Ordinal regression models made easy. A tutorial on parameter interpretation, data simulation, and power analysis.

Filippo Gambarota[1] & Gianmarco Altoè[1]

[1] Department of Developmental Psychology and Socialization, University of Padova, Italy

Author Note

Add complete departmental affiliations for each author here. Each new line herein must be indented, like this line.

Enter author note here.

The authors made the following contributions. Filippo Gambarota: Conceptualization, Writing - Original Draft Preparation, Writing - Review & Editing; Gianmarco Altoè: Writing - Review & Editing, Supervision.

Correspondence concerning this article should be addressed to Filippo Gambarota, Postal address. E-mail: filippo.gambarota@unipd.it

Abstract

Abstract here

Keywords: ordinal, likert, simulations, power

Word count: X

Ordinal regression models made easy. A tutorial on parameter interpretation, data simulation, and power analysis.

Introduction

Psychological research make an extensive use of ordinal data. One of the main reason is probably the usage of Likert scales (Likert, 1932). Ordinal data refers to a specific type of measurement scale (Stevens, 1946) where ordered numbers are assigned to a variable. Compared to nominal scale and as the name suggest the labels are ordered. Compared to interval or ratio scales there is no explicit assumption about the distance between labels. (rivedi questo). An example is asking people the degree of agreement about a certain statement using a scale from 1 (no agreement) to 7 (total agreement). Answering 4 compared to 2 suggest an higher agreement but we cannot affirm that there is two times the agreement compared to the second answer. Stevens (1946) and Kemp and Grace (2021) suggested that for ordinal variables is appropriate to calculate ranks-based statistics (e.g., median or percentiles) instead of metric statistics (e.g., mean or standard deviation). This distinction in terms of the appropriateness of certain descriptive statistics is also relevant when modeling data. Treating ordinal data as metric refers to assuming the labels as actual integer numbers thus assuming a fixed and know distance between levels (Liddell & Kruschke, 2018).

In Psychology especially when using item-based measures (questionnaires, surveys, etc.) the common practice is using a normal linear regression that makes an explicit assumption about metric features of the response variable. Liddell and Kruschke (2018) reviewed the psychological literature using likert-based measures and reported how the majority of papers used metric-based statistical models. In the same work, Liddell and Kruschke (2018) showed extensive examples and simulations about the potential pitfalls of treating an ordinal variable as metric (but see Robitzsch, 2020 for an alternative perspective). They reported problems in terms of lack of power, inversion of the effects and distortion in estimating the effect size (rivedi meglio qui). (vedi se c'è qualche altro lavoro che fa vedere questa cosa). Some authors suggested that despite individual items being ordinal, averaging multiple items could solve the problem of applying metric models (e.g., Carifio & Perla, 2007, 2008 metti queste citazioni). Liddell and Kruschke (2018) suggested that in some conditions even averaging multiple items and applying metric models could be problematic. Cliff (2016) suggest that most of the research questions in behavioral sciences can be considered as ordinal (*is the score x higher than the score y*?) concerning variables where the most appropriate measurement scale is probably ordinal.

qualcosa ancora qui

Ordinal regression models

Despite the actual modeling proposal by Liddell and Kruschke (2018), there is a class of regression models taking into account the ordinal nature of the response variable without making metric assumptions. We can class this general class of models as *ordinal regression*. The nomenclature of these models can be confusing mainly because there are several subclasses of models with different assumptions and structures (Gerhard Tutz, 2022). Gerhard Tutz (2022) and Bürkner and Vuorre (2019) provide a clear and updated taxonomy of ordinal regression models. We can identify three main classes: *cumulative models* (McCullagh, 1980), *sequential models* (Gerhard Tutz, 1990) and *adjacent category models*. The cumulative is the mostly used model assuming the existence of a latent variable that when segmented using thresholds produces the observed ordinal variable. The psychological process underlying the response is clearly formalized in the signal detection theory framework where the respondent (vedi come sistemare). The sequential model as suggested by the name is appropriate when modelling sequential processes. Assuming to have 5 response options, the sequential model assume that responding 3 means already reaching the states 1 and 2. A clear example is proposed by Bürkner and Vuorre (2019) where the marriage duration in years is predicted as a function of some explanatory variables. For each level of the response variable there is a latent distribution where the step between a marriage year $k = 1$ and the next years $k > 1$ is modeled by the sequential regression. When comparing k with $k > 1$, everything lower than k is assumed to be already reached (Gerhard Tutz & Berger, 2020). The adjacent category model compare the category k with $k + 1$ still assuming a latent distribution for each k. As suggested by Gerhard Tutz (2022) the adjacent-category model can be seen as a series of binary binomial regressions taking into account the order of the categories. Bürkner and Vuorre (2019) suggested that adjacent-category model can be chosen for its mathematical convenience and there is no a clear empirical distinction as for the cumulative vs sequential model.

In the current paper we put the focus on the cumulative model for several reasons. The first reason is that the latent formulation of the model is particularly convenient both for parameter interpretation and data simulation. The second reason is that several psychological variables can be formalized as a latent continuous construct that when measured is collected using ordinal items. Furthermore, also the signal detection theory framework models the decision process assuming a latent distribution for the signal and noise where thresholds are the response criteria (e.g., DeCarlo, 2010).

Model notation

metti qualcosa sulla nomenclatura

> The name, cumulative link models is adopted from Agresti (2002), but the model class has been referred to by several other names in the literature, such as ordered logit models and ordered probit models (Greene and Hensher 2010) for the logit and probit link functions. The cumulative link model with a logit link is widely known as the proportional odds model due to McCullagh (1980) and with a complementary log-log link, the model is sometimes referred to as the proportional hazards model for grouped survival times.

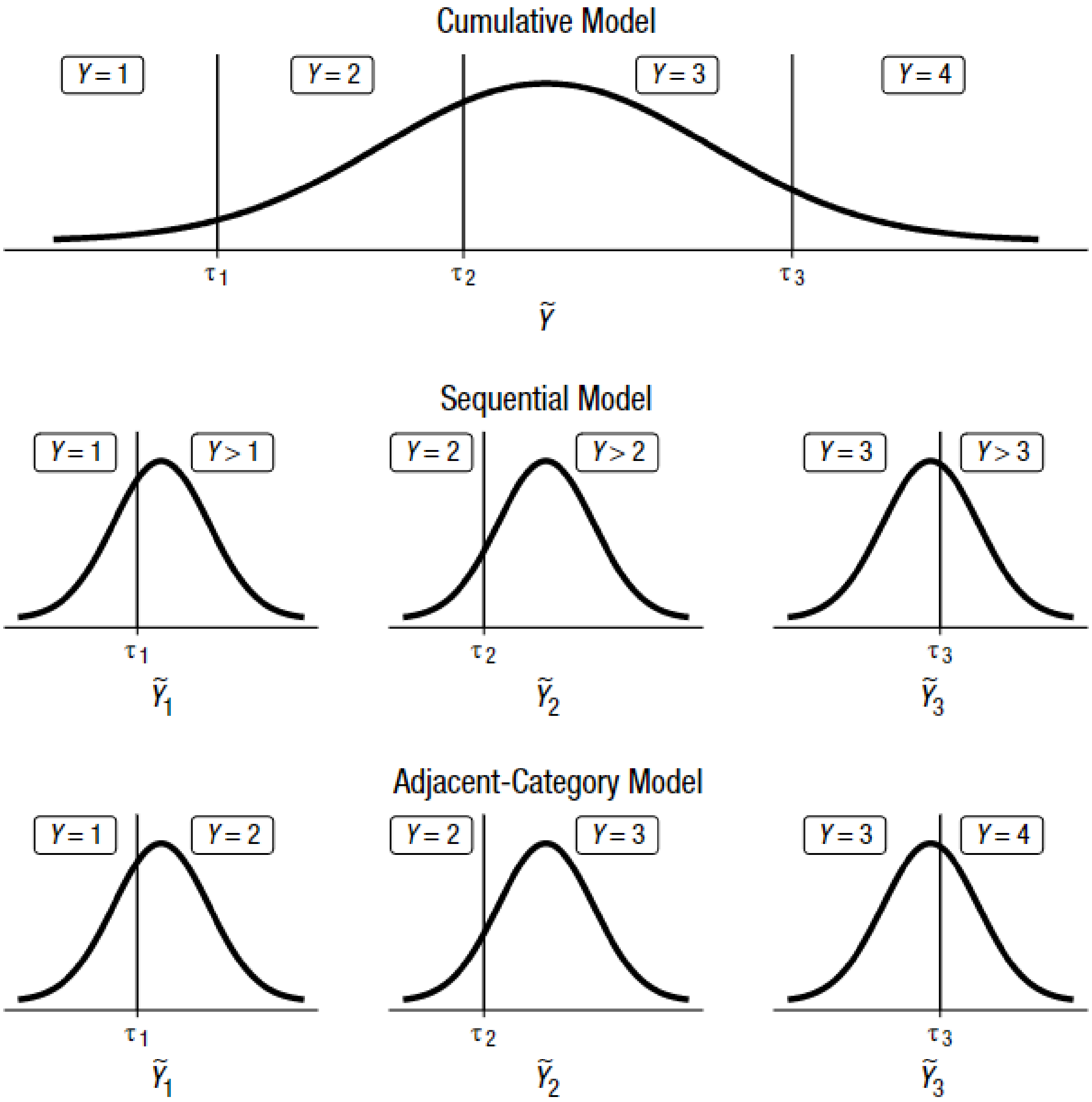

Figure 1. Theoretical figure about ordinal models. Voglio fare una figura come questa in Bürkner and Vuorre (2019) ma con qualche modifica per fare vedere i tre modelli.

