

Design of ordinal CRM

John Kirkpatrick and Wojciech Wójciak

Last run on 2023-06-26 by kirkpatj on
rstudio-deployment-kirkpatj-p5571g-865bf846c6-jxvp6

1 The statistical model and theory about it.

The patient's toxicity response is not binary (no DLT, DLT) but categorical (for example: no DLT, sub-DLT, DLT). Furthermore, multicategory toxicity responses are treated as ordinal due to their ordinal characteristics such as a monotone trend.

1.1 The model: Cumulative Link Models

These type of responses can be modeled using Generalized Linear Model (GLM), where the model is expressed in terms of the cumulative response probabilities. Hence, the name: Cumulative Link Model, or CLM. The CLM assumes the form:

$$h[p_{ij}] = h[P(Y_i \leq j)] = \alpha_j + \mathbf{x}_i^T \boldsymbol{\beta}, \quad j = 1, \dots, c-1; i \geq 1$$

where $c \in \{2, 3, \dots\}$ denotes a number of toxicity categories on ordinal scale, and the components of the model are as follows:

1. h - link function. The most common link functions are “logit” and “probit”, which result in Cumulative Logit Model and Cumulative Probit Model respectively.
2. Y_i - random variable that represents the response outcome category for subject $i \geq 1$, i.e. $\text{supp}(Y_i) = \{1, 2, \dots, c\}$, $i \geq 1$. Probabilities $P(Y_i = j)$, $j = 1, \dots, c$, for $i \geq 1$, are called category probabilities, and the cumulative response probabilities for subject $i \geq 1$ are defined as

$$p_{ij} = P(Y_i \leq j) = P(Y_i = 1) + \dots + P(Y_i = j), \quad j = 1, \dots, c; i \geq 1,$$

3. \mathbf{x}_i - a vector of covariates corresponding to main effects and interactions for subject $i \geq 1$.
4. α_j , $j = 1, \dots, c-1$ are “cutpoints” and $\boldsymbol{\beta}$ is a vector of effects (without intercept term). Although each response category has a corresponding cutpoint, the regression coefficients $\boldsymbol{\beta}$ are constant across categories. Each cumulative logs has its own intercept (“cutpoint”). Furthermore, if link function h is strictly increasing function of $P(Y_i \leq j)$, $i \geq 1$ (such as logit or probit), then

$$-\infty < \alpha_1 < \alpha_2 < \dots < \alpha_{c-1} < \infty, \quad (1)$$

because $P(Y_i \leq j)$ increases in j at any fixed \mathbf{x}_i .

Note that a model for $\text{logit}[p_{ij}]$ alone is an ordinary logistic model for a binary response in which categories 1 to j represent “success” and categories $j+1$ to c represent “failure”.

1.1.1 Why cumulative probabilities rather than category probabilities?

As explained by McCullagh and Nelder (1989), the choice and definition of response categories is often arbitrary. It is essential, therefore, if we are to arrive at valid conclusions, that the nature of those conclusions should not be affected by the number of choice of response categories. Such considerations lead to models that based on the cumulative probabilities rather than category probabilities. These two sets of probabilities are equivalent, but simple models for the cumulative probabilities are likely to have better properties for ordinal response scales than equally simple models based on the category probabilities.

1.1.2 Backward compatible CLM model

In current version of the `crmPack`, we support only two toxicity categories, i.e. no DLT and DLT. We code it as 0 (no DLT) and 1 (DLT) and we compute the probability of the DLT, i.e. $0 < P(DLT) < 1$. To preserve this schema for multi-category toxicities, we propose the following form of the CLM model:

$$h[p_{ij}] = h[P(Y_i \geq j)] = \alpha_j + \mathbf{x}_i^T \boldsymbol{\beta}, \quad j = 0, \dots, c-1, i \geq 1, \quad (2)$$

where $c \in \mathbb{N}$, such that $c \geq 2$, denotes a number of toxicity categories on ordinal scale. This yields:

$$\infty > \alpha_0 > \dots > \alpha_{c-1} > -\infty. \quad (3)$$

This convention directly simplifies to existing `crmPack` coding for $c = 2$, i.e.

$$P(DLT) = P(Y \geq 1) = P(Y = 1)$$

Also, since $P(Y \geq 0) = 1$, we are not interested in specifying α_0 .

