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

notation

- idea di mettere F o qualcosa di simile per la pdf (Φ e.g., per la normale)
- $Y^{\star}$ per la latente
- k numero di livelli ordinali
- α per le soglie

per notazione https://people.vcu.edu/~dbandyop/BIOS625/CLM_R.pdf e Bürkner and Vuorre (2019)

per la parte di notazione e quella più formale anche DeCarlo (2010)

cumulative vs others

- cumulative quando una variabile continua latente viene tagliata in modo osservato. Le likert possono essere gestite in questo modo

from Bürkner and Vuorre (2019):

> For many ordinal variables, the assumption of a single underlying continuous variable, as in cumulative models, may not be appropriate. If the response can be understood as being the result of a sequential pro- cess, such that a higher response category is possible only after all lower categories are achieved, the sequen- tial model proposed by Tutz (1990) is usually appropri- ate.

vedi se fare k-1 logistiche è più o meno la stessa cosa

in realtà credo che l'adiaject category e sequential model comunque abbiamo la proportional odds assumption

come sottolinea Bürkner and Vuorre (2019) e Gerhard Tutz and Berger (2020) i sequential model sono adatti per un certo tipo di dati dove l'interesse è vedere l'impatto dei predittori sulla transizione ad un categoria r considerando che la categoria $r-1$ è stata raggiunta.

Bürkner and Vuorre (2019) suggest that while cumulative and sequentials models have a clear empirical distinction the adjiacent category model is usually used for computational advantages. The probability of passing from one category to the other is modeled as a function of predictors.

Risolvere un pochino la nomenclatura

https://cran.r-project.org/web/packages/ordinal/vignettes/clm_article.pdf

> 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.

Da citare in qualche modo Kemp and Grace (2021)

> "As a matter of fact, most of the scales used widely and effectively by psychologists are ordinal scales. In the strictest propriety the ordinary statistics involving means and standard deviations ought not to be used with these scales, for these statistics imply a knowledge of something more than the relative rank-order of data. On the other hand for this 'illegal' statisticizing there can

> be invoked a kind of pragmatic sanction: In numerous instances it leads to fruitful results." (Stevens, 1946, p. 679.)

> It is straightforward to construct ordinal scales that do not involve rank ordering. For example, one can take the first element encountered and arbitrarily assign it the number 100. If the next element encountered is smaller it is given a smaller arbitrary number, 53 say. If the third element is between these two, it can be given the number 86, and so on. If this construction method is used, the difference between the elements assigned the numbers 70 and 80 will not in any important sense be equal to the difference between the elements assigned 90 and 100, and the intervals between the numbers are not really interpretable. Note, too, that monotonic transformations of the scale essentially leave the mea- sure unaffected.

interessante come Cliff (2016) descriva che la maggiorparte dellle domande di ricerca in psicologia siano in riferimento alla location di un costrutto e possano essere gestite in modo ordinale

The use of ordinal data is widespread in Psychology. Usually items from questionnaires are created using Likert scales where a certain psychological traits is measured using an intuitive scale with 3-5 or more anchor points. These measure cannot be considered *metric* measure on interval or ratio scales [stevens] but the categories are ordered.

Other examples here

Despite the usage of ordinal variables, statistical models made for these type of data are rarely used in Psychology. Liddell and Kruschke (2018) reported that the majority of published papers using likert-like measures used standard methods to analyze the data. In practical terms, they use *metric* models where the response variable cannot be considered fully numeric (vedi se c'è un termine tipo scala a rapporti).

altro sui modelli

Theoretically, using a *metric* for ordinal data is not appropriate but understanding the actual impact is not straightforward. Liddell and Kruschke (2018) did a comprehensive work about pitfalls of analzying ordinal data as metric. They showed that metric models produce higher type-1 and type-2 errors compared to the ordinal models. In particular, given the bounded nature of ordinal data, difference in the underlying latent distribution are not always captured by the metric model that simply estimate the mean of the ordinal variable. This is even more relevant when the underluing variance of the ordinal variables are not homogeneous. Furthermore they presented some situations where the metric model could be wrong in the opposite direction, finding an effect with the wrong sign (type-s error).

Finally, they also demonstrated that even in the best condition where e.g. comparing two groups the variance are equal, the ordinal model is more powerful than the metric model given the underestimation of the true effect size from the latter.

> Some authors have argued that, despite the ordinal character of individual Likert items, averaged ordinal items can have an emergent property of an interval scale and so it is appropriate to apply metric methods to the averaged values (e.g., Carifio & Perla, 2007, 2008).

vedi come gestire questo

anche questo è importante > Ordered-probit models typically assume that the thresholds (k) are the same across all groups because the thresholds are theoretically linked to the response measure, not to the predictor value. For example, when asked, "How happy are you?" with response options '1' = very unhappy, '2' = mildly unhappy, '3' =

neutral, ‘4’ = mildly happy, ‘5’ = very happy,” the latent thresholds between ordinal levels are assumed to be implicit in the phrasing of the question, regardless of other aspects of the respondent or situation. In other words, the thresholds are assumed to be part of the measurement procedure, not dependent on the value of the predictor or covariate. This can be technically referred to as a type of measurement invariance.

