A tutorial on fitting Generalized Linear Models for categorical responses with the glmcat package

Lorena León* Jean Peyhardi† Catherine Trottier‡

Abstract

In statistical modeling, there is a wide variety of regression models for categorical responses. Yet, no software encapsulates all of these models in a standardized format. We introduce and illustrate the utility of glmcat, the R package we developed to estimate generalized linear models implemented under the unified specification (r, F, Z), where r represents the ratio of probabilities (reference, cumulative, adjacent, or sequential), F the cumulative cdf function for the linkage, and Z the design matrix. We present the properties of the four families of models, which must be investigated when selecting the components r, F, and Z. The functions are user-friendly and fairly intuitive; offering the possibility to choose from a large range of models through a combination (r, F, Z).

Introduction to the *(r,F,Z)* methodology:

A generalized linear model is characterized by three components: 1) the *random component* that defines the conditional cdf of the response variable Y_i given the realization of the explanatory variables $\boldsymbol{x_i}$; 2) the *systematic component* which is determined by the *linear predictor* η (that specifies the linear entry of the independent variables), and 3) the link function g that relates the expected response and the linear predictor.

The random component of a GLM for a categorical response with J categories is the multinomial cdf with vector of probabilities $(\pi_1, ..., \pi_J)$ where $\sum \pi_r = 1$. The linear predictor $(\eta_1, ..., \eta_{J-1})$ can be written as the product of the design matrix Z and the unknown parameter vector β. The link function which characterizes this model is given by the equation $g(\pi) = Z\beta$, with $J-1$ equations $g_j = \eta_j$. Peyhardi, Trottier, and Guédon (2015) proposed to write the link function as

$$g_j = F^{-1} \circ r_j \Leftrightarrow r_j = F(\eta_j) \qquad j = 1, 2, ..., J-1 \tag{1}$$

where F is a cumulative cdf function and $r = (r_1, ..., r_{J-1})$ is a transformation of the expected value vector. In the following, we will describe in more details the components *(r,F,Z)* and their modalities.

Ratio of probabilities *r*

The linear predictor is not directly related to the expectation $\boldsymbol{\pi}$ instead they are related through a particular transformation $\boldsymbol{r}$ of the vector $\boldsymbol{\pi}$ which is called the ratio. Peyhardi, Trottier, and Guédon (2015) proposed four ratios that gather the alternatives to model categorical response data:

$r_j(\boldsymbol{\pi})$	Cumulative $\pi_1 + ... + \pi_j$	Sequential $\frac{\pi_j}{\pi_j + ... + \pi_J}$	Adjacent $\frac{\pi_j}{\pi_j + \pi_{j+1}}$	Reference $\frac{\pi_j}{\pi_j + \pi_J}$
Y	ordinal			nominal

*Université de Montpellier, ylorenaleonv@gmail.com
†Université de Montpellier, jean.peyhardi@umontpellier.fr
‡Université de Montpellier, catherine.trottier@umontpellier.fr

Each component $r_j(\boldsymbol{\pi})$ can be viewed as a (conditional) probability. For the reference ratio, each category j is compared to the reference category J. For the adjacent ratio, each category j is compared to its adjacent category $j+1$. For the cumulative ratio, the probabilities of categories are cumulated. For the sequential ratio, each category j is compared to its following category, $j+1,\ldots,J$. The adjacent, cumulative and sequential ratios all rely on an ordering assumption among categories. The reference ratio is devoted to nominal responses.

Cumulative cdf function *F*

The cumulative cdf functions (distributions) available in `glmcat` to fit the models are: *logistic*, *normal*, *Cauchy*, *Student (with any df)*, *Gompertz* and *Gumbel*. The *logistic* and *normal* distributions are the symmetric distributions most commonly used to define link functions in generalized linear models. However, for specific scenarios, the use of other distributions may result in a more accurate fit. An example is presented by Bouscasse, Joly, and Peyhardi (2019), where the employment of the *Student* cdf leaded to a better fit for a modeling exercise on travel choice data. For the asymmetric case, the *Gumbel* and *Gompertz* distributions are the most commonly used.

