Fitting Proportional Odds Models for Ordinal Response Variables in Educational Research: A Comparison of Multiple Packages in R

Xing Liu
Eastern Connecticut State University

Ann A. O'Connell
Rutgers University

Haiyan Bai
University of Central Florida

Hongwei Yang
University of West Florida

Lucy Liu
University of Connecticut

Although researchers assume that software packages should produce the same results (e.g., model parameter estimates) for the same model, it is not always the case for ordinal regression models. The `MASS`, `ordinal`, `rms`, and `VGAM` packages in `R` may use different forms to express the proportional odds model (PO) for ordinal response variables and parameterize it differently, so researchers can easily become confused when they interpret the results. However, there are no studies specifically to address this issue. The purpose of the study was to investigate how to implement the PO model in educational research with multiple packages in `R` and compare the differences among these packages. Furthermore, it compares the results from the `R` packages and those from other general purpose software programs, SAS, SPSS Statistics, and Stata with the aim to help researchers to understand the performance of each package.

Ordinal logistic regression is a modeling technique for predicting ordinal response variables. The proportional odds (PO) model (Agresti, 2010, 2013, 2019; Ananth & Kleinbaum, 1997; Armstrong & Sloan, 1989; Hilbe, 2009; Liu, 2009, 2016, 2023; Long, 1997; Long & Freese, 2014; McCullagh, 1980; McCullagh & Nelder, 1989; O'Connell, 2000, 2006; O'Connell & Liu, 2011; Powers & Xie, 2008) is one of the most popular models for ordinal regression analysis. This model estimates the cumulative odds of being at or below a particular level of the ordinal response variable, or the inversed odds of being above that level. Thus, it is also called the cumulative odds model.

Although researchers currently have a variety of statistical software options (e.g., SAS, SPSS Statistics, and Stata) when fitting ordinal logistic regression models, they have been increasingly interested in the free software package `R`. `R` is not only general-purpose statistical software but also a programming language environment. It is powerful, flexible, and freely available with rising popularity in various disciplines and research fields. Compared with other statistical packages usually developed and maintained by a single company, `R` tends to have more extensive analytic capabilities for a variety of models including ordinal regression thanks to the contributions from all around the globe. Several packages in `R` can be used to fit the PO model. For example, the `polr()` function in the `MASS` package (Venable & Ripley, 2002), the `clm()` function in the `ordinal` package (Christensen, 2015, 2024), the `lrm()` function in the `rms` package (Harrell, 2001, 2015), and the `vglm()` function in the `VGAM` package (Yee, 2010, 2024) are all capable of estimating the PO model.

Although we assume that software packages should produce the same results (e.g., model parameter estimates) for the same model, it is not always the case for ordinal regression models since software packages may use different forms to express the PO model and parameterize it differently. Liu (2009) compared the features for ordinal logistic regression among Stata, SAS, and SPSS Statistics and found that these three packages parameterized the PO model differently and thus produced inconsistent output. These differences in model parameterizations may also exist in the `MASS`, `ordinal`, `rms`, and `VGAM` packages in `R`. In addition, methods used in `R` packages are not all well documented. For example, not all four packages provide the parameterization for the PO model in the `R` documentation and manuals. When provided, it lacks thorough explanation and the different parameterizations used by other software packages are not noted, which may confuse researchers when interpreting the results from these packages.

To our knowledge, no study has been conducted to fit the PO model by using and synthesizing multiple packages in `R`, nor comparing differences among them. Therefore, it is critical to assist educational researchers in understanding the methods for fitting the ordinal logistic regression model with these `R` packages, recognizing their differences, making a sound choice, and correctly interpreting the results. Our study aims to address this research gap.

The purpose of the study was to investigate how to implement the PO model for ordinal response variables in educational research by using multiple packages in `R`. In addition, this study compared the differences and identified similarities in model fitting using the `MASS`, `ordinal`, `rms`, and `VGAM` packages in `R`. Furthermore, it compared the results from the `R` packages and those from other general-purpose software such as SAS, SPSS Statistics, and Stata. To illustrate the uses of these `R` packages, the empirical data from the High School Longitudinal Study of 2009 (HSLS:09) were employed to conduct the ordinal regression analysis.

Theoretical Framework

The PO model can mainly be parameterized in two different ways. One is the latent variable model, and the other is a direct extension of the binary logistic regression model.

