MultOrd: An R Package for Fitting Multivariate Ordinal Regression Models

Rainer Hirk **Kurt Hornik** **Laura Vana**
WU Wirtschaftsuniversität Wien

Abstract

The R package MultOrd implements composite likelihood estimation in the class of multivariate ordinal regression models. A flexible modeling framework for cross-sectional as well as longitudinal observations is set up, which allows different error structures. Two different link functions are employed by assuming a multivariate normal and a multivariate logistic distribution for the latent variables underlying the ordinal outcomes. In order to deal with the issue that absolute location and absolute scale are not identifiable in ordinal models, several restrictions on the parameter space are imposed either by fixing location parameters and/or by restricting the full covariance matrix to be a correlation matrix. In addition, constraints on both coefficient as well as threshold parameters can be imposed. Standard errors are computed using the Godambe information matrix.

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

1. Model Class

Multivariate ordinal regression models are based on *cumulative link models* 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 panel multivariate ordinal regression model.

1.1. Model formulation

Let us suppose to have J repeated measurements on n different subjects i, where each repeated ordinal observation (indexed by $j \in J$) is denoted by Y_{ij}. Each observable categorical outcome Y_{ij} and the unobservable latent variable $\widetilde{Y}_{ij}$ are connected by:

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

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: $-\infty \equiv \theta_{j,0} < \theta_{j,1} < \cdots < \theta_{j,K_j} \equiv \infty$.

The number of threshold categories K_j as well as the threshold parameters themselves are allowed to vary across outcome dimensions $j \in J$ in order to account for differences in the repeated measurements. Given an $n \times p$ matrix X_j of covariates for each $j \in J$, where each $\boldsymbol{x}_{ij}$ is a p-dimensional vector (i-th row of X_j) for subject i and repeated measurement j, the following linear model for the relationship between $\widetilde{Y}_{ij}$ and the vector of covariates $\boldsymbol{x}_{ij}$ is assumed:

$$\widetilde{Y}_{ij} = \beta_{j0} + \boldsymbol{x}_{ij}^{\top}\boldsymbol{\beta}_j + \epsilon_{ij}, \qquad \boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})^{\top} \sim F_J(\mathbf{0}, \boldsymbol{\Sigma}), \tag{1}$$

where

- β_{j0} is an intercept term corresponding to outcome j,
- $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jp})^{\top}$ is a vector of regression coefficients corresponding to outcome j,
- ϵ_{ij} is an error term with mean zero and distributed according to a J-dimensional distribution function F_J.

The regression parameters $\boldsymbol{\beta}_j$ are allowed to vary between the repeated measurements j and account for heterogeneity in the latent processes of the outcome dimensions. The errors are assumed to be independent across subjects and orthogonal to the covariates $\boldsymbol{x}_{ij}$. Typical multivariate distribution functions F are the multivariate normal distribution yielding a probit link between the linear predictor and the cumulative probabilities and the multivariate logistic distribution yielding the logit link. In addition, we allow for different error structures which will be discussed in the following subsections.

The notation above corresponds to a cross-sectional setting, where repeated ordinal observations Y_{ij} on the same subject i are observed at the same time. Similarly, in a panel setting we assume to have T repeated measurements Y_{it} at consecutive equi-spaced time points with index $t \in T$ of subject i . In principle, we can replace every j by t, in order to obtain the panel model notation. In the following we resume with the cross-sectional notation und use $j \in J$ as an index for the repeated measurements.

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 a full covariance matrix $\boldsymbol{\Sigma}$ with diagonal elements σ_j^2, only the quantities $\boldsymbol{\beta}_j/\sigma_j$ and $(\theta_{j,r_{ij}} - \beta_{j0})/\sigma_j$ are identifiable in model (1). The scale can be fixed either by restricting the full variance-covariance matrix $\boldsymbol{\Sigma}$ to be a correlation matrix $\boldsymbol{R}$, 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 following typical constraints on the parameter set are imposed:

- Fixing the intercept β_{j0} (e.g., to zero), using flexible thresholds $\boldsymbol{\theta}_j$ and fixing σ_j (e.g., to unity) for all $j \in J$.
- Leaving the intercept β_{j0} unrestricted, fixing one threshold parameter (e.g., $\theta_{j,1} = 0$) and fixing σ_j (e.g., to unity) for all $j \in J$.

- Leaving the intercept β_{j0} unrestricted, fixing two threshold parameters (e.g., $\theta_{j,1} = 0$ and $\theta_{j,2} = 1$) and leaving σ_j unrestricted for all $j \in J$.

- Fixing the intercept β_{j0} (e.g., to zero), fixing one threshold parameter (e.g., $\theta_{j,1} = 0$) and leaving σ_j unrestricted for all $j \in J$.

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 error structure and an $AR(1)$ error structure.

Correlation error structure

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

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})^\top \sim F_J \left(\mathbf{0}, \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1J} \\ \rho_{12} & 1 & \cdots & \rho_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1J} & \rho_{2J} & \cdots & 1 \end{pmatrix} \right). \tag{2}$$

  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 some 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 constant across repeated measurements j. The factor dependent error structure for a correlation structure has the following form:

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

- **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 J dimensions, but differ across subjects i. The correlation parameter ρ_i of each subject i is assumed to depend on a vector of covariates $\boldsymbol{w}_i$. Fisher's z-transformation allows

us to re-parameterize the linear term $\alpha_0 + \boldsymbol{w}_i^\top \boldsymbol{\alpha}$ in terms of a correlation parameter for each subject:

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

Solving for ρ_i gives us the following re-transformation:

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

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:

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

- $AR(1)$ **correlation structure**
  For given consecutive equi-spaced time points $t_1, \ldots, t_T$ 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:

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

  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:

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})^\top \sim F_J\left(\mathbf{0}, \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1J}\sigma_1\sigma_J \\ \rho_1\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2J}\sigma_2\sigma_J \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1J}\sigma_1\sigma_J & \rho_{2J}\sigma_2\sigma_J & \cdots & \sigma_J^2 \end{pmatrix}\right). \quad (3)$$

  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:

$$\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \epsilon_{i2}, \ldots, \epsilon_{iJ})^\top \sim F_J(\mathbf{0}, \boldsymbol{\Sigma}_{f(i)}).$$

Correlation	Covariance	Intercept	flexible thresholds	one fixed threshold	two fixed thresholds	all fixed thresholds
✓			✓			
✓		✓		✓		
✓		✓				✓
	✓	✓			✓	
	✓	✓				✓

Table 1: This table displays different model parameterizations.

1.4. Composite Likelihood Estimation

For a given parameter vector $\boldsymbol{\Gamma}$ which contains the threshold parameters $\boldsymbol{\Theta}$, the regression coefficients $\boldsymbol{B}$ and the correlation (or variance-covariance) parameters $\boldsymbol{R}$ that have to be estimated. The likelihood of all implemented multivariate ordinal regression models (in case of the panel notation j and $\boldsymbol{x}_{ij}$ have to be replaced by t and $\boldsymbol{x}_{it}$) can be represented in the following form:

$$\begin{aligned} L(\boldsymbol{\Gamma}) &= \prod_{i=1}^{n} \mathbb{P}(Y_{i1} = r_{i1}, Y_{i2} = r_{i2}, \ldots, Y_{iJ} = r_{iJ})^{w_i} \\ &= \prod_{i=1}^{n} \left(\int_{\theta_{1,r_{i1}-1}-\beta_{j0}-\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_j}^{\theta_{1,r_{i1}}-\beta_{j0}-\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_j} \cdots \int_{\theta_{J_i,r_{iJ_i}-1}-\beta_{j0}-\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_j}^{\theta_{J_i,r_{iJ_i}}-\beta_{j0}-\boldsymbol{x}_{ij}^\top\boldsymbol{\beta}_j} f_J(v_{i1}, \ldots, v_{iJ}; \boldsymbol{R}) dv_{i1} \ldots dv_{iJ} \right)^{w_i} \end{aligned}$$

where f_q denotes the density of a q-dimensional distribution F_q. In order to estimate the model parameters we approximate the full likelihood by a composite likelihood, where a pseudolikelihood is constructed from bivariate marginal distributions F_2. Using transformed upper $U_{ij} = \theta_{j,r_{ij}} - \beta_{j0} - \boldsymbol{x}_i^\top\boldsymbol{\beta}_j$ and lower $L_{ij} = \theta_{j,r_{ij}-1} - \beta_{j0} - \boldsymbol{x}_i^\top\boldsymbol{\beta}_j$ integration bounds, the

pairwise log-likelihood function is obtained by

$$\begin{aligned} c\ell^{PL}(\boldsymbol{\Gamma}) &= \sum_{i=1}^{n}\sum_{k=1}^{J-1}\sum_{l=k+1}^{J} w_i \log\left(\mathbb{P}(Y_{ik}=r_{ik}, Y_{il}=r_{il})\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{J-1}\sum_{l=k+1}^{J} w_i \log\left(\int_{L_{ik}}^{U_{ik}}\int_{L_{il}}^{U_{il}} f_2(v_{ik}, v_{il}|\rho_{kl})dv_{ik}dv_{il}\right) \end{aligned} \tag{4}$$

