity. Second, for all unrestricted correlation (and covariance) matrices we use the spherical parameterization of Pinheiro and Bates (1996). This parameterization has the advantage that it can be easily applied to correlation matrices. Third, if we assume to have equicorrelated or $AR(1)$ errors, we use the hyperbolic tangent transformation.

Computation of the standard errors is needed in order to quantify the uncertainty of the maximum pairwise likelihood estimates. Under certain regularity conditions, the maximum pairwise likelihood estimates are consistent as the number of responses is fixed and $n \to \infty$. In addition, the maximum pairwise likelihood estimator is asymptotically normal with asymptotic mean $\mathbf{\Gamma}$ and a covariance matrix which equals the inverse of the Godambe information matrix:

$$G(\mathbf{\Gamma})^{-1} = H^{-1}(\mathbf{\Gamma})V(\mathbf{\Gamma})H^{-1}(\mathbf{\Gamma}),$$

where $G(\mathbf{\Gamma})$ denotes the Godambe information matrix, $H(\mathbf{\Gamma})$ the Hessian (sensitivity matrix) and $V(\mathbf{\Gamma})$ the variability matrix. The Hessian $H(\mathbf{\Gamma})$ and variability matrix $V(\mathbf{\Gamma})$ can be estimated as follows:

$$\hat{V}(\mathbf{\Gamma}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial p\ell_i(\hat{\mathbf{\Gamma}}_{\mathrm{PL}}|\boldsymbol{Y}_i)}{\partial \mathbf{\Gamma}} \left(\frac{\partial p\ell_i(\hat{\mathbf{\Gamma}}_{\mathrm{PL}}|\boldsymbol{Y}_i)}{\partial \mathbf{\Gamma}}\right)^{\top}, \quad \hat{H}(\mathbf{\Gamma}) = -\frac{1}{n}\sum_{i=1}^{n} \frac{\partial^2 p\ell_i(\hat{\mathbf{\Gamma}}_{\mathrm{PL}}|\boldsymbol{Y}_i)}{\partial \mathbf{\Gamma}\partial \mathbf{\Gamma}^{\top}},$$

where $p\ell_i(\mathbf{\Gamma}|\boldsymbol{Y}_i)$ is the component of the pairwise log-likelihood corresponding to subject i. It is possible to avoid the computation of the second-order derivatives, as the Hessian can be computed as:

$$\hat{H}(\mathbf{\Gamma}) = \frac{1}{n}\sum_{i=1}^{n} \sum_{k<l,k,l\in J_i} \left(\frac{\partial p\ell_i(\hat{\mathbf{\Gamma}}_{\mathrm{PL}}|Y_{ik}, Y_{il})}{\partial \mathbf{\Gamma}}\right) \left(\frac{\partial p\ell_i(\hat{\mathbf{\Gamma}}_{\mathrm{PL}}|Y_{ik}, Y_{il})}{\partial \mathbf{\Gamma}}\right)^{\top}.$$

In order to compare different models, the composite likelihood information criterion can be used: $\mathrm{CLIC}(\mathbf{\Gamma}) = -2\, p\ell(\hat{\mathbf{\Gamma}}_{\mathrm{PL}}|X, Y) + k\ \mathrm{tr}(\hat{V}(\mathbf{\Gamma})\hat{H}(\mathbf{\Gamma})^{-1})$ (where $k = 2$ corresponds to CLAIC and $k = \log(n)$ corresponds to CLBIC). A comprehensive overview and further details on the properties of the maximum composite likelihood estimates is provided in Varin (2008).

2. Implementation

Multivariate ordinal regression models in the R package **mvord** are fitted using the function `multord()`

```
R> multord(formula,
+          error.structure = corGeneral(~1),
+          link = c("probit", "logit"),
+          data,
+          index = NULL,
```

```
+           response.names = NULL,
+           response.levels = NULL,
+           coef.constraints = NULL,
+           coef.values = NULL,
+           threshold.constraints = NULL,
+           threshold.values = NULL,
+           weights = NULL,
+           se = TRUE,
+           start.values = NULL,
+           solver = "BFGS",
+           control = list(maxit=200000, trace = 1, kkt = FALSE)
+  )
```

Two link functions and different error structures are implemented in `multord()`. By default, threshold parameters and regression coefficients are allowed to be outcome specific. However, this can be restricted by the user, who can specify constraints on the threshold parameters and/or on the regression coefficients.

All features are illustrated by means of a simulated data set which corresponds to an application in credit risk modeling.

```
R> head(data_cr_multord, n = 3)
```

```
  firm_id rater_id rating      ICR        LR     LEV1       LEV2
1       1       R1      D 1.546318 0.2484137 3.782934 0.92053787
2       2       R1      B 8.723779 0.1506502 1.033042 0.05305052
3       3       R1      D 4.726520 0.5187664 8.942818 0.97001785
         PR     lRSIZE     lSYSR  BSEC
1 0.2743184 -11.202807 -3.691023 BSEC3
2 0.1182763  -8.815116 -4.270618 BSEC3
3 0.2871493  -9.548691 -3.895642 BSEC6
```

```
R> str(data_cr_multord, vec.len = 3)
```

```
'data.frame':        4566 obs. of  11 variables:
 $ firm_id : Factor w/ 1665 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 ...
 $ rater_id: Factor w/ 4 levels "R1","R2","R3",..: 1 1 1 1 1 1 1 1 ...
 $ rating  : chr  "D" "B" "D" ...
 $ ICR     : num  1.55 8.72 4.73 4.08 ...
 $ LR      : num  0.248 0.151 0.519 0.168 ...
 $ LEV1    : num  3.78 1.03 8.94 2.19 ...
 $ LEV2    : num  0.9205 0.0531 0.97 2.8743 ...
 $ PR      : num  0.2743 0.1183 0.2871 0.0821 ...
 $ lRSIZE  : num  -11.2 -8.82 -9.55 -8.66 ...
 $ lSYSR   : num  -3.69 -4.27 -3.9 -5.13 ...
 $ BSEC    : Factor w/ 8 levels "BSEC1","BSEC2",..: 3 3 6 4 3 1 6 4 ...
```

2.1. Data structure

We use the long format for the input of `data`, where each row contains a subject index i (`firm_id`), a repeated measurement index j (`rater_id`), an ordinal response (`rating`) and all the covariates (`ICR`, `LR`, `LEV1`, `LEV2`, `PR`, `lRSIZE` and `lSYSR`). This long format data stucture is internally transformed to a matrix of responses Y (which contains `NA` in the case of missing entries) and a list of covariate matrices X_j for all $j \in J$[1]. In order to construct these objects, subject index i and the repeated measurement index j should be specified. This can be performed by an optional argument `index`, a character vector of length two, specifying the column names of the subject index and the repeated measurement index in `data`. In the credit risk example we set:

```
R> index <- c("firm_id", "rater_id")
R> index

[1] "firm_id"  "rater_id"
```

The default value of `index` is `NULL` assuming that the first column of `data` contains the subject index i and the second column the repeated measurement index j. If specific constraints are imposed on the threshold parameters and/or on the regression coefficients, it is important to know which level of the repeated measurement index j corresponds to the first dimension, second dimension and so on. Hence, a well defined index $j \in J$ for the repeated measurements is needed. Therefore, a vector `response.names` is used to define the index number of the repeated measurements:

```
R> response.names <- c("R1", "R2", "R3", "R4")
R> response.names

[1] "R1" "R2" "R3" "R4"
```

The default value of `response.names` is `NULL` giving the natural ordering of the levels of the factor variable for all the repeated measurements. The ordering of `response.names` always specifies the index of the repeated measurement unit $j \in J$. This ordering is essential when putting constraints on the parameters and when setting `response.levels`:

```
R> response.levels <-  list(rev(LETTERS[1:6]),
+                            rev(LETTERS[1:6]),
+                            rev(LETTERS[7:13]),
+                            rev(LETTERS[14:15]))
R> names(response.levels) <- response.names
R> response.levels

$R1
[1] "F" "E" "D" "C" "B" "A"

$R2
```

[1] In order to avoid numerical instabilities we suggest to standardize the covariates $\boldsymbol{x}_{ij}$.

```
[1] "F" "E" "D" "C" "B" "A"

$R3
[1] "M" "L" "K" "J" "I" "H" "G"

$R4
[1] "O" "N"
```

If the categories differ across repeated measurements (either the number of categories and/or the category labels) one needs to specify the `response.levels` explicitly. This is performed by a list of length J (number of repeated measurements), where each element contains the names of the levels of the ordered categories in ascending (or if desired descending) order.