Design Matrix *Z*

It is possible to impose restrictions on the thresholds, or on the effects of the covariates, for example, for them to vary or not according to the response categories.

- Constraints on the effects:

It is plausible for a predictor to have specific level of impact on the different categories of the response. Thus, the $J-1$ linear predictors are of the form: $\eta_j = \alpha_j + x'\delta_j$ with $\beta = (\alpha_1, ..., \alpha_{J-1}, \delta'_1, ..., \delta'_{J-1})$. And, its associated design matrix is:

$$Z_c = \begin{pmatrix} 1 & & & x^t & & \\ & \ddots & & & \ddots & \\ & & 1 & & & x^t \end{pmatrix}_{(J-1)\times(J-1)(1+p)} \tag{2}$$

Another case is to constrain the effects of the covariates to be constant across the response categories. Therefore, there is only a global effect that is not specific to the response categories, this is known as the parallelism assumption, for which the constrained space is represented by:

$$Z_p = \begin{pmatrix} 1 & & & x^t \\ & \ddots & & \vdots \\ & & 1 & x^t \end{pmatrix}_{(J-1)\times(J-1+p)} \tag{3}$$

The first case (Z_c) is named by Peyhardi, Trottier, and Guédon (2015) as the *complete* design, whereas the second (Z_p) as the *parallel* design. These two matrices are sufficient to define all the classical models. A third option is to consider both kind of effects, complete and parallel, this in known as *partial parallel design*

$$Z = \begin{pmatrix} 1 & & x_k^t & & & x_l^t \\ & \ddots & & & \ddots & \vdots \\ & & 1 & & x_k^t & x_l^t \end{pmatrix}_{(J-1)\times((J-1)(1+K)+L)} \tag{4}$$

- Constraints on the intercepts:
  For the particular case of the *cumulative* ratio the *equidistant* constraint considers that the distances

between adjacent intercepts are the same for all the pairs $(j, j+1)$, therefore we can write the intercepts as

$$\alpha_j = \alpha_1 + (j-1)\theta \tag{5}$$

this restriction implies that only two parameters (α_1, the first threshold, and, θ the spacing) have to be estimated regardless the number of categories.

All the classical models for categorical response data, can be written as an *(r,F,Z)* triplet, as examples:

- The multinomimal model $\equiv (Reference, Logistic, Complete)$
- The odds parallel logit model $\equiv (Cumulative, Logistic, parallel)$
- The parallel hazard model $\equiv (Sequential, Gompertz, parallel)$
- The continuation ratio logit model $\equiv (Sequential, Logistic, Complete)$
- The adjacent logit model $\equiv (Adjacent, Logistic, Complete)$

Fitting *(r,F,Z)* with the `glmcat` package

Family of reference models

We used the 223 observations of the *boy's disturbed dreams* benchmark dataset drawn from a study that cross-classified boys by their age x and the severity of their disturbed dreams y (Maxwell 1961). The data is available as the object `DisturbedDreams` in the package `glmcat`.

For more information see the manual entry for the `DisturbedDreams` data: `help(DisturbedDreams)`.

```
data("DisturbedDreams")
summary(DisturbedDreams)
```

```
##       Age               Level
##  Min.   : 6.00   Not.severe :100
##  1st Qu.: 8.50   Severe.1   : 42
##  Median :10.50   Severe.2   : 41
##  Mean   :10.96   Very.severe: 40
##  3rd Qu.:12.50
##  Max.   :14.50
```

We will fit the model $(Reference, Logistic, Complete)$ to the `DisturbedDreams` data using the function `glmcat`. We save the fitted `glmcat` model in the object `mod_ref_log_c` and we print it by simply typing its name:

```
mod_ref_log_c <- glmcat(
  formula = Level ~ Age, ratio = "reference",
  cdf = "logistic", ref_category = "Very.severe",
  data = DisturbedDreams
)
```

The most common **R** functions which describe different model features are available for the objects in `glmcat`