The maximum composite likelihood estimates $\hat{\boldsymbol{\Gamma}}_{c\ell}$ are obtained by direct maximization of the composite likelihood given in Equation (4) using general purpose optimizers of the R package `optimx` (Nash and Varadhan 2011; Nash 2014). Numerical differentiation is used to compute the standard errors in order to quantify the uncertainty of the maximum composite likelihood estimates. Under certain regularity conditions, the maximum composite likelihood estimates are consistent as J is fixed and $n \to \infty$. In addition, the maximum composite likelihood estimator is asymptotically normal with asymptotic mean $\boldsymbol{\Gamma}$ and a covariance matrix which equals the inverse of the Godambe information matrix:

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

where $G(\boldsymbol{\Gamma})$ denotes the Godambe information matrix, $H(\boldsymbol{\Gamma})$ the Hessian (sensitivity matrix) and $V(\boldsymbol{\Gamma})$ the variability matrix (Varin 2008). The Hessian $H(\boldsymbol{\Gamma})$ and variability matrix $V(\boldsymbol{\Gamma})$ can be estimated as follows: (or more like Reusens)

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

where $c\ell_i(\boldsymbol{\Gamma}|\boldsymbol{Y}_i)$ is the i-th component of the composite log likelihood. In order to compare different models, the composite likelihood information criterion can be used: $\mathrm{CLIC}(\boldsymbol{\Gamma}) = -2\, c\ell(\hat{\boldsymbol{\Gamma}}_{\mathrm{CL}}|X,Y) + k\ \mathrm{tr}(\hat{V}(\boldsymbol{\Gamma})\hat{H}(\boldsymbol{\Gamma})^{-1})$ (where $k=2$ corresponds to CLIC-AIC and $k=\log(n)$ corresponds to CLIC-BIC).

We use three reparameterizations in our implementation. First, in order to achieve monotonicity we use an unconstrained parameterization of the threshold parameters.Second, for all full correlation and covariance matrices we use the spherical parameterization of Pinheiro and Bates (1996). This parameterization for covariance matrices 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 to obtain an unconstrained correlation parameter.

2. Implementation

Multivariate ordinal regression models in the R package **MultOrd** are fitted using the function

```
multord(formula, data , index, response.names = NULL, response.levels = NULL,
        link = c("probit", "logit"), error.structure = corGeneral(~ 1),
```

```
        coef.constraints = NULL, coef.values = NULL,
        threshold.constraints = NULL, threshold.values = NULL,
        se = TRUE, start.values = NULL, solver = "newuoa")
```

Different model types are implemented and can be chosen by selecting different error structures in `multord`. Constraints on the threshold parameters as well as on the regression coefficients can be imposed. All features are explained by means of an example in credit risk with the following simulated dataset:

```
> head(data_multord)

  firmID raterID rating         X1          X2          X3          X4
1      1  rater1      B -1.7253621  0.23686361  0.68404628 -0.85374252
2      2  rater1     BB  0.6910390  0.24387582 -0.03467597 -1.13908602
3      3  rater1      B -1.4880963  0.24746024 -0.31505879  0.45284491
4      4  rater1    BBB -2.1109710  0.31692055 -0.08755990 -0.46870265
5      5  rater1      B -0.9454859 -0.25000228 -0.29887960 -1.39374063
6      6  rater1    BBB -0.2285612 -0.08547767  0.11796657 -0.03109767
          X5 X6
1 -0.4447102  X
2  1.6427284  Y
3  1.4863629  Y
4 -1.3563313  Z
5  1.1312291  X
6 -1.0999933  X

> str(data_multord)

'data.frame':        4000 obs. of  9 variables:
 $ firmID : int  1 2 3 4 5 6 7 8 9 10 ...
 $ raterID: Factor w/ 4 levels "rater1","rater2",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ rating : Ord.factor w/ 14 levels "C"<"B"<"BB"<"BBB"<..: 2 3 2 4 2 4 6 2 4 1 ...
 $ X1     : num  -1.725 0.691 -1.488 -2.111 -0.945 ...
 $ X2     : num  0.237 0.244 0.247 0.317 -0.25 ...
 $ X3     : num  0.684 -0.0347 -0.3151 -0.0876 -0.2989 ...
 $ X4     : num  -0.854 -1.139 0.453 -0.469 -1.394 ...
 $ X5     : num  -0.445 1.643 1.486 -1.356 1.131 ...
 $ X6     : Factor w/ 3 levels "X","Y","Z": 1 2 2 3 1 1 3 2 2 3 ...
```

2.1. Data structure

We use the long format for the input of `data`, where each row contains a subject index i (`firmID`), a repeated measurement index j (`raterID`), an ordinal response (`rating`) and all the covariates (`X1, ..., X6`). This long format data stucture is internally transformed to matrix of covariates Y and a list of covariate matrices X_j for all $j \in J$ by a matching according to the subject index i and the repeated measurement index j, which are passed by an optional argument `index`. This is usually performed by a character vector of length two specifying the column names of the subject index and the repeated measurement index in `data`.

```
> index = c("firmID", "raterID")
```

The default value of `index` is `NULL` assuming that the first column of `data` contains the subject index and the second column the repeated measurement index. In order to avoid numerical instabilities we suggest to standardize the covariates $\boldsymbol{x}_{ij}$.

If specific constraints are imposed on the parameter set, 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 measurement.

```
> response.names = c("rater1", "rater2", "rater3", "defaultInd")
```

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`.

```
> response.levels = list(c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                        c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                        c("D","C","B","Ba", "Baa", "A", "Aa", "Aaa"),
+                        c("default", "no default"))
```

If the categories differ across repeated measurements (either the number of categories 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 descending order.

2.2. Formula

The ordinal responses Y (`rating`) are passed by a `formula` object. Intercepts can be included or excluded in the model depending on the model paramterization:

Model without intercept: If the intercept should be removed the `formula` for a given response (`rating`) and covariates (`X1` to `Xp`) has the following form:

```
> formula <- rating ~ 0 + X1 + X2 + ... + Xp
```

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

```
> formula <- rating ~ 1 + X1 + X2 + ... + Xp
```

or by

```
> formula <- rating ~ X1 + X2 + ... + Xp
```

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 a multivariate logistic distribution introduced by O'Brien and Dunson (2004) is used.

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).

    ```
    > 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`. This 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.

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

  - `corEqui`
    A covariate dependent equicorrelation structure, where the correlations are equal across all J dimensions and depend on some covariates `X1, ..., Xp`. It has to be noted that these covariates `X1, ..., Xp` as well as the factor `f` are not allowed to vary across repeated measurements j for the same subject i.

    ```
    > error.structure <- corEqui(~ X1 + ... + Xp)
    ```

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

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

    In order to account for some heterogeneity the $AR(1)$ error structure is allowed to depend on covariates `X1, ..., Xp` that are constant over time for each subject i

    ```
    > error.structure = corAR1(~ X1 + ... + Xp)
    ```

- **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:

    ```
    > 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`:

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

`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(~ X)`		✓		✓
`corAR1(~ 1)`		✓		
`corAR1(~ X)`		✓		✓

Table 2: This table gives an overview of the error structures.

2.5. Constraints on Coefficients

In order to achieve a more flexible framework we allow for constraints on the regression coefficients. These constraints can be specified by a vector or a matrix `coef.constraints`. First, a simple and less flexible way by specifying a vector `coef.constraints` of dimension J[1]. This vector is allocated in the following way: The first element of the vector `coef.constraints` gets a value of 1. If the coefficients of the repeated measurement $j = 2$ should be equal to the coefficients of the first dimension ($j = 1$) again a value of 1 is set. If the coefficients should be different to the coefficients of the first dimension a value of 2 is set. In analogy, if the coefficients of dimensions two and three should be the same one sets both values to 2 and if they should be different, a value of 3 is set. Constraints on the regression coefficients of the remaining repeated measurements are set analogously.

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

[1] 1 1 2 3
```

This vector `coef.constraints` sets the coefficients of the first two raters equal ($\beta_{1\cdot} = \beta_{2\cdot}$).

A more advanced way to specify constraints on the regression coefficients is a matrix with J rows, where each column specifies constraints on one of the p coefficients in the same way as above. In addition, a value of `NA` excludes a corresponding coefficient.