2.2. Formula

The ordinal responses Y (`rating`) and the covariates are passed by a `formula` object. Intercepts can be included or excluded in the model depending on the model parameterization chosen in order to ensure identifiability:

Model without intercept If the intercept should be removed, the `formula` of a given response (`rating`) and covariates (`ICR`, `LR`, `LEV1`, `LEV2`, `PR`, `lRSIZE` and `lSYSR`) has the following form:

```
R> formula <- rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
```

Model with intercept If one wants to include an intercept in the model, there are two equivalent possibilities to set the model `formula`. Either the intercept is included explicitly by:

```
R> formula <- rating ~ 1 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
```

or by

```
R> formula <- rating ~ ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
```

2.3. Link function

We allow for two different link functions, the probit link (`link = "probit"`) and the logit link (`link = "logit"`). For the probit link a multivariate normal distribution for the errors is applied, while for the logit link an approximate multivariate logistic distribution is used. The normal bivariate probabilities which enter the pairwise log-likelihood are computed with the R package **pbivnorm** (Genz and Kenkel 2015). The bivariate t probabilities are computed using Fortran code from Alan Genz (Genz and Bretz 2009).

2.4. Error structures

We allow for several different error structures depending on the model parameterization:

- **Correlation**

  - `corGeneral`
    The most common parameterization is the general correlation matrix given in Equation 2. This error structure is applied by:

    ```
    R> error.structure <- corGeneral(~ 1)
    ```

    This paramterization can be extended by allowing a factor dependent correlation structure, where the correlation of each subject i depends on a given factor `f`.

    ```
    R> error.structure = corGeneral(~ f)
    ```

    The factor `f` is not allowed to vary across repeated measurements j for the same subject i and due to numerical constraints only up to maximum 30 levels are allowed.

  - `corEqui`
    A covariate dependent equicorrelation structure, where the correlations are equal across all q dimensions and depend on some covariates `S1, ..., Sm`, is used by:

    ```
    R> error.structure <- corEqui(~ S1 + ... + Sm)
    ```

    It has to be noted that these covariates `S1, ..., Sm` as well as the factor `f` are not allowed to vary across repeated measurements j for the same subject i.

  - `corAR1`
    An autoregressive error structure of order one $AR(1)$ is obtained by:

    ```
    R> error.structure = corAR1(~ 1)
    ```

    In order to account for some heterogeneity the $AR(1)$ error structure is allowed to depend on covariates `S1, ..., Sm` that are constant over time for each subject i

    ```
    R> error.structure = corAR1(~ S1 + ... + Sm)
    ```

- **Covariance**

  - `covGeneral`
    In case of a full variance-covariance parameterization given in Equation 3 the standard parameterization with a full variance-covariance is obtained by:

    ```
    R> error.structure = covGeneral(~ 1)
    ```

    This parameterization can be extended to the factor dependent covariance structure, where the covariance of each subject depends on a given factor `f`:

    ```
    R> error.structure = covGeneral(~ f)
    ```

2.5. Constraints on threshold coefficients

The package supports constraints on the threshold parameters. Firstly, the user can specify whether the threshold parameters should be equal across some or all response dimensions. Secondly, the values of some of the threshold parameters can be fixed. This feature is important for the users who wish to further restrict the parameter space of the thresholds or who wish to specify values for the threshold parameters other than the default values used in the package. Note that fixing some of the thresholds is needed for some of the parameterizations presented in Table 2.5.2 in order to ensure identifiability of the model.

`error.structure`	Cov. structure ($\boldsymbol{\Sigma}$)	Corr. structure ($\boldsymbol{R}$)	Factor dependent	Covariate dependent
`corGeneral(~1)`		✓		
`corGeneral(~f)`		✓	✓	
`covGeneral(~1)`	✓			
`covGeneral(~f)`	✓		✓	
`corEqui(~1)`		✓		
`corEqui(~S)`		✓		✓
`corAR1(~1)`		✓		
`corAR1(~S)`		✓		✓

Table 1: This table gives an overview on the error structures in **mvord**.

Threshold constraints across responses

Such constraints can be imposed by a vector of positive integers `threshold.constraints`, where dimensions with equal threshold parameters obtain the same integer. When restricting two outcome dimensions to be the same, one has to be careful that the number of categories in the two outcome dimensions must be the same. In our example with $q = 4$ different outcomes, if one wishes to restrict the threshold parameters of `R1` and `R2` to be equal, i.e.:

- $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$;
- $\boldsymbol{\theta}_3$, $\boldsymbol{\theta}_4$ arbitrary.

These constraints on the threshold parameters are specified by:

```
R> threshold.constraints <- c(1, 1, 2, 3)
R> names(threshold.constraints) <- response.names
R> threshold.constraints

R1 R2 R3 R4
 1  1  2  3
```

Fixing threshold values

Values for the threshold parameters can be specified by the argument `threshold.values`. For this purpose the user can pass a `list` with q elements, where each element is a `vector` of length $K_j - 1$ (where K_j is number of ordered categories for ordinal outcome j). A numeric value in this vector fixes the corresponding threshold parameter to the specified value while `NA` leaves the parameter flexible and indicates it should be estimated.

After specifying the error structure (through the `error.structure` argument) and whether an intercept should be estimated or not (in the `formula` argument), the user can choose among five possible options for fixing the thresholds:

- leaving all thresholds flexible;
- fixing, for all $j \in J$, the first threshold $\theta_{j,1}$ to a constant a_j;

- fixing, for all outcomes with $K_j > 2$, the first and second thresholds $\theta_{j,1} = a_j$, $\theta_{j,2} = b_j$;
- fixing, for all outcomes with $K_j > 2$, the first and last thresholds $\theta_{j,1} = a_j$, $\theta_{j,K_j-1} = b_j$;
- an extra option is fixing all of the threshold parameters, for all $j \in J$.

Note that the option chosen needs to be consistent across the different outcomes (e.g., it is not allowed to fix first and last threshold for one outcome and first and second threshold for a different). Table 2.5.2 provides information about the options available for each combination error structure and intercept, as well as about the default values in case the user does not specify any threshold values.

Error Structure	Intercept	Thresholds				
		all flexible	one fixed $\theta_{j,1} = a_j$	two fixed $\theta_{j,1} = a_j$ $\theta_{j,2} = b_j$	two fixed $\theta_{j,1} = a_j$ $\theta_{j,K_j-1} = b_j$	all fixed
`cor`	no	✓	✓	✓	✓	✓
	yes		✓	✓	✓	✓
`cov`	no		✓	✓	✓	✓
	yes			✓	✓	✓

Table 2: This table displays different model parameterizations in the presence of truly ordinal observations ($K_j > 2 \ \forall j \in J$). The row `cor` includes error structures `corGeneral`, `corEqui` and `corAR1`, while row `cov` includes the error structure `covGeneral`. The minimal restrictions (default) to ensure identifiability are given in green. The default threshold values (in case `threshold.values = NULL`) are always $a_j = 0$ and $b_j = 1$.

In the presence of binary observations ($K_j = 2$) in connection with a covariance error structure, the intercept has always to be fixed to some value due to identifiability constraints. In a correlation structure setting no further restrictions are required.

For example, the following restrictions on the threshold parameters

- $\theta_{11} = -4 \leq \theta_{12} \leq \theta_{13} \leq \theta_{14} \leq \theta_{15} \leq \theta_{16}$;
- $\theta_{21} = -4 \leq \theta_{22} \leq \theta_{23} \leq \theta_{24} \leq \theta_{25} \leq \theta_{26}$;
- $\theta_{31} = -5 \leq \theta_{32} \leq \theta_{33} \leq \theta_{34} \leq \theta_{35} \leq \theta_{36} \leq \theta_{37}$;
- $\theta_{41} = 0$.

are implemented as:

```
R> threshold.values <- list(c(-4, NA, NA, NA, NA, NA),
+                           c(-4, NA, NA, NA, NA, NA),
+                           c(-5, NA, NA, NA, NA, NA, NA),
+                           c(0))
R> names(threshold.values) <- response.names
R> threshold.values
```

```
$R1
[1] -4 NA NA NA NA NA

$R2
[1] -4 NA NA NA NA NA

$R3
[1] -5 NA NA NA NA NA NA

$R4
[1] 0
```

2.6. Constraints on Coefficients

Similar to the threshold parameters, the package supports constraints on the regression coefficients. Firstly, the user can specify whether the regression coefficients should be equal across some or all response dimensions. Secondly, the values of some of the regression coefficients can be fixed.

Coefficient constraints across responses

Such constraints can be specified by a vector or a matrix `coef.constraints`, which can be either a vector or a matrix of integer values.

If vector constraints of the type $\boldsymbol{\beta}_k = \boldsymbol{\beta}_l$, are desired, which should hold for all p regression coefficients corresponding to outcome k and l, the easiest way to specify this is by means of a vector of integers of dimension q, where outcomes with equal vectors of regression coefficients get the same integer.

For example, for $q = 4$, a model where the regression coefficients of the first and second outcomes are equal ($\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$), while the coefficients of outcomes three and four are unrestricted, can be specified as:

```
R> coef.constraints <- c(1, 1, 2, 3)
R> names(coef.constraints) <- response.names
R> coef.constraints

R1 R2 R3 R4
 1  1  2  3
```

A more flexible framework allows the user to specify such constraints for each of the regression coeffiecients of the p covariates, not only for the whole vector. Such constraints will be specified by means of a matrix of dimension $q \times p$, where each column specifies constraints for one of the p covariates in the same way presented above. Moreover, a value of `NA` indicates that the corresponding coefficient is fixed (as we will show below) and should not be estimated.