1.2 The prior

Specifying prior for $\boldsymbol{\beta}$ is straightforward. Let $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_{c-1}]^T$ be a random vector, where $c \geq 2$ denotes a number of toxicity categories. Specifying prior for $\boldsymbol{\alpha}$ is challenging because of the ordering restriction (3) (or (1)). In the following sections we present most popular approaches that are discussed in currently available subject domain literature, including our own comments and considerations.

1.2.1 Truncated prior distribution (we propose to implement this approach in `crmPack`)

We specify this prior for model (2).

$$\begin{aligned} \alpha_1 &\sim \mathcal{N}(\mu_1, \sigma_1^2), \\ \alpha_j | \alpha_{j-1} &\sim \mathcal{N}(\mu_j, \sigma_j^2) T(-\infty, \alpha_{j-1}), \quad j = 2, \dots, c-1 \geq 2, \end{aligned} \quad (4)$$

where $\mu_j \in \mathbb{R}$, $\sigma_j^2 \in \mathbb{R}_+$. This yields $\text{supp}(\alpha_j) = (-\infty, \alpha_{j-1})$, $j = 2, \dots, c-1 \geq 2$, ensuring (3).

See McKinley et al. (2013) (equation (11)) and the references given therein, i.e. Albert and Chib (1993), Johnson and Albert (1999), Congdon (2005). Eventually see James et al. (2022)

However, we see the following issues (drawbacks) with this model:

1. If α_k for some $k \in \{1, \dots, c-2\}$ is small by a chance, then it forces that α_{k+1} will be even smaller, regardless of whether it is practically justified. This may become an issue, especially if the number of categories c is high. So the alternative could be doubly-truncated distributions, which is somehow given in Congdon (2005), but it is unclear to me how this should be used exactly. John's experience is that it has not been an issue in the models he has fitted to date.

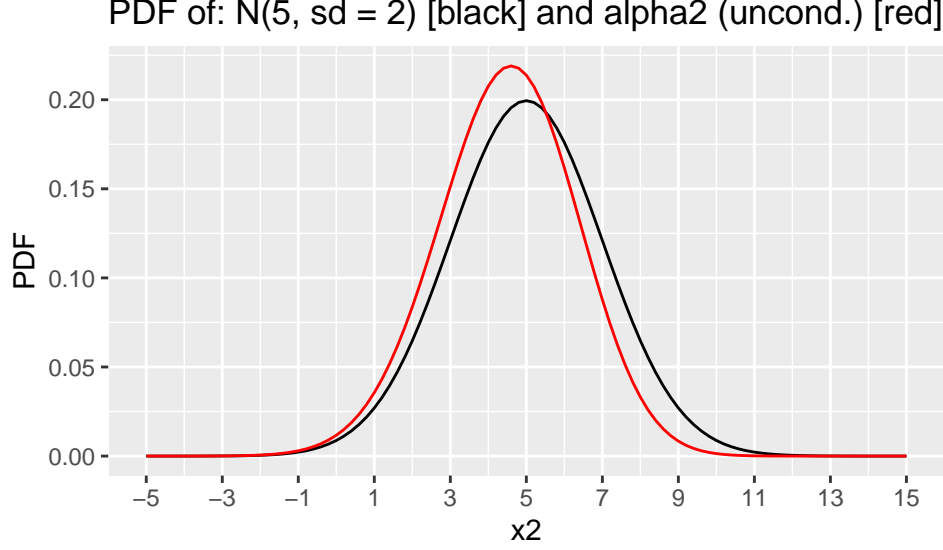
2. It might not be entirely clear for the user what the exact (unconditional) distributions of $\alpha_2, \dots, \alpha_{c-1}$ are, as well as what the var-cov structure for α is. The user can obviously work it out, but to do that, he needs to know about conditional probability distributions tools and methods and it is still not that trivial. For instance, see just the beginning of the computations for a bivariate α .