```
> coef.constraints <- cbind(c(1,2,3,4), c(1,1,1,2), c(NA,NA,NA,1),
+                           c(1,1,1,NA),c(1,2,3,4),c(1,2,3,4))
> coef.constraints

     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1   NA    1    1    1
[2,]    2    1   NA    1    2    2
[3,]    3    1   NA    1    3    3
[4,]    4    2    1   NA    4    4
```

[1]The ordering $j \in J$ of the responses is given by `response.names`

This matrix `coef.constraints` gives the following constraints:

- $\beta_{12} = \beta_{22} = \beta_{32}$
- $\beta_{13} = 0$
- $\beta_{23} = 0$
- $\beta_{33} = 0$
- $\beta_{44} = 0$
- $\beta_{14} = \beta_{24} = \beta_{34}$

Set specific values on coefficients

In addition, specific values on regression coefficients can be set in the matrix `coef.values`. Parameters are removed if the value is set to zero (default for `NA`'s in `coef.constraints`) or to some fixed value. If constraints on parameters are set, these dimensions need to have the same value in `coef.values`. Again each column corresponds to one regression coefficient. In the credit risk example, the constraints given by `coef.constraints` and `coef.values`

```
> coef.constraints <- cbind(c(1,2,3,4), c(1,1,1,2), c(NA,NA,NA,1),
+                           c(1,1,1,NA), c(1,2,3,4), c(1,2,3,4))
> coef.constraints

     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1   NA    1    1    1
[2,]    2    1   NA    1    2    2
[3,]    3    1   NA    1    3    3
[4,]    4    2    1   NA    4    4

> coef.values <- cbind(c(NA,NA,NA,NA), c(2,2,2,NA), c(0,0,0,NA),
+                      c(NA,NA,NA,0), c(NA,NA,NA,NA), c(NA,NA,NA,NA))
> coef.values

     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   NA    2    0   NA   NA   NA
[2,]   NA    2    0   NA   NA   NA
[3,]   NA    2    0   NA   NA   NA
[4,]   NA   NA   NA    0   NA   NA
```

result in the following model:

$$
\begin{aligned}
\widetilde{Y}_{i1} &= \beta_{11}x_{i1} + 2x_{i2} &&+ x_{i4} + \beta_{15}x_{i5} + \beta_{16}x_{i6},\\
\widetilde{Y}_{i2} &= \beta_{21}x_{i1} + 2x_{i2} &&+ x_{i4} + \beta_{25}x_{i5} + \beta_{26}x_{i6},\\
\widetilde{Y}_{i3} &= \beta_{31}x_{i1} + 2x_{i2} &&+ x_{i4} + \beta_{35}x_{i5} + \beta_{36}x_{i6},\\
\widetilde{Y}_{i4} &= \beta_{41}x_{i1} + \beta_{42}x_{i2} + \beta_{43}x_{i3} &&+ \beta_{45}x_{i5} + \beta_{46}x_{i6}.
\end{aligned}
$$

Note: Interaction terms
When constraints on the regression coefficient should be specified in models with interaction terms, the `coef.constraints` matrix has to be expanded manually. In case of interaction terms (specified either by `X1 + X2 + X1:X2` or equivalently by `X1*X2`), one additional column at the end of `coef.constraints` for the interaction term has to be specified for numerical variables. For interaction terms including factor variables suitably more columns have to be added to the `coef.constraints` matrix.

2.6. Constraints on threshold coefficients

Similarly, constraints on the threshold parameters can be imposed by a vector of positive integers, where dimensions with equal threshold parameters get 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 $J = 4$ different outcomes we impose:

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

gives the following restrictions:

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

Set specific threshold values

In addition, threshold parameter values can be specified by `threshold.values` in accordance with identifiability constraints. For this purpose we use a `list` with J elements, where each element specifies the constraints of the particular dimension by a vector of length of the number of threshold parameters (number of categories - 1). A number specifies a threshold parameter to a specific value and `NA` leaves the parameter flexible. All threshold parameters values need to be fixed according to the identifiablity constraints given in Table 1.3.2. These constraints have to be consistent across all dimensions. In the credit risk example we have

```
> threshold.constraints <- NULL
> threshold.constraints

NULL

> threshold.values <- list(c(-4,NA,NA,NA,4.5),
+                          c(-4,NA,NA,NA,4),
+                          c(-5,NA,NA,NA,NA,4.5),
+                          c(-2.5))
```

result in

- $\theta_{11} = -4 \leq \theta_{12} \leq \theta_{13} \leq \theta_{14} \leq \theta_{15} \leq \theta_{16} = 4.5$,

- $\theta_{21} = -4 \leq \theta_{22} \leq \theta_{23} \leq \theta_{24} \leq \theta_{25} \leq \theta_{26} = 4$,
- $\theta_{31} = -5 \leq \theta_{32} \leq \theta_{33} \leq \theta_{34} \leq \theta_{35} \leq \theta_{36} \leq \theta_{37} = 4.5$,
- $\theta_{41} = -2.5$.

2.7. Additional arguments

Starting values

Starting values can be chosen manually by:

`start.values = c(start.values.theta, start.values.beta, start.values.sigma).`

This parameter vector has the length of the number of parameters that have to be optimized (e.g., fixed threshold parameters are not included). If no starting values are passed to the function, they are set automatically.

Weights

Weights on each subject i can be chosen in a way that they are constant across repeated measurements. This is performed by the column name of the weights in `data`. 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).

Standard errors

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

2.8. Methods for class 'multord'

In addition, 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 `threshold` function to extract the threshold coefficients and a function `get.error.struct` to present the estimation of the error structure.

2.9. Output

The function `multord` returns an object of `class "multord"`.

The functions `summary` and `print` can be used to display the results. The function `coef` extracts the regression coefficients, a function `threshold` the threshold coefficients and the function `get.error.struct` represents the estimation of the corresponding error structure.

An object of class `"multord"` is a list containing the following components:

`beta`	a named `matrix` of regression coefficients
`theta`	a named `list` of threshold parameters
`sigmas`	a named `list` of correlation (covariance) matrices, or a ?vector of coefficients in the `error.structure = corEqui` 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
`sesigmas`	a named `list` of the standard errors of the correlation (covariance) matrices, or a vector of the standard errors of the coefficients in the `error.structure = corEqui` setting
`rho`	a `list` of all objects that are used in `multord`
`rho$optRes`	a `vector` of the optimizer output

2.10. Implementation CMOR

In addition, for a setting where the covariates do not vary between the repeated measurements j, a second function `CMOR`

```
CMOR(formula, data, link = c("probit", "logit"), error.structure =
     corGeneral(~1), weights = NULL, coef.constraints = NULL,
     coef.values = NULL, threshold.constraints = NULL,
     threshold.values = NULL, se = FALSE, start.values = NULL,
     solver = "newuoa")
```

is implemented with a slightly simplified data structure. In this setting the repeated ordinal observations as well as the covariates are stored as columns in a `data.frame` with corresponding names as labels. Each subject i corresponds to one row of the data frame, where all repeated observations $(Y_{i1}, \ldots, Y_{iJ})$ with all the covariates $(x_{i1}, \ldots, x_{ip})$ are stored in different columns.

```
> head(data_CMOR)
```

```
  firmID defaultInd rater1 rater2 rater3         X1          X2          X3
1      1 no default      B      B      B -1.7253621  0.23686361  0.68404628
2      2 no default     BB     BB      B  0.6910390  0.24387582 -0.03467597
3      3 no default      B      B      B -1.4880963  0.24746024 -0.31505879
4      4 no default    BBB    BBB      A -2.1109710  0.31692055 -0.08755990
5      5 no default      B      B      B -0.9454859 -0.25000228 -0.29887960
6      6 no default    BBB    BBB    Baa -0.2285612 -0.08547767  0.11796657
           X4         X5 X6
1 -0.85374252 -0.4447102  X
2 -1.13908602  1.6427284  Y
3  0.45284491  1.4863629  Y
4 -0.46870265 -1.3563313  Z
5 -1.39374063  1.1312291  X
6 -0.03109767 -1.0999933  X
```

In order to specify the responses as well as the covariates we use a multivariate formula object of the form

```
> formula <- cbind(rater1, rater2, rater3) ~ 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`, `start.values` and the `solver` can be chosen in the same way as in `multord`.

3. Examples

We use different examples in credit risk, where several raters assign ratings on different firms in order to explain some of the features of the R package **MultOrd**.