The following constraints on the regression coefficients:

- $\beta_{12} = \beta_{22} = \beta_{32}$;

- $\beta_{13} = 0$, $\beta_{23} = 0$, $\beta_{33} = 0$;

- $\beta_{14} = \beta_{24} = \beta_{34}$, $\beta_{44} = 0$;

- $\beta_{15} = \beta_{25} = \beta_{35} = \beta_{45} = 2$;

give rise to the following model:

$$\begin{aligned}
\widetilde{Y}_{i1} &= \beta_{11}x_{i1} + \beta_{12}x_{i2} && + \beta_{14}x_{i4} + 2x_{i5} + \beta_{16}x_{i6} + \beta_{17}x_{i7},\\
\widetilde{Y}_{i2} &= \beta_{21}x_{i1} + \beta_{12}x_{i2} && + \beta_{14}x_{i4} + 2x_{i5} + \beta_{26}x_{i6} + \beta_{27}x_{i7},\\
\widetilde{Y}_{i3} &= \beta_{31}x_{i1} + \beta_{12}x_{i2} && + \beta_{14}x_{i4} + 2x_{i5} + \beta_{36}x_{i6} + \beta_{37}x_{i7},\\
\widetilde{Y}_{i4} &= \beta_{41}x_{i1} + \beta_{42}x_{i2} + \beta_{43}x_{i3} && + 2x_{i5} + \beta_{46}x_{i6} + \beta_{47}x_{i7}.
\end{aligned}$$

These restrictions on the parameter set of the regression coefficients are imposed by:

```
R> coef.constraints = cbind(c(1, NA, 1, NA),
+                           c(NA, NA, NA, 1),
+                           c(1, 1, 1, NA),
+                           c(1, 2, 3, 4),
+                           c(1, 1, 1, 4),
+                           c(1, 2, 3, 4),
+                           c(NA, NA, NA, 1))
R> rownames(coef.constraints) <- response.names
R> colnames(coef.constraints) <- c("ICR", "LR", "LEV1", "LEV2", "PR",
+                                  "lRSIZE", "lSYSR")
R> coef.constraints
```

```
   ICR LR LEV1 LEV2 PR lRSIZE lSYSR
R1   1 NA    1    1  1      1    NA
R2  NA NA    1    2  1      2    NA
R3   1 NA    1    3  1      3    NA
R4  NA  1   NA    4  4      4     1
```

Specific values of coefficients can be fixed through the `coef.values` argument, as we will show in the following.

Fixing coefficient values

In addition, specific values on regression coefficients can be set in the $q \times p$ matrix `coef.values`. Again each column corresponds to the regression coefficients of one covariate. This feature is to be used if some of the covariates have known slopes, but also for excluding covariates from the mean model of some of the outcomes (by fixing the regression coefficient to zero).

By default, if no `coef.values` are passed by the user, all the regression coefficients which receive an `NA` in `coef.constraints` will be set to zero. `NA` in the `coef.values` matrix indicates the regression coefficient ought to be estimated. For the example above, we have:

```
R> coef.values <- cbind(c(NA, NA, NA, NA),
+                       c(NA, NA, NA, NA),
+                       c(0, 0, 0, NA),
+                       c(NA, NA, NA, 0),
+                       c(2, 2, 2, 2),
+                       c(NA, NA, NA, NA),
+                       c(NA, NA, NA, NA))
R> rownames(coef.values) <- response.names
R> colnames(coef.values) <- c("ICR", "LR", "LEV1", "LEV2", "PR",
+                             "lRSIZE", "lSYSR")
R> coef.values

   ICR LR LEV1 LEV2 PR lRSIZE lSYSR
R1  NA NA    0   NA  2     NA    NA
R2  NA NA    0   NA  2     NA    NA
R3  NA NA    0   NA  2     NA    NA
R4  NA NA   NA    0  2     NA    NA
```

Note on interaction terms and factor covariates When constraints on the regression coefficients should be specified in models with interaction terms or factor covariates, the `coef.constraints` matrix has to be constructed appropriately. If the order of the terms in the covariate matrix is not clear to the user, it is helpful to call the function `model.matrix()` before constructing the `coef.constraints` and `coef.values` matrices. The command

```
R> formula <- rating ~ 0 + ICR : LR + LEV1 + LEV2 + PR + lRSIZE * lSYSR
R> colnames(model.matrix(formula, data = data_cr_multord))

[1] "LEV1"         "LEV2"         "PR"           "lRSIZE"
[5] "lSYSR"        "ICR:LR"       "lRSIZE:lSYSR"
```

will give the names of each column in the covariate matrix and should be used when setting up the coefficient constraints.

2.7. Additional arguments

Weights

Weights on each subject i can be chosen in a way that they are constant across repeated measurements. Weights should be part of the `data`. The column name of the weights in `data` should be passed to this argument. Negative weights are not allowed.

Solver

All general-purpose optimizers of the R package **optimx** can be used for maximization of the composite log-likelihood. These are `"Nelder-Mead"`, `"BFGS"`, `"CG"`, `"L-BFGS-B"`, `"nlm"`, `"nlminb"`, `"spg"`, `"ucminf"`, `"newuoa"`, `"bobyqa"`, `"nmkb"`, `"hjkb"`, `"Rcgmin"` and `"Rvmmin"` (Nash and Varadhan 2011; Nash 2014). The default is the `"BFGS"` solver. However, also the "newuoa" solver performed very well in terms of convergence in our experiments. Moreover, if the user desires a specific solver which is not implemented in the R package **optimx**, other applicable solvers can be used by using a wrapper function with arguments `starting.values`, `objFun`, `control` of the following form:

```
R> solver = function(starting.values, objFun, control){
+    optRes <- solver.function(...)
+    list(optpar = , optRes$optpar, # a vector of length equal to number of parameters to
+         objvalue = optRes$objvalue) # value of objective function
+  }
```

The output of the `solver.function` has to be a list of vector of length of the `starting.values` (threshold parameters, regression coefficients and error structure parameters) and value of the objective function.

Standard errors

If `se = TRUE` standard errors are computed using the Godambe information matrix (see Section 1.4).

Starting values

A list of starting values for threshold as well as regression coefficients can be passed by the argument `start.values`. This list contains a list (with a vector of starting values for each dimension) `theta` of all flexible threshold parameters and a list `beta` of all flexible regression parameters. All fixed values need to be excluded and in case of constraints on a whole dimension (e.g., `threshold.constraints = c(1,1,2,3)` or `coef.constraints = c(1,1,2,3)`), the element can be either skipped or a vector of length zero can be set. Starting values for Example 1 in Section 3 are for example:

```
R> start.values = list(theta = list(c(-3,-1,0,0.5,2.5),
+                                    c(-3,-1,0,0.5,2,3.5),
+                                    c(0)),
+                      beta = list(c(0.05,-0.05,-0.8,1,0.2),
+                                  c(-0.5,0.2),
+                                  c(-0.3,0.3),
+                                  c(0.5,-1.1,0.7,0.3,-1.2)))
```

2.8. Methods for class "multord"

Several methods are implemented for the class "multord". These methods include a `summary()` and a `print()` function to represent the estimation results, a `coef()` function to extract the regression coefficients, a `thresholds()` function to extract the threshold coefficients and a function `get.error.struct()` to extract the estimated parameters of the correlation/covariance structure of the errors. In addition, the pairwise log-likelihood can be extracted by `logPL()` as well as information critera like CLAIC by `claic()` and CLBIC by `clbic()`.

2.9. Output

The function `multord` returns an object of class `"multord"`, which is a list containing the following components:

`beta`	a named `matrix` of regression coefficients
`theta`	a named `list` of threshold parameters
`error.struct`	a named `list` of correlation (covariance) matrices, or a vector of coefficients in the `corEqui` or `corAR1` setting
`sebeta`	a named `matrix` of the standard errors of the regression coefficients
`setheta`	a named `list` of the standard errors of the threshold parameters
`seerror.struct`	a named `list` of the standard errors of the correlation (covariance) matrices, or a vector of the standard errors of the coefficients in the `corEqui` or `corAR1` setting
`rho`	a `list` of all objects that are used in `multord`

2.10. Implementation multord2()

Additionally, a second function `multord2()` is implemented, for the setting where the covariates do not vary between the repeated measurements ($\boldsymbol{x}_{i1} = \cdots = \boldsymbol{x}_{iq}$):

```
R> multord2(formula,
+           error.structure = corGeneral(~1),
+           data,
+           link = c("probit", "logit"),
+           coef.constraints = NULL,
+           coef.values = NULL,
+           threshold.constraints = NULL,
+           threshold.values = NULL,
+           weights = NULL,
+           se = TRUE,
+           start.values = NULL,
+           solver = "BFGS",
+           control = list(maxit = 200000, trace = 1, kkt = FALSE))
```