$$\begin{aligned}
 f_{\alpha_2}(x_2) &= \int_{\mathbb{R}} f_{\alpha_1, \alpha_2}(x_1, x_2) dx_1 \\
 &= \int_{\mathbb{R}} f_{\alpha_2|\alpha_1=x_1}(x_2) f_{\alpha_1}(x_1) dx_1 \\
 &= \int_{\mathbb{R}} \frac{1}{\sigma_2} \frac{\varphi(\frac{x_2-\mu_2}{\sigma_2})}{\Phi(\frac{x_1-\mu_2}{\sigma_2})} I(x_2 < x_1) \frac{1}{\sigma_1} \varphi(\frac{x_1-\mu_1}{\sigma_1}) dx_1 \\
 &= \frac{1}{\sigma_1 \sigma_2} \varphi(\frac{x_2-\mu_2}{\sigma_2}) \int_{x_2}^{\infty} \frac{\varphi(\frac{x_1-\mu_1}{\sigma_1})}{\Phi(\frac{x_1-\mu_2}{\sigma_2})} dx_1 \\
 &= \dots
 \end{aligned}$$

for $x_2 \in \mathbb{R}$, where φ is the probability density function of the standard normal distribution and Φ is its cumulative distribution function. Below is the example shape of f_{α_2} function.

```
f <- function(x2, mu1 = 8, mu2 = 5, sd1 = 2, sd2 = 2) {
  phi <- dnorm
  Phi <- pnorm
  to_integrate <- function(x1, mu1, mu2, sd1, sd2) {
    phi((x1 - mu1) / sd1) / Phi((x1 - mu2) / sd2)
  }
  intgr1 <- integrate(
    to_integrate,
    lower = x2, upper = Inf,
    mu1 = mu1, mu2 = mu2, sd1 = sd1, sd2 = sd2
  )
  1 / (sd1 * sd2) * phi((x2 - mu2) / sd2) * intgr1$value
}
f <- Vectorize(f)

ggplot() +
  geom_function(fun = dnorm, colour = "black", args = list(mean = 5, sd = 2)) +
  geom_function(fun = f, colour = "red") +
  scale_x_continuous(breaks = seq(-5, 15, 2), limits = c(-5, 15)) +
  xlab("x2") +
  ylab("PDF") +
  ggtitle("PDF of: N(5, sd = 2) [black] and alpha2 (uncond.) [red]")
```



```
optimize(f, interval = c(1, 7), maximum = TRUE)
```

```
## $maximum
## [1] 4.600697
##
## $objective
## [1] 0.2189738
```

3. In JAGS, we can specify α_j , $j = 2, \dots, c-1$ with

```
alpha[j] ~ dnorm(meanAlpha[j], precAlpha[j]) T(, alpha[j-1])
```

However, I think we need to make sure the sample that is being obtained by this is indeed from an unconditional distribution of α_j , $j = 2, \dots, c-1$.

1.2.2 Multivariate Truncated Normal Distribution

We specify this prior for model (2).

This approach allows a user for a full control of var-cov structure of random vector α . Following [15], we specify distribution of α as multivariate truncated normal distribution. Consider first $\alpha^* \sim N_{c-1}(\mu, \Sigma)$. Now, let α be truncation of α^* above $\mathbf{k} \in \mathbb{R}^{c-1}$. α has a $(c-1)$ -variate truncated normal distribution given by

$$f_{\alpha}(\mathbf{x}; \mu, \Sigma, \mathbf{k}) = \frac{\exp\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma (\mathbf{x} - \mu)\}}{\int \cdots \int_{R_{\leq \mathbf{k}}^{c-1}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma (\mathbf{x} - \mu)\} dx_1 \dots dx_{c-1}}$$

where

$$\begin{aligned} \mathbf{x} &\in \mathbb{R}_{\leq \mathbf{k}}^{c-1} \\ \mu &\in \mathbb{R}^{c-1} \\ \Sigma &\text{ is } (c-1) \times (c-1) \text{ var-cov matrix,} \\ \mathbf{k} &\in \mathbb{R}^{c-1}, \end{aligned}$$

with $c \geq 2$. Due to CLM model restriction (3), we further require that $\mathbf{k} = [k_1, k_2, \dots, k_{c-1}]^T$ is such that $\infty = k_1 > k_2 > \dots > k_{c-1} > -\infty$. Note that for $c = 2$ we obtain a univariate normal distribution without truncation, which somehow corresponds to α_1 in (4).

Unfortunately, the marginals of multivariate truncated normal variates do not need to be truncated normal in general. The exact formula for the marginal probability densities functions are given in [16].