3.1. Example 1 - corGeneral

In the first setting, a simulated dataset where four different raters (`rater1, rater2` and`rater3`) assign ordinal ratings on different firms. `rater3` uses a different rating scale compared to `rater1` and `rater2`. In addition, a default indicator (`defaultInd`) is observed. The ID's for each subject i of the $n = 1000$ firms are stored in the column `firmID`. The ID's of the raters as well as the default information are stored in the column `raterID`. The ordinal ratings are provided in the column `rating` and all the covariates in the remaining columns.

```
> head(data_multord)
```

```
  firmID raterID rating         X1          X2          X3          X4
1      1  rater1      B -1.7253621  0.23686361  0.68404628 -0.85374252
2      2  rater1     BB  0.6910390  0.24387582 -0.03467597 -1.13908602
3      3  rater1      B -1.4880963  0.24746024 -0.31505879  0.45284491
4      4  rater1    BBB -2.1109710  0.31692055 -0.08755990 -0.46870265
5      5  rater1      B -0.9454859 -0.25000228 -0.29887960 -1.39374063
6      6  rater1    BBB -0.2285612 -0.08547767  0.11796657 -0.03109767
          X5 X6
1 -0.4447102  X
2  1.6427284  Y
3  1.4863629  Y
4 -1.3563313  Z
5  1.1312291  X
6 -1.0999933  X
```

```
> str(data_multord)
```

```
'data.frame':        4000 obs. of  9 variables:
 $ firmID : int  1 2 3 4 5 6 7 8 9 10 ...
 $ raterID: Factor w/ 4 levels "rater1","rater2",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ rating : Ord.factor w/ 14 levels "C"<"B"<"BB"<"BBB"<..: 2 3 2 4 2 4 6 2 4 1 ...
 $ X1     : num  -1.725 0.691 -1.488 -2.111 -0.945 ...
 $ X2     : num  0.237 0.244 0.247 0.317 -0.25 ...
 $ X3     : num  0.684 -0.0347 -0.3151 -0.0876 -0.2989 ...
```

```
 $ X4     : num  -0.854 -1.139 0.453 -0.469 -1.394 ...
 $ X5     : num  -0.445 1.643 1.486 -1.356 1.131 ...
 $ X6     : Factor w/ 3 levels "X","Y","Z": 1 2 2 3 1 1 3 2 2 3 ...
```

We include the covariates `X1, ..., X5` in a model without intercept by the `formula`:

```
> formula <- rating ~ 0 + X1 + X2 + X3 + X4 + X5
```

The subject index i (`firmID`) and the repeated measurement index j (`raterID`) are set by a character vector of length two in order to obtain a matching for the repeated observations:

```
> index <- c("firmID", "raterID")
```

We need to specify the raters which should be included in the model. The ordering of the vector `response.names` is essential when constraints on the parameter set want to be imposed:

```
> response.names <- c("rater1", "rater2", "rater3")
```

Due to the fact that the categories differ across raters we specify the `response.levels`. Otherwise the natural ordering is used and could lead to an incorrect labeling:

```
> response.levels <- list(c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                         c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                         c("D","C","B","Ba", "Baa", "A", "Aa", "Aaa"))
```

The general correlation error structure is chosen by `error.structure`:

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

We impose the following constraints on the regression coefficients by using the more advanced method:

```
> coef.constraints <- cbind(c(1,2,2),
+                           c(1,1,2),
+                           c(NA,1,2),
+                           c(NA,NA,NA),
+                           c(1,1,2))#either a vector or a matrix
> coef.constraints

     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1   NA   NA    1
[2,]    2    1    1   NA    1
[3,]    2    2    2   NA    2
```

Together with `coef.values`

```
> coef.values <- cbind(c(NA,NA,NA),
+                      c(NA,NA,NA),
+                      c(0,NA,NA),
+                      c(1,1,1),
+                      c(NA,NA,NA))
> coef.values
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]   NA   NA    0    1   NA
[2,]   NA   NA   NA    1   NA
[3,]   NA   NA   NA    1   NA
```

this gives the following model:

$$\begin{aligned}
\widetilde{Y}_{i1} &= \beta_{11}x_{i1}+\beta_{12}x_{i2} \qquad\qquad +x_{i4}+\beta_{15}x_{i5},\\
\widetilde{Y}_{i2} &= \beta_{21}x_{i1}+\beta_{12}x_{i2}+\beta_{23}x_{i3}+x_{i4}+\beta_{15}x_{i5},\\
\widetilde{Y}_{i3} &= \beta_{21}x_{i1}+\beta_{22}x_{i2}+\beta_{33}x_{i3}+x_{i4}+\beta_{35}x_{i5}.
\end{aligned}$$

In addition, we set the threshold parameters of `rater1` and `rater2` equal by:

```
> threshold.constraints <- c(1,1,2)
> threshold.constraints

[1] 1 1 2
```