In this section we introduce some notations for the cumulative model that is used through the paper and in the R code. The notation is as consistent as possible with the main textbooks and with the `ordinal` R package used in the tutorial. We define Y_k as the observed ordinal response with $1, 2, \dots, k$ options and $Y^\star$ is the corresponding latent variable. The latent variable is segmented using $k - 1$ thresholds $\alpha_1, \alpha_2, \dots, \alpha_{k-1}$. With $g()$ we define the cumulative probability function assumed for the specific model. For example, when using a Gaussian distribution we are fitting a model with a `probit` link function thus $f() = \Phi()$. With $g^{-1}()$ we define the inverse of the cumulative probability function. The minus sign in $\mathbf{X}$ is used to give the usual interpretation of standard regression models where an increase in β corresponds to an increase in the probability of higher categories of Y (metti il libro di agresti sui dati ordinali).

$$P(Y \leq k) = g(\alpha_k - \mathbf{X}\ ), \quad k = 1, 2, \dots, k \tag{1}$$

We can call the $\mathbf{X}$ as the linear predictor η. Equation (num) describe how the probability of a single outcome Y_k.

$$P(Y = k) = g^{-1}(\alpha_k - \eta) - g^{-1}(\alpha_{k-1} - \eta) \tag{2}$$

The same model can be formalized using the latent formulation (see Equation). This is more similar to a standard linear regression.

$$Y_i^\star = \eta + \epsilon_i \tag{3}$$

Where ϵ is the error part and comes from the distribution that is assumed from the latent model. For a *probit* model, errors are sampled from a standard Gaussian distribution while for a *logit* model from a standard logistic distribution. Then the observed ordinal value Y_i comes from $Y_i^\star$ being between the thresholds $\alpha_{k-1} < Y_i^\star < \alpha_k$ with thresholds organized as $-\infty \equiv \alpha_1 < \cdots < \alpha_{k-1} < \alpha_k \equiv \infty$.

In the basic version of the model, the thresholds α are considered as fixed and do not vary as a function of predictors. The thresholds are part of the measurement procedure (Liddell & Kruschke, 2018). In a more sophisticated version of the model called location-shift (Gerhard Tutz, 2022), both the location μ and the thresholds α can vary as a function of the predictors. In the following sections another version of the model called location-scale (Cox, 1995; Rigby & Stasinopoulos, 2005; Gerhard Tutz, 2022) will be presented as a more flexible way to model latent distributions with heterogeneous variance (spiega meglio).

lm on latent vs ordinal

as suggested https://people.vcu.edu/~dbandyop/BIOS625/CLM_R.pdf, running a lm on the latent variables gives similar parameter as the clm. Of course, we are able to do this with real data given that the ordinal variable is the observed version of an unobserved latent variable. But in simulation this is useful to understand what the cumulative model is doing.

```
## 
## Call:
## lm(formula = ys ~ x1 + x2, data = dat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.4772 -0.6765 -0.0005  0.6742  4.0796 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.0007926  0.0031780   0.249    0.803    
## x1          1.0009911  0.0031859 314.192   <2e-16 ***
## x2          0.4998991  0.0031800 157.203   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.005 on 99997 degrees of freedom
## Multiple R-squared:  0.5514, Adjusted R-squared:  0.5513
## F-statistic: 6.144e+04 on 2 and 99997 DF,  p-value: < 2.2e-16
```

```
## formula: y ~ x1 + x2
## data:    dat
##
##  link   threshold nobs  logLik     AIC       niter max.grad cond.H
##  probit flexible  1e+05 -122909.88 245831.76 5(0)  2.04e-07 9.1e+00
##
## Coefficients:
##     x1     x2
## 0.9949 0.4965
##
## Threshold coefficients:
##     1|2     2|3     3|4     4|5
## -0.8360 -0.2568  0.2515  0.8421
```

```
##  (Intercept)           x1           x2
## 0.0007886378 0.9960398259 0.4974264662
```

Kruscke parametrization

Liddell and Kruschke (2018) and Kruschke (2015) proposed an alternative parametrization to understand the model parameters. They used a probit model where thresholds and regression parameters are estimated on the scale of the ordinal variable compared to standard ordinal regression where they refers to the quantile of the latent variables for the threshold and z score or odds ratio for the regression coefficients. They implemented the model in Jags and provided some equations and R functions to convert from the standard parametrization to the proposed one.

The proposed model is fitted within a Bayesian framework using either Jags (citation) or Stan (citation). Kruskche proposed a simple method to convert the parameters fitted with a standard model within the proposed parametriazion. The main improvement regards mapping the values (for thresholds α_i and slopes β) from latent standard distribution (gaussian or logistic) into the scale of the y ordinal value. The scale of the variable depend on the numeric labels assigned to ordered categories. There is an additional feature of the proposed parametrization where the first and the last thresholds are fixed respectively to $\alpha_1 + 0.5$ and $\alpha_k - 0.5$ where other thresholds are estimated.

Kurz (2023) explain how to convert a model fitted with the `brms` package (citation) (an R package for regression modeling using *stan*) into the Kruschke (2015) parametrization.

Using the function proposed by Kruschke (2015) (metti ref ad osf) and the equations presented in Kurz (2023) the `clm_to_ord()` function convert parameters fitted with the `clm` function into the corresponding parameters for the latent $Y^\star$ variable considering the actual range of values (from 1 to k).

```r
clm_to_ord <- function(fit){
  th <- unname(fit$alpha) # estimated thresholds
  coefs <- fit$coefficients
  coefs <- unname(coefs[(length(th) + 1):length(coefs)])
  k <- length(th) + 1 # number of levels for y
  y <- 1:k # ordinal values
  stdm <- 0 # latent mean
  stds <- 1 # latent sigma
  th_y <- scales::rescale(th, to = c(y[1] + 0.5, y[k] - 0.5))
  lats <- (th_y[2] - th_y[k - 2]) / (th[2] - th[k - 2]) # real latent sigma
  b0 <- th_y[1] - th[1] * lats
  betas <- lats * coefs
  out <- list(sigma = lats, b0 = b0, beta = betas, alpha = th_y)
  return(out)
}
```

```r
dat <- data.frame(
  x = rnorm(1e5)
)

dat <- sim_ord_latent(~x, By = 1, probs = rep(1/5, 5), link = "probit", data = dat)
fit <- clm(y ~ x, data = dat, link = "probit")
```

```r
clm_to_ord(fit)
```

```
## $sigma
## [1] 1.78472
##
## $b0
## [1] 2.993037
##
## $beta
## [1] 1.797978
##
## $alpha
## [1] 1.500000 2.560652 3.446266 4.500000
```

Gelman parametrization

rivedi questa parametrizzazione in Gelman, Hill, and Vehtari (2020)

Interpreting parameters

Odds and odds ratio

una piccola spiegazione di odds e odds ratio

Proportional odds assumption

Following again the taxonomy by Gerhard Tutz (2022), each of the presented ordinal regression model has a basic version making the proportional odds assumption. There are more advanced versions of the model relaxing this assumption completely (*non proportional odds* models) and partially (*partial proportional odds* models) (rivedi se la nomenclatura va bene). The proportional odds assumption is a crucial point when intepreting model parameters. Can be formalized as in Equation (). Basically if we use the *logit* link function $g()$ the βs are interpreted as standard odds ratio in the logistic regression. Given that we have $k > 2$ alternatives there is the need of $k - 1$ equations (as reported in Equation ()). The porportional odds is assuming that the regression coefficients β are independent from the thresholds α thus the effect is the same regardless of the Y level. Thus $\beta_1 = \beta_2 \cdots = \beta_{k-1}$. The Figure () depicts the proportional odds assumption for the $k - 1$ logistic curves both for probabilities and linear predictors η.