Da citare come paper iniziale sui modelli (McCullagh, 1980).

questo per la tassonomia dei diversi modelli ordinali (Gerhard Tutz, 2022)

da vedere magari per una critica tipo Gomila sul binary model (Robitzsch, 2020)

Metriv vs ordinal models

- the main difference is that the metric model assign a number to each label of the discrete variable assuming that the distance is the same. (Liddell & Kruschke, 2018)

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
## -4.0357 -0.6735 -0.0024  0.6761  4.0977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0004437  0.0031671    0.14    0.889
## x1          0.9995687  0.0031605  316.27   <2e-16 ***
## x2          0.5046426  0.0031633  159.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.002 on 99997 degrees of freedom
## Multiple R-squared:  0.5577, Adjusted R-squared:  0.5576
## F-statistic: 6.303e+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 -121998.29 244008.57 5(0)  6.33e-08 9.1e+00
##
## Coefficients:
##     x1     x2
## 1.0025 0.5039
##
## Threshold coefficients:
##     1|2     2|3     3|4     4|5
## -0.8418 -0.2514  0.2543  0.8413
##  (Intercept)           x1           x2
## 0.0004430131 0.9980388055 0.5038702203
```

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")
clm_to_ord(fit)
```

```
## $sigma
## [1] 1.776447
##
## $b0
## [1] 2.999548
##
## $beta
## [1] 1.777515
##
## $alpha
## [1] 1.500000 2.549889 3.449757 4.500000
```

Gelman parametrization

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

Odds ratios and cumulative odds ratios

- https://online.stat.psu.edu/stat504/lesson/4/4.1

In this section we introduce the main concept to understand the effects of a logit model.

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, He, Tu, & Tang, 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:

```
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 0.20301120 0.1911931 0.1969380 0.2054224 0.2034352
## 2 0.07967757 0.1014347 0.1483835 0.2414729 0.4290314
```

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:

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

```
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
## -0.9379058 -0.9379058
```

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 1 depicts the assumption of proportional odds in the probability and logit space.