As a link function we choose the probit link:

```
> link <- "probit"
```

```
> res_cor <- multord(
+   formula = rating ~ 0 + X1 + X2 + X3 + X4 + X5,
+   index = c("firmID", "raterID"),
+   data = data_multord,
+   response.levels = list(c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                          c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                          c("D","C","B","Ba", "Baa", "A", "Aa", "Aaa")),
+   response.names = c("rater1", "rater2", "rater3"),
+   link = "probit",
+   error.structure = corGeneral(~1),
+   weights = NULL,
+   coef.constraints = cbind(c(1,2,2),
+                            c(1,1,2),
+                            c(NA,1,2),
+                            c(NA,NA,NA),
+                            c(1,1,2)),
+   coef.values = cbind(c(NA,NA,NA),
+                       c(NA,NA,NA),
+                       c(0,NA,NA),
+                       c(1,1,1),
+                       c(NA,NA,NA)),
+   threshold.constraints = c(1,1,2),
+   threshold.values = NULL,
+   se = TRUE,
```

```
+   start.values = NULL,
+   solver = "newuoa")#,
>                       #PL.lag = NULL) #only for corAR1

> print(res_cor)

Call:
multord(formula = rating ~ 0 + X1 + X2 + X3 + X4 + X5, data = data_multord,
    index = c("firmID", "raterID"), response.levels = list(c("C",
        "B", "BB", "BBB", "A", "AA", "AAA"), c("C", "B", "BB",
        "BBB", "A", "AA", "AAA"), c("D", "C", "B", "Ba", "Baa",
        "A", "Aa", "Aaa")), response.names = c("rater1", "rater2",
        "rater4"), link = "probit", error.structure = corGeneral(~1),
    weights = NULL, coef.constraints = cbind(c(1, 2, 2), c(1,
        1, 2), c(NA, 1, 2), c(NA, NA, NA), c(1, 1, 2)), coef.values = cbind(c(NA,
        NA, NA), c(NA, NA, NA), c(0, NA, NA), c(1, 1, 1), c(NA,
        NA, NA)), threshold.constraints = c(1, 1, 2), threshold.values = NULL,
    se = TRUE, start.values = NULL, solver = "newuoa")

Threshold parameters:
$rater1
      C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-4.087598 -2.460453 -1.024200  1.034102  2.537427  4.286016

$rater2
      C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-4.087598 -2.460453 -1.024200  1.034102  2.537427  4.286016

$rater4
       D|C        C|B        B|Ba      Ba|Baa       Baa|A        A|Aa      Aa|Aaa
-4.7739978 -3.4516243 -1.8314461 -0.3928419  1.2299339  2.6060985  4.5595852

Coefficients:
              X1         X2         X3 X4         X5
rater1 0.8664923 -0.5439454  0.0000000  1 -0.6290721
rater2 0.6363396 -0.5439454  0.1840961  1 -0.6290721
rater4 0.6363396 -1.0715577 -0.2125230  1 -0.8363725

Sigma:
[[1]]
          rater1    rater2    rater4
rater1 1.0000000 0.8708983 0.8316842
rater2 0.8708983 1.0000000 0.6992996
rater4 0.8316842 0.6992996 1.0000000

> summary(res_cor)
```

```
Call: multord(formula = rating ~ 0 + X1 + X2 + X3 + X4 + X5, data = data_multord,
    index = c("firmID", "raterID"), response.levels = list(c("C",
        "B", "BB", "BBB", "A", "AA", "AAA"), c("C", "B", "BB",
        "BBB", "A", "AA", "AAA"), c("D", "C", "B", "Ba", "Baa",
        "A", "Aa", "Aaa")), response.names = c("rater1", "rater2",
        "rater4"), link = "probit", error.structure = corGeneral(~1),
    weights = NULL, coef.constraints = cbind(c(1, 2, 2), c(1,
        1, 2), c(NA, 1, 2), c(NA, NA, NA), c(1, 1, 2)), coef.values = cbind(c(NA,
        NA, NA), c(NA, NA, NA), c(0, NA, NA), c(1, 1, 1), c(NA,
        NA, NA)), threshold.constraints = c(1, 1, 2), threshold.values = NULL,
    se = TRUE, start.values = NULL, solver = "newuoa")

Formula: rating ~ 0 + X1 + X2 + X3 + X4 + X5

  link threshold nsubjects ndim  ?logPL CLIC-AIC CLIC-BIC fevals
probit  flexible      1000    3 5222.59 10546.34 10794.59   4529

Threshold parameters:
                Estimate Std. Error    z value      Pr(>|z|) signif
rater1 C|B    -4.0875983 0.14952030 -27.338082 1.496595e-164    ***
rater1 B|BB   -2.4604530 0.07171101 -34.310672 5.439940e-258    ***
rater1 BB|BBB -1.0242005 0.04918048 -20.825346  2.550758e-96    ***
rater1 BBB|A   1.0341018 0.04949681  20.892292  6.292617e-97    ***
rater1 A|AA    2.5374271 0.07150083  35.488081 7.506367e-276    ***
rater1 AA|AAA  4.2860159 0.17482517  24.516012 9.970426e-133    ***
rater2 C|B    -4.0875983 0.14952030 -27.338082 1.496595e-164    ***
rater2 B|BB   -2.4604530 0.07171101 -34.310672 5.439940e-258    ***
rater2 BB|BBB -1.0242005 0.04918048 -20.825346  2.550758e-96    ***
rater2 BBB|A   1.0341018 0.04949681  20.892292  6.292617e-97    ***
rater2 A|AA    2.5374271 0.07150083  35.488081 7.506367e-276    ***
rater2 AA|AAA  4.2860159 0.17482517  24.516012 9.970426e-133    ***
rater4 D|C    -4.7739978 0.17214974 -27.731658 2.900247e-169    ***
rater4 C|B    -3.4516243 0.12552235 -27.498086 1.850664e-166    ***
rater4 B|Ba   -1.8314461 0.06943612 -26.375985 2.584547e-153    ***
rater4 Ba|Baa -0.3928419 0.05275499  -7.446534  9.582401e-14    ***
rater4 Baa|A   1.2299339 0.05940148  20.705443  3.093755e-95    ***
rater4 A|Aa    2.6060985 0.08356957  31.184779 1.713729e-213    ***
rater4 Aa|Aaa  4.5595852 0.16147877  28.236437 2.088707e-175    ***

Coefficients:
            Estimate Std. Error    z value      Pr(>|z|) signif
X1 rater1  0.8664923 0.04046706  21.412287 1.026501e-101    ***
X1 rater2  0.6363396 0.03526199  18.046047  8.474259e-73    ***
X1 rater4  0.6363396 0.03526199  18.046047  8.474259e-73    ***
X2 rater1 -0.5439454 0.03383699 -16.075468  3.791192e-58    ***
X2 rater2 -0.5439454 0.03383699 -16.075468  3.791192e-58    ***
X2 rater4 -1.0715577 0.04287918 -24.990164 7.820415e-138    ***
```

```
X3 rater1  0.0000000 0.00000000         NA            NA   <NA>
X3 rater2  0.1840961 0.02727706   6.749119  1.487455e-11    ***
X3 rater4 -0.2125230 0.02839917  -7.483422  7.241174e-14    ***
X4 rater1  1.0000000 0.00000000         NA            NA   <NA>
X4 rater2  1.0000000 0.00000000         NA            NA   <NA>
X4 rater4  1.0000000 0.00000000         NA            NA   <NA>
X5 rater1 -0.6290721 0.03488752 -18.031438  1.103830e-72    ***
X5 rater2 -0.6290721 0.03488752 -18.031438  1.103830e-72    ***
X5 rater4 -0.8363725 0.03759956 -22.244208 1.283378e-109    ***

Correlation/Covariance Structure:
                     Estimate Std. Error  z value      Pr(>|z|) signif
corr 1 rater1 rater2 0.8708983 0.01535715 56.70964  0.000000e+00    ***
corr 1 rater1 rater4 0.8316842 0.01654426 50.27026  0.000000e+00    ***
corr 1 rater2 rater4 0.6992996 0.02379638 29.38681 8.095538e-190    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> thresholds(res_cor)

$rater1
      C|B      B|BB    BB|BBB     BBB|A      A|AA    AA|AAA
-4.087598 -2.460453 -1.024200  1.034102  2.537427  4.286016

$rater2
      C|B      B|BB    BB|BBB     BBB|A      A|AA    AA|AAA
-4.087598 -2.460453 -1.024200  1.034102  2.537427  4.286016

$rater4
       D|C        C|B       B|Ba     Ba|Baa      Baa|A       A|Aa     Aa|Aaa
-4.7739978 -3.4516243 -1.8314461 -0.3928419  1.2299339  2.6060985  4.5595852

> coefficients(res_cor)

             X1         X2         X3 X4         X5
rater1 0.8664923 -0.5439454  0.0000000  1 -0.6290721
rater2 0.6363396 -0.5439454  0.1840961  1 -0.6290721
rater4 0.6363396 -1.0715577 -0.2125230  1 -0.8363725

> get.error.struct(res_cor)

[[1]]
          rater1    rater2    rater4
rater1 1.0000000 0.8708983 0.8316842
rater2 0.8708983 1.0000000 0.6992996
rater4 0.8316842 0.6992996 1.0000000
```

3.2. Example 2 - covGeneral

In the second setting, we use the same dataset as in the example above.

We include the covariates `X1, ..., X5` in a model with intercept by the `formula`:

```
> formula <- rating ~ 1 + X1 + X2 + X3 + X4 + X5
```

The subject index i (`firmID`) and the repeated measurement index j (`raterID`) are set by a character vector of length two in order to obtain a matching for the repeated observations:

```
> index <- c("firmID", "raterID")
```

We need to specify the raters which should be included in the model. The ordering of the vector `response.names` is essential when constraints on the parameter set want to be imposed:

```
> response.names <- c("rater1", "rater2", "rater3")
```

Due to the fact that the categories differ across raters we specify the `response.levels`. Otherwise the natural ordering is used and could lead to an incorrect labeling:

```
> response.levels <- list(c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                         c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                         c("D","C","B","Ba", "Baa", "A", "Aa", "Aaa"))
```

The general covariance error structure is chosen by `error.structure`:

```
> error.structure <- covGeneral(~1)
```

In order to fix scale and location we fix the first and the last threshold for all raters by `threshold.values`:

```
> threshold.values = list(c(-4,NA,NA,NA,NA,4.5),
+                         c(-4,NA,NA,NA,NA,4),
+                         c(-5,NA,NA,NA,NA,NA,4.5))
> threshold.values

[[1]]
[1] -4.0   NA   NA   NA   NA  4.5

[[2]]
[1] -4 NA NA NA NA  4

[[3]]
[1] -5.0   NA   NA   NA   NA   NA  4.5
```