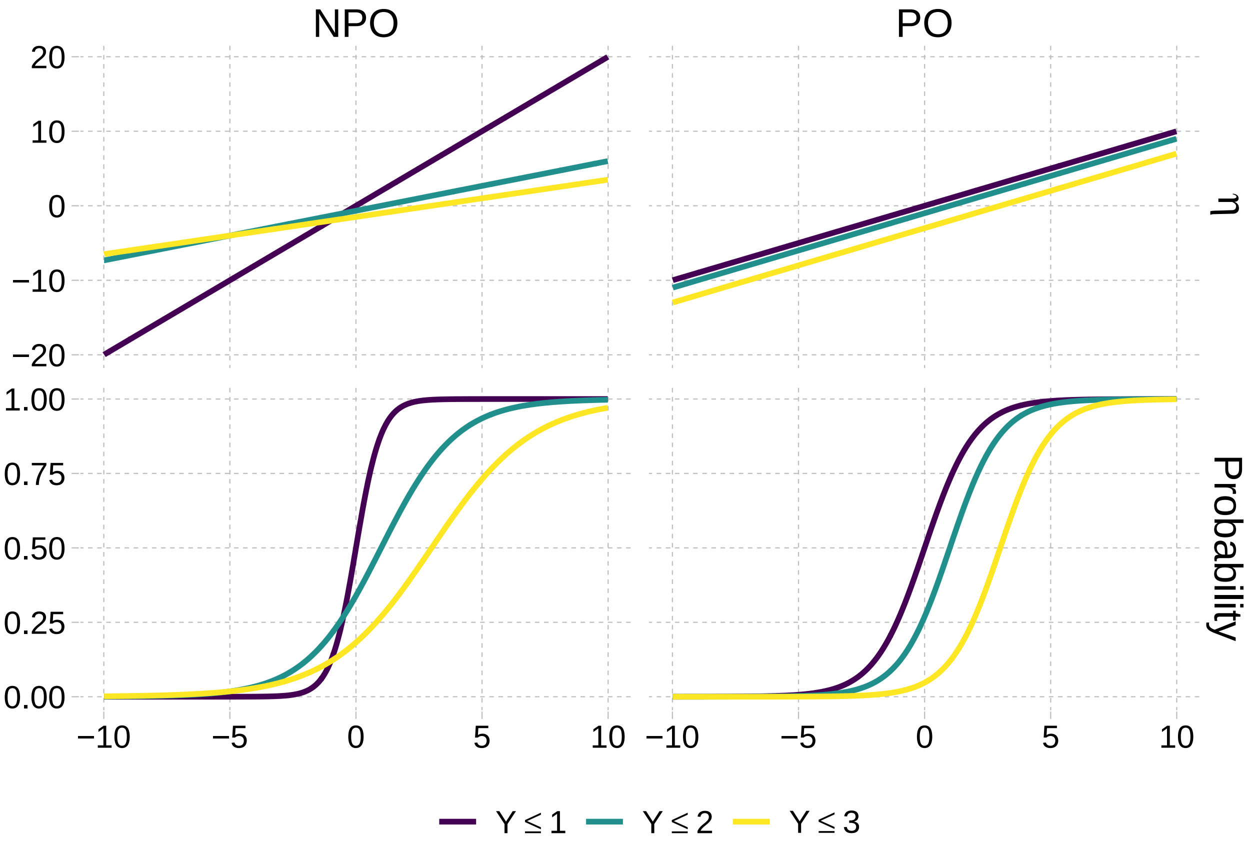

Figure 2. caption here

The proportional odds assumption is convenient because regardless the number of k, the β_j (j being the number of regression coefficients) effect is assumed to be the same. The model is more parsimonious compared to estimating $k - 1$ coefficients for each β_j as in the multinomial regression or the non-proportional version of the ordinal regression. Clearly this can be a strong assumption in some conditions. There are several methods to test the proportional odds assumption (see Liu, He, Tu, & Tang, 2023 for an overview). Gerhard Tutz and Berger (2020) suggested that using alternative as the location-shift or location-scale models, the assumption can be made (improving model parsimony) but still having a more flexible modeling framework. In the next section, we provide more details about the location-scale model in terms of parameters interpretation and data simulation.

Demonstrating the proportional odds assumption

basically given that the effect of β is constant the odds ratio is independent from the thresholds α_j (Liu et al., 2023). In the current paper we are simulating data using a single set of βs thus the proportional odds assumption is true. Liu et al. (2023) reviewed the available methods to test this assumption.

- https://hbiostat.org/ordinal/impactpo.pdf blog su proportional odds, mi pare di capire che non sia così problematica come assunzione

sarebbe da capire quali sono i rischi di assumere questo ed eventualmente se simulare non proportional odds è troppo complicato.

Let's simulate the effect of a binary predictor on ordinal scale 1-5:

```r
b1 <- log(3) # log odds ratio
n <- 1e4
x <- rep(c("a", "b"), each = n/2)
dat <- data.frame(x = x)
probs <- rep(1/5, 5) # for the group "a", uniform probabilities
dat <- sim_ord_latent(~x, By = b1, probs = probs, data = dat, link = "logit")
fit <- clm(y ~ x, data = dat, link = "logit")
pr <- predict(fit, data.frame(x = unique(x)))$fit
pr
```

```
##            1         2         3         4         5
## 1 0.20141151 0.2052163 0.2038036 0.1939345 0.1956341
## 2 0.07031792 0.1001600 0.1492126 0.2324907 0.4478188
```

Basically the proportional odds suggest that:

$$log(\frac{P(y \leq 1)}{P(y > 1)})$$

Is the same regardless the level of the x predictor. Thus:

$$\log\left(\frac{\frac{P(y \leq 1|x_0)}{P(y > 1|x_0)}}{\frac{P(y \leq 2|x_0)}{P(y > 2|x_0)}}\right) = \log\left(\frac{\frac{P(y \leq 1|x_1)}{P(y > 1|x_1)}}{\frac{P(y \leq 2|x_1)}{P(y > 2|x_1)}}\right)$$

```r
a_or1vs2345 <- filor::odds(pr[1, 1]) # 1 vs 2 3 4 5
a_or12vs345 <- filor::odds(sum(pr[1, 1:2])) # 1 vs 2 3 4 5

b_or1vs2345 <- filor::odds(pr[2, 1]) # 1 vs 2 3 4 5
b_or12vs345 <- filor::odds(sum(pr[2, 1:2])) # 1 vs 2 3 4 5

c(xa = log(a_or1vs2345 / a_or12vs345), xb = log(b_or1vs2345 / b_or12vs345))
```

```
##         xa         xb
## -0.9995719 -0.9995719
```

Proportional odds can be also visualized by plotting the cumulative probabilities of y, in terms of $g(P(y \leq j))$ (where $g()$ is the logit link function) as a function of the predictor x. If the proportional odds assumptions holds the slopes are parallel (also known as parallel regression assumption). The Figure 3 depicts the assumption of proportional odds in the probability and logit space.

Harrell (2015) proposed a very intuitive plot to assess the proportional odds assumption and eventually the degree of deviation from the ideal case. Basically predictor is plotted against the logit of the cumulative probability. Distances between pairs of symbols should be similar across levels of the predictors. Numerical predictors can be binned before plotting the corresponding logit. Figure 4 depicts an example with simulated data satisfying the proportional odds assumption.

From a statistical point of view, the proportional odds assumption can be assessed by fitting $k - 1$ binomial regressions and checking if the estimated β is similar between

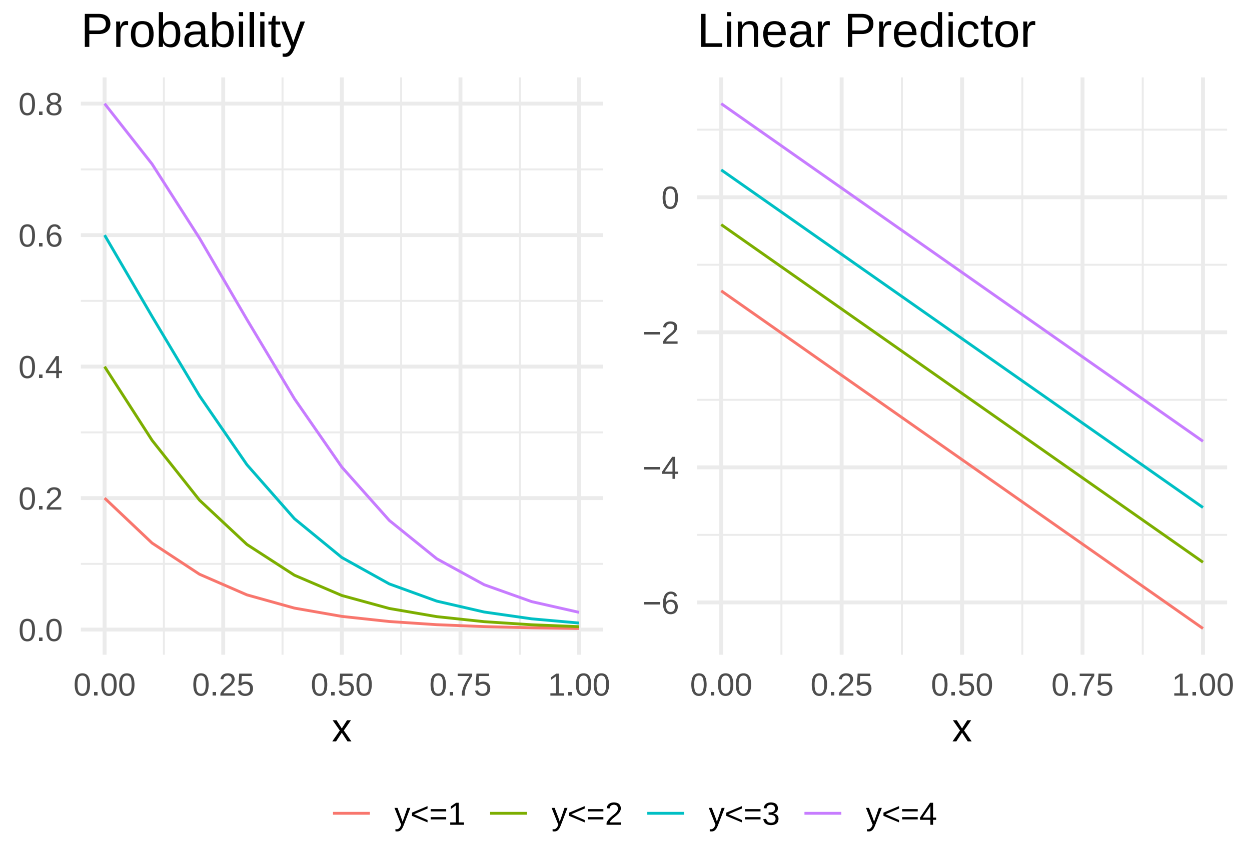

Figure 3. POA assumption

regressions. The regressions are estimated by creating $k - 1$ dummy variables from the ordinal y. The code below show that simulating data with the proportional odds create $k - 1$ binomial regression with similar βs. In fact, fitting $k - 1$ binomial regressions can be considered also an alternative to fitting an ordinal regression (see Gelman et al., 2020) with more flexibility in parameters (there will be $k - 1$ regression coefficients instead of a single one) but losing the latent intepretation (magari qualcosa di più qui).

```r
k <- 4
n <- 1e5
dat <- data.frame(
  x = runif(n)
)

dat <- sim_ord_latent(~x, By = 5, probs = rep(1/k, k), data = dat, link = "logit")
```