A Latent Variable Model

The latent variable model (Agresti, 2013; Liu, 2009; Long & Freese, 2014) assumes that a latent variable, Y*, exists. $Y^* = \mathbf{x}\boldsymbol{\beta} + \varepsilon$, where **x** is a row vector of predictors, **β** is a column vector of coefficients, and ε is the error term. Let Y* be divided by some cut points: $\alpha_1, \alpha_2, \ldots, \alpha_j$, and $\alpha_1 < \alpha_2 < \ldots < \alpha_j$. The observed variable $Y = j$ if the latent variable Y* falls in the interval between α_{j-1} and α_j, $\alpha_{j-1} < Y^* \le \alpha_j$. For example, Y = 1 if $y^* \le \alpha_1$ and Y = 2 if $\alpha_1 < Y^* \le \alpha_2$. Therefore, $P(Y = 1) = P(Y^* \le \alpha_1) = P(\mathbf{x}\boldsymbol{\beta} + \varepsilon \le \alpha_1) = F(\alpha_1 - \mathbf{x}\boldsymbol{\beta})$, and then $P(Y = j) = P(\alpha_{j-1} < Y^* \le \alpha_j) = F(\alpha_j - \mathbf{x}\boldsymbol{\beta}) - F(\alpha_{j-1} - \mathbf{x}\boldsymbol{\beta})$.

The cumulative probabilities can be obtained using the following function:

$$P(Y \le j) = F(\alpha_j - \mathbf{x}\boldsymbol{\beta}), \quad (1)$$

where F is the cumulative distribution function; and $j = 1, 2, \ldots, J-1$. Since the PO model estimates the cumulative probabilities of being at or below a particular category, this model can be expressed on the logit scale as follows (Fullerton & Xu, 2016; Liu, 2009, 2016, 2023; Long, 1997; Long & Freese, 2014):

$$\text{logit}\,[\pi(Y \le j \mid x_1, x_2, \ldots, x_p)] = \ln\left(\frac{\pi(Y \le j \mid x_1, x_2, \ldots x_p)}{\pi(Y > j \mid x_1, x_2, \ldots x_p)}\right) = \alpha_j + (-\beta_1 X_1 - \beta_2 X_2 - \ldots - \beta_p X_p), \quad (2)$$

where $\pi(Y \le j \mid x_1, x_2, \ldots, x_p)$ is the cumulative probability of being at or below a category j, given a set of predictors; $j = 1, 2, \ldots, J-1$. α_j are the cut points; and $\beta_1, \beta_2, \ldots, \beta_p$ are the logit coefficients. The signs before both logit coefficients on the right side of the equation are negative so that an increase in a predictor is associated with the odds of being above a particular category.

The Proportional Odds Model: An Extension of Binary Logistic Regression

In addition to the latent variable model, the PO model can be expressed as an extension of the binary logistic regression model as follows (Agresti, 2010; Liu, 2016, 2022; O'Connell, 2006; Yee, 2010):

$$\text{logit}\,[\pi_j(x)] = \ln\left(\frac{\pi_j(x)}{1-\pi_j(x)}\right) = \ln\left(\frac{\pi(Y \le j \mid x_1, x_2, \ldots x_p)}{\pi(Y > j \mid x_1, x_2, \ldots x_p)}\right) = \alpha_j + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p, \quad (3)$$

where $\pi_j(x) = \pi(Y \le j \mid x_1, x_2, \ldots, x_p)$, the cumulative probability of being at or below a category j; $\ln\left[\frac{\pi_j(x)}{1-\pi_j(x)}\right]$ is the ln(odds), where the cumulative odds are the ratio of the cumulative probability of being at or below a particular category to the cumulative probability of above that category.

When estimating the cumulative probability and odds of being above a category, the modified form of the PO model can be expressed as follows (Agresti, 2010).

$$\text{logit}\,[\pi(Y > j \mid x_1, x_2, \ldots, x_p)] = \ln\left(\frac{\pi(Y > j \mid x_1, x_2, \ldots x_p)}{\pi(Y \le j \mid x_1, x_2, \ldots x_p)}\right) = \alpha_j + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p, \quad (4)$$

where $j = 1, 2, \ldots, J-1$. Please note that the cumulative odds of being above a particular category are the inversed odds of being at or below that category.