As a link function we choose the probit link:

```
> link <- "probit"
```

```
> res_cov <- multord(
+   formula = rating ~ 1 + X1 + X2 + X3 + X4 + X5,
+   index = c("firmID", "raterID"),
+   data = data_multord,
+   response.levels = list(c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                          c("C","B","BB", "BBB", "A", "AA", "AAA"),
+                          c("D","C","B","Ba", "Baa", "A", "Aa", "Aaa")),
+   response.names = c("rater1", "rater2", "rater3"),
+   link = "probit",
+   error.structure = covGeneral(~1),
+   weights = NULL,
+   coef.constraints = NULL,
+   coef.values = NULL,
+   threshold.constraints = NULL,
+   threshold.values = list(c(-4,NA,NA,NA,NA,4.5),
+                           c(-4,NA,NA,NA,NA,4),
+                           c(-5,NA,NA,NA,NA,NA,4.5)),
+   se = TRUE,
+   start.values = NULL,
+   solver = "newuoa")#, PL.lag = NULL)

> print(res_cov)

Call:
multord(formula = rating ~ 1 + X1 + X2 + X3 + X4 + X5, data = data_multord,
    index = c("firmID", "raterID"), response.levels = list(c("C",
        "B", "BB", "BBB", "A", "AA", "AAA"), c("C", "B", "BB",
        "BBB", "A", "AA", "AAA"), c("D", "C", "B", "Ba", "Baa",
        "A", "Aa", "Aaa")), response.names = c("rater1", "rater2",
        "rater4"), link = "probit", error.structure = covGeneral(~1),
    weights = NULL, coef.constraints = NULL, coef.values = NULL,
    threshold.constraints = NULL, threshold.values = list(c(-4,
        NA, NA, NA, NA, 4.5), c(-4, NA, NA, NA, NA, 4), c(-5,
        NA, NA, NA, NA, NA, 4.5)), se = TRUE, start.values = NULL,
    solver = "newuoa")

Threshold parameters:
$rater1
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-4.0000000 -2.4603748 -1.0864130  0.9347148  2.3598290  4.5000000

$rater2
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-4.0000000 -2.3029142 -0.8823006  1.1014094  2.6139308  4.0000000

$rater4
      D|C       C|B      B|Ba    Ba|Baa     Baa|A      A|Aa    Aa|Aaa
```

```
-5.000000 -3.657335 -2.014900 -0.554772  1.092901  2.488899  4.500000

Coefficients:
       (Intercept)        X1         X2          X3        X4         X5
rater1 -0.07475049 0.8486579 -0.5490828 -0.06274753 0.9782579 -0.6073374
rater2  0.10035091 0.6253225 -0.5128118  0.14435713 0.9500605 -0.6145800
rater4 -0.15455631 0.6403679 -1.0908901 -0.24903895 1.0019150 -0.8484080

Sigma:
[[1]]
          rater1    rater2    rater4
rater1 0.9286473 0.8338070 0.8249538
rater2 0.8338070 0.9731125 0.7147333
rater4 0.8249538 0.7147333 1.0552407

> summary(res_cov)

Call: multord(formula = rating ~ 1 + X1 + X2 + X3 + X4 + X5, data = data_multord,
    index = c("firmID", "raterID"), response.levels = list(c("C",
        "B", "BB", "BBB", "A", "AA", "AAA"), c("C", "B", "BB",
        "BBB", "A", "AA", "AAA"), c("D", "C", "B", "Ba", "Baa",
        "A", "Aa", "Aaa")), response.names = c("rater1", "rater2",
        "rater4"), link = "probit", error.structure = covGeneral(~1),
    weights = NULL, coef.constraints = NULL, coef.values = NULL,
    threshold.constraints = NULL, threshold.values = list(c(-4,
        NA, NA, NA, NA, 4.5), c(-4, NA, NA, NA, NA, 4), c(-5,
        NA, NA, NA, NA, NA, 4.5)), se = TRUE, start.values = NULL,
    solver = "newuoa")

Formula: rating ~ 1 + X1 + X2 + X3 + X4 + X5

  link     threshold nsubjects ndim  ?logPL CLIC-AIC CLIC-BIC fevals
probit fix2firstlast      1000    3 5202.61 10662.98 11295.45   5229

Threshold parameters:
                Estimate Std. Error   z value      Pr(>|z|) signif
rater1 C|B    -4.0000000 0.00000000        NA            NA   <NA>
rater1 B|BB   -2.4603748 0.03892671 -63.20531  0.000000e+00    ***
rater1 BB|BBB -1.0864130 0.04672864 -23.24940 1.442399e-119    ***
rater1 BBB|A   0.9347148 0.05816758  16.06934  4.184869e-58    ***
rater1 A|AA    2.3598290 0.06618179  35.65677 1.850351e-278    ***
rater1 AA|AAA  4.5000000 0.00000000        NA            NA   <NA>
rater2 C|B    -4.0000000 0.00000000        NA            NA   <NA>
rater2 B|BB   -2.3029142 0.05017519 -45.89747  0.000000e+00    ***
rater2 BB|BBB -0.8823006 0.04900175 -18.00549  1.764301e-72    ***
rater2 BBB|A   1.1014094 0.04690720  23.48060 6.438273e-122    ***
rater2 A|AA    2.6139308 0.04402913  59.36821  0.000000e+00    ***
```

```
rater2 AA|AAA  4.0000000 0.00000000        NA            NA   <NA>
rater4 D|C    -5.0000000 0.00000000        NA            NA   <NA>
rater4 C|B    -3.6573348 0.04122828 -88.70937  0.000000e+00    ***
rater4 B|Ba   -2.0148999 0.04216189 -47.78960  0.000000e+00    ***
rater4 Ba|Baa -0.5547720 0.04053295 -13.68694  1.215190e-42    ***
rater4 Baa|A   1.0929009 0.03897605  28.04032 5.242063e-173    ***
rater4 A|Aa    2.4888990 0.03760398  66.18712  0.000000e+00    ***
rater4 Aa|Aaa  4.5000000 0.00000000        NA            NA   <NA>

Coefficients:
                     Estimate Std. Error     z value     Pr(>|z|) signif
(Intercept) rater1 -0.07475049 0.45190111  -0.1654134 8.686186e-01
(Intercept) rater2  0.10035091 0.39042769   0.2570282 7.971570e-01
(Intercept) rater4 -0.15455631 0.38842057  -0.3979097 6.906967e-01
X1 rater1           0.84865788 0.06468221  13.1204227 2.515279e-39    ***
X1 rater2           0.62532250 0.04528665  13.8080977 2.277670e-43    ***
X1 rater4           0.64036795 0.04490726  14.2597865 3.896450e-46    ***
X2 rater1          -0.54908279 0.04937110 -11.1215430 9.855265e-29    ***
X2 rater2          -0.51281183 0.04182682 -12.2603585 1.478443e-34    ***
X2 rater4          -1.09089014 0.05569826 -19.5857125 2.047224e-85    ***
X3 rater1          -0.06274753 0.03329500  -1.8845929 5.948483e-02      .
X3 rater2           0.14435713 0.03696277   3.9054740 9.404088e-05    ***
X3 rater4          -0.24903895 0.03727961  -6.6802984 2.384563e-11    ***
X4 rater1           0.97825786 0.07470180  13.0955066 3.493357e-39    ***
X4 rater2           0.95006048 0.05758732  16.4977378 3.808971e-61    ***
X4 rater4           1.00191502 0.05230472  19.1553466 8.734194e-82    ***
X5 rater1          -0.60733736 0.05489198 -11.0642286 1.870655e-28    ***
X5 rater2          -0.61457996 0.04530642 -13.5649630 6.461230e-42    ***
X5 rater4          -0.84840800 0.04683338 -18.1154564 2.406685e-73    ***

Correlation/Covariance Structure:
                     Estimate Std. Error  z value      Pr(>|z|) signif
corr 1 rater1 rater2 0.8771192 0.01550348 56.57564  0.000000e+00    ***
corr 1 rater1 rater4 0.8333520 0.01709118 48.75917  0.000000e+00    ***
corr 1 rater2 rater4 0.7053209 0.02411848 29.24400 5.353269e-188    ***
sigma 1 rater1       0.9636635 0.06708384 14.36506  8.572924e-47    ***
sigma 1 rater2       0.9864646 0.05097531 19.35181  1.968018e-83    ***
sigma 1 rater4       1.0272491 0.04842874 21.21156 7.469805e-100    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> thresholds(res_cov)

$rater1
       C|B       B|BB      BB|BBB       BBB|A        A|AA      AA|AAA
-4.0000000 -2.4603748 -1.0864130  0.9347148  2.3598290  4.5000000
```

```
$rater2
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-4.0000000 -2.3029142 -0.8823006  1.1014094  2.6139308  4.0000000

$rater4
      D|C       C|B      B|Ba    Ba|Baa     Baa|A      A|Aa    Aa|Aaa
-5.000000 -3.657335 -2.014900 -0.554772  1.092901  2.488899  4.500000

> coefficients(res_cov)

       (Intercept)        X1         X2          X3        X4         X5
rater1 -0.07475049 0.8486579 -0.5490828 -0.06274753 0.9782579 -0.6073374
rater2  0.10035091 0.6253225 -0.5128118  0.14435713 0.9500605 -0.6145800
rater4 -0.15455631 0.6403679 -1.0908901 -0.24903895 1.0019150 -0.8484080

> get.error.struct(res_cov)

[[1]]
          rater1    rater2    rater4
rater1 0.9286473 0.8338070 0.8249538
rater2 0.8338070 0.9731125 0.7147333
rater4 0.8249538 0.7147333 1.0552407
```