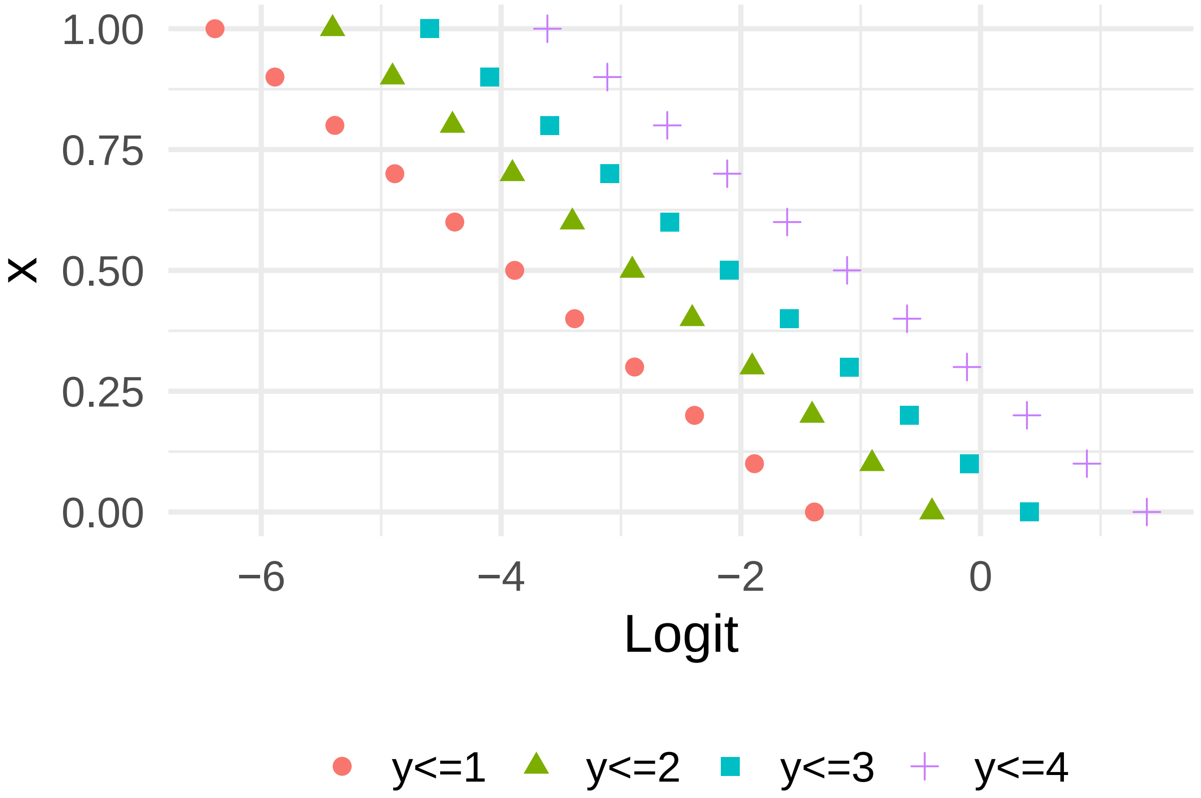

Figure 4. Harrel Poa Plot

```
# create dummy variables

dat$y1vs234 <- ifelse(dat$y <= 1, 1, 0)
dat$y12vs34 <- ifelse(dat$y <= 2, 1, 0)
dat$y123vs4 <- ifelse(dat$y <= 3, 1, 0)

fit1vs234 <- glm(y1vs234 ~ x, data = dat, family = binomial(link = "logit"))
fit12vs34 <- glm(y12vs34 ~ x, data = dat, family = binomial(link = "logit"))
fit123vs4 <- glm(y123vs4 ~ x, data = dat, family = binomial(link = "logit"))

car::compareCoefs(fit123vs4, fit12vs34, fit1vs234)
```

```
## Calls:
## 1: glm(formula = y123vs4 ~ x, family = binomial(link = "logit"), data = dat)
## 2: glm(formula = y12vs34 ~ x, family = binomial(link = "logit"), data = dat)
## 3: glm(formula = y1vs234 ~ x, family = binomial(link = "logit"), data = dat)
##
```

```
##             Model 1 Model 2 Model 3
## (Intercept)  1.1146  0.0128 -1.0869
## SE           0.0151  0.0163  0.0216
##
## x           -5.0266 -5.0021 -5.0291
## SE           0.0365  0.0482  0.0736
##
```

spiegare qui la differenza tra proportional and non proportional odds Gerhard Tutz and Berger (2020).

Logit vs Probit model

When fitting an ordinal regression the two mostly used link functions are the *probit* and *logit*. From the distribution point of view the two functions are very similar. The *probit* model is based on a cumulative Gaussian distribution while the *logit* model is based on a logistic distribution. Figure 5 depict the two cumulative distributions.

Given the different underlying distribution, the parameters have a different interpretation under the two models. The probit model assume a latent standard normal distribution with and $\sigma = 1$. The logit model assume a logistic distribution with $\sigma^2 = \pi^2/3$. Thus regression coefficients β represents the increase in standard deviations units. The interpretation in terms of latent distribution is particularly useful for the probit model where βs can be interpreted in a Cohen's d like manner. Furthermore, there is the possibility to directly map parameters from signal detection theory (Green, Swets, & Others, 1966; Stanislaw & Todorov, 1999) into an ordinal probit regression. In practical terms, the thresholds are the criterion cut-offs and the β_1 is the d$'$ parameter (DeCarlo, 1998; Knoblauch & Maloney, 2012).

The latent formulation of the ordinal regression model allow to think in standard linear regression terms for choosing and interpreting model parameters before converting into the probability space. Both the standard Normal and the logistic distribution are

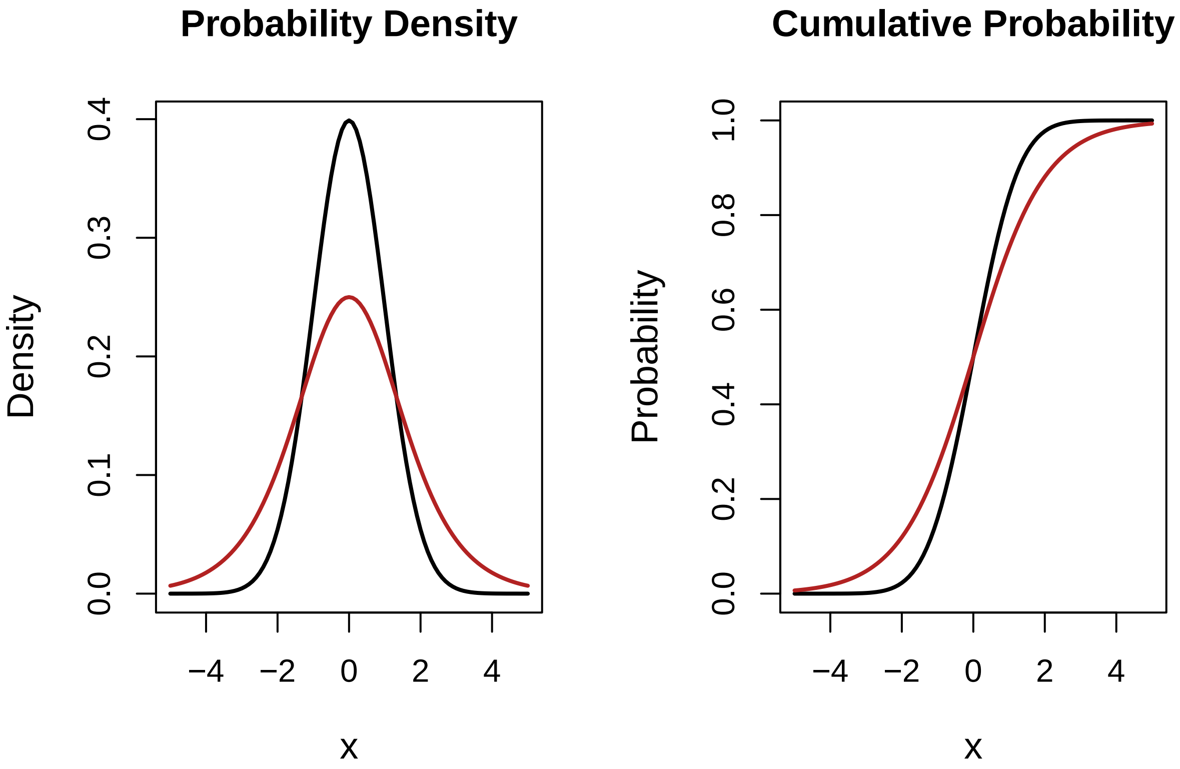

Figure 5. Logit vs Probit

defined with a location (μ) and a scale (s) parameter. For the Normal distribution, μ is the mean and scale is the variance (σ^2). For the logistic distribution the variance is $\sigma^2 = \frac{s^2\pi^2}{3}$. Thus fixing μ and $\sigma^2 = 1$ (the default), the two distributions are similar with the logistic having more variance. For this reason, the latent formulation for parameter intepretation is particularly useful for the probit model because the β is directly interpreted in standard deviaton units that by default are fixed to 1. With a binary predictor x, β is the shift of the latent distribution increasing x by one unit. This can be directly interpreted as a standardized mean difference effect size (e.g., Cohen's d). This is the same for the logistic distribution but a unit increase in x shift the latent distribution by $\sigma = \sqrt{\frac{s^2\pi^2}{3}} = \frac{s\pi}{\sqrt{3}}$.[1].