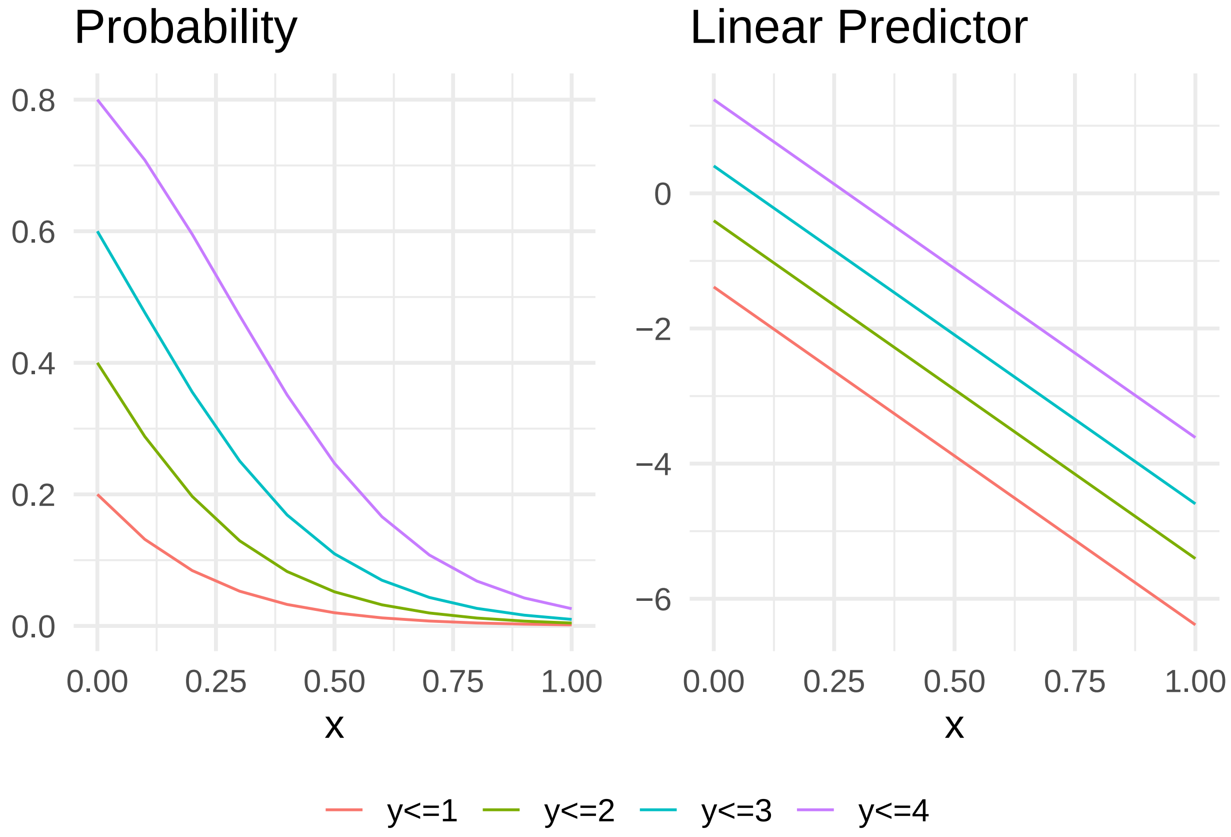

Figure 1. POA assumption

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 2 depicts an example with simulated data satisfying the proportional odds assumption.

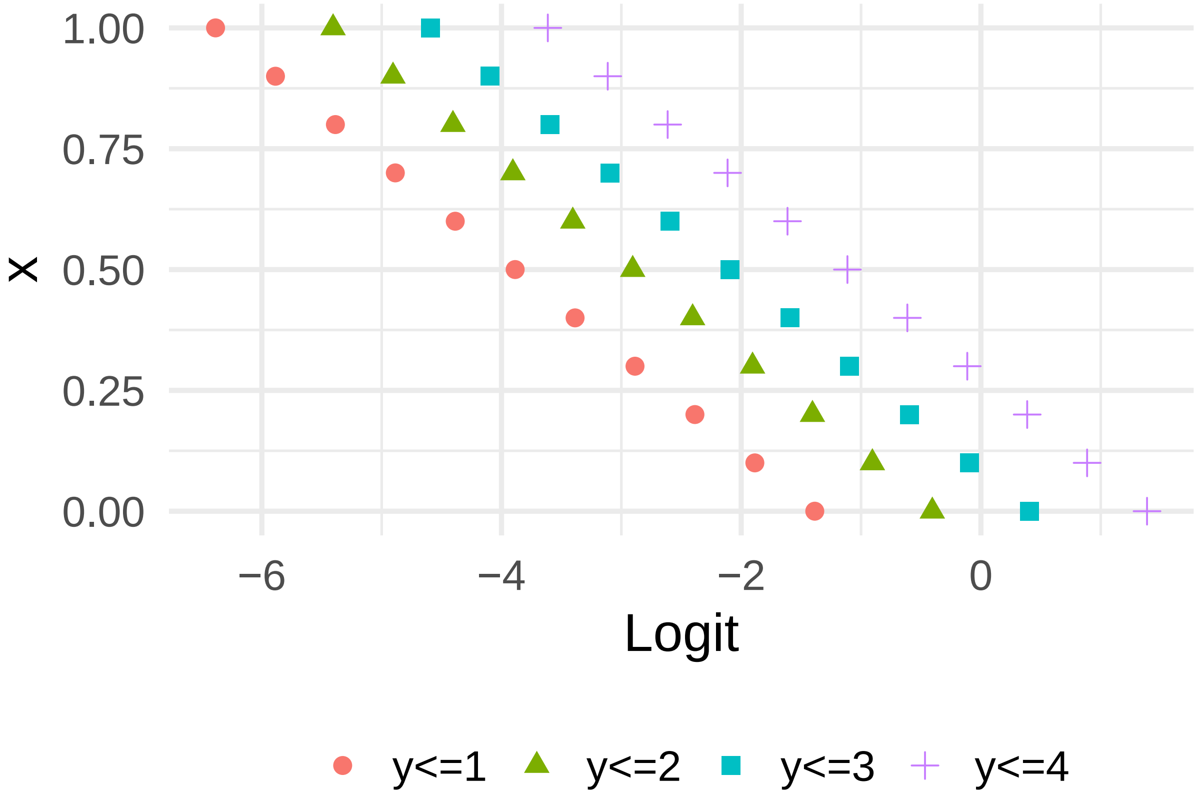

Figure 2. Harrel Poa Plot

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 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")