3.3. Example 3 - AR1

In a third setting, only one rater assigns ratings over the years 2001 to 2010 for a set of firms. The ID's for each subject i of the $n = 1000$ firms are stored in the column `firmID`. The year of the rating observation is stored in the column `year`. The ordinal ratings are provided in the column `rating` and all the covariates in the remaining columns.

```
> head(data_multord_panel)

  firmID year rating          X1         X2          X3          X4         X5
1      1 2001     BB  0.84075950  1.5268907 -1.17083386 -0.08391745  0.7046638
2      2 2001    BBB  1.06122627 -0.7825659 -0.86175032  0.29414138  0.8323862
3      3 2001    BBB -0.04015277 -1.8660206  1.57883027  1.11663741 -0.3892742
4      4 2001     AA -0.17157274 -1.1979696 -0.01585399 -0.67814327 -1.1806786
5      5 2001     AA  1.33023651 -1.0663252  0.58567481  1.54923638 -2.5318378
6      6 2001    BBB  0.97805496 -0.3829815  0.35293911  0.14177743  0.9379203
  X6
1  X
2  Y
3  Y
4  Z
5  X
6  X
```

```
> str(data_multord_panel)

'data.frame':        10000 obs. of  9 variables:
 $ firmID: int  1 2 3 4 5 6 7 8 9 10 ...
 $ year  : int  2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
 $ rating: Factor w/ 7 levels "A","AA","AAA",..: 5 6 6 2 2 6 6 6 5 6 ...
 $ X1    : num  0.8408 1.0612 -0.0402 -0.1716 1.3302 ...
 $ X2    : num  1.527 -0.783 -1.866 -1.198 -1.066 ...
 $ X3    : num  -1.1708 -0.8618 1.5788 -0.0159 0.5857 ...
 $ X4    : num  -0.0839 0.2941 1.1166 -0.6781 1.5492 ...
 $ X5    : num  0.705 0.832 -0.389 -1.181 -2.532 ...
 $ X6    : Factor w/ 3 levels "X","Y","Z": 1 2 2 3 1 1 3 2 2 3 ...
```

We include the covariates `X1, ..., X5` in a model without intercept by the `formula`:

```
> formula <- rating ~ 0 + X1 + X2 + X3 + X4 + X5
```

The subject index i (`firmID`) and the repeated measurement index j (`year`) are set by a character vector of length two in order to obtain a matching for the repeated observations:

```
> index <- c("firmID", "year")
```

We need to specify the raters which should be included in the model. The ordering of the vector `response.names` is essential when constraints on the parameter set are desired:

```
> response.names <- c(2001:2005)
```

The categories do not vary over years. Nevertheless we define `response.levels` by:

```
> response.levels <- rep(list(c("C","B","BB", "BBB", "A", "AA", "AAA")), 5)
```

The error structure is chosen by `error.structure` and depends on the factor variable `X6`:

```
> error.structure <- corAR1(~X6)
```

We impose no constraints on the parameter set. As a link function we choose the probit link:

```
> link <- "probit"
```

```
> res_AR1 <- multord(
+   formula = rating ~ 0 + X1 + X2 + X3 + X4 + X5,
+   index = c("firmID", "year"),
+   data = data_multord_panel,
+   response.levels = rep(list(c("C","B","BB", "BBB", "A", "AA", "AAA")), 5),
+   response.names = c(2001:2005),
+   link = "probit",
+   error.structure = corAR1(~1),
+   weights = NULL,
```

```
+   coef.constraints = NULL,
+   coef.values = NULL,
+   threshold.constraints = NULL,
+   threshold.values = NULL,
+   se = TRUE,
+   start.values = NULL,
+   solver = "newuoa")#, PL.lag = NULL)

> print(res_AR1)

Call:
multord(formula = rating ~ 0 + X1 + X2 + X3 + X4 + X5, data = data_multord_panel,
    index = c("firmID", "year"), response.levels = rep(list(c("C",
        "B", "BB", "BBB", "A", "AA", "AAA")), 5), response.names = c(2001:2005),
    link = "probit", error.structure = corAR1(~1), weights = NULL,
    coef.constraints = NULL, coef.values = NULL, threshold.constraints = NULL,
    threshold.values = NULL, se = TRUE, start.values = NULL,
    solver = "newuoa")

Threshold parameters:
$`2001`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.9336318 -2.5731420 -0.9394342  1.0388899  2.5337040  4.4255617

$`2002`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.9391016 -2.4535215 -0.9942456  0.9679743  2.5936608  4.5869266

$`2003`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.8590243 -2.4322687 -0.9785237  0.9936912  2.6374055  4.2132439

$`2004`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.9050425 -2.4790889 -0.9855307  0.9922006  2.5947111  4.8000279

$`2005`
      C|B      B|BB    BB|BBB     BBB|A      A|AA    AA|AAA
-3.843655 -2.454172 -1.018689  1.044926  2.534620  4.334503

Coefficients:
            X1         X2           X3        X4         X5
2001 0.7892333 -0.4583754 -0.021555494 1.0191690 -0.5595444
2002 0.7857250 -0.4887868 -0.023095750 0.9621466 -0.6531238
2003 0.7542494 -0.5321538  0.025989752 0.9680971 -0.6179504
2004 0.7965603 -0.4586160 -0.007481102 1.0068089 -0.5968277
2005 0.7978689 -0.5176147 -0.001121708 0.9974046 -0.5792169
```

```
correlation parameter:
[1] 0.8877651

> summary(res_AR1)

Call: multord(formula = rating ~ 0 + X1 + X2 + X3 + X4 + X5, data = data_multord_panel,
    index = c("firmID", "year"), response.levels = rep(list(c("C",
        "B", "BB", "BBB", "A", "AA", "AAA")), 5), response.names = c(2001:2005),
    link = "probit", error.structure = corAR1(~1), weights = NULL,
    coef.constraints = NULL, coef.values = NULL, threshold.constraints = NULL,
    threshold.values = NULL, se = TRUE, start.values = NULL,
    solver = "newuoa")

Formula: rating ~ 0 + X1 + X2 + X3 + X4 + X5

  link threshold nsubjects ndim   ?logPL CLIC-AIC CLIC-BIC fevals
probit  flexible      1000    5 16852.63 34106.26 35090.28  10525

Threshold parameters:
              Estimate Std. Error   z value      Pr(>|z|) signif
2001 C|B    -3.9336318 0.17113736 -22.98523 6.550120e-117    ***
2001 B|BB   -2.5731420 0.09568784 -26.89100 2.798392e-159    ***
2001 BB|BBB -0.9394342 0.05302327 -17.71740  3.078410e-70    ***
2001 BBB|A   1.0388899 0.05856469  17.73918  2.089450e-70    ***
2001 A|AA    2.5337040 0.09924243  25.53045 9.052971e-144    ***
2001 AA|AAA  4.4255617 0.22315899  19.83143  1.594458e-87    ***
2002 C|B    -3.9391016 0.14837610 -26.54809 2.702234e-155    ***
2002 B|BB   -2.4535215 0.08883700 -27.61824 6.720563e-168    ***
2002 BB|BBB -0.9942456 0.05586330 -17.79783  7.345822e-71    ***
2002 BBB|A   0.9679743 0.05818189  16.63704  3.757484e-62    ***
2002 A|AA    2.5936608 0.09472400  27.38124 4.587403e-165    ***
2002 AA|AAA  4.5869266 0.20118173  22.79992 4.593939e-115    ***
2003 C|B    -3.8590243 0.15119439 -25.52359 1.078751e-143    ***
2003 B|BB   -2.4322687 0.09349702 -26.01440 3.403480e-149    ***
2003 BB|BBB -0.9785237 0.05376981 -18.19839  5.315052e-74    ***
2003 BBB|A   0.9936912 0.05590902  17.77336  1.136700e-70    ***
2003 A|AA    2.6374055 0.09540298  27.64490 3.214199e-168    ***
2003 AA|AAA  4.2132439 0.17494008  24.08393 3.684373e-128    ***
2004 C|B    -3.9050425 0.15436975 -25.29668 3.474357e-141    ***
2004 B|BB   -2.4790889 0.09876975 -25.09968 5.013746e-139    ***
2004 BB|BBB -0.9855307 0.05601980 -17.59254  2.809868e-69    ***
2004 BBB|A   0.9922006 0.05589276  17.75186  1.667345e-70    ***
2004 A|AA    2.5947111 0.09378626  27.66622 1.780970e-168    ***
2004 AA|AAA  4.8000279 0.29586610  16.22365  3.431809e-59    ***
2005 C|B    -3.8436554 0.15791622 -24.33984 7.427430e-131    ***
2005 B|BB   -2.4541724 0.09268645 -26.47822 1.727065e-154    ***
```

```
2005 BB|BBB -1.0186893 0.05740157 -17.74671  1.827403e-70    ***
2005 BBB|A   1.0449255 0.05841526  17.88789  1.465638e-71    ***
2005 A|AA    2.5346204 0.09209870  27.52070 9.927624e-167    ***
2005 AA|AAA  4.3345025 0.19144663  22.64079 1.719411e-113    ***

Coefficients:
           Estimate Std. Error      z value      Pr(>|z|) signif
X1 2001  0.789233254 0.03581301  22.03761367 1.255842e-107    ***
X1 2002  0.785724954 0.03363039  23.36353669 1.004004e-120    ***
X1 2003  0.754249391 0.03229485  23.35509620 1.223261e-120    ***
X1 2004  0.796560323 0.03257701  24.45160658 4.838222e-132    ***
X1 2005  0.797868905 0.03404946  23.43264705 1.986924e-121    ***
X2 2001 -0.458375436 0.02936277 -15.61076768  6.149261e-55    ***
X2 2002 -0.488786814 0.02746912 -17.79404875  7.859060e-71    ***
X2 2003 -0.532153823 0.02940075 -18.10001038  3.186077e-73    ***
X2 2004 -0.458615964 0.02901651 -15.80534297  2.858305e-56    ***
X2 2005 -0.517614650 0.02886916 -17.92967426  6.918381e-72    ***
X3 2001 -0.021555494 0.02497065  -0.86323316  3.880093e-01
X3 2002 -0.023095750 0.02384011  -0.96877685  3.326565e-01
X3 2003  0.025989752 0.02287500   1.13616396  2.558880e-01
X3 2004 -0.007481102 0.02442595  -0.30627677  7.593939e-01
X3 2005 -0.001121708 0.02719104  -0.04125285  9.670943e-01
X4 2001  1.019169030 0.03862944  26.38322191 2.134791e-153    ***
X4 2002  0.962146578 0.03721278  25.85527350 2.122572e-147    ***
X4 2003  0.968097093 0.03717830  26.03930453 1.778234e-149    ***
X4 2004  1.006808937 0.03815041  26.39051555 1.760569e-153    ***
X4 2005  0.997404637 0.03950946  25.24470459 1.294840e-140    ***
X5 2001 -0.559544357 0.03061882 -18.27452178  1.320423e-74    ***
X5 2002 -0.653123828 0.03179265 -20.54323237  8.847387e-94    ***
X5 2003 -0.617950429 0.02978103 -20.74979759  1.231086e-95    ***
X5 2004 -0.596827706 0.03045951 -19.59413038  1.735249e-85    ***
X5 2005 -0.579216877 0.03061364 -18.92021932  7.773038e-80    ***

Correlation/Covariance Structure:
     Estimate  Std. Error z value Pr(>|z|) signif
corr 0.8877651 0.007219067 122.975        0    ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> thresholds(res_AR1)

$`2001`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.9336318 -2.5731420 -0.9394342  1.0388899  2.5337040  4.4255617

$`2002`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
```

```
-3.9391016 -2.4535215 -0.9942456  0.9679743  2.5936608  4.5869266

$`2003`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.8590243 -2.4322687 -0.9785237  0.9936912  2.6374055  4.2132439

$`2004`
       C|B       B|BB     BB|BBB      BBB|A       A|AA     AA|AAA
-3.9050425 -2.4790889 -0.9855307  0.9922006  2.5947111  4.8000279

$`2005`
      C|B      B|BB    BB|BBB     BBB|A      A|AA    AA|AAA
-3.843655 -2.454172 -1.018689  1.044926  2.534620  4.334503
```

```
> coefficients(res_AR1)

            X1         X2           X3        X4         X5
2001 0.7892333 -0.4583754 -0.021555494 1.0191690 -0.5595444
2002 0.7857250 -0.4887868 -0.023095750 0.9621466 -0.6531238
2003 0.7542494 -0.5321538  0.025989752 0.9680971 -0.6179504
2004 0.7965603 -0.4586160 -0.007481102 1.0068089 -0.5968277
2005 0.7978689 -0.5176147 -0.001121708 0.9974046 -0.5792169
```

```
> get.error.struct(res_AR1)

(Intercept)
   1.411277
```

```
> head(get.error.struct(res_AR1, type = "r"))

       [,1]
1 0.8877651
2 0.8877651
3 0.8877651
4 0.8877651
5 0.8877651
6 0.8877651
```

```
> head(get.error.struct(res_AR1, type = "sigmas"))

[[1]]
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8877651 0.7881268 0.6996714 0.6211439
[2,] 0.8877651 1.0000000 0.8877651 0.7881268 0.6996714
[3,] 0.7881268 0.8877651 1.0000000 0.8877651 0.7881268
[4,] 0.6996714 0.7881268 0.8877651 1.0000000 0.8877651
```

```
[5,] 0.6211439 0.6996714 0.7881268 0.8877651 1.0000000

[[2]]
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8877651 0.7881268 0.6996714 0.6211439
[2,] 0.8877651 1.0000000 0.8877651 0.7881268 0.6996714
[3,] 0.7881268 0.8877651 1.0000000 0.8877651 0.7881268
[4,] 0.6996714 0.7881268 0.8877651 1.0000000 0.8877651
[5,] 0.6211439 0.6996714 0.7881268 0.8877651 1.0000000

[[3]]
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8877651 0.7881268 0.6996714 0.6211439
[2,] 0.8877651 1.0000000 0.8877651 0.7881268 0.6996714
[3,] 0.7881268 0.8877651 1.0000000 0.8877651 0.7881268
[4,] 0.6996714 0.7881268 0.8877651 1.0000000 0.8877651
[5,] 0.6211439 0.6996714 0.7881268 0.8877651 1.0000000

[[4]]
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8877651 0.7881268 0.6996714 0.6211439
[2,] 0.8877651 1.0000000 0.8877651 0.7881268 0.6996714
[3,] 0.7881268 0.8877651 1.0000000 0.8877651 0.7881268
[4,] 0.6996714 0.7881268 0.8877651 1.0000000 0.8877651
[5,] 0.6211439 0.6996714 0.7881268 0.8877651 1.0000000

[[5]]
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8877651 0.7881268 0.6996714 0.6211439
[2,] 0.8877651 1.0000000 0.8877651 0.7881268 0.6996714
[3,] 0.7881268 0.8877651 1.0000000 0.8877651 0.7881268
[4,] 0.6996714 0.7881268 0.8877651 1.0000000 0.8877651
[5,] 0.6211439 0.6996714 0.7881268 0.8877651 1.0000000

[[6]]
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8877651 0.7881268 0.6996714 0.6211439
[2,] 0.8877651 1.0000000 0.8877651 0.7881268 0.6996714
[3,] 0.7881268 0.8877651 1.0000000 0.8877651 0.7881268
[4,] 0.6996714 0.7881268 0.8877651 1.0000000 0.8877651
[5,] 0.6211439 0.6996714 0.7881268 0.8877651 1.0000000
```