[1] In the case of a standard logistic distribution ($s^2 = 1$), the standard deviation is $\frac{\pi}{\sqrt{3}}$

Simulating data

There are mainly two ways to simulate data. The first method concerns simulating starting from the latent formulation of the ordinal model. Basically we can simulate the underlying latent distribution and then fixing the thresholds converting the latent continuous variable into the observed ordinal variable.

simulating using probabilities. The first method to simulate ordinal data as a function of predictors is by calculating the true probabilities $g^{-1}(\eta)$ as a function of predictors and then sample from a multinomial distribution. This is similar to the general method to simulate data for a generalized linear model where the linear predictors η are calculated and data are generated from the corresponding probability distribution of the random component. The following code box simulate a n binary trials with a continuous predictor x.

We can apply the same idea to an ordinal outcome but we need $k-1$ equations where k is the number of ordinal levels. Let's simulate a similar design with a continuous x predictor and $k=4$ options. We fix the baseline probabilities where $x=0$ as uniform thus $p(y_1)=p(y_2)=...p(y_k)=1/k$. The general workflow for the simulation is:

1. Define n (number of observations) and k (number of ordinal outcomes)
2. Define the regression coefficients βs
3. Define the baseline probabilities when $x=0$. These probabilities can be converted to the corresponding thresholds α choosing the appropriate link function (*logit* or *probit*).
4. Calculate the linear predictors η using $k-1$ equations combining the predictors $\mathbf{X}$ and regression coefficients .
5. Apply the inverse of the link function $g^{-1}(\eta)$ on the linear predictor and calculate the cumulative probabilities $p(y \leq 1|x), p(y \leq 2|x), ...p(y \leq k-1|x)$.

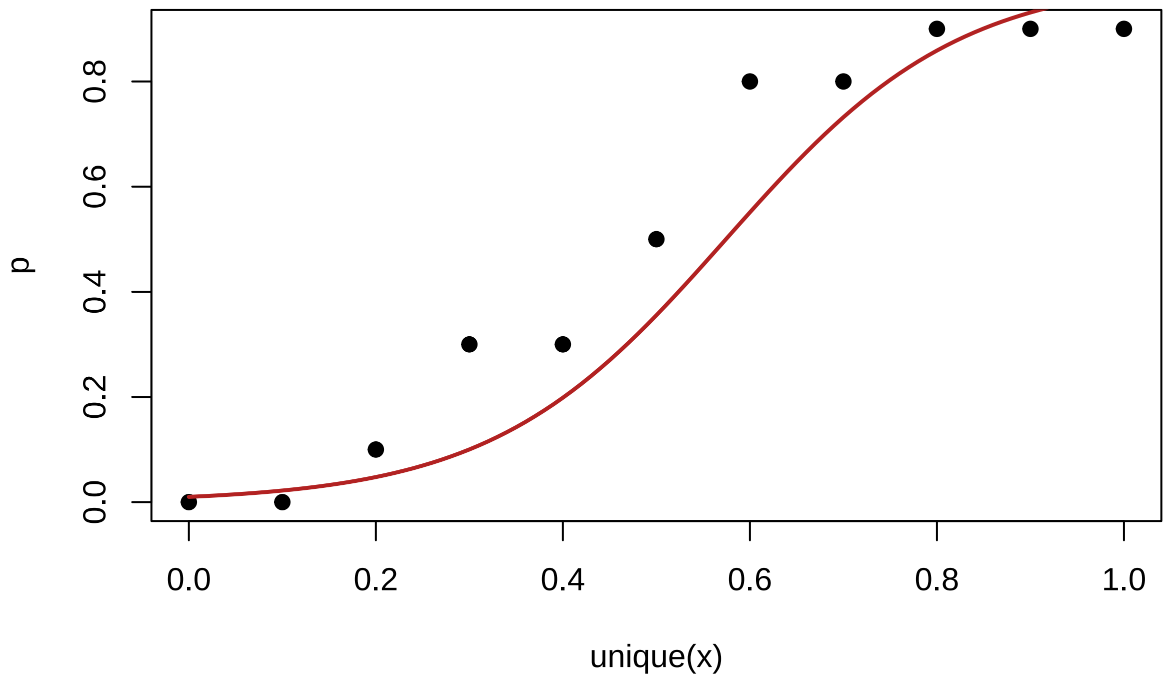

Figure 6

6. Calculate for each observation the probability of the k outcome as the difference between cumulative probabilities.

   $p(y = 1) = 0 - p(y <= 1), p(y = 2) = p(y <= 2) - p(y <= 1)$ (sistema qua). This is implemented in R by adding a columns of zeros and ones respectively at the beginning and end of the matrix of cumulative probabilities.

7. Sample n outcomes from a multinomial distribution, choosing between k alternatives with the calculated probabilities.

(rivedi se fare con for) This can be easily implemented in R. We are using a `for` loop to improve the clarity of the workflow but using the `*apply` family of functions can improve the efficiency of the computations.

The linear predictors can be seen in the Figure ref. The g in this case is the cumulative *probit* function Φ.

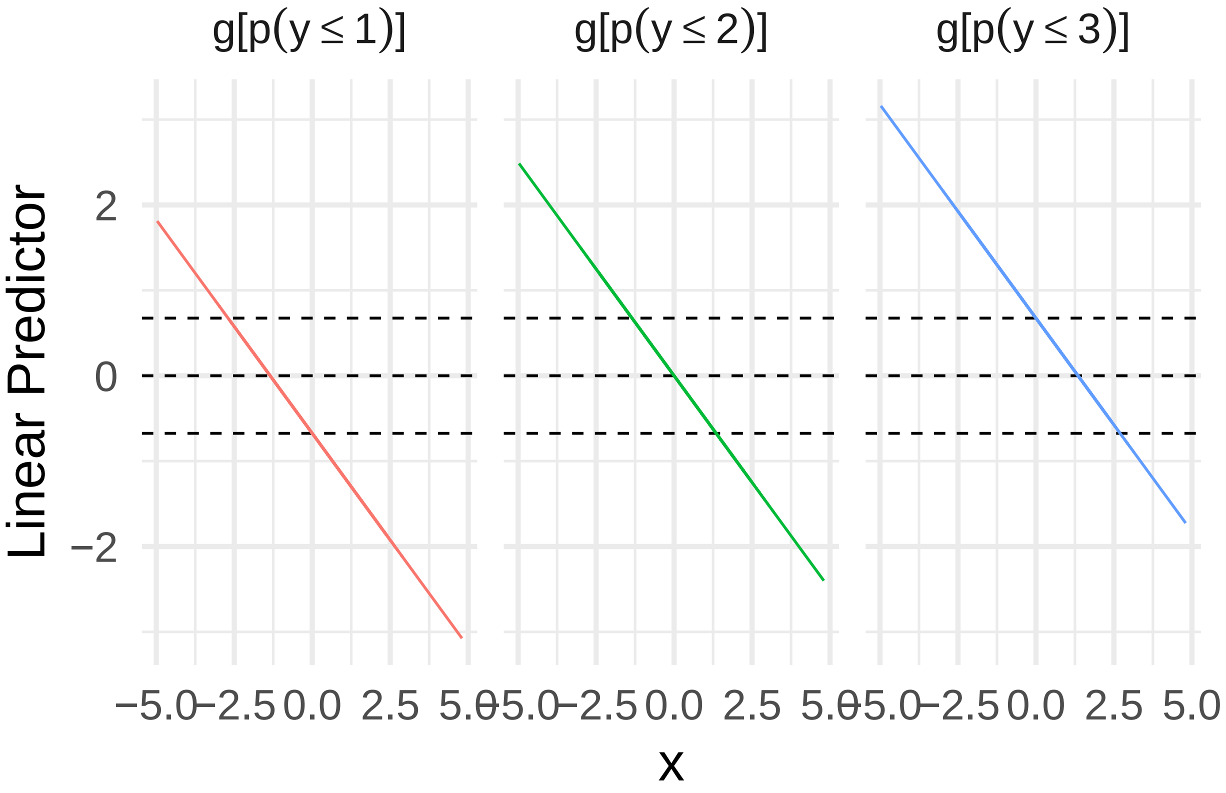

Figure 7

Now we can apply the inverse of the link function $g^{-1} = \Phi^{-1}$ to calculate the corresponding cumulative probabilities.

```
##    p_y1  p_y2  p_y3  p_y4
## 1 0.529 0.243  0.15 0.077
## 2 0.242 0.248 0.252 0.258
## 3 0.553 0.237 0.141 0.069
## 4 0.166 0.218 0.264 0.352
## 5   ...   ...   ...   ...
## 6  0.22 0.241 0.257 0.282
## 7 0.357 0.264 0.216 0.163
## 8 0.464 0.256 0.175 0.104
## 9 0.173 0.221 0.263 0.342
```

We can plot the expected effect of β_1 on the k probabilities using the `num_latent_plot()` function (see Figure 8). Finally we sample using the `sample()` function using the calculated probabilities.