Then, in JAGS we would simply need to generate these marginal r.vs. (see also `tmvtnorm` R package)

1.2.3 Transformation of the intercepts to an unconstrained space

$\delta_1 := \log(\alpha_1)$, $\delta_j := \log(\alpha_j - \alpha_{j-1})$, $j = 2, \dots, c-1 \geq 2$. Then, e.g. $\boldsymbol{\delta} \sim \mathcal{N}_{c-1}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Note that it is required that $\alpha_1 > 0$.

See Albert and Chib (1997) for more details.

1.2.4 An ordered uniform distribution (Ishwaran, 2000)

This approach was mentioned in few sources, but it is still a bit unclear to me. For instance, in Congdon (2005), on page 239, the author writes: “An alternative suggested by Ishwaran (2000) is a uniform density

$$0 = \alpha_1 < \alpha_2 < \dots < \alpha_{c-1} = U,$$

where U is equal to or less than c .” I think this approach is about to a simple sequence of non overlapping uniform distributions.

1.2.5 Latent response model

With this approach, the observed response Y is often taken to reflect an underlying continuous random variable Y^* with $c-1$ thresholds or cut points. Albert and Chib (1993) presented a Bayesian analysis that utilizes the latent (response) variable model:

$$Y_i^* = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i, \quad i \geq 1,$$

where ϵ_i are iid and $\epsilon_i \sim \mathcal{N}(0, 1)$, $i \geq 1$. Here we have that

$$Y_i = j \quad \text{iff} \quad \alpha_{j-1} < Y_i^* \leq \alpha_j, \quad j = 2, \dots, c-1,$$

where $\alpha_0 := -\infty$ and $\alpha_c := \infty$.

Following Congdon (2005) (see Chapter 7.2) that is based on Johnson and Albert (1999) (Chapter 4.3), the cut points are sampled in a way that takes account of the sampled Y^* . Thus, at iteration t :

$$\alpha_j^{(t)} \sim \mathcal{N}(0, V_\alpha) T(L_j, U_j), \quad j = 1, \dots, c-1,$$

where V_α is preset and

$$L_j = \max(Y^{*(t)}, Y_i = j) U_j = \min(Y^{*(t)}, Y_i = j+1)$$

and

$$Y_i^* \sim \mathcal{N}(\mathbf{x}_i^T \boldsymbol{\beta}, \gamma_i) T(\alpha_{y_{i-1}}, \alpha_{y_i}),$$

where $\gamma_i \sim Ga(0.5v, 0.5v)$, with v preset.

1.3 References

- [1] Agresti, A. (2010). Analysis of Ordinal Categorical Data. Hoboken, NJ: John Wiley & Sons.
- [2] Agresti, A. (2013). Categorical Data Analysis. Hoboken, NJ: John Wiley & Sons.
- [3] Agresti, A. (2015). Foundations of Linear and Generalized Linear Models. Hoboken, NJ: John Wiley & Sons.
- [4] Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. Journal of the American Statistical Association, 88(422): 669–679.
- [5] Albert, J. and Chib, S. (1997). Bayesian methods for cumulative, sequential and twostep ordinal data regression models. Technical report.
- [6] Congdon, P. (2005). Bayesian Model for Categorical Data, Wiley.
- [7] Ishwaran, H. (2000). Univariate and Multivariate Ordinal Cumulative Link Regression with Covariate Specific Cutpoints. The Canadian Journal of Statistics / La Revue Canadienne de Statistique, Vol. 28, No. 4 (Dec., 2000), pp. 715–730
- [8] James, N. T., Harrell Jr, F. E., Shepherd, B. E. (2022) Bayesian Cumulative Probability Models for Continuous and Mixed Outcomes. arXiv:2102.00330v2 [stat.ME]
- [9] Johnson, V. E. and Albert, J. H. (1999). Ordinal Data Modeling. Springer-Verlag, New York.
- [10] McKinley, T. J., Morders, M., Wood, J. L. N. (2015) Bayesian Model Choice in Cumulative Link Ordinal Regression Models.
- [11] McCullagh, P., and Nelder, J. A. (1989). Generalized Linear Models. 2nd ed. London: Chapman & Hall <https://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf>
- [12] Van Meter EM, Garrett-Mayer E, Bandyopadhyay D. Dose-finding clinical trial design for ordinal toxicity grades using the continuation ratio model: an extension of the continual reassessment method. Clin Trials. 2012 Jun;9(3):303-13. doi: 10.1177/1740774512443593. Epub 2012 Apr 30. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5531273/>
- [13] Wikipedia, https://en.wikipedia.org/wiki/Truncated_normal_distribution.
- [14] Greene, William H. (2003). Econometric Analysis (5th ed.). Prentice Hall.
- [15] Horrace, William C., “Some Results on the Multivariate Truncated Normal Distribution” (2005). Economics - Faculty Scholarship. 19.
- [16] Jack Cartinhour (1990) One-dimensional marginal density functions of a truncated multivariate normal density function, Communications in Statistics - Theory and Methods, 19:1, 197-203, DOI: 10.1080/03610929008830197