References

Agresti A (2010). *Analysis of ordinal categorical data*. Wiley Series in Probability and Statistics, 2 edition. John Wiley & Sons.

Azzalini A, Genz A (2016). *The R package `mnormt`: The multivariate normal and t distributions (version 1.5-4)*.

Bhat CR, Varin C, Ferdous N (2010). "A comparison of the maximum simulated likelihood and composite marginal likelihood estimation approaches in the context of the multivariate ordered-response model." In W Greene, RC Hill (eds.), *Maximum Simulated Likelihood Methods and Applications*, volume 26 of *Advances in Econometrics*, pp. 65–106. Emerald Group Publishing Limited.

Christensen RHB (2015). "ordinal—Regression Models for Ordinal Data." R package version 2015.6-28. http://www.cran.r-project.org/package=ordinal/.

Fahrmeir L, Tutz G, Hennevogl W (2001). *Multivariate statistical modelling based on generalized linear models*. Springers series in statistics. Springer, New York.

Greene WH, Hensher DA (2010). *Modeling Ordered Choices: A Primer*. Cambridge University Press.

Nash JC (2014). "On Best Practice Optimization Methods in R." *Journal of Statistical Software*, **60**(2), 1–14.

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

O'Brien SM, Dunson DB (2004). "Bayesian multivariate logistic regression." *Biometrics*, **60**(3), 739–746.

Pagui K, Clovis E, Canale A (2015). "Pairwise likelihood inference for multivariate ordinal responses with applications to customer satisfaction." *Applied Stochastic Models in Business and Industry*.

Pinheiro JC, Bates DM (1996). "Unconstrained parametrizations for variance-covariance matrices." *Statistics and Computing*, **6**(3), 289–296.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Scott DM, Kanaroglou PS (2002). "An activity-episode generation model that captures interactions between household heads: development and empirical analysis." *Transportation Research Part B: Methodological*, **36**(10), 875–896.

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), 1–28.

Varin C, Reid N, Firth D (2011). "An Overview of Composite Likelihood Methods." *Statistica Sinica*, **21**(1), 5–42.

Affiliation:

Rainer Hirk
Institute for Statistics and Mathematics
WU Vienna University of Economics and Business
1020 Vienna, Austria
E-mail: rainer.hirk@wu.ac.at