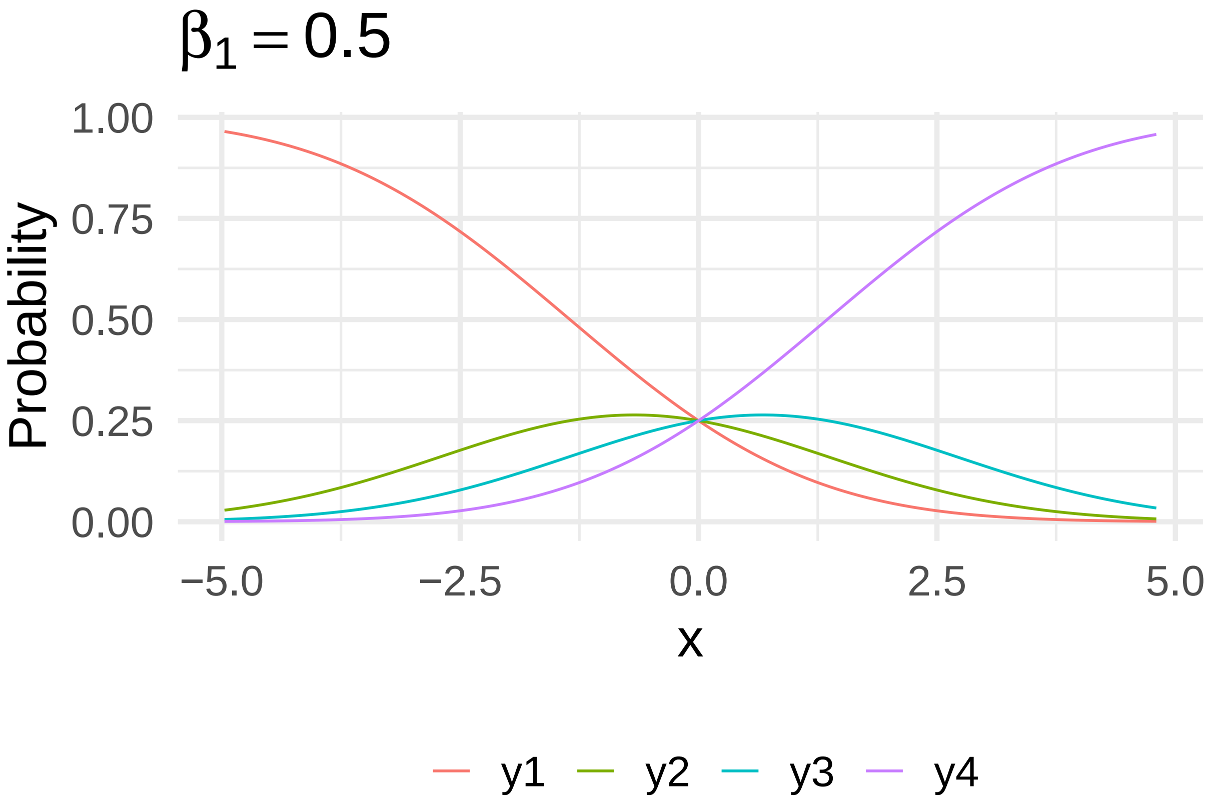

Figure 8. sim-with-probs-lat-plot

To check the simulation result we can increase the number of observations, fit the model using the `ordinal::clm()` function and assess the recovery of simulated parameters.

simulating using latent distribution. An equivalent but more efficient way to simulate an ordinal outcome is using the latent formulation of the model. This require simulating a standard linear regression using the appropriate data generation function (*logistic* or *normal*) and the cutting the latent values according to the thresholds α. The workflow is slighlty different compared to the previous paragraph:

1. Define n (number of observations) and k (number of ordinal outcomes)

2. Define the regression coefficients β
3. Define the baseline probabilities when $x = 0$. These probabilities can be converted to the corresponding thresholds α choosing the appropriate link function (*logit* or *probit*).
4. Calculate the linear predictors η using $k - 1$ equations combining the predictors $\mathbf{X}$, the regression coefficients   and the error   sampled from the appropriate probability distribution.
5. Cut the latent variable into k areas using the thresholds α and assign the corresponding ordinal value. This step simply checks if the latent value $Y_i^\star$ is between two threshold values and assign the corresponding value. This can be done using the `cut()` or the `findInterval()` functions.

The Figure (put crossref) depict the simulated Y^* and the corresponding ordinal value. As for the previous simulation we can fit the model using `clm()` and check the estimated parameters.

The simulation using the latent formulation of the model is implemented in the `sim_ord_latent()` function.

```
sim_ord_latent <- function(location,
                           scale = NULL,
                           By,
                           Bscale = NULL,
                           probs = NULL,
                           alphas = NULL,
                           data,
                           link = c("logit", "probit")){

  # get all the correct functions according to the link
  lf <- get_link(link)
  if(is.null(alphas)){
    # calculate thresholds if not provided
    alphas <- prob_to_alpha(probs, link = link)
  }
```

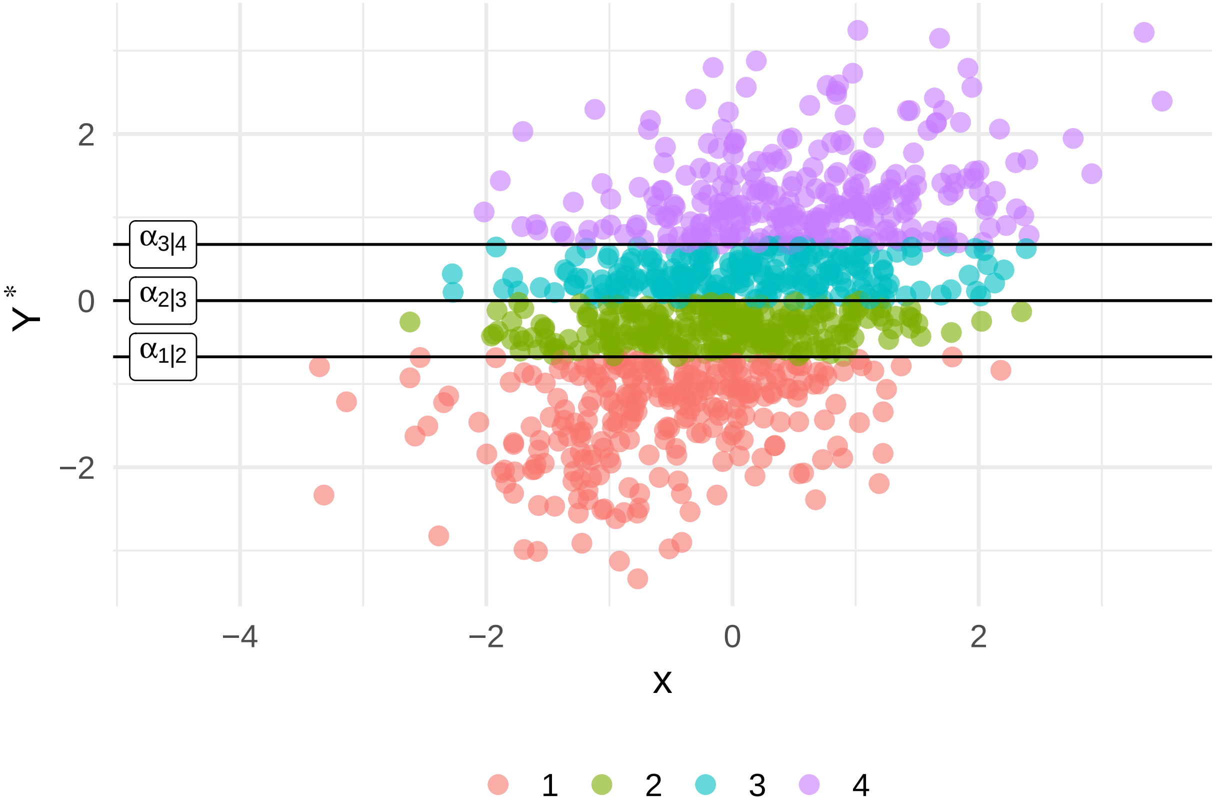

Figure 9

```
k <- length(alphas) + 1 # number of ordinal outcomes
n <- nrow(data) # number of observations

# model matrix for the location effect
Xy <- model.matrix(location, data = data)[, -1, drop = FALSE] # remove intercept
lpy <- c(Xy %*% By) # linear predictor for the location
lps <- 0 # default scale effect 0, exp(0) = 1 (the default scale)

# check predictors on the scale parameter
if(!is.null(scale)){
  # model matrix for the scale effect
  Xsigma <- model.matrix(scale, data = data)[, -1, drop = FALSE] # remove intercept
  lps <- c(Xsigma %*% Bscale) # linear predictor for the scale
}

# latent variable with appropriate error function
ystar <- lf$rfun(n, lpy, exp(lps))
```

```
  # cut according to thresholds
  y <- findInterval(ystar, alphas) + 1 # to start from 1

  data$y <- ordered(y) # to ordered factor
  data$ys <- ystar # save also the latent
  return(data)
}
```

Scale Effects

The default ordinal regression model assume that the variance of th underlying latent distribution is the same across condition. This is similar to a standard linear regression assuming homogeneity of variance. For example, when comparing two groups or conditions we can run a standard linear model (i.e., a t-test) assuming homogeneity of variances or using the Welch t-test (see Delacre, Lakens, & Leys, 2017). In addition, there are the so-called location-scale models that allows to include predictors also for the scale (e.g., the variance) of the distribution. This can be done also in ordinal regression where instead of assuming the same variance between conditions, the linear predictors can be included. Figure **??** depict an example of a comparison between two groups where the two underlyning latent distributions have unequal variance.