This function uses a slightly simplified data structure, where the repeated ordinal observations as well as the covariates are stored as columns in a `data.frame`. Each subject i corresponds to one row of the data frame, where all outcomes $Y_{i1}, \ldots, Y_{iq}$ (with missing observations set to `NA`) and all the covariates $x_{i1}, \ldots, x_{ip}$ are stored in different columns. Each outcome must be of type `Ord.factor`.

```
R> head(data_cr_multord2, n = 3)

  firm_id R1   R2   R3 R4      ICR        LR     LEV1       LEV2
1       1  D <NA>    K  N 1.546318 0.2484137 3.782934 0.92053787
2       2  B <NA> <NA>  N 8.723779 0.1506502 1.033042 0.05305052
3       3  D <NA> <NA>  N 4.726520 0.5187664 8.942818 0.97001785
         PR     lRSIZE     lSYSR  BSEC
1 0.2743184 -11.202807 -3.691023 BSEC3
```

```
2 0.1182763  -8.815116 -4.270618 BSEC3
3 0.2871493  -9.548691 -3.895642 BSEC6
```

In order to specify the mean model we use a multivariate formula object of the form:

```
R> formula <- cbind(R1, R2, R3) ~ 0 + X1 + ... + Xp
```

The `error.structure` and the constraints on the regression and threshold parameters are set in analogy to `multord()`, however, the ordering of the responses is given by the ordering in the model `formula`. In addition, the `link`, subject `weights`, `se` and the `solver` are chosen in the same way as in `multord()`.

3. Examples

The motivation of this package lies in a credit risk application, where multiple credit ratings are assigned by various credit rating agencies (CRAs) to firms over several years. CRAs have an important role in financial markets, as they deliver (subjective) assessments or opinions of an entity's (typically firm or sovereign) creditworthiness, which are then used by other players on the market, such as investors and regulators, in their decision making process. Entities are assigned to rating classes by CRAs on an ordinal scale by using both quantitative and qualitative criteria. This setting is an example of an application where correlated ordinal data arises naturally. On the one hand, multiple ratings for one firm at the same point in time can be assumed to be correlated and on the other hand, given the longitudinal dimension of the data, for each rater, there is serial dependence in the ratings assigned over several periods.

The data set used in the original credit risk application cannot be made available due to proprietary reasons. We therefore resort to the simulation of data sets which have a similar structure to the original data.

3.1. Example 1 – ratings assigned by multiple raters to a cross-section of firms

The first example presents a multivariate ordinal logit regression model with a general correlation error structure (`corGeneral(~ 1)`). The simulated data set contains the credit risk measure `rating` (ratings assigned by raters `R1`, `R2`, `R3` and `R4`) and 8 covariates for a cross-section of 1665 firms. The number of firm-ratings is 4566.

```
R> head(data_cr_multord, n = 3)

  firm_id rater_id rating      ICR        LR     LEV1       LEV2
1       1       R1      D 1.546318 0.2484137 3.782934 0.92053787
2       2       R1      B 8.723779 0.1506502 1.033042 0.05305052
3       3       R1      D 4.726520 0.5187664 8.942818 0.97001785
         PR     lRSIZE     lSYSR  BSEC
1 0.2743184 -11.202807 -3.691023 BSEC3
2 0.1182763  -8.815116 -4.270618 BSEC3
3 0.2871493  -9.548691 -3.895642 BSEC6
```

```
R> str(data_cr_multord, vec.len = 3)

'data.frame':	4566 obs. of  11 variables:
 $ firm_id : Factor w/ 1665 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 ...
 $ rater_id: Factor w/ 4 levels "R1","R2","R3",..: 1 1 1 1 1 1 1 1 ...
 $ rating  : chr  "D" "B" "D" ...
 $ ICR     : num  1.55 8.72 4.73 4.08 ...
 $ LR      : num  0.248 0.151 0.519 0.168 ...
 $ LEV1    : num  3.78 1.03 8.94 2.19 ...
 $ LEV2    : num  0.9205 0.0531 0.97 2.8743 ...
 $ PR      : num  0.2743 0.1183 0.2871 0.0821 ...
 $ lRSIZE  : num  -11.2 -8.82 -9.55 -8.66 ...
 $ lSYSR   : num  -3.69 -4.27 -3.9 -5.13 ...
 $ BSEC    : Factor w/ 8 levels "BSEC1","BSEC2",..: 3 3 6 4 3 1 6 4 ...
```

The panel of ratings is unbalanced:

```
R> nfirms <- length(unique(data_cr_multord$firm_id))
R> table(data_cr_multord$rater_id)/nfirms

       R1        R2        R3        R4 
0.9459459 0.1387387 0.6576577 1.0000000 
```

We observe that rater `R1` rates 95% of the firms, rater `R2` rates only 14% of the firms, rater `R3` rates 66% of the firms and rater `R4` rates all the firms in the sample.

The distribution of the ratings classes for the four raters is:

```
R> by(data_cr_multord,  data_cr_multord$rater_id,
+      function(x) table(x$rating))

data_cr_multord$rater_id: R1

  A   B   C   D   E   F 
 89 450 605 281  89  61 
------------------------------------------------ 
data_cr_multord$rater_id: R2

 A  B  C  D  E  F 
12 74 79 51  9  6 
------------------------------------------------ 
data_cr_multord$rater_id: R3

  G   H   I   J   K   L   M 
 40 163 169 209 404  88  22 
------------------------------------------------ 
data_cr_multord$rater_id: R4
```

```
   N    O
1067  598
```

We include 7 financial ratios as covariates in a model without intercept by the formula:

```
R> formula <- rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
R> formula

rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
```

The subject index i is stored in the column `firm_id` and the multiple measurement index j, which indicates the rater, is given by `rater_id`:

```
R> index <- c("firm_id", "rater_id")
R> index

[1] "firm_id"  "rater_id"
```

An optional vector `response.names` is used to specify all the raters to be included in the model. The ordering of this vector is essential when constraints on the parameter set want to be imposed:

```
R> response.names <- c("R1", "R2", "R3", "R4")
R> response.names

[1] "R1" "R2" "R3" "R4"
```

Due to the fact that the categories differ across raters we specify the `response.levels` by:

```
R> response.levels <-  list(rev(LETTERS[1:6]),
+                            rev(LETTERS[1:6]),
+                            rev(LETTERS[7:13]),
+                            rev(LETTERS[14:15]))
R> names(response.levels) <- response.names
R> response.levels

$R1
[1] "F" "E" "D" "C" "B" "A"

$R2
[1] "F" "E" "D" "C" "B" "A"

$R3
[1] "M" "L" "K" "J" "I" "H" "G"

$R4
[1] "O" "N"
```

If no `response.levels` are passed, the natural ordering is used and could lead to an incorrect labeling. The rating classes assigned by the raters are here in order from worst to best indicating that lower values of the latent variables indicate lower creditworthiness or increased credit risk.

We fit a model to these data which:

- assumes that R1 and R2 use the same rating scale by setting the following constraints on the threshold parameters:

```
R>  threshold.constraints <- c(1,1,2,3)
R>  names(threshold.constraints) <- response.names
R>  threshold.constraints

R1 R2 R3 R4
 1  1  2  3
```

- assumes that some covariates are equal for some raters. For example, we assume that the coefficient of ICR is equal for R1 and R3, or that the coeffiecients of LEV1 and PR are the same for the raters R1, R2 and R3. In addition, some of the regression coefficients are set to zero like ICR for R1 and R3, or lSYSR for the raters R1, R2 and R3. All the constraints above and some additional constraints are performed by the following restrictions on the regression coefficients by using the advanced method:

```
R> coef.constraints = cbind(c(1,NA,1,NA),
+                           c(NA,NA,NA,1),
+                           c(1,1,1,NA),
+                           c(1,2,3,4),
+                           c(1,1,1,4),
+                           c(1,2,3,4),
+                           c(NA,NA,NA,1))
R> rownames(coef.constraints) <- response.names
R> colnames(coef.constraints) <- c("ICR", "LR", "LEV1", "LEV2",
+                                  "PR", "lRSIZE", "lSYSR")
R> coef.constraints

   ICR LR LEV1 LEV2 PR lRSIZE lSYSR
R1   1 NA    1    1  1      1    NA
R2  NA NA    1    2  1      2    NA
R3   1 NA    1    3  1      3    NA
R4  NA  1   NA    4  4      4     1
```

The NAs in `coef.constraints` have to be fixed to some value. If no matrix `coef.values` is provided, the coefficients are set by default to zero automatically. This automatically generated `coef.values` matrix, looks like:

```
R> coef.values <- cbind(c(NA, 0, NA, 0),
+                       c(0, 0, 0, NA),
+                       c(NA, NA, NA, NA),
```

```
+                        c(NA, NA, NA, 0),
+                        c(NA, NA, NA, NA),
+                        c(NA, NA, NA, NA),
+                        c(0, 0, 0, NA))
R> rownames(coef.values) <- response.names
R> colnames(coef.values) <- c("ICR", "LR", "LEV1", "LEV2",
+                              "PR", "lRSIZE", "lSYSR")
R> coef.values

   ICR LR LEV1 LEV2 PR lRSIZE lSYSR
R1  NA  0   NA   NA NA     NA     0
R2   0  0   NA   NA NA     NA     0
R3  NA  0   NA   NA NA     NA     0
R4   0 NA   NA    0 NA     NA    NA
```