A modified form of Equation (4) estimates the cumulative probability of being at or above a category and is expressed as follows (Yee, 2010).

$$\text{logit}\,[\pi(Y \geq j \mid x_1, x_2, \ldots, x_p)] = \ln\left(\frac{\pi(Y \geq j \mid x_1, x_2, \ldots x_p)}{\pi(Y < j \mid x_1, x_2, \ldots x_p)}\right) = \alpha_j + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p, \quad (5)$$

where $j = 2, \ldots, J$.

These two equations (i.e., Equations 4 and 5) are equivalent since Equation 4 estimates the cumulative probabilities of being above $J-1$ categories starting from category 1, whereas Equation 5 estimates the cumulative probabilities starting from category 2.

Why Compare Multiple Packages in `R`?

Although different parameterizations do not affect model estimation, they do influence the signs for the cut points or intercepts and the logit coefficients, thereby impacting the interpretations of the output of different software packages. The `R` documentation and manuals for various packages for the PO model do not address or explain different parameterizations by other software packages. Therefore, it is important for researchers to understand different parameterizations in various `R` packages for ordinal regression.

Methodology

Sample

The High School Longitudinal Study of 2009 (HSLS:09), conducted by the NCES (Ingels, et. al., 2011), kept track of high school students from ninth grade to postsecondary education and their choice of future careers. It surveyed students, their parents, and school personnel, and assessed 9th graders' mathematics achievement. In the 2009 base-year data, 21,444 high school students, from a national sample of 944 schools, participated in the study. Students were asked to provide basic demographic information, school and home experience, mathematics and science attitude, mathematics and science self-efficacy, and future educational and life plans. The ordinal outcome variable of interest is students' mathematics proficiency, and the predictors are students' math identity (MTHID), students' math self-efficacy (MTHEFF), and math teachers' self-efficacy (X1TMEFF).

The outcome variable, students' mathematics proficiency levels in high schools, was ordinal with five levels, from level 1, students can answer questions in algebraic expressions, to level 5, students can understand linear functions. Students who failed to pass through level 1 were assigned to level 0. Table 1 provides the frequency of six mathematics proficiency levels (i.e., levels 0-5).

Data Analysis

First, the `polr()` function in the `R` `MASS` package was used to fit the PO model. Then, the same model was fitted using the `clm()` function in the `ordinal` package, the `lrm()` function in the `rms` package, and the `vglm()` function in the `VGAM` package, respectively. The similarities and differences of the results from these four packages were compared. Finally, the results from the `R` packages were compared with those from SAS, SPSS Statistics, and Stata.

Results

The PO Model with the `polr()` Function in the `MASS` Package

The `polr()` function in the `MASS` package (Venable & Ripley, 2002) was used to fit the PO model. This function can be used to fit ordinal logistic regression and ordinal probit models. It uses Equation 2 to express the PO model with the negative signs for the logit coefficients in the linear predictor.

$$\text{logit}\,[\pi(Y \leq j)] = \alpha_j + (-\beta_1 X_1 - \beta_2 X_2 - \ldots - \beta_p X_p).$$

Table 1: Proficiency Levels and Frequencies (Percentages) for the Study Sample, HSLS: 09 Base-Year Data (n = 21,444)