- The summary of the object:

```
summary(mod_ref_log_c)
```

```
## Level ~ Age
##                 ratio      cdf nobs niter    logLik
## Model info: reference logistic  223     5 -277.1345
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -2.45444    0.84559  -2.903   0.0037 **
## (Intercept) Severe.1   -0.55464    0.89101  -0.622   0.5336
## (Intercept) Severe.2   -1.12464    0.91651  -1.227   0.2198
## Age Not.severe          0.30999    0.07804   3.972 7.13e-05 ***
## Age Severe.1            0.05997    0.08582   0.699   0.4847
## Age Severe.2            0.11228    0.08684   1.293   0.1960
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The number or observations:

```
nobs(mod_ref_log_c)
```

```
## [1] 223
```

- The coefficients of the model

```
coef(mod_ref_log_c)
```

```
##                               [,1]
## (Intercept) Not.severe -2.45443827
## (Intercept) Severe.1   -0.55463962
## (Intercept) Severe.2   -1.12464112
## Age Not.severe          0.30998759
## Age Severe.1            0.05997162
## Age Severe.2            0.11228063
```

- The LogLikelihood

```
logLik(mod_ref_log_c)
```

```
## 'log Lik.' -277.1345 (df=6)
```

- Information criteria

```
AIC(mod_ref_log_c)
```

```
## [1] 566.2691
```

```
BIC(mod_ref_log_c)
```

```
## [1] 586.7121
```

It is possible to do predictions in `glmcat` using the function `predict_glmcat`. We are going to predict the response for 3 random observations:

```
# Random observations
set.seed(13)
ind <- sample(x = 1:nrow(DisturbedDreams), size = 3)
# Probabilities

predict(mod_ref_log_c, newdata = DisturbedDreams[ind, ], type = "prob")
```

```
##      Not.severe  Severe.1  Severe.2 Very.severe
## [1,]  0.5392049 0.1583321 0.1721826   0.1302804
## [2,]  0.1832207 0.2732519 0.2115046   0.3320229
## [3,]  0.2996414 0.2391879 0.2110034   0.2501674
```

```
# Linear predictor
predict(mod_ref_log_c, newdata = DisturbedDreams[ind, ], type = "linear.predictor")
```

```
##      Not.severe    Severe.1   Severe.2
## [1,]  1.4204066  0.19500566  0.2788668
## [2,] -0.5945128 -0.19480988 -0.4509573
## [3,]  0.1804562 -0.04488083 -0.1702558
```

Now we illustrate how to predict in a set of new observations. Suppose we want to predict the severity of dreams for 3 individuals whose ages are *5, 9.5* and *15* respectively:

```
# New data
# Age <- c(5, 9.5, 15)
# predict(mod_ref_log_c, newdata = Age, type = "prob")
```

Assume that we are interested in making the effect of the predictor variable parallel, to that end, we type the name of the predictor variable as the input for the parameter `parallel`. The model to fit corresponds to the triplet *(Reference, Logistic, parallel)*:

```
mod2 <- glmcat(
  formula = Level ~ Age, cdf = "logistic",
  parallel = "Age", ref_category = "Very.severe",
  data = DisturbedDreams
)
summary(mod2)
```

```
## Level ~ Age
##                  ratio      cdf nobs niter  logLik
## Model info: reference logistic  223     5 -284.24
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -1.14696    0.71912  -1.595  0.11073
```

```
## (Intercept) Severe.1   -2.01446    0.72866  -2.765  0.00570 **
## (Intercept) Severe.2   -2.03856    0.72906  -2.796  0.00517 **
## Age                     0.19585    0.06845   2.861  0.00422 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
logLik(mod2)
```

```
## 'log Lik.' -284.24 (df=4)
```

Another variation of the reference model is obtained at changing the cdf function. Let's now fit the model *(Reference, Student (0.5), Complete)*:

```
mod3 <- glmcat(
  formula = Level ~ Age, ref_category = "Very.severe",
  data = DisturbedDreams, cdf = list("student",0.5)
)
summary(mod3)
```