# 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)  1.0844 -0.0176 -1.1199
## SE          0.0151  0.0163  0.0216
##
## x          -4.9826 -4.9374 -4.9444
## SE          0.0364  0.0481  0.0733
##
```

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 3 depict the two cumulative distributions.

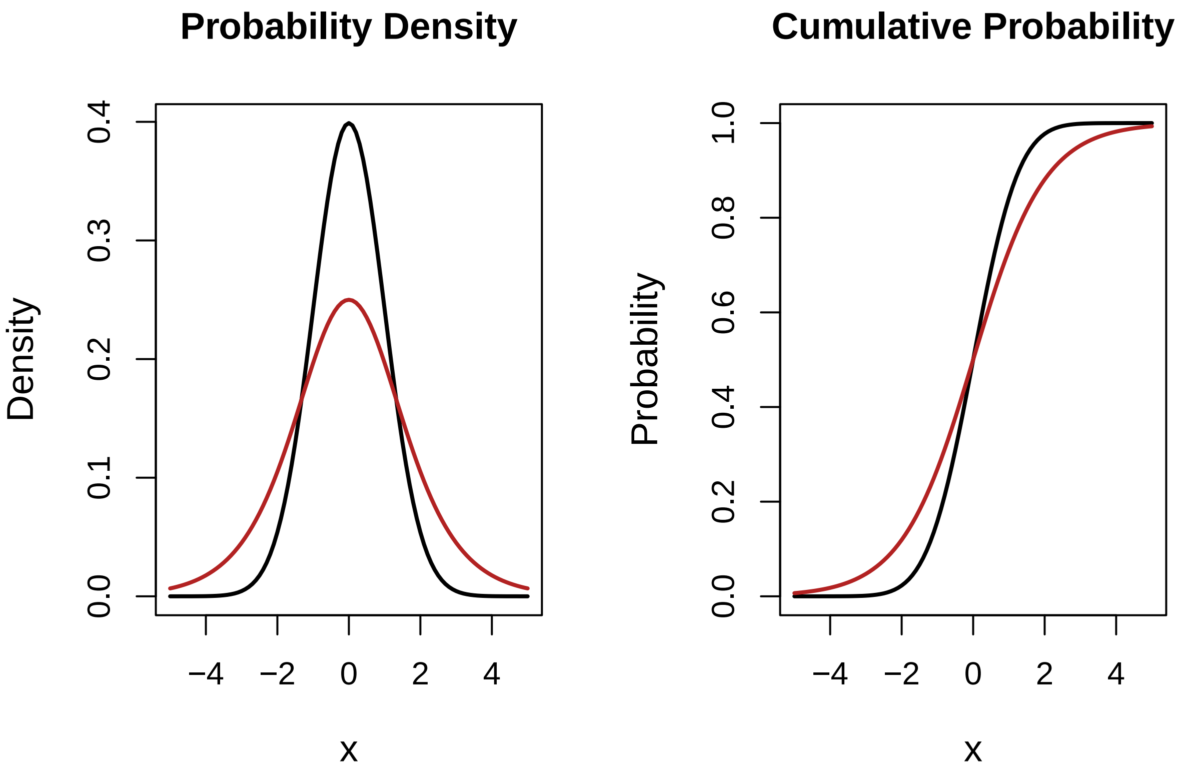

Figure 3. Logit vs Probit

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 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].

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

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

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.

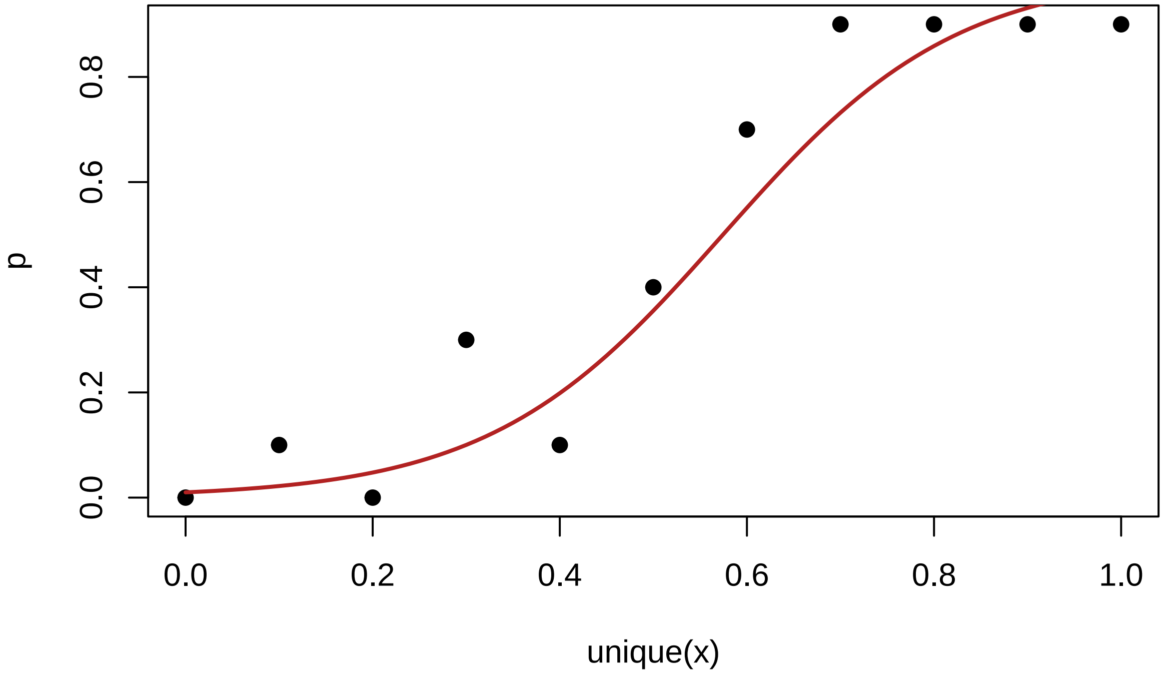

Figure 4

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 $p(y \leq 1|x), p(y \leq 2|x), ...p(y \leq k - 1|x)$.
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.

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 0.481 0.254 0.169 0.096
## 2 0.092 0.165 0.252 0.491
```