Proficiency Levels	*Description*	*Frequency*
0	Did not pass level 1	2263 (10.6%)
1	Algebraic expressions	4933 (23%)
2	Multiplicative and proportional thinking	5495 (25.6%)
3	Algebraic equivalents	5761 (26.9%)
4	Systems of equations	2396 (11.2%)
5	Linear functions	596 (2.8%)

```
> library(MASS)
> polr.po<-polr(as.factor(Mathprof)~ MTHID + MTHEFF + X1TMEFF, data = hsls)
> summary(polr.po)

Re-fitting to get Hessian

Call:
polr(formula = as.factor(Mathprof) ~ MTHID + MTHEFF + X1TMEFF, data = hsls)

Coefficients:
         Value Std. Error t value
MTHID   0.6264    0.02044  30.647
MTHEFF  0.2431    0.02009  12.098
X1TMEFF 0.1330    0.01706   7.795

Intercepts:
    Value    Std. Error t value
0|1  -2.5795   0.0335   -76.9260
1|2  -0.8870   0.0206   -43.1017
2|3   0.3435   0.0192    17.9164
3|4   1.9532   0.0266    73.5244
4|5   3.7734   0.0528    71.4298

Residual Deviance: 38025.22
AIC: 38041.22
(8970 observations deleted due to missingness)

> ctable <- coef(summary(polr.po))

Re-fitting to get Hessian

> p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
> ctable <- cbind(ctable, "p value" = p)
> ctable
             Value Std. Error    t value       p value
MTHID    0.6264123 0.02043988  30.646581 2.934950e-206
MTHEFF   0.2430702 0.02009105  12.098430  1.076504e-33
X1TMEFF  0.1329641 0.01705704   7.795263  6.427430e-15
0|1     -2.5795005 0.03353225 -76.925959  0.000000e+00
1|2     -0.8869798 0.02057878 -43.101678  0.000000e+00
2|3      0.3435181 0.01917336  17.916428  8.778764e-72
3|4      1.9531969 0.02656528  73.524413  0.000000e+00
4|5      3.7734393 0.05282723  71.429821  0.000000e+00

> cbind(exp(coef(polr.po)), exp(confint(polr.po)))
Waiting for profiling to be done...

Re-fitting to get Hessian

                    2.5 %   97.5 %
MTHID   1.870886 1.797514 1.947466
MTHEFF  1.275158 1.225947 1.326404
X1TMEFF 1.142209 1.104669 1.181056
```

Figure 1. The PO Model with the `polr` Function in the `MASS` Package

In the model formula for this function, the ordinal response variable needs to be specified as a factor or categorical variable with the `as.factor()` function. Figure 1 displays the `R` syntax and the output.

Since the output from the `summary()` function did not provide the *p*-values for the tests of the logit coefficients, we used the `pnorm()` function to compute them. We also ran the `exp()` function to compute the odds ratios by exponentiating the logit coefficients.

In the results of the estimated PO model, the logit coefficients of all three predictors were significant in predicting the mathematics proficiency levels. They were positively associated with the odds of being above a proficiency level. In terms of the odds ratios (OR), the odds of being above a proficiency level increased by 1.871 with a one-unit increase in students' mathematics identity, increased by 1.275 with a one-unit increase in students' mathematics self-efficacy, and increased by 1.142 with a one-unit increase in teachers' mathematics self-efficacy. Alternatively, the results can also be interpreted in terms of the odds of being at or below a proficiency level when the inversed odds are obtained with the `vglm()` function in the `VGAM` package (see Table 2 and Figure 4).

The PO Model with the `clm()` Function in the `ordinal` Package

The `clm()` function in the `ordinal` package (Christensen, 2015, 2024) was also used to fit the PO model. This function can be used to fit a variety of ordinal regression models, also called cumulative link models as the function name implies. Multiple link functions, such as logit, probit, cloglog, and loglog and different type of thresholds or cut points, can be specified for various models. Same as the `polr()` function, the `clm()` function also uses Equation 2 to express the PO model where there are negative signs before the logit coefficients.

In the model formula, as with the `polr()` function, the ordinal response variable needs to be specified as a categorical variable with the `as.factor()` function. Figure 2 displays the `R` syntax and the output. To compute the odds ratios, we again used the `exp()` function to exponentiate the coefficients. The results were the same as those estimated by the `polr()` function in the preceding section.

The PO Model with the `lrm()` Function in the `rms` Package

The same PO model was fitted using the `lrm()` function in the `rms` package (Harrell, 2015). The `lrm()` function can be used to fit both logistic regression models and proportional odds models but does not allow other link functions. It uses Equation 5 to express the PO model where the signs for logit coefficients are positive: $\text{logit}[\pi(Y \geq j)] = \alpha_j + \beta_1X_1 + \beta_2X_2 + \ldots +\beta_pX_p$. The `R` syntax and the output are displayed in Figure 3.

In the output, the intercepts or thresholds have the same magnitude as those estimated by the `polr()` and `clm()` functions but have negative signs because the PO model for the `lrm()` function estimates the cumulative odds of at or above a category of an ordinal response variable (see Equation 5). For example, the log odds of being at or above category 1, logit[P(Y>=1)], compares the probabilities of categories 1, 2, 3, 4, and 5 to the probability of being at category 0.