```
## Level ~ Age
##                  ratio     cdf nobs niter    logLik
## Model info: reference student  223    11 -279.7941
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -4.48612    1.84226  -2.435  0.01489 *
## (Intercept) Severe.1   -0.32543    0.76910  -0.423  0.67220
## (Intercept) Severe.2   -0.81958    0.84449  -0.971  0.33180
## Age Not.severe          0.51793    0.19912   2.601  0.00929 **
## Age Severe.1            0.02467    0.07223   0.342  0.73272
## Age Severe.2            0.07042    0.07720   0.912  0.36166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
logLik(mod3)
```

```
## 'log Lik.' -279.7941 (df=6)
```

Family of adjacent models

The equivalence between $(Adjacent, Logistic, Complete)$ and $(Reference, Logistic, Complete)$ models is shown by comparing the associated LogLikelihood of both models:

```
logLik(mod_ref_log_c) # recall (ref,logit,com)
```

```
## 'log Lik.' -277.1345 (df=6)
```

```
mod_adj_log_c <- glmcat(
  formula = Level ~ Age, ratio = "adjacent",
  data = DisturbedDreams, cdf = "logistic"
)
logLik(mod_adj_log_c)
```

```
## 'log Lik.' -279.5628 (df=4)
```

```
summary(mod_adj_log_c)
```

```
## Level ~ Age
##                ratio      cdf nobs niter    logLik
## Model info: adjacent logistic  223     5 -279.5628
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -0.23611    0.33657  -0.702  0.48297
## (Intercept) Severe.1   -1.01875    0.33535  -3.038  0.00238 **
## (Intercept) Severe.2   -0.95464    0.31524  -3.028  0.00246 **
## Age                     0.09730    0.02405   4.045 5.23e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Remark that despite the fact that the LogLikelihoods are equal, the parameters estimations are different ($\alpha \neq \alpha'$). Defining the matrix A^T as follows:

$$A^T = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

we can check that $A^T * \alpha = \alpha'$.

Note: The adjacent models are stable under the reverse permutation.

$(Adjacent, Cauchy, Complete)$

```
mod_adj_cau_c <- glmcat(
  formula = Level ~ Age,
  ratio = "adjacent", cdf = "cauchy",
  categories_order = c("Not.severe", "Severe.1", "Severe.2", "Very.severe"),
  data = DisturbedDreams
)
logLik(mod_adj_cau_c)
```

```
## 'log Lik.' -280.116 (df=4)
```

```
summary(mod_adj_cau_c)
```

```
## Level ~ Age
##                ratio    cdf nobs niter   logLik
## Model info: adjacent cauchy  223     6 -280.116
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -0.18005    0.28499  -0.632 0.527526
## (Intercept) Severe.1   -0.83297    0.29215  -2.851 0.004356 **
## (Intercept) Severe.2   -0.78360    0.26287  -2.981 0.002874 **
## Age                     0.08008    0.02083   3.845 0.000121 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Adjacent, Cauchy, Complete) with reversed order

```r
mod_adj_cau_c_rev <- glmcat(
  formula = Level ~ Age,
  ratio = "adjacent", cdf = "cauchy",
  categories_order = c("Very.severe", "Severe.2", "Severe.1", "Not.severe"),
  data = DisturbedDreams
)
logLik(mod_adj_cau_c_rev)
```

```
## 'log Lik.' -280.116 (df=4)
```

```r
summary(mod_adj_cau_c_rev)
```

```
## Level ~ Age
##                ratio    cdf nobs niter   logLik
## Model info: adjacent cauchy  223     6 -280.116
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept) Very.severe  0.78360    0.26287   2.981 0.002874 **
## (Intercept) Severe.2     0.83297    0.29215   2.851 0.004356 **
## (Intercept) Severe.1     0.18005    0.28499   0.632 0.527526
## Age                     -0.08008    0.02083  -3.845 0.000121 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The LogLikelihoods of the last two models are the same, this is because the *Cauchy* cdf is symmetric; for non symmetric distributions this is not longer true. Note that if the Gumbel cdf is used with the reverse order, then, its LogLikelihood is equal to the model using *Gompertz* as the cdf, this is because the *Gumbel* cdf is the symmetric of the *Gompertz* cdf. Otherwise, the parameter estimations are reversed:

(Adjacent, Gumbel, parallel)

```r
adj_gumbel_p <- glmcat(
  formula = Level ~ Age,
  ratio = "adjacent", cdf = "gumbel",
  categories_order = c("Not.severe", "Severe.1", "Severe.2", "Very.severe"),
  parallel = c("(Intercept)", "Age"),
  data = DisturbedDreams
)
logLik(adj_gumbel_p)
```

```
## 'log Lik.' -284.0416 (df=2)
```

```r
summary(adj_gumbel_p)
```

```
## Level ~ Age
##                ratio    cdf nobs niter    logLik
## Model info: adjacent gumbel  223     5 -284.0416
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.28023    0.20340  -1.378    0.168
## Age          0.08385    0.01909   4.392 1.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Adjacent, Gompertz, parallel)

```
adj_gompertz_rev <- glmcat(
  formula = Level ~ Age,
  ratio = "adjacent", cdf = "gompertz",
  categories_order = c("Very.severe", "Severe.2", "Severe.1", "Not.severe"),
  parallel = c("(Intercept)", "Age"),
  data = DisturbedDreams
)
logLik(adj_gompertz_rev)
```

```
## 'log Lik.' -284.0416 (df=2)
```

```
summary(adj_gompertz_rev)
```

```
## Level ~ Age
##                ratio      cdf nobs niter    logLik
## Model info: adjacent gompertz  223     5 -284.0416
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.28023    0.20340   1.378    0.168
## Age         -0.08385    0.01909  -4.392 1.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Family of sequential models

The sequential ratio, which assumes a binary process at each transition, higher levels can be reached only if previous levels where reached at a earlier stage.

(Sequential, Normal, Complete)

```
seq_probit_c <- glmcat(
  formula = Level ~ Age,
  ratio = "sequential", cdf = "normal",
  data = DisturbedDreams
)
logLik(seq_probit_c)
```

```
## 'log Lik.' -280.5465 (df=4)
```

```
summary(seq_probit_c)
```

```
## Level ~ Age
##                  ratio    cdf nobs niter    logLik
## Model info: sequential normal  223     6 -280.5465
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -1.20313    0.27969  -4.302 1.69e-05 ***
## (Intercept) Severe.1   -1.41347    0.28345  -4.987 6.14e-07 ***
## (Intercept) Severe.2   -0.98393    0.28252  -3.483 0.000496 ***
## Age                     0.09752    0.02414   4.039 5.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Family of cumulative models

(Cumulative, Logistic, Complete)

```
cum_log_co <- glmcat(
  formula = Level ~ Age,
  cdf = "logistic",
  ratio = "cumulative",
  data = DisturbedDreams
)
logLik(cum_log_co)
```

```
## 'log Lik.' -278.4682 (df=4)
```

```
summary(cum_log_co)
```

```
## Level ~ Age
##                   ratio      cdf nobs niter    logLik
## Model info: cumulative logistic  223     6 -278.4682
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -2.60639    0.56166  -4.640 3.48e-06 ***
## (Intercept) Severe.1   -1.78157    0.54641  -3.260  0.00111 **
## (Intercept) Severe.2   -0.77714    0.53923  -1.441  0.14953
## Age                     0.21875    0.04949   4.420 9.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The function *glmcat* has special features for the cumulative models. The option for the thresholds to be equidistant is a characteristic of interest for the family of cumulative models:

(Cumulative, Logistic, Equidistant)

```
cum_log_co_e <- glmcat(
  formula = Level ~ Age,
  cdf = "logistic",
  ratio = "cumulative",
  data = DisturbedDreams,
  parallel = "Age",
  threshold = "equidistant",
)
logLik(cum_log_co_e)
```

```
## 'log Lik.' -278.892 (df=3)
```

```
summary(cum_log_co_e)
```

```
## Level ~ Age
##                   ratio      cdf nobs niter   logLik
## Model info: cumulative logistic  223     6 -278.892
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -2.63769    0.56148  -4.698 2.63e-06 ***
```

```
## (Intercept) distance    0.90366    0.08860  10.199  < 2e-16 ***
## Age                     0.21995    0.04947   4.446 8.75e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we have a preliminary idea of the coefficients of the model, we can specify an initialization vector through the parameter `beta_init`:

```
cum_log_c <- glmcat(
  formula = Level ~ Age,
  cdf = list("student",0.8),
  ratio = "cumulative",
  data = DisturbedDreams,
  control = control_glmcat(beta_init = coef(cum_log_co))
)
logLik(cum_log_c)
```

```
## 'log Lik.' -280.5428 (df=4)
```

```
summary(cum_log_c)
```

```
## Level ~ Age
##                   ratio     cdf nobs niter    logLik
## Model info: cumulative student  223     7 -280.5428
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -2.15258    0.56970  -3.778 0.000158 ***
## (Intercept) Severe.1   -1.39169    0.52035  -2.675 0.007483 **
## (Intercept) Severe.2   -0.07141    0.57835  -0.123 0.901740
## Age                     0.17909    0.04957   3.613 0.000302 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The equivalence between the $(Cumulative, Gompertz, parallel)$ and $(Sequential, Gompertz, parallel)$ models has been demonstrated by Läärä and Matthews (1985) and it is hereby tested using the functions:

```
cum_gom_p <- glmcat(
  formula = Level ~ Age,
  cdf = "gompertz",
  ratio = "cumulative",
  data = DisturbedDreams,
  parallel = "Age"
)
logLik(cum_gom_p)
```

```
## 'log Lik.' -280.0788 (df=4)
```

```
summary(cum_gom_p)
```

```
## Level ~ Age
##                   ratio      cdf nobs niter    logLik
## Model info: cumulative gompertz  223     6 -280.0788
```

```
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -1.88781    0.36046  -5.237 1.63e-07 ***
## (Intercept) Severe.1   -1.33515    0.35206  -3.792 0.000149 ***
## (Intercept) Severe.2   -0.78551    0.34252  -2.293 0.021828 *
## Age                     0.12434    0.03009   4.133 3.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
seq_gom_p <- glmcat(
  formula = Level ~ Age,
  cdf = "gompertz",
  ratio = "sequential",
  data = DisturbedDreams,
  parallel = "Age"
)
logLik(seq_gom_p)
```

```
## 'log Lik.' -280.0788 (df=4)
```

```
summary(seq_gom_p)
```

```
## Level ~ Age
##                   ratio      cdf nobs niter    logLik
## Model info: sequential gompertz  223     6 -280.0788
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept) Not.severe -1.88781    0.36046  -5.237 1.63e-07 ***
## (Intercept) Severe.1   -2.19180    0.36851  -5.948 2.72e-09 ***
## (Intercept) Severe.2   -1.64626    0.35822  -4.596 4.31e-06 ***
## Age                     0.12434    0.03009   4.133 3.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conclusion

The models for categorical response data have been evolved in different fields of research under different names. Some of them are fairly similar or are even the same. Until recently, there was no methodology that encompassed these models in a comparable scheme. `glmcat` is based on the new specification of a generalized linear model given by the *(r,F,Z)*-triplet, which groups together all the proposed methodologies for modelling categorical responses. `glmcat` offers a full picture of the spectrum of models where the user has three components to combine in order to obtain a model that meets the specifications of the problem.

References

Bouscasse, Hélène, Iragaël Joly, and Jean Peyhardi. 2019. "A new family of qualitative choice models: An application of reference models to travel mode choice." *Transportation Research Part B: Methodological* 121 (C): 74–91.

Läärä, E., and J. N. S. Matthews. 1985. "The equivalence of two models for ordinal data." *Biometrika* 72 (1): 206–7. https://doi.org/10.1093/biomet/72.1.206.

Maxwell, A. E. 1961. *Analyzing Qualitative Data*. Methuen.

Peyhardi, J., C. Trottier, and Y. Guédon. 2015. "A new specification of generalized linear models for categorical responses." *Biometrika* 102 (4): 889–906. https://doi.org/10.1093/biomet/asv042.