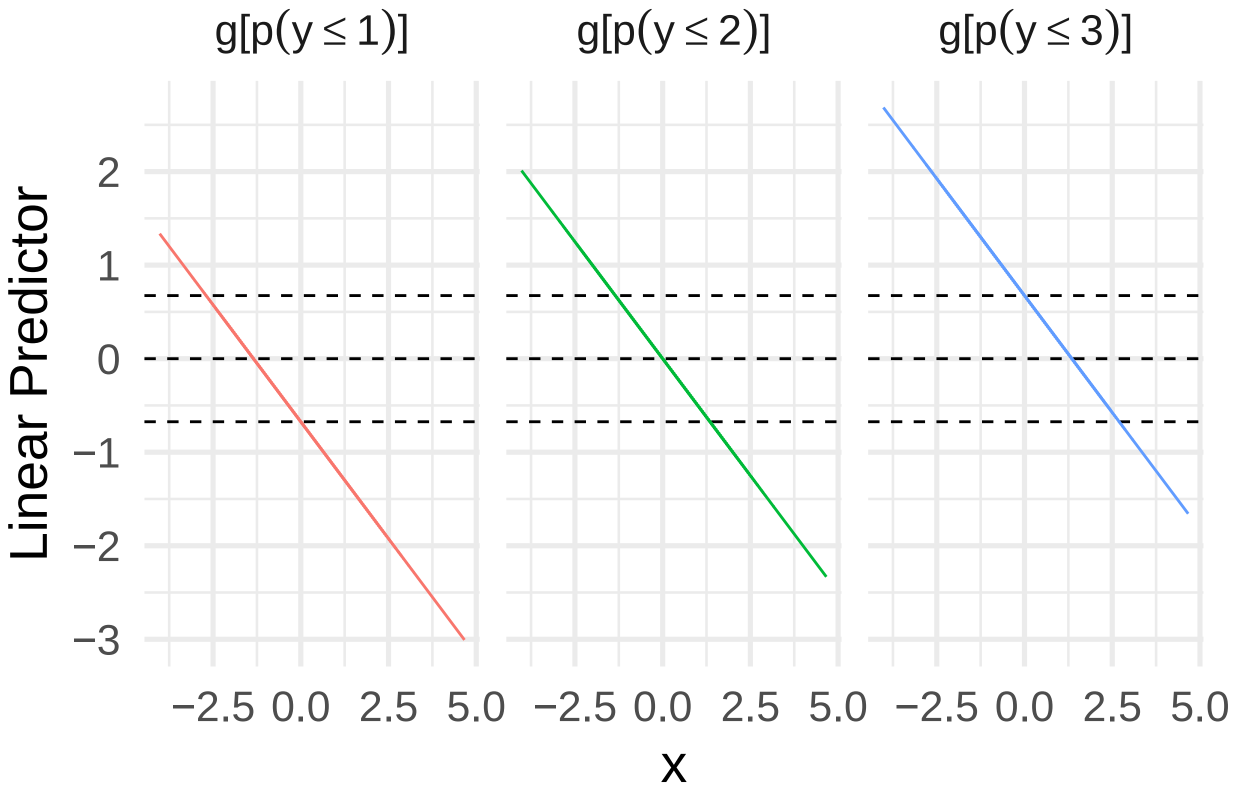

Figure 5

```
## 3 0.098  0.17 0.254 0.478
## 4 0.426 0.261  0.19 0.122
## 5   ...   ...   ...   ...
## 6  0.11  0.18 0.258 0.452
## 7 0.315 0.261 0.231 0.193
## 8 0.108 0.179 0.258 0.456
## 9 0.065 0.136 0.234 0.565
```

We can plot the expected effect of β_1 on the k probabilities using the `num_latent_plot()` function (see Figure 6). Finally we sample using the `sample()` function using the calculated probabilities.

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.