The PO Model with the `vglm()` Function in the `VGAM` Package

The `vglm()` function in the `VGAM` package (Yee, 2010, 2024) was also used to fit the PO model. This function can fit various generalized linear models for binary, ordinal, nominal, and count response variables. It uses Equations 3 and 5 to express the PO model where the signs for the logit coefficients are positive.

In the model formula, the ordinal response variable does not need to be specified as a categorical variable since it will be converted to a factor variable internally. To fit a PO model or a cumulative odds model, the argument `cumulative(parallel = TRUE)` needs to be used, where the parallel odds or proportional odds are specified. To estimate the cumulative odds of being at or below a particular category of an ordinal response variable, we also need to specify that the ordinal categories are not reversed with the argument, `reverse = FALSE`. The `R` syntax and the output are displayed in Figure 4.

In the output, although the intercepts are the same as those estimated by the `polr()` and `clm()` functions, the estimated logit coefficients have the same magnitude with negative signs since the `vglm()` function uses a different equation (i.e., Equation 3) to express the PO model. The estimated logit coefficients for the three predictor variables were –0.626, –0.243, and –0.133, respectively.

The `exp()` function was used to exponentiate the coefficients to obtain the odds ratios of being at or below a category. The odds of being at or below a proficiency level decreased by 0.535 with a one-unit increase in students' mathematics identity, decreased by 0.784 with a one-unit increase in students' mathematics self-efficacy, and decreased by 0.875 with a one-unit increase in teachers' mathematics self-efficacy.

```
> library(ordinal)
> clm.po<-clm(as.factor(Mathprof) ~ MTHID + MTHEFF + X1TMEFF, data = hsls,
na.action="na.omit")
> summary(clm.po)
formula: as.factor(Mathprof) ~ MTHID + MTHEFF + X1TMEFF
data:    hsls

 link  threshold nobs  logLik    AIC      niter max.grad cond.H
 logit flexible  12474 -19012.61 38041.22 6(0)  8.21e-13 2.0e+01

Coefficients:
        Estimate Std. Error z value Pr(>|z|)
MTHID    0.62641    0.02044  30.647  < 2e-16 ***
MTHEFF   0.24307    0.02009  12.099  < 2e-16 ***
X1TMEFF  0.13296    0.01706   7.795 6.43e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
    Estimate Std. Error z value
0|1 -2.57935    0.03353  -76.92
1|2 -0.88695    0.02058  -43.10
2|3  0.34356    0.01917   17.92
3|4  1.95321    0.02657   73.53
4|5  3.77342    0.05283   71.43

> cbind(exp(coef(clm.po)), exp(confint(clm.po, type="Wald")))
                          2.5 %      97.5 %
0|1       0.07582343  0.07100062  0.08097383
1|2       0.41191166  0.39562848  0.42886503
2|3       1.40995557  1.35795409  1.46394840
3|4       7.05128348  6.69354026  7.42814667
4|5      43.52855055 39.24714538 48.27700702
MTHID     1.87088821  1.79741970  1.94735971
MTHEFF    1.27515852  1.22592176  1.32637277
X1TMEFF   1.14220743  1.10465355  1.18103799
```

Figure 2. The PO Model with the `clm()` Function in the `ordinal` Package

We used the `reverse = TRUE` argument to estimate the logit coefficients of being at or above a particular level of the mathematics proficiency. The `R` syntax and output are displayed in Figure 4. Compared to the results from the `vglm()` function with the `reverse = FALSE` option, the results from the same function with the `reverse = TRUE` argument had the same intercepts and logit coefficients in magnitude but with opposite signs. The estimated logit coefficients for the three predictor variables were 0.626, 0.243, and 0.133, respectively. We obtained the odds ratios of being at or above a level of the mathematical proficiency by exponentiating the logit coefficients. The results are provided in Table 2. For example, the odds ratio for MTHID was 1.871, indicating that the odds of being at or above a proficiency level increased by 1.871 with a one-unit increase in students' mathematics identity.