The specified `coef.constraints` together with `coef.values` give the following model:

$$
\begin{aligned}
\widetilde{Y}_1 &= \beta_{11}\texttt{ICR}+ \quad\quad \beta_{13}\texttt{LEV1}+\beta_{14}\texttt{LEV2}+\beta_{15}\texttt{PR}+\beta_{16}\texttt{lRSIZE},\\
\widetilde{Y}_2 &= \quad\quad\quad\quad\quad\quad \beta_{13}\texttt{LEV1}+\beta_{24}\texttt{LEV2}+\beta_{15}\texttt{PR}+\beta_{26}\texttt{lRSIZE},\\
\widetilde{Y}_3 &= \beta_{11}\texttt{ICR}+ \quad\quad \beta_{13}\texttt{LEV1}+\beta_{34}\texttt{LEV2}+\beta_{15}\texttt{PR}+\beta_{36}\texttt{lRSIZE},\\
\widetilde{Y}_4 &= \quad\quad \beta_{42}\texttt{LR}+ \quad\quad \beta_{44}\texttt{LEV2}+\beta_{45}\texttt{PR}+\beta_{46}\texttt{lRSIZE}+\beta_{47}\texttt{lSYSR}.
\end{aligned}
$$

As a link function we choose the logit link:

```
R> link <- "logit"
```

For simplicity, we use a general correlation structure which is constant for all subjects:

```
R> error.structure <- corGeneral(~ 1)
R> error.structure

$type
[1] "corGeneral"

$formula
~1
```

In order to avoid numerical instabilities, we standardize our data for each rater:

```
R> covar_names <- c("ICR", "LR","LEV1","LEV2", "PR","lRSIZE","lSYSR")
R> data_cr_multord_scaled <- do.call("rbind.data.frame",
+    by(data_cr_multord, data_cr_multord$rater_id,
+      function(x){x[, covar_names] <- scale(x[, covar_names]); x}))
```

The estimation can now be performed by the function `multord()`:

```
R> res_cor_logit <- multord(
+    formula = rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR,
+    error.structure = corGeneral(~ 1),
+    link = "logit",
+    data = data_cr_multord_scaled,
+    index = c("firm_id", "rater_id"),
+    response.names = c("R1", "R2", "R3", "R4"),
+    response.levels = list(rev(LETTERS[1:6]),
+                           rev(LETTERS[1:6]),
+                           rev(LETTERS[7:13]),
+                           rev(LETTERS[14:15])),
+    coef.constraints = cbind(c(1,NA,1,NA),
+                             c(NA,NA,NA,1),
+                             c(1,1,1,NA),
+                             c(1,2,3,4),
+                             c(1,1,1,4),
+                             c(1,2,3,4),
+                             c(NA,NA,NA,1)),
+    threshold.constraints = c(1,1,2,3))
```

The results are displayed either by the function `summary()`:

```
R> summary(res_cor_logit, call = FALSE)

Formula: rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR

 link threshold nsubjects ndim    logPL   CLAIC    CLBIC fevals
logit  flexible      1665    4 -7995.33 16121.2 16474.83    358

Threshold parameters:
          Estimate Std. Error     z value      Pr(>|z|) signif
R1 F|E -4.84882521 0.21246069 -22.8222224 2.759174e-115    ***
R1 E|D -3.19261718 0.13597584 -23.4792974 6.638998e-122    ***
R1 D|C -1.10858835 0.07317162 -15.1505243  7.516458e-52    ***
R1 C|B  0.94717024 0.06555248  14.4490366  2.542168e-47    ***
R1 B|A  3.38129642 0.12803640  26.4088682 1.083763e-153    ***
R3 M|L -6.14130711 0.32082572 -19.1421908  1.124414e-81    ***
R3 L|K -3.18193062 0.14945190 -21.2906663 1.385443e-100    ***
R3 K|J  0.02328815 0.07060703   0.3298277  7.415302e-01
R3 J|I  1.06403373 0.07516952  14.1551230  1.736338e-45    ***
R3 I|H  2.05389139 0.08979632  22.8727781 8.673229e-116    ***
R3 H|G  3.96195911 0.17336527  22.8532459 1.356722e-115    ***
R4 O|N -0.98345892 0.08487255 -11.5874799  4.769600e-31    ***

Coefficients:
            Estimate Std. Error   z value     Pr(>|z|) signif
ICR R1     0.3542920 0.03839647  9.227203 2.777908e-20    ***
```

```
ICR R2      0.0000000 0.00000000         NA           NA   <NA>
ICR R3      0.3542920 0.03839647   9.227203 2.777908e-20    ***
ICR R4      0.0000000 0.00000000         NA           NA   <NA>
LR R1       0.0000000 0.00000000         NA           NA   <NA>
LR R2       0.0000000 0.00000000         NA           NA   <NA>
LR R3       0.0000000 0.00000000         NA           NA   <NA>
LR R4      -0.3808375 0.06241727  -6.101476 1.050931e-09    ***
LEV1 R1    -0.1795457 0.04588253  -3.913160 9.109630e-05    ***
LEV1 R2    -0.1795457 0.04588253  -3.913160 9.109630e-05    ***
LEV1 R3    -0.1795457 0.04588253  -3.913160 9.109630e-05    ***
LEV1 R4     0.0000000 0.00000000         NA           NA   <NA>
LEV2 R1    -1.3901375 0.07034796 -19.760878 6.466964e-87    ***
LEV2 R2    -1.0021188 0.09383600 -10.679471 1.269931e-26    ***
LEV2 R3    -1.4644950 0.08241302 -17.770189 1.202840e-70    ***
LEV2 R4    -1.7185313 0.12278884 -13.995827 1.652955e-44    ***
PR R1       0.5546747 0.05009754  11.071895 1.717304e-28    ***
PR R2       0.5546747 0.05009754  11.071895 1.717304e-28    ***
PR R3       0.5546747 0.05009754  11.071895 1.717304e-28    ***
PR R4       2.1770281 0.12171745  17.885917 1.518342e-71    ***
lRSIZE R1   0.2522814 0.05161464   4.887787 1.019756e-06    ***
lRSIZE R2   0.4092883 0.09066342   4.514371 6.350495e-06    ***
lRSIZE R3   0.4750829 0.05675575   8.370656 5.729847e-17    ***
lRSIZE R4   0.1652422 0.07448297   2.218523 2.651922e-02      *
lSYSR R1    0.0000000 0.00000000         NA           NA   <NA>
lSYSR R2    0.0000000 0.00000000         NA           NA   <NA>
lSYSR R3    0.0000000 0.00000000         NA           NA   <NA>
lSYSR R4   -0.1985668 0.06601729  -3.007801 2.631458e-03     **

Error Structure:
             Estimate Std. Error   z value      Pr(>|z|) signif
corr R1 R2 0.9035414 0.02189504 41.266946  0.000000e+00    ***
corr R1 R3 0.6929071 0.02182114 31.753936 2.801830e-221    ***
corr R1 R4 0.4933093 0.03841571 12.841340  9.619454e-38    ***
corr R2 R3 0.7340554 0.04366159 16.812382  1.980517e-63    ***
corr R2 R4 0.5983776 0.10503523  5.696923  1.219892e-08    ***
corr R3 R4 0.7702408 0.03152657 24.431478 7.919559e-132    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

or by the function `print()`:

```
R> print(res_cor_logit, call = FALSE)

Threshold parameters:
$R1
       F|E        E|D        D|C        C|B        B|A
-4.8488252 -3.1926172 -1.1085883  0.9471702  3.3812964
```

```
$R2
      F|E        E|D        D|C        C|B        B|A 
-4.8488252 -3.1926172 -1.1085883  0.9471702  3.3812964 

$R3
        M|L         L|K         K|J         J|I         I|H 
-6.14130711 -3.18193062  0.02328815  1.06403373  2.05389139 
        H|G 
 3.96195911 

$R4
      O|N 
-0.9834589 

Coefficients:
        ICR         LR       LEV1      LEV2        PR    lRSIZE
R1 0.354292  0.0000000 -0.1795457 -1.390137 0.5546747 0.2522814
R2 0.000000  0.0000000 -0.1795457 -1.002119 0.5546747 0.4092883
R3 0.354292  0.0000000 -0.1795457 -1.464495 0.5546747 0.4750829
R4 0.000000 -0.3808375  0.0000000 -1.718531 2.1770281 0.1652422
        lSYSR
R1  0.0000000
R2  0.0000000
R3  0.0000000
R4 -0.1985668

Sigma:
          R1        R2        R3        R4
R1 1.0000000 0.9035414 0.6929071 0.4933093
R2 0.9035414 1.0000000 0.7340554 0.5983776
R3 0.6929071 0.7340554 1.0000000 0.7702408
R4 0.4933093 0.5983776 0.7702408 1.0000000
```

An extended summary, where all thresholds and regression coefficients are shown, even though they are duplicated, can be obtained by:

```
R> summary(res_cor_logit, short = FALSE, call = FALSE)

Formula: rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR

 link threshold nsubjects ndim    logPL   CLAIC    CLBIC fevals
logit  flexible      1665    4 -7995.33 16121.2 16474.83    358

Threshold parameters:
          Estimate Std. Error     z value      Pr(>|z|) signif
R1 F|E -4.84882521 0.21246069 -22.8222224 2.759174e-115    ***
```

```
R1 E|D -3.19261718 0.13597584 -23.4792974 6.638998e-122    ***
R1 D|C -1.10858835 0.07317162 -15.1505243  7.516458e-52    ***
R1 C|B  0.94717024 0.06555248  14.4490366  2.542168e-47    ***
R1 B|A  3.38129642 0.12803640  26.4088682 1.083763e-153    ***
R2 F|E -4.84882521 0.21246069 -22.8222224 2.759174e-115    ***
R2 E|D -3.19261718 0.13597584 -23.4792974 6.638998e-122    ***
R2 D|C -1.10858835 0.07317162 -15.1505243  7.516458e-52    ***
R2 C|B  0.94717024 0.06555248  14.4490366  2.542168e-47    ***
R2 B|A  3.38129642 0.12803640  26.4088682 1.083763e-153    ***
R3 M|L -6.14130711 0.32082572 -19.1421908  1.124414e-81    ***
R3 L|K -3.18193062 0.14945190 -21.2906663 1.385443e-100    ***
R3 K|J  0.02328815 0.07060703   0.3298277  7.415302e-01
R3 J|I  1.06403373 0.07516952  14.1551230  1.736338e-45    ***
R3 I|H  2.05389139 0.08979632  22.8727781 8.673229e-116    ***
R3 H|G  3.96195911 0.17336527  22.8532459 1.356722e-115    ***
R4 O|N -0.98345892 0.08487255 -11.5874799  4.769600e-31    ***

Coefficients:
             Estimate Std. Error    z value     Pr(>|z|) signif
ICR R1      0.3542920 0.03839647   9.227203 2.777908e-20    ***
ICR R2      0.0000000 0.00000000         NA           NA   <NA>
ICR R3      0.3542920 0.03839647   9.227203 2.777908e-20    ***
ICR R4      0.0000000 0.00000000         NA           NA   <NA>
LR R1       0.0000000 0.00000000         NA           NA   <NA>
LR R2       0.0000000 0.00000000         NA           NA   <NA>
LR R3       0.0000000 0.00000000         NA           NA   <NA>
LR R4      -0.3808375 0.06241727  -6.101476 1.050931e-09    ***
LEV1 R1    -0.1795457 0.04588253  -3.913160 9.109630e-05    ***
LEV1 R2    -0.1795457 0.04588253  -3.913160 9.109630e-05    ***
LEV1 R3    -0.1795457 0.04588253  -3.913160 9.109630e-05    ***
LEV1 R4     0.0000000 0.00000000         NA           NA   <NA>
LEV2 R1    -1.3901375 0.07034796 -19.760878 6.466964e-87    ***
LEV2 R2    -1.0021188 0.09383600 -10.679471 1.269931e-26    ***
LEV2 R3    -1.4644950 0.08241302 -17.770189 1.202840e-70    ***
LEV2 R4    -1.7185313 0.12278884 -13.995827 1.652955e-44    ***
PR R1       0.5546747 0.05009754  11.071895 1.717304e-28    ***
PR R2       0.5546747 0.05009754  11.071895 1.717304e-28    ***
PR R3       0.5546747 0.05009754  11.071895 1.717304e-28    ***
PR R4       2.1770281 0.12171745  17.885917 1.518342e-71    ***
lRSIZE R1   0.2522814 0.05161464   4.887787 1.019756e-06    ***
lRSIZE R2   0.4092883 0.09066342   4.514371 6.350495e-06    ***
lRSIZE R3   0.4750829 0.05675575   8.370656 5.729847e-17    ***
lRSIZE R4   0.1652422 0.07448297   2.218523 2.651922e-02      *
lSYSR R1    0.0000000 0.00000000         NA           NA   <NA>
lSYSR R2    0.0000000 0.00000000         NA           NA   <NA>
lSYSR R3    0.0000000 0.00000000         NA           NA   <NA>
lSYSR R4   -0.1985668 0.06601729  -3.007801 2.631458e-03     **
```

```
Error Structure:
            Estimate Std. Error   z value      Pr(>|z|) signif
corr R1 R2 0.9035414 0.02189504 41.266946  0.000000e+00    ***
corr R1 R3 0.6929071 0.02182114 31.753936 2.801830e-221    ***
corr R1 R4 0.4933093 0.03841571 12.841340  9.619454e-38    ***
corr R2 R3 0.7340554 0.04366159 16.812382  1.980517e-63    ***
corr R2 R4 0.5983776 0.10503523  5.696923  1.219892e-08    ***
corr R3 R4 0.7702408 0.03152657 24.431478 7.919559e-132    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The threshold coefficients can be extracted by the function `thresholds()`:

```
R> thresholds(res_cor_logit)

$R1
       F|E        E|D        D|C        C|B        B|A
-4.8488252 -3.1926172 -1.1085883  0.9471702  3.3812964

$R2
       F|E        E|D        D|C        C|B        B|A
-4.8488252 -3.1926172 -1.1085883  0.9471702  3.3812964

$R3
        M|L         L|K         K|J         J|I         I|H
-6.14130711 -3.18193062  0.02328815  1.06403373  2.05389139
        H|G
 3.96195911

$R4
       O|N
-0.9834589
```

The regression coefficients are obtained by the function `coef()`:

```
R> coef(res_cor_logit)

        ICR         LR       LEV1      LEV2        PR    lRSIZE
R1 0.354292  0.0000000 -0.1795457 -1.390137 0.5546747 0.2522814
R2 0.000000  0.0000000 -0.1795457 -1.002119 0.5546747 0.4092883
R3 0.354292  0.0000000 -0.1795457 -1.464495 0.5546747 0.4750829
R4 0.000000 -0.3808375  0.0000000 -1.718531 2.1770281 0.1652422
        lSYSR
R1  0.0000000
R2  0.0000000
R3  0.0000000
R4 -0.1985668
```

The error structure is displayed by the function `get.error.struct()`:

```
R> get.error.struct(res_cor_logit)

          R1        R2        R3        R4
R1 1.0000000 0.9035414 0.6929071 0.4933093
R2 0.9035414 1.0000000 0.7340554 0.5983776
R3 0.6929071 0.7340554 1.0000000 0.7702408
R4 0.4933093 0.5983776 0.7702408 1.0000000
```

Fitting the model with the function multord2()

Due to the fact that the covariates do not change across the repeated measurements (the covariates are firm specific and do not vary across raters), we can alternatively fit the model by the function `multord2()`. In `multord2()`, a slightly different format of `data` is used and the ordering of the responses is defined by a multivariate `formula` object. The repeated measurements are stored in different columns as ordered factors:

```
R> head(data_cr_multord2, n = 3)

  firm_id R1   R2   R3 R4      ICR        LR     LEV1       LEV2
1       1  D <NA>    K  N 1.546318 0.2484137 3.782934 0.92053787
2       2  B <NA> <NA>  N 8.723779 0.1506502 1.033042 0.05305052
3       3  D <NA> <NA>  N 4.726520 0.5187664 8.942818 0.97001785
         PR     lRSIZE     lSYSR  BSEC
1 0.2743184 -11.202807 -3.691023 BSEC3
2 0.1182763  -8.815116 -4.270618 BSEC3
3 0.2871493  -9.548691 -3.895642 BSEC6
```

```
R> str(data_cr_multord2, vec.len = 2)

'data.frame':        1665 obs. of  13 variables:
 $ firm_id: Factor w/ 1665 levels "1","2","3","4",..: 1 2 3 4 5 ...
 $ R1     : Ord.factor w/ 6 levels "F"<"E"<"D"<"C"<..: 3 5 3 1 5 ...
 $ R2     : Ord.factor w/ 6 levels "F"<"E"<"D"<"C"<..: NA NA NA NA NA ...
 $ R3     : Ord.factor w/ 7 levels "M"<"L"<"K"<"J"<..: 3 NA NA 1 NA ...
 $ R4     : Ord.factor w/ 2 levels "O"<"N": 2 2 2 1 2 ...
 $ ICR    : num  1.55 8.72 ...
 $ LR     : num  0.248 0.151 ...
 $ LEV1   : num  3.78 1.03 ...
 $ LEV2   : num  0.9205 0.0531 ...
 $ PR     : num  0.274 0.118 ...
 $ lRSIZE : num  -11.2 -8.82 ...
 $ lSYSR  : num  -3.69 -4.27 ...
 $ BSEC   : Factor w/ 8 levels "BSEC1","BSEC2",..: 3 3 6 4 3 ...
```

Again, we standardize the data to avoid numerical instabilities:

```
R> data_cr_multord2[, covar_names] <- scale(data_cr_multord2[, covar_names])
```

The estimation is performed by calling the function `multord2()`:

```
R> res_cor_logit <- multord(
+    formula = cbind(R1, R2, R3, R4) ~ 0 + ICR + LR + LEV1 + LEV2 + PR +
+                                       lRSIZE + lSYSR,
+    error.structure = corGeneral(~ 1),
+    link = "logit",
+    data = data_cr_multord_scaled,
+    coef.constraints = cbind(c(1,NA,1,NA),
+                             c(NA,NA,NA,1),
+                             c(1,1,1,NA),
+                             c(1,2,3,4),
+                             c(1,1,1,4),
+                             c(1,2,3,4),
+                             c(NA,NA,NA,1)),
+    threshold.constraints = c(1,1,2,3))
```

yielding equivalent results to the fit of `multord()`.