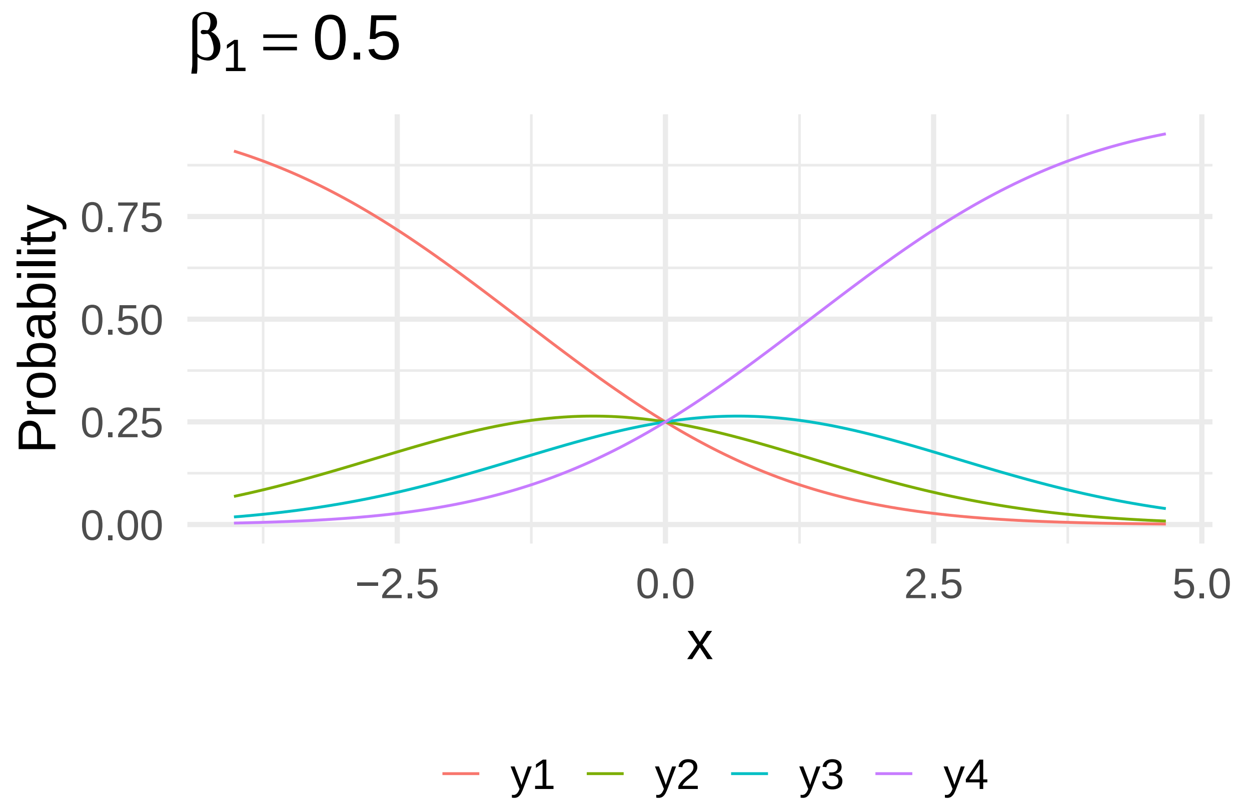

Figure 6. sim-with-probs-lat-plot

coef.type	term	estimate	std.error	statistic	p.value	row
$$\alpha$$	1|2	-0.680 (*-0.674*)	0.004	−152.618	0.000	1.000
	2|3	-0.005 (*0.000*)	0.004	−1.179	0.238	2.000
	3|4	0.666 (*0.674*)	0.004	149.925	0.000	3.000
$$\mu$$	x	0.495 (*0.500*)	0.004	132.696	0.000	1.000

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.

coef.type	term	estimate	std.error	statistic	p.value	row
$$\alpha$$	1|2	-0.681 (*-0.674*)	0.004	−152.568	0.000	1.000
	2|3	-0.007 (*0.000*)	0.004	−1.708	0.088	2.000
	3|4	0.664 (*0.674*)	0.004	149.541	0.000	3.000
$$\mu$$	x	0.498 (*0.500*)	0.004	133.483	0.000	1.000

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

```
sim_ord_latent <- function(location,
                           scale = NULL,
```