2 Prototype code

2.1 Fictitious observed data

```
dose_grid <- c(5, 15, 45, 70, 100, 220, 300, 600, 1000, 1800, 4000, 10000, 16000)

x <- c(
  rep(5, 1),
  rep(15, 4),
  rep(45, 5),
  rep(70, 5),
```

```

rep(100, 5),
rep(220, 8),
rep(300, 6),
rep(600, 15),
rep(1000, 8),
rep(1800, 9),
rep(4000, 10),
rep(10000, 14),
rep(16000, 2)
)

y <- c(
  rep(0, 1),
  rep(0, 4),
  rep(0, 5),
  rep(0, 5),
  rep(0, 5),
  rep(0, 7), 1,
  rep(0, 5), 1,
  rep(0, 10), 1, 1, 1, 1, 1,
  rep(0, 5), 1, 1, 1,
  rep(0, 8), 1,
  rep(0, 8), 1, 1,
  rep(0, 10), 1, 1, 1, 1,
  rep(1, 1),
  rep(2, 1)
)

```

2.2 Data-class

We need to adopt a new class for data as the existing one `Data` does not allow for categorical (>2 categories) toxicity. We propose to create new data class, named e.g. `DataOrdinal`, which inherits from existing `GeneralData` class and is a parent of existing `Data` class. The new `DataOrdinal` class:

- gets its slots from current `Data` and
- adds one new slot `yCategories`. It is a named integer vector that keeps all the possible values for `y`. Default to `c("No DLT" = 0L, "DLT" = 1L)`.

Eventually, instead of having `yCategories` as a named vector, we can introduce a new slot, e.g. `yLabels`. Below is the prototype code.

```

.DataOrdinal <- setClass(
  Class = "DataOrdinal",
  contains = "GeneralData",
  slots = c(
    x = "numeric",
    y = "integer",
    doseGrid = "numeric",
    nGrid = "integer",
    xLevel = "integer",
    yCategories = "integer",
    placebo = "logical"
  )
)

```

```

),
prototype = prototype(
  x = numeric(),
  y = integer(),
  doseGrid = numeric(),
  nGrid = 0L,
  xLevel = integer(),
  yCategories = c("No DLT" = 0L, "DLT" = 1L),
  placebo = FALSE
),
validity = v_data_cat_tox
)

.Data <- setClass(
  Class = "Data",
  contains = "DataOrdinal",
  validity = v_data # same as already existing v_data + minor update due to `y` labels.
)

```

Another option we considered was modification of existing `Data` class. The modification is done by:

- allowing `y` to take on values from 0, 1, 2, 3,
- adding new slot `yCategories`, defined in the same way as in `DataOrdinal` proposal, above.

However, the main disadvantages of this approach is that if we modify existing `Data` class rather than subclasse it, we need to make sure that the `Data` instances that are passed to work with existing models have `length(yCategories) <= 2`, which would be quite unpractical. Not to mentioned some other changes to `Data` specific methods, such as, `update`, `plot` or validation function.