3.2. Example 2 – ratings assigned by one rater to a panel of firms

In a second example we present a longitudinal multivariate ordinal probit regression model with a covariate dependent $AR(1)$ error structure. The simulated data set contains the credit risk measure `rating` (ratings assigned by rater `R1`) and 8 covariates for a panel of 1665 firms over ten years. The number of firm-year observations is 11431:

```
R> str(data_cr_panel, vec.len = 3)

'data.frame':        11431 obs. of  11 variables:
 $ firm_id: Factor w/ 1665 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 ...
 $ year   : Factor w/ 10 levels "year1","year2",..: 1 1 1 1 1 1 1 1 ...
 $ rating : Ord.factor w/ 6 levels "F"<"E"<"D"<"C"<..: 3 5 3 1 5 3 4 5 ...
 $ ICR    : num  1.55 8.72 4.73 4.08 ...
 $ LR     : num  0.248 0.151 0.519 0.168 ...
 $ LEV1   : num  3.78 1.03 8.94 2.19 ...
 $ LEV2   : num  0.9205 0.0531 0.97 2.8743 ...
 $ PR     : num  0.2743 0.1183 0.2871 0.0821 ...
 $ lRSIZE : num  -11.2 -8.82 -9.55 -8.66 ...
 $ lSYSR  : num  -3.69 -4.27 -3.9 -5.13 ...
 $ BSEC   : Factor w/ 8 levels "BSEC1","BSEC2",..: 3 3 6 4 3 1 6 4 ...
```

```
R> head(data_cr_panel, n = 3)

  firm_id  year rating      ICR        LR     LEV1       LEV2
1       1 year1      D 1.546318 0.2484137 3.782934 0.92053787
```

```
2       2 year1      B 8.723779 0.1506502 1.033042 0.05305052
3       3 year1      D 4.726520 0.5187664 8.942818 0.97001785
         PR     lRSIZE     lSYSR  BSEC
1 0.2743184 -11.202807 -3.691023 BSEC3
2 0.1182763  -8.815116 -4.270618 BSEC3
3 0.2871493  -9.548691 -3.895642 BSEC6
```

The panel is highly unbalanced. The distribution of the number of ratings per firm assigned by rater `R1` over the 10 years is given by:

```
R> summary(rowSums(with(data_cr_panel, table(firm_id, year))))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   4.000   8.000   6.865  10.000  10.000
```

Per year the number of ratings decreases:

```
R> with(data_cr_panel, table(year))
```

```
year
 year1  year2  year3  year4  year5  year6  year7  year8  year9
  1665   1487   1377   1250   1135   1048    948    888    832
year10
   801
```

We include the 7 financial ratios as covariates in a model without intercept by the formula:

```
R> formula <- rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
R> formula
```

```
rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR
```

The subject index+i is stored in the column `firm_id` while the repeated measurement index j is given in the column `year`:

```
R> index <- c("firm_id", "year")
R> index
```

```
[1] "firm_id" "year"
```

If we wish to estimate the model only for the last eight years of the sample, this can be done by specifing the names of each dimension ordered response which should enter the model:

```
R> response.names <- paste0("year", 3:10)
R> response.names
```

```
[1] "year3"  "year4"  "year5"  "year6"  "year7"  "year8"
[7] "year9"  "year10"
```

The rating classes assigned by rater R1 are:

```
R> levels(data_cr_panel$rating)

[1] "F" "E" "D" "C" "B" "A"
```

with the sixth rating class F being the worst class and the first rating class A being the best rating class. We specify the response levels, in the order from worst to best, for each of the 10 outcome dimensions through the `response.level` argument. Ordering the classes from worst to best indicates that lower values of the latent variables indicate lower creditworthiness or increased credit risk. The rating classes and labels do not change over the ten years:

```
R> response.levels <- rep(list(levels(data_cr_panel$rating)),
+                         length(response.names))
R> names(response.levels) <- response.names
R> response.levels

$year3
[1] "F" "E" "D" "C" "B" "A"

$year4
[1] "F" "E" "D" "C" "B" "A"

$year5
[1] "F" "E" "D" "C" "B" "A"

$year6
[1] "F" "E" "D" "C" "B" "A"

$year7
[1] "F" "E" "D" "C" "B" "A"

$year8
[1] "F" "E" "D" "C" "B" "A"

$year9
[1] "F" "E" "D" "C" "B" "A"

$year10
[1] "F" "E" "D" "C" "B" "A"
```

Additionally, the model has the following features:

- we assume the rating agencies do not change their methodology over the sample period. This means the threshold parameters are constant over the years. This can be specified through the argument `threshold.constraints`:

```
R> threshold.constraints <- rep(1, length(response.names))
R> names(threshold.constraints) <- response.names
R> threshold.constraints

 year3  year4  year5  year6  year7  year8  year9 year10
     1      1      1      1      1      1      1      1
```

- we assume there is a breakpoint in the regression coefficients after `year5` in the sample (this could correspond to the beginning of a crisis in a real case application). Hence, we use one set of regression coefficients for years `year3`, `year4` and `year5` and a different set for `year6`, `year7`, `year8`, `year9`, `year10`. This can be specified through the argument `coef.constraints`:

```
R> coef.constraints = c(rep(1, 3),  rep(2, 5))
R> names(coef.constraints) <- response.names
R> coef.constraints

 year3  year4  year5  year6  year7  year8  year9 year10
     1      1      1      2      2      2      2      2
```

- allows for different correlation parameters in the $AR(1)$ structure for the different business sectors.

```
R> error.structure = corAR1(~ BSEC)
R> error.structure

$type
[1] "corAR1"

$formula
~BSEC
```

The estimation is performed by calling the function `multord()`: As before, we standardize our covariates on a yearly basis:

```
R> data_cr_panel_scaled <- do.call("rbind.data.frame",
+    by(data_cr_panel, data_cr_panel$year,
+      function(x){x[, covar_names] <- scale(x[, covar_names]); x}))

R> res_AR1_probit <- multord(
+      formula = rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR,
+      index = c("firm_id", "year"),
+      data = data_cr_panel_scaled,
+      response.levels = rep(list(levels(data_cr_panel$rating)), 8),
+      response.names = paste0("year", 3:10),
+      link = "probit",
+      error.structure = corAR1(~ BSEC),
+      coef.constraints = c(rep(1, 3),  rep(2, 5)),
+      threshold.constraints = rep(1, 8),
+      solver = "BFGS")
```

The results are displayed either by the function `summary()`:

```
R> summary(res_AR1_probit, short = TRUE, call = FALSE, digits = 6)

Formula: rating ~ 0 + ICR + LR + LEV1 + LEV2 + PR + lRSIZE + lSYSR

  link threshold nsubjects ndim     logPL     CLAIC     CLBIC fevals
probit  flexible      1392    8 -52751.13 105820.97 106655.76    222

Threshold parameters:
            Estimate Std. Error   z value     Pr(>|z|) signif
year3 F|E -4.0902846  0.0994663 -41.12230  0.00000e+00    ***
year3 E|D -1.0098087  0.0325721 -31.00229 5.02033e-211    ***
year3 D|C  0.0203523  0.0278756   0.73011  4.65323e-01
year3 C|B  1.0260948  0.0326125  31.46328 2.76294e-217    ***
year3 B|A  3.0498425  0.0695938  43.82347  0.00000e+00    ***

Coefficients:
               Estimate Std. Error   z value     Pr(>|z|) signif
ICR year3     0.1663783  0.0137447  12.10492  9.94626e-34    ***
ICR year6     0.0919790  0.0120313   7.64495  2.09024e-14    ***
LR year3      0.0242913  0.0132655   1.83116  6.70774e-02      .
LR year6     -0.1887875  0.0127358 -14.82340  1.03408e-49    ***
LEV1 year3   -0.0906768  0.0135194  -6.70715  1.98461e-11    ***
LEV1 year6   -0.2941070  0.0126544 -23.24156 1.73168e-119    ***
LEV2 year3   -0.7025465  0.0217684 -32.27375 1.63368e-228    ***
LEV2 year6   -1.6211075  0.0340118 -47.66310  0.00000e+00    ***
PR year3      0.2732929  0.0140069  19.51129  8.80340e-85    ***
PR year6      0.4133250  0.0141526  29.20498 1.67643e-187    ***
lRSIZE year3  0.1104956  0.0134466   8.21734  2.08064e-16    ***
lRSIZE year6  0.5331242  0.0154265  34.55895 1.04576e-261    ***
lSYSR year3  -0.0309864  0.0132937  -2.33091  1.97583e-02      *
lSYSR year6  -0.1976489  0.0126467 -15.62843  4.66113e-55    ***

Error Structure:
              Estimate Std. Error    z value    Pr(>|z|) signif
(Intercept)  1.3700044  0.0668721  20.486921 2.81652e-93    ***
BSECBSEC2   -0.6047990  0.0851659  -7.101423 1.23478e-12    ***
BSECBSEC3    0.1152617  0.0794137   1.451408 1.46666e-01
BSECBSEC4    0.1429737  0.0764200   1.870892 6.13600e-02      .
BSECBSEC5    0.0194572  0.0928413   0.209575 8.34000e-01
BSECBSEC6   -0.4795777  0.0878203  -5.460901 4.73724e-08    ***
BSECBSEC7   -0.9067004  0.0852025 -10.641714 1.90590e-26    ***
BSECBSEC8   -0.5922317  0.1110170  -5.334602 9.57544e-08    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

or by the function `print()`:

```
R> print(res_AR1_probit, call = FALSE, digits = 4)

Threshold parameters:
$year3
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year4
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year5
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year6
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year7
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year8
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year9
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

$year10
     F|E      E|D      D|C      C|B      B|A
-4.09028 -1.00981  0.02035  1.02609  3.04984

Coefficients:
           ICR       LR     LEV1    LEV2     PR lRSIZE    lSYSR
year3  0.16638  0.02429 -0.09068 -0.7025 0.2733 0.1105 -0.03099
year4  0.16638  0.02429 -0.09068 -0.7025 0.2733 0.1105 -0.03099
year5  0.16638  0.02429 -0.09068 -0.7025 0.2733 0.1105 -0.03099
year6  0.09198 -0.18879 -0.29411 -1.6211 0.4133 0.5331 -0.19765
year7  0.09198 -0.18879 -0.29411 -1.6211 0.4133 0.5331 -0.19765
year8  0.09198 -0.18879 -0.29411 -1.6211 0.4133 0.5331 -0.19765
year9  0.09198 -0.18879 -0.29411 -1.6211 0.4133 0.5331 -0.19765
year10 0.09198 -0.18879 -0.29411 -1.6211 0.4133 0.5331 -0.19765

alpha parameters error.structure:
```

```
(Intercept)   BSECBSEC2   BSECBSEC3   BSECBSEC4   BSECBSEC5
    1.37000    -0.60480     0.11526     0.14297     0.01946
  BSECBSEC6   BSECBSEC7   BSECBSEC8
   -0.47958    -0.90670    -0.59223
```

An extended summary, where all thresholds and regression coefficients are shown, even though they are duplicated, can be obtained by:

```
R> summary(res_AR1_probit, short = FALSE, call = FALSE)
```

The threshold coefficients can be extracted by the function `thresholds()`:

```
R> thresholds(res_AR1_probit)

$year3
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year4
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year5
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year6
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year7
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year8
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year9
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253

$year10
        F|E         E|D         D|C         C|B         B|A
-4.09028458 -1.00980868  0.02035227  1.02609482  3.04984253
```

The regression coefficients are obtained by the function `coef()`:

```
R> coef(res_AR1_probit)

              ICR          LR       LEV1       LEV2        PR
year3  0.16637829  0.02429127 -0.0906768 -0.7025465 0.2732929
year4  0.16637829  0.02429127 -0.0906768 -0.7025465 0.2732929
year5  0.16637829  0.02429127 -0.0906768 -0.7025465 0.2732929
year6  0.09197896 -0.18878750 -0.2941070 -1.6211075 0.4133250
year7  0.09197896 -0.18878750 -0.2941070 -1.6211075 0.4133250
year8  0.09197896 -0.18878750 -0.2941070 -1.6211075 0.4133250
year9  0.09197896 -0.18878750 -0.2941070 -1.6211075 0.4133250
year10 0.09197896 -0.18878750 -0.2941070 -1.6211075 0.4133250
          lRSIZE       lSYSR
year3  0.1104956 -0.03098645
year4  0.1104956 -0.03098645
year5  0.1104956 -0.03098645
year6  0.5331242 -0.19764887
year7  0.5331242 -0.19764887
year8  0.5331242 -0.19764887
year9  0.5331242 -0.19764887
year10 0.5331242 -0.19764887
```

The error structure is displayed by the function `get.error.struct()`:

```
R> get.error.struct(res_AR1_probit)

(Intercept)   BSECBSEC2   BSECBSEC3   BSECBSEC4   BSECBSEC5
 1.37000442 -0.60479901  0.11526167  0.14297368  0.01945718
  BSECBSEC6   BSECBSEC7   BSECBSEC8
-0.47957770 -0.90670039 -0.59223175
```

In addition, the correlation parameters ρ_i for each firm are obtained by:

```
R> head(get.error.struct(res_AR1_probit, type = "corr"), n = 3)

  Correlation
1   0.9024499
2   0.9024499
3   0.7116044
```

Moreover, the correlation matrices for each specific firm are obtained by:

```
R> head(get.error.struct(res_AR1_probit, type = "sigmas"), n = 1)

$`1`
          year3     year4     year5     year6     year7
year3  1.0000000 0.9024499 0.8144159 0.7349696 0.6632733
year4  0.9024499 1.0000000 0.9024499 0.8144159 0.7349696
```

```
year5  0.8144159 0.9024499 1.0000000 0.9024499 0.8144159
year6  0.7349696 0.8144159 0.9024499 1.0000000 0.9024499
year7  0.6632733 0.7349696 0.8144159 0.9024499 1.0000000
year8  0.5985709 0.6632733 0.7349696 0.8144159 0.9024499
year9  0.5401803 0.5985709 0.6632733 0.7349696 0.8144159
year10 0.4874857 0.5401803 0.5985709 0.6632733 0.7349696
           year8     year9    year10
year3  0.5985709 0.5401803 0.4874857
year4  0.6632733 0.5985709 0.5401803
year5  0.7349696 0.6632733 0.5985709
year6  0.8144159 0.7349696 0.6632733
year7  0.9024499 0.8144159 0.7349696
year8  1.0000000 0.9024499 0.8144159
year9  0.9024499 1.0000000 0.9024499
year10 0.8144159 0.9024499 1.0000000
```

References

Agresti A (2002). *Categorical Data Analysis (2nd ed.)*. Wiley series in Probability and mathematical Statistics.

Genz A, Bretz F (2009). *Computation of Multivariate Normal and T Probabilities*. 1st edition. Springer Publishing Company, Incorporated. ISBN 364201688X, 9783642016882.

Genz A, Kenkel B (2015). *pbivnorm: Vectorized Bivariate Normal CDF*. R package version 0.6.0, URL https://CRAN.R-project.org/package=pbivnorm.

Hirk R, Hornik K, Vana L (2017). "Multivariate Ordinal Regression Models: An Analysis of Corporate Credit Ratings." *Research Report Series / Department of Statistics and Mathematics 132*, WU Vienna University of Economics and Business, Vienna. URL http://epub.wu.ac.at/5389/.

Nash JC (2014). "On Best Practice Optimization Methods in R." *Journal of Statistical Software*, **60**(2), 1–14. doi:10.18637/jss.v060.i02.

Nash JC, Varadhan R (2011). "Unifying Optimization Algorithms to Aid Software System Users: optimx for R." *Journal of Statistical Software*, **43**(9), 1–14. doi:10.18637/jss.v043.i09.

O'Brien SM, Dunson DB (2004). "Bayesian Multivariate Logistic Regression." *Biometrics*, **60**(3), 739–746. ISSN 1541-0420. doi:10.1111/j.0006-341X.2004.00224.x.

Pinheiro JC, Bates DM (1996). "Unconstrained parametrizations for variance-covariance matrices." *Statistics and Computing*, **6**(3), 289–296. ISSN 1573-1375. doi:10.1007/BF00140873.

Tutz G (2012). *Regression for Categorical Data*. Springers Series in Statistics. Cambridge University Press.

Varin C (2008). "On composite marginal likelihoods." *AStA Advances in Statistical Analysis*, **92**(1), 1. ISSN 1863-818X. doi:10.1007/s10182-008-0060-7.

Varin C, Reid N, Firth D (2011). "An Overview of Composite Likelihood Methods." *Statistica Sinica*, **21**(1), 5–42. ISSN 10170405, 19968507. URL http://www.jstor.org/stable/24309261.

Affiliation:

Rainer Hirk
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
1020 Vienna, Austria
E-mail: rainer.hirk@wu.ac.at

Kurt Hornik
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
1020 Vienna, Austria
E-mail: kurt.hornik@wu.ac.at

Laura Vana
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
1020 Vienna, Austria
E-mail: laura.vana@wu.ac.at