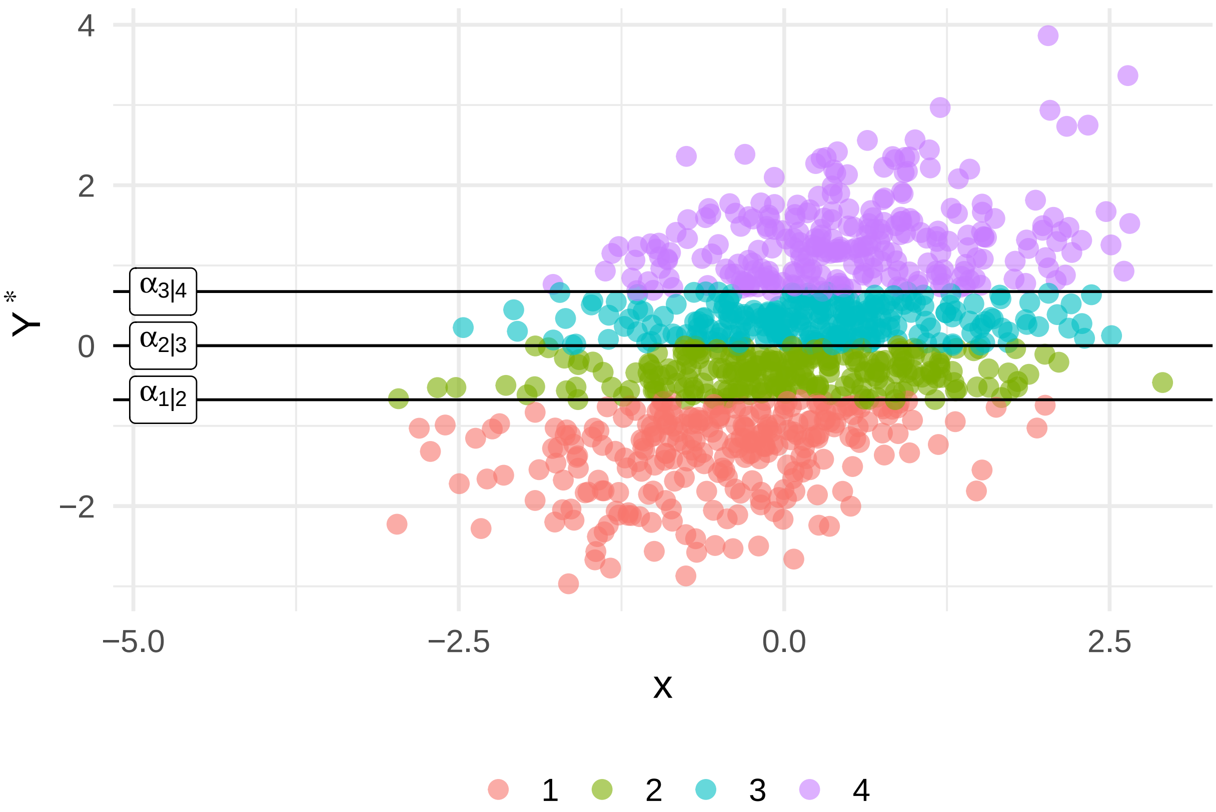

Figure 7

```
                         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)
```

```r
  }
  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.

```r
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)) +
  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])))), simp
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 = "valu
  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 = "valu
  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 8 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 9 depicts the effect of a continuous predictor x sampled from a standard Normal distribution assuming a $\beta = 1$.