A Comparison of the Results Using Different `R` Packages

Table 2 provides the results of the PO Models with the `MASS`, `ordinal`, `rms`, and `VGAM` packages in `R`. Comparing the results using the `MASS` and `ordinal` packages in `R`, we found that they produced the same logit coefficients and intercepts or thresholds. Compared to the output from both the `polr()` and `clm()` functions, the estimated logit coefficients from the `lrm()` function in the `rms` package were the same. However, the intercepts were the same in magnitude with reversed signs. In addition, the `vglm()` function in the `VGAM` package with the `reverse = FALSE` and `reverse = TRUE` options produced the same intercepts and coefficients in magnitude with reversed signs. Further, the `lrm()` function and the `vglm()` function with the `reverse = TRUE` option produced the same results. Finally, compared to the results from both the `polr()` and `clm()` functions, the `VGAM` package with the `reverse = FALSE` option produced the same intercepts, but the coefficients had reversed signs.

```
> library(rms)
> lrm.po<-lrm(as.factor(Mathprof) ~ MTHID + MTHEFF + X1TMEFF, data = hsls)
> lrm.po
Frequencies of Missing Values Due to Each Variable
as.factor(Mathprof)            MTHID         MTHEFF          X1TMEFF
                  0              285           2685             7371

Logistic Regression Model

 lrm(formula = as.factor(Mathprof) ~ MTHID + MTHEFF + X1TMEFF,
     data = hsls)

 Frequencies of Responses
    0    1    2    3    4    5
 1059 2760 3249 3493 1524  389

                      Model Likelihood     Discrimination    Rank Discrim.
                         Ratio Test           Indexes           Indexes
 Obs          12474    LR chi2    2264.84    R2       0.173    C       0.672
 max |deriv| 7e-13     d.f.             3    g        0.919    Dxy     0.344
                       Pr(> chi2) <0.0001    gr       2.507    gamma   0.344
                                             gp       0.203    tau-a   0.269
                                             Brier    0.215

        Coef    S.E.   Wald Z Pr(>|Z|)
 y>=1     2.5793 0.0335  76.93 <0.0001
 y>=2     0.8869 0.0206  43.10 <0.0001
 y>=3    -0.3436 0.0192 -17.92 <0.0001
 y>=4    -1.9532 0.0266 -73.53 <0.0001
 y>=5    -3.7734 0.0528 -71.43 <0.0001
 MTHID    0.6264 0.0204  30.65 <0.0001
 MTHEFF   0.2431 0.0201  12.10 <0.0001
 X1TMEFF  0.1330 0.0171   7.80 <0.0001

> exp(coefficients(lrm.po))
   y>=1         y>=2        y>=3         y>=4        y>=5       MTHID
13.18853589  2.42770499  0.70924221  0.14181815  0.02297343  1.87088821
     MTHEFF     X1TMEFF
 1.27515852  1.14220743
```

Figure 3. The PO Model with the `lrm()` Function in the `rms` Package

A Comparison of the Results of the PO Models Using SAS, SPSS Statistics, and Stata

Table 3 provides the results of the PO Models using SAS (ascending and descending), SPSS Statistics, and Stata. The results by SPSS Statistics and Stata were the same as those by the `polr()` and `clm()` functions in `R`. In addition, SAS `proc logistic` with the ascending option produced the same results as those by the `VGAM` package with the `reverse = FALSE` option. Correspondingly, SAS `proc logistic` with the descending option, the `lrm()` function and the `vglm()` function with the `reverse = TRUE` option produced the same results.

A Comparison of the Results of the PO Models Using SAS, SPSS Statistics, and Stata