3 Implementation

3.1 Data-class (`DataOrdinal`)

Add to `data-validity.R`

```

#' @description `r lifecycle::badge("experimental")`
#' @describeIn v_data_objects validates that the [DataOrdinal] object contains
#'   valid elements with respect to their types, dependency and length.

v_data_ordinal <- function(object) {
  v <- Validate()
  v$check(
    test_double(object@x, len = object@nObs, any.missing = FALSE),
    "Doses vector x must be of type double and length nObs"
  )
  v$check(
    test_integer(object@y, lower = 0, upper = length(object@yCategories)-1, len = object@nObs, any.missing = FALSE),
    paste(
      "DLT vector y must be nObs long and contain integers in the range 0 to",
      length(object@yCategories) - 1,
      "only"
    )
  )
}

```



```

    )
  )
  v$check(
    test_double(object@doseGrid, len = object@nGrid, any.missing = FALSE, unique = TRUE, sorted = TRUE),
    "doseGrid must be of type double and length nGrid and contain unique, sorted values"
  )
  v$check(
    test_int(object@nGrid),
    "Number of dose grid values nGrid must be scalar integer"
  )
  v$check(
    test_integer(object@xLevel, len = object@nObs, any.missing = FALSE),
    "Levels xLevel for the doses the patients have been given must be of type integer and length nObs"
  )
  v$check(
    test_flag(object@placebo),
    "The placebo flag must be scalar logical"
  )
  v$check(
    test_subset(object@x, object@doseGrid),
    "Dose values in x must be from doseGrid"
  )
  v$check(
    h_all_equivalent(object@x, object@doseGrid[object@xLevel]),
    "x must be equivalent to doseGrid[xLevel] (up to numerical tolerance)"
  )
  if (object@placebo) {
    is_placebo <- object@x == object@doseGrid[1]
    v$check(
      test_set_equal(object@cohort, object@cohort[!is_placebo]),
      "A cohort with only placebo is not allowed"
    )
    v$check(
      h_doses_unique_per_cohort(dose = object@x[!is_placebo], cohort = object@cohort[!is_placebo]),
      "There must be only one dose level, other than placebo, per cohort"
    )
  } else {
    v$check(
      h_doses_unique_per_cohort(dose = object@x, cohort = object@cohort),
      "There must be only one dose level, per cohort"
    )
  }
  v$check(
    test_set_equal(object@yCategories, as.integer(0:(length(object@yCategories) - 1))),
    paste0(
      "Values of yCategories must be integers in the range 0 to ",
      length(object@yCategories) - 1
    )
  )
  v$result()
}

```

Add to Data-class.R

```

# DataOrdinal ----

## class ----

#' `DataOrdinal`
#'
#' @description `r lifecycle::badge("experimental")`
#'
#' [DataOrdinal] is a class ordinal toxicity data.
#' It inherits from [GeneralData] and it describes toxicity responses on an
#' ordinal rather than binary scale.
#'
#' @note This class has been implemented as a sibling of the existing Data class
#' (rather than as a parent or child) to minimise the risk of unintended side
#' effects on existing classes and methods.
#'
#' @note The default setting for the yCategories slot replicates the behaviour
#' of the existing Data class.
#'
#' @inheritParams Data
#' @aliases DataOrdinal
#' @export
.DataOrdinal <- setClass(
  Class = "DataOrdinal",
  contains = "GeneralData",
  slots = c(
    x = "numeric",
    y = "integer",
    doseGrid = "numeric",
    nGrid = "integer",
    xLevel = "integer",
    yCategories = "integer",
    placebo = "logical"
  ),
  prototype = prototype(
    x = numeric(),
    y = integer(),
    doseGrid = numeric(),
    nGrid = 0L,
    xLevel = integer(),
    yCategories = c("No DLT" = 0L, "DLT" = 1L),
    placebo = FALSE
  ),
  validity = v_data_ordinal
)

## constructor ----

#' @rdname DataOrdinal-class
#' @param yCategories (named integer vector)\cr the names and codes for the
#' toxicity categories used in the data. Category labels are taken from the
#' names of the vector. The names of the vector must be unique and its values
#' must be sorted and take the values 0, 1, 2, ...##'