spiegare qui che i location-scale models sono sostanzialmente un cambiamento delle soglie (e non dei beta) in funzione di predittori Gerhard Tutz and Berger (2020)

dire che gli scale effects possono essere visti come un'alterativa più parsimoniosa a modelli non-proportional odds Christensen (2019)

se citi brms per modelli più complicati, ricordati di specificare che:

> That is, s = 1/disc. Further, because disc must be strictly positive, it is by default modeled on the log scale.

un modo intuitivo per spiegare gli scale effects è di far capire la relazione tra slope e scale. La slope della funzione cumulative corrisponde alla varianza. Se la slope è piuù steep, la varianza sarà minore e la discrimizione maggiore (in `brms` si chiama desc)

si può citare anche questo Cox (1995) come riferimento per i modelli

una distinzione tra location-shift and location-scale models G. Tutz and Berger (2017) dove i primi oltre ad un cambiamento di location prevedono un cambiamento di soglie mentre i secondi modellano la differenza di soglie come un cambiamento della scala (varianza) che quindi permette di aumetare o diminuire l'area associata ad un determinato intervallo.

sempre in G. Tutz and Berger (2017) propongono il location-scale/shift come una via di mezzo tra il proportional odds ed il non proportional odds

Simulating scale effects. The location-scale model can be simulated using the `sim_ord_latent()` function and providing the predictors for the scale parameters. Given the `log` link function, predictors are provided on the `log` scale. For example, we simulate the effect of a binary variable x representing two independent groups predicting the $k = 5$ response. We simulate a *location* effect of $\beta_1 = 0.5$ (in *probit* scale) and $\zeta_1 = \log(2) = 0.70$. The first group has a $\sigma = 1$ and the second group has $\sigma = 2$. Again we simulate that the baseline probabilities are uniform for the first group.

The table (put ref) reports the simulation results.

Visualizing effects

Before starting with the actual simulation it is useful to plot the predicted probabilities of the response variable y as a function of the predictors. The `cat_latent_plot()` and the `num_latent_plot()` functions are able to visualize the effect of a single categorical or numerical predictor. Specifying the mean and standard deviations of the latent variable in each condition (for the categorical version) or the β (for the numerical version) the function compute the expected probability for each level of the ordinal variable y and visualize the result. In this way the β can be choose in a meaningful way.

```
cat_latent_plot <- function(m = 0,
                            s = 1,
                            th = NULL,
                            probs = NULL,
                            link = c("logit", "probit"),
                            plot = c("probs", "latent", "both")){
  require(ggplot2)
  plot <- match.arg(plot)
  lf <- get_link(link)
  x <- paste0("g", 1:length(m))
  lat <- data.frame(
    x = x,
    m = m,
    s = s
  )

  if(is.null(th)){
    th <- prob_to_alpha(probs, link)
  }

  thl <- latex2exp::TeX(sprintf("$\\alpha_{%s}$", 1:length(th)))

  if(link == "logit"){
    dfun <- distributional::dist_logistic
    s <- sqrt(vlogit(s))
    title <- "Logistic Distribution"
  }else{
    dfun <- distributional::dist_normal
    s <- lat$s
    title <- "Normal Distribution"
  }
  lat$dist <- dfun(lat$m, lat$s)
  range_y <- c(min(lat$m) - max(s) * 5, max(lat$m) + max(s) * 5)

  lat_plot <- ggplot(lat,
         aes(x = factor(x), y = m, dist = dist)) +
```

```r
  ggdist::stat_halfeye(aes(fill = after_stat(factor(findInterval(y, th) + 1))),
               alpha = 0.85) +
  ylim(range_y) +
  geom_hline(yintercept = th, linetype = "dashed", col = "black", alpha = 0.7) +
  geom_line(aes(x = factor(x), y = m, group = 1)) +
  annotate("label", x = 0.7, y = th, label = thl, size = 5) +
  theme_minimal(15) +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        axis.title.x = element_blank()) +
  ylab(latex2exp::TeX("\\mu")) +
  ggtitle(title)



th <- c(-Inf, th, Inf)
lat_ms <- lat[, c("m", "s")]
ps <- apply(lat_ms, 1, function(x) data.frame(t(diff(lf$pfun(th, x[1], x[2])))), simplify = FALSE)
ps <- do.call(rbind, ps)
names(ps) <- paste0("y", 1:ncol(ps))
latl <- cbind(lat, ps)
latl <- tidyr::pivot_longer(latl, starts_with("y"), names_to = "y", values_to = "value")
probs_plot <- ggplot(latl, aes(x = factor(x), y = value, fill = y)) +
  geom_col(position = position_dodge()) +
  theme_minimal(15) +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        axis.title.x = element_blank()) +
  ylab("Probability")
if(plot == "probs"){
  probs_plot
}else if(plot == "latent"){
  lat_plot
}else{
  legend_b <- cowplot::get_legend(
    probs_plot
  )
  plts <- cowplot::plot_grid(
    lat_plot + theme(legend.position = "none"),
    probs_plot + theme(legend.position = "none")
  )
  cowplot::plot_grid(plts, legend_b, ncol = 1, rel_heights = c(1, .1))
  }
}
num_latent_plot <- function(x,
                            b1,
                            th = NULL,
                            probs = NULL,
                            link = c("logit", "probit"),
                            nsample = 1e3,
                            size = 20){
  lf <- get_link(link)
  if(is.null(th)){
    th <- prob_to_alpha(probs, link)
  }
  data <- data.frame(x = seq(min(x), max(x), length.out = nsample))
  data <- get_probs(~x, b1, probs, data, link, append = TRUE)
  datal <- tidyr::pivot_longer(data, starts_with("y"), names_to = "y", values_to = "value")
  ggplot(datal, aes(x = x, y = value, color = y)) +
    geom_line() +
    theme_minimal(size) +
    theme(legend.position = "bottom",
          legend.title = element_blank()) +
    ylab("Probability") +
    ggtitle(latex2exp::TeX(sprintf("$\\beta_1 = %s$", b1)))
}
```

For example, the Figure 10 depicts the effect of a categorical predictor on a 5-level y assuming uniform probabilities for the reference level and a mean difference on a standardized Normal distribution of 1. The Figure 11 depicts the effect of a continuous predictor x sampled from a standard Normal distribution assuming a $\beta = 1$.

Disclaimer about the functions

The current paper proposed a simplified way with some functions to generate ordinal data. For more complex simulations such as simulating correlated ordinal data the `simstudy` package https://kgoldfeld.github.io/simstudy/articles/ordinal.html proposed a very comprehensive set of data simulation function also for ordinal data.