Table 3 provides the results of the PO Models using SAS (ascending and descending), SPSS Statistics, and Stata. The results by SPSS Statistics and Stata were the same as those by the `polr()` and `clm()` functions in `R`. In addition, SAS `proc logistic` with the ascending option produced the same results as those by the VGAM package with the `reverse = FALSE` option. Correspondingly, SAS `proc logistic` with the descending option, the `lrm()` function and the `vglm()` function with the `reverse = TRUE` option produced the same results.

```
> library(VGAM)
> vglm.po<-vglm(Mathprof ~ MTHID + MTHEFF + X1TMEFF, cumulative(parallel =
TRUE, reverse = FALSE), data = hsls)
> summary(vglm.po)

Call:
vglm(formula = Mathprof ~ MTHID + MTHEFF + X1TMEFF, family =
cumulative(parallel = TRUE,
    reverse = FALSE), data = hsls)

Pearson residuals:
                   Min       1Q  Median      3Q    Max
logit(P[Y<=1])  -0.967 -0.30658 -0.1721 -0.1159 8.3973
logit(P[Y<=2])  -2.179 -0.79749 -0.2591  0.5256 4.1206
logit(P[Y<=3])  -3.518 -0.84827  0.2343  0.8166 2.5004
logit(P[Y<=4])  -6.904  0.12411  0.2096  0.5984 1.1675
logit(P[Y<=5]) -12.815  0.07708  0.1073  0.1522 0.9156

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1 -2.57935    0.03356 -76.866  < 2e-16 ***
(Intercept):2 -0.88695    0.02061 -43.027  < 2e-16 ***
(Intercept):3  0.34356    0.01921  17.881  < 2e-16 ***
(Intercept):4  1.95321    0.02651  73.665  < 2e-16 ***
(Intercept):5  3.77342    0.05265  71.674  < 2e-16 ***
MTHID         -0.62641    0.02025 -30.932  < 2e-16 ***
MTHEFF        -0.24307    0.01992 -12.201  < 2e-16 ***
X1TMEFF       -0.13296    0.01694  -7.851 4.13e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors:  5

Names of linear predictors:
logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3]), logit(P[Y<=4]), logit(P[Y<=5])

Residual deviance: 38025.22 on 62362 degrees of freedom

Log-likelihood: -19012.61 on 62362 degrees of freedom
Number of iterations: 5
> cbind(exp(coef(vglm.po)), exp(confint(vglm.po)))
                                 2.5 %      97.5 %
(Intercept):1  0.07582343  0.0709970  0.08097795
(Intercept):2  0.41191171  0.3956014  0.42889444
(Intercept):3  1.40995585  1.3578460  1.46406555
(Intercept):4  7.05128466  6.6942020  7.42741482
(Intercept):5 43.52854836 39.2609979 48.25996845
MTHID          0.53450553  0.5137053  0.55614795
MTHEFF         0.78421620  0.7541847  0.81544356
X1TMEFF        0.87549776  0.8469136  0.90504663

> vglm.po2<-vglm(Mathprof ~ MTHID + MTHEFF + X1TMEFF, cumulative(parallel = TRUE,
reverse = TRUE), data = hsls)
> summary(vglm.po2)
Call:
vglm(formula = Mathprof ~ MTHID + MTHEFF + X1TMEFF, family = cumulative(parallel =
TRUE,
    reverse = TRUE), data = hsls)

Pearson residuals:
                   Min      1Q  Median       3Q    Max
logit(P[Y>=2]) -8.3973  0.1159  0.1721  0.30658  0.967
logit(P[Y>=3]) -4.1206 -0.5256  0.2591  0.79749  2.179
logit(P[Y>=4]) -2.5004 -0.8166 -0.2343  0.84827  3.518
logit(P[Y>=5]) -1.1675 -0.5984 -0.2096 -0.12411  6.904
logit(P[Y>=6]) -0.9156 -0.1522 -0.1073 -0.07708 12.815
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1  2.57935    0.03356  76.866  < 2e-16 ***
(Intercept):2  0.88695    0.02061  43.027  < 2e-16 ***
(Intercept):3 -0.34356    0.01921 -17.881  < 2e-16 ***
(Intercept):4 -1.95321    0.02651 -73.665  < 2e-16 ***
(Intercept):5 -3.77342    0.05265 -71.674  < 2e-16 ***
MTHID          0.62641    0.02025  30.932  < 2e-16 ***
MTHEFF         0.24307    0.01992  12.201  < 2e-16 ***
X1TMEFF        0.13296    0.01694   7.851 4.13e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors:  5

Names of linear predictors:
logit(P[Y>=2]), logit(P[Y>=3]), logit(P[Y>=4]), logit(P[Y>=5]), logit(P[Y>=6])

Residual deviance: 38025.22 on 62362 degrees of freedom

Log-likelihood: -19012.61 on 62362 degrees of freedom

Number of iterations: 5
```

Figure 4. The PO Model with the `vglm()` Function in the `VGAM` Package

Table 2. Comparison of Results from the PO Models with the `MASS`, `ordinal`, `rms`, and `VGAM` `R` packages.