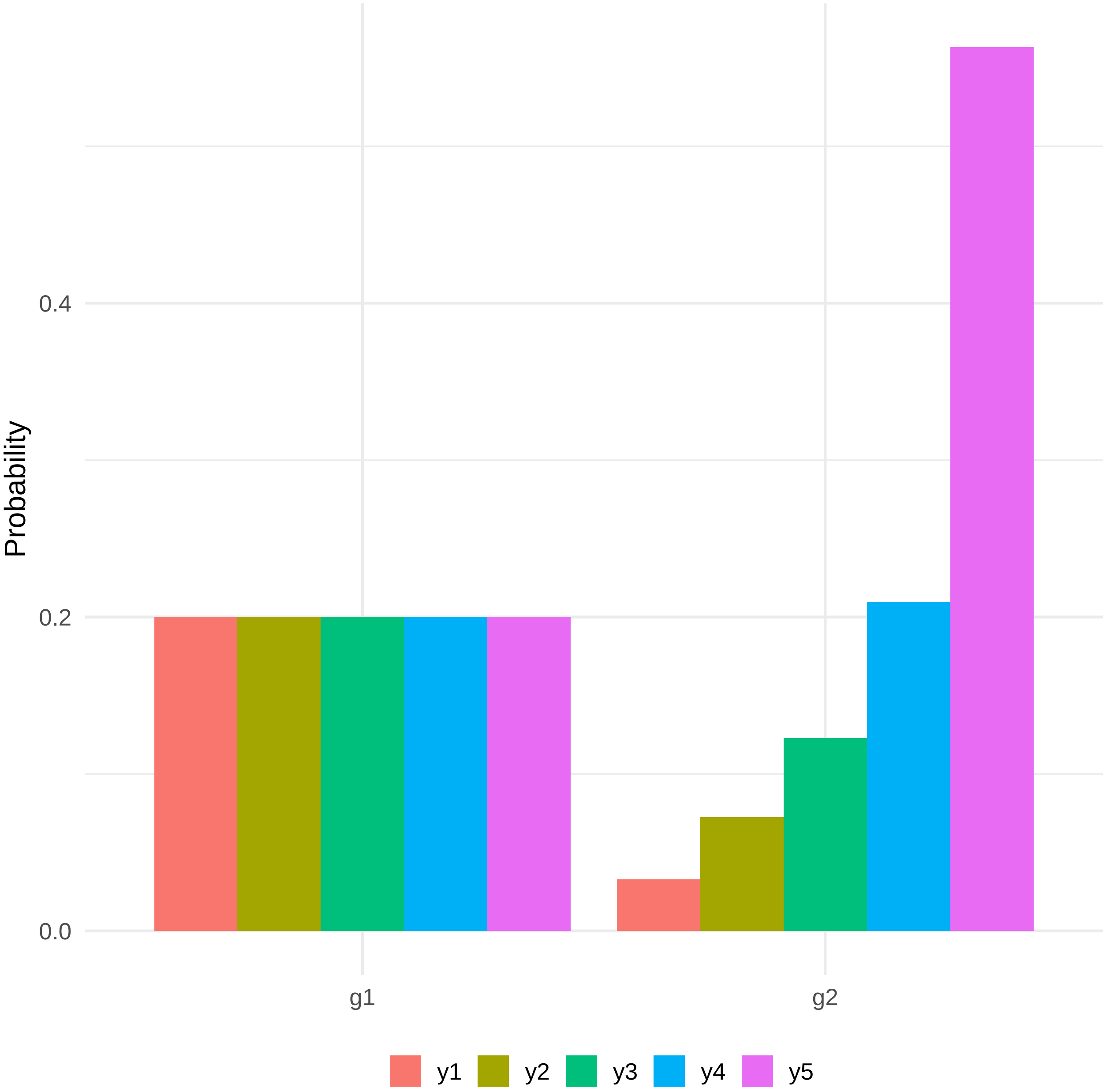

Figure 8. Example categorical

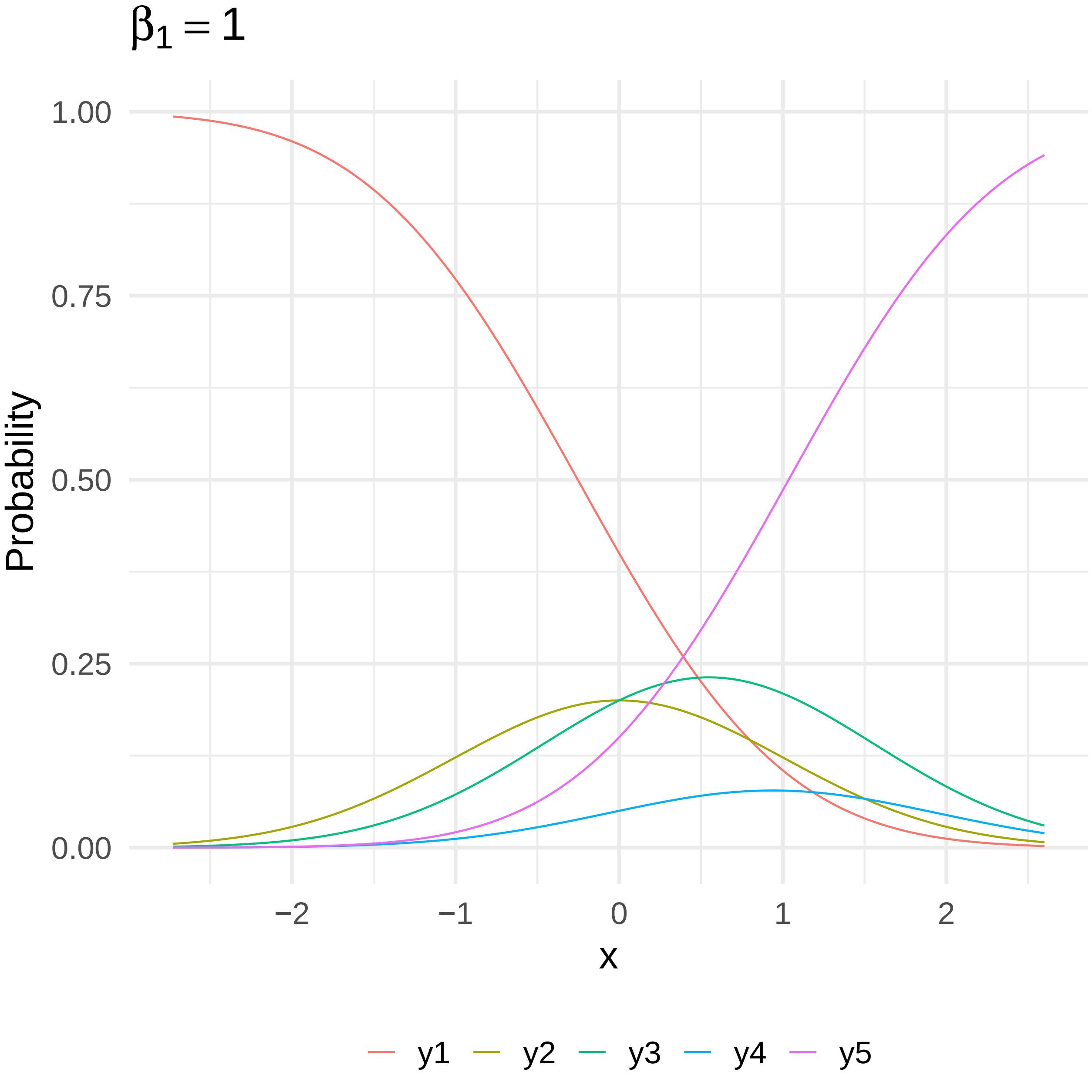

Figure 9. Example numerical

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.

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

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

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

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