```

```

#' @inherit Data details note params
#' @example examples/Data-class-DataOrdinal.R
#' @export
DataOrdinal <- function(
  x = numeric(),
  y = integer(),
  ID = integer(),
  cohort = integer(),
  doseGrid = numeric(),
  placebo = FALSE,
  yCategories = c("No DLT" = 0L, "DLT" = 1L),
  ...
) {
  assert_numeric(x)
  assert_numeric(y)
  assert_numeric(ID)
  assert_numeric(cohort)
  assert_numeric(doseGrid, any.missing = FALSE, unique = TRUE)
  assert_numeric(yCategories, any.missing = FALSE, unique = TRUE)
  assert_character(names(yCategories), any.missing = FALSE, unique = TRUE)
  assert_flag(placebo)

  doseGrid <- as.numeric(sort(doseGrid))

  if (length(ID) == 0 && length(x) > 0) {
    message("Used default patient IDs!")
    ID <- seq_along(x)
  }

  if (!placebo && length(cohort) == 0 && length(x) > 0) {
    message("Used best guess cohort indices!")
    # This is just assuming that consecutive patients
    # in the data set are in the same cohort if they
    # have the same dose. Note that this could be wrong,
    # if two subsequent cohorts are at the same dose.
    cohort <- as.integer(c(1, 1 + cumsum(diff(x) != 0)))
  }

  .DataOrdinal(
    x = as.numeric(x),
    y = safeInteger(y),
    ID = safeInteger(ID),
    cohort = safeInteger(cohort),
    doseGrid = doseGrid,
    nObs = length(x),
    nGrid = length(doseGrid),
    xLevel = matchTolerance(x = x, table = doseGrid),
    placebo = placebo,
    yCategories = yCategories
  )
}

```

Create a test object

```
ordinal_data <- DataOrdinal(
  doseGrid = seq(10, 100, 10),
  x = c(10, 20, 30, 40, 50, 50, 50),
  y = c(0L, 0L, 0L, 0L, 0L, 1L, 2L),
  ID = 1L:7L,
  cohort = as.integer(c(1:4), 5, 5, 5)
)
```

3.2 Model-class (LogisticLogNormOrd)

For simplicity of notation, unless we need to refer to particular subjects or to particular values of the explanatory variables, we replace p_{ij} (i.e. $P(X_i \geq j)$) in such equations by p_j (i.e. $P(Y \geq j)$), keeping in mind that in the model this is actually a conditional probability at each fixed value for the explanatory variables.

The following basic CLM can be considered as a single-agent toxicity model, in which the covariate is the natural logarithm of the dose x divided by the reference dose x^* , i.e.:

$$\text{logit}[p_j] = \alpha_j + \beta \log\left(\frac{x}{x^*}\right), \quad j = 1, \dots, c-1.$$

The prior is:

$$\log(\beta) \sim N(\mu_\beta, \sigma_\beta^2)$$

and for $\{\alpha_j\}_{j \in \{1, \dots, c-1\}}$ we propose prior as in (4).

```
LogisticLogNormOrd <- function(meanAlpha, precAlpha, meanBeta, precBeta, ref_dose = 0) {
  .LogisticLogNormOrd(
    meanAlpha = meanAlpha,
    precAlpha = precAlpha,
    meanBeta = meanBeta,
    precBeta = precBeta,
    datamodel = function() {
      for (i in 1:nObs) {
        x_rel[i] <- log(x[i] / ref_dose)
        for (j in 1:(yCategories - 1)) {
          logit(p[i, j]) <- alpha[j] + beta * x_rel[i]
          y[i, j] ~ dbern(p[i, j])
        }
      }
    },
    priormodel = function() {
      alpha[1] ~ dnorm(meanAlpha[1], precAlpha[1])
      if(yCategories >= 3) {
        for (j in 2:(yCategories-1)) {
          alpha[j] ~ dnorm(meanAlpha[j], precAlpha[j]) T(, alpha[j-1])
        }
      }
      beta ~ dnorm(meanBeta, precBeta)
    },
    modelspecs = function(from_prior) {
      ms <- list(
        meanAlpha = meanAlpha,
        precAlpha = precAlpha,
        meanBeta = meanBeta,
        precBeta = precBeta
      )
    }
  )
}
```

```

    )
    if (!from_prior) {
      ms$ref_dose <- ref_dose
    }
    ms
  },
  init = function(yCategories) {
    list(
      alpha = seq(from = 5, to = 3, length.out = length(yCategories) - 1),
      beta = 0
    )
  },
  datanames = c("nObs", "yCategories", "y", "x"),
  sample = c("alpha", "beta")
)
}

```