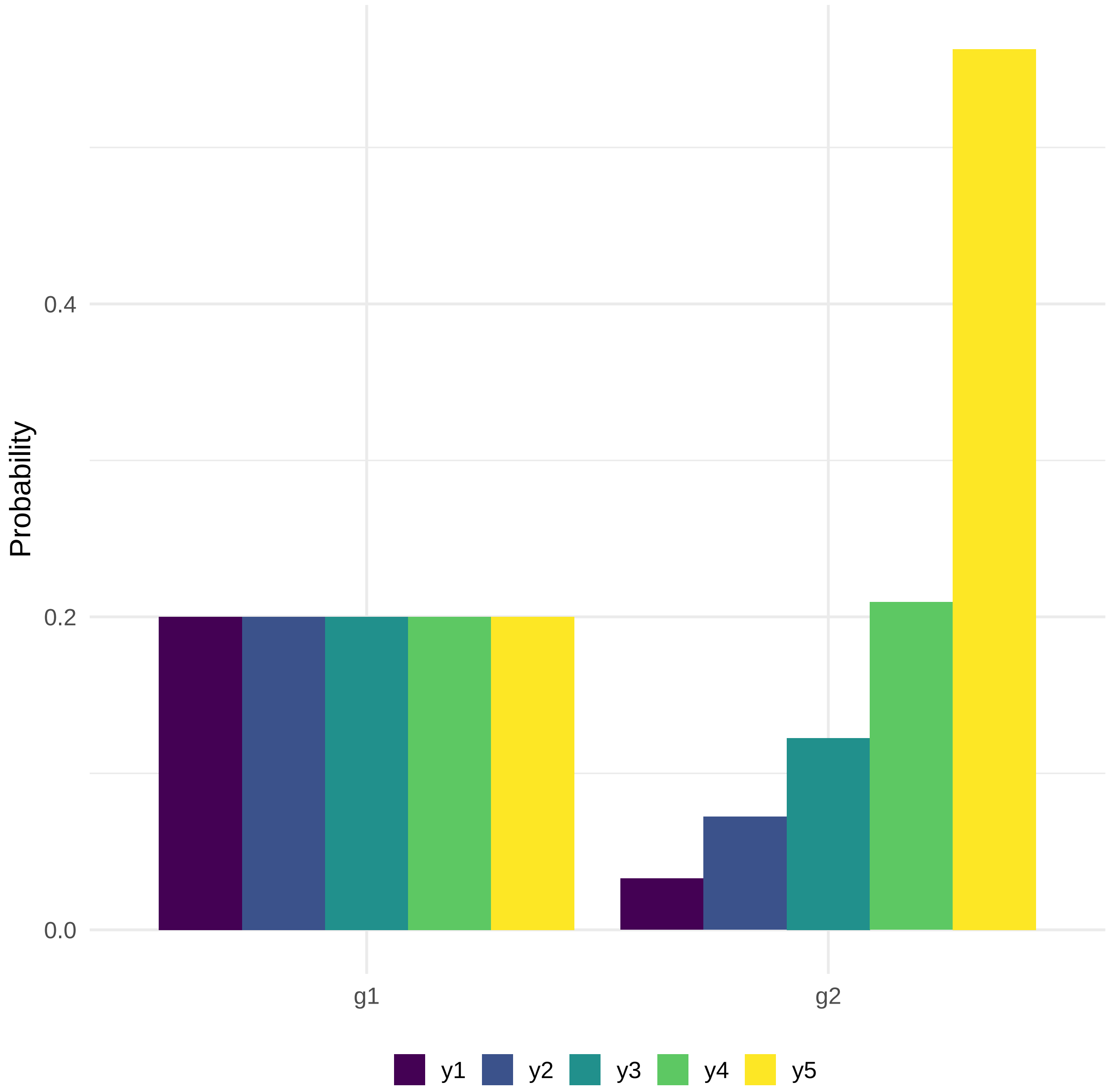

Figure 10. Example categorical

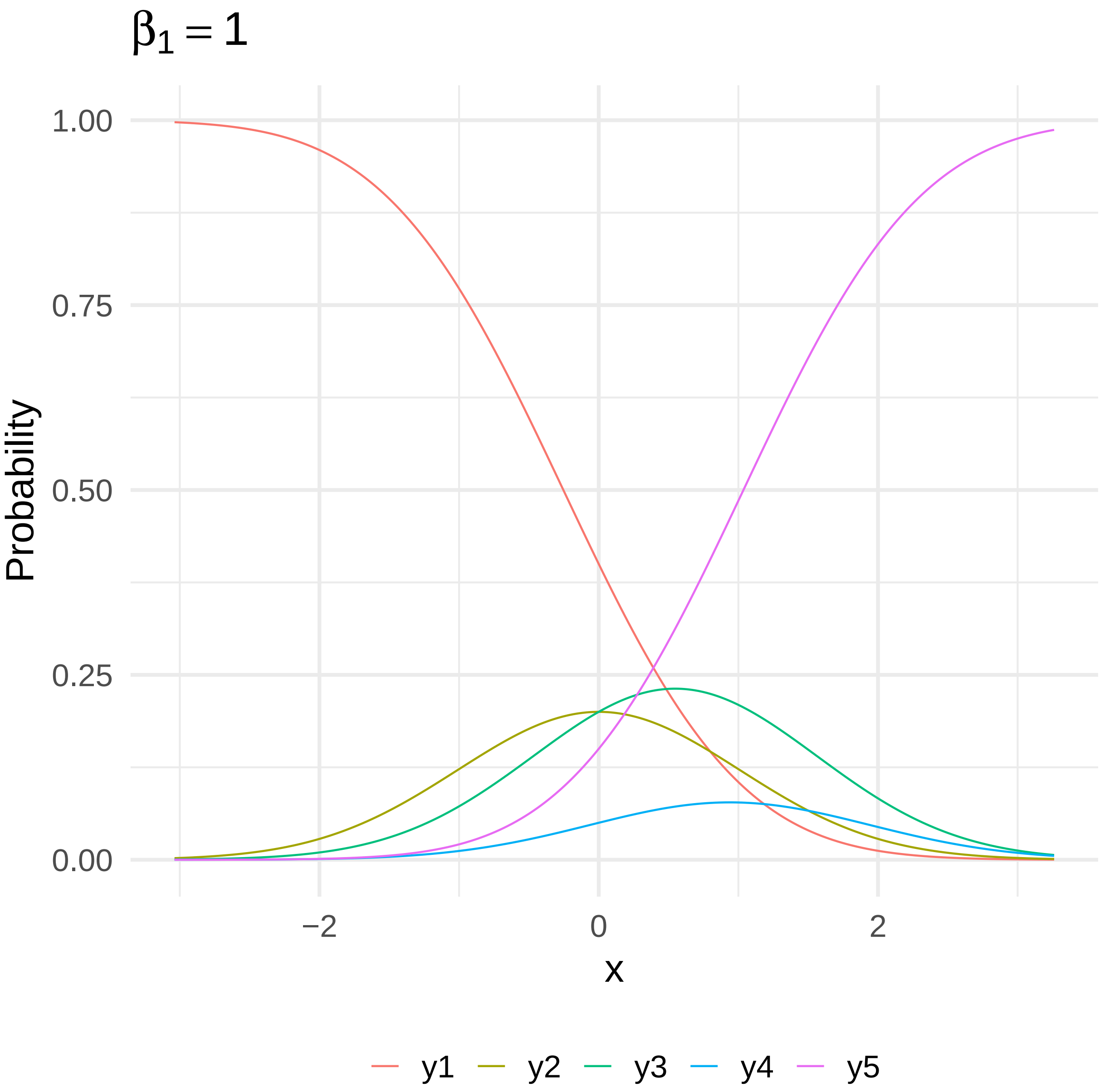

Figure 11. Example numerical

References

Bürkner, P.-C., & Vuorre, M. (2019). Ordinal regression models in psychology: A tutorial. *Advances in Methods and Practices in Psychological Science*, *2*, 77–101. https://doi.org/10.1177/2515245918823199

Christensen, R. H. B. (2019). *Ordinal—regression models for ordinal data.*

Cliff, N. (2016). *Ordinal methods for behavioral data analysis.* London, England: Psychology Press.

Cox, C. (1995). Location-scale cumulative odds models for ordinal data: A generalized non-linear model approach. *Statistics in Medicine*, *14*, 1191–1203. https://doi.org/10.1002/sim.4780141105

DeCarlo, L. T. (1998). Signal detection theory and generalized linear models. *Psychological Methods*, *3*, 186–205. https://doi.org/10.1037/1082-989X.3.2.186

DeCarlo, L. T. (2010). On the statistical and theoretical basis of signal detection theory and extensions: Unequal variance, random coefficient, and mixture models. *Journal of Mathematical Psychology*, *54*, 304–313. https://doi.org/10.1016/j.jmp.2010.01.001

Delacre, M., Lakens, D., & Leys, C. (2017). Why psychologists should by default use welch's *t*-test instead of student's *t*-test. *International Review of Social Psychology*, *30*, 92. https://doi.org/10.5334/irsp.82

Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories.* Cambridge University Press. https://doi.org/10.1017/9781139161879

Green, D. M., Swets, J. A., & Others. (1966). *Signal detection theory and psychophysics* (Vol. 1, pp. xi, 455–xi, 455). Oxford, England: Wiley New York.

Harrell, F. E. (2015). *Regression modeling strategies.* Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-19425-7

Kemp, S., & Grace, R. C. (2021). Using ordinal scales in psychology. *Methods in Psychology (Online)*, *5*, 100054. https://doi.org/10.1016/j.metip.2021.100054

Knoblauch, K., & Maloney, L. T. (2012). *Modeling psychophysical data in r.* New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-4475-6

Kruschke, J. K. (2015). *Doing bayesian data analysis.* https://doi.org/10.1016/c2012-0-00477-2

Kurz, A. S. (2023). *Doing bayesian data analysis in brms and the tidyverse* (Version 1.1.0). Retrieved from https://bookdown.org/content/3686/

Liddell, T. M., & Kruschke, J. K. (2018). Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology*, *79*, 328–348. https://doi.org/10.1016/j.jesp.2018.08.009

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, *22*, 55. Retrieved from https://psycnet.apa.org/record/1933-01885-001

Liu, A., He, H., Tu, X. M., & Tang, W. (2023). On testing proportional odds assumptions for proportional odds models. *General Psychiatry*, *36*, e101048. https://doi.org/10.1136/gpsych-2023-101048

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society*, *42*, 109–127. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x

Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society. Series C, Applied Statistics*, *54*, 507–554. https://doi.org/10.1111/j.1467-9876.2005.00510.x

Robitzsch, A. (2020). Why ordinal variables can (almost) always be treated as continuous variables: Clarifying assumptions of robust continuous and ordinal factor analysis estimation methods. *Frontiers in Education*, *5*, 589965. https://doi.org/10.3389/feduc.2020.589965

Stanislaw, H., & Todorov, N. (1999). Calculation of signal detection theory measures. *Behavior Research Methods, Instruments, & Computers: A Journal of the Psychonomic Society, Inc*, *31*, 137–149. https://doi.org/10.3758/bf03207704

Stevens, S. S. (1946). On the theory of scales of measurement. *Science (New York, N.Y.)*, *103*, 677–680. https://doi.org/10.1126/science.103.2684.677

Tutz, Gerhard. (1990). Sequential item response models with an ordered response. *The British Journal of Mathematical and Statistical Psychology*, *43*, 39–55. https://doi.org/10.1111/j.2044-8317.1990.tb00925.x

Tutz, Gerhard. (2022). Ordinal regression: A review and a taxonomy of models. *Wiley Interdisciplinary Reviews. Computational Statistics*, *14*. https://doi.org/10.1002/wics.1545

Tutz, G., & Berger, M. (2017). Separating location and dispersion in ordinal regression models. *Econometrics and Statistics*, *2*, 131–148. https://doi.org/10.1016/j.ecosta.2016.10.002

Tutz, Gerhard, & Berger, M. (2020). Non proportional odds models are widely dispensable – sparser modeling based on parametric and additive location-shift approaches. Retrieved from http://arxiv.org/abs/2006.03914