Model Estimates	`polr` in `MASS`		`clm` in `ordinal`		`lrm` in `rms`		`vglm` in `VGAM` `(reverse=FALSE)`		`vglm` in `VGAM` `(reverse=TRUE)`	
Variables	$P(Y \leq j)$		$P(Y \leq j)$		$P(Y \geq j)$		$P(Y \leq j)$		$P(Y \geq j)$	
α_1	-2.580		-2.580		2.579		-2.580		2.579	
α_2	-0.887		-0.887		0.887		-0.887		0.887	
α_3	0.344		0.344		-0.344		0.344		-0.344	
α_4	1.953		1.953		-1.953		1.953		-1.953	
α_5	3.773		3.773		-3.773		3.773		-3.773	
Variables	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**
MTHID	0.626** (0.020)	1.871	0.626** (0.020)	1.871	0.626** (0.020)	1.871	-0.626** (0.020)	0.535	0.626** (0.020)	1.871
MTHEFF	0.243** (0.020)	1.275	0.243** (0.020)	1.275	0.243** (0.020)	1.275	-0.243** (0.020)	0.784	0.243** (0.020)	1.275
X1TMEFF	0.133** (0.017)	1.142	0.133** (0.017)	1.142	0.133** (0.017)	1.142	-0.0133** (0.017)	0.875	0.133** (0.017)	1.142
Model Fit	AIC		AIC		χ^2_3		AIC		AIC	
	38,041.22		38,041.22		2,264.84**		38,041.22		38,041.22	

Significant at ** p <0.01.

Feature Comparisons of Fitting the PO Model Using Multiple `R` Packages

In addition to the different parameterizations in expressing PO models among the `R` packages above, we also identified other differences when fitting the PO model with those four `R` packages. The comparison of the features between those packages is provided in Table 4. We compared the model specification, parameter estimates, model fit statistics, test of the PO assumption, predicted probabilities, and extension to multilevel models. The differences in the model specification were discussed in the preceding section. Both the `polr()` and `clm()` functions parameterize the PO model with negative signs before the logit coefficients, whereas the signs before logit coefficients in the model equation used by the `lrm()` and `vglm()` functions are positive.

Table 3. Comparison of Results from the PO Models SAS, SPSS Statistics, and Stata

Model Estimates	SAS (Ascending)		SAS (Descending)		SPSS Statistics		Stata	
Variables	$P(Y \leq j)$		$P(Y \leq j)$		$P(Y \geq j)$		$P(Y \leq j)$	
α_1	-2.579		2.579		-2.580		-2.580	
α_2	-0.887		0.887		-0.887		-0.887	
α_3	0.344		-0.344		0.344		0.344	
α_4	1.953		-1.953		1.953		1.953	
α_5	3.773		-3.773		3.773		3.773	
Variables	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**	**b ($SE_{(b)}$)**	**OR**
MTHID	-0.626** (0.020)	0.535	0.626** (0.020)	10.871	0.626** (0.020)	10.871	0.626** (0.020)	10.871
MTHEFF	-0.243** (0.020)	0.784	0.243** (0.020)	10.275	0.243** (0.020)	10.275	0.243** (0.020)	10.275
X1TMEFF	-0.133** (0.017)	0.875	0.133** (0.017)	10.142	0.133** (0.017)	10.142	0.133** (0.017)	10.142
Model Fit	χ^2_3		χ^2_3		χ^2_3		χ^2_3	
	2,264.84**		2,264.84**		2,264.84**		2,264.84**	

Significant at ** p <0.01.

As summarized in Table 4, unlike the other three functions, the `polr()` function does not provide the significance test for logit coefficients for predictor variables in the parameter estimates. To computer the *p* values, we need to use additional `R` functions. In addition, all four functions provide either the *t*-statistics or *z*-statistics for parameter estimates. While the `polr()` function provides the *t*-values, the other three functions provide the *z*-values. After fitting the PO models with the `polr()`, `clm()`, and `vglm()` functions, the profile likelihood confidence intervals for the parameter estimates can be easily computed with the `confint()` function, but the `lrm()` function does not work with `confint()` function.

Those four functions provide limited fit statistics for the PO model. The `clm()` and `vglm()` functions provide the log-likelihood, whereas the `polr()` function provides the residual deviance, and the `lrm()` function provided the model likelihood ratio test, the discrimination indices, and the rank discrimination indices.

Although we can use the `anova()` function to conduct the likelihood ratio test after fitting the PO model with the `polr()`, `clm()` and `lrm()` functions, we cannot use it with the `vglm()` function. We need to use the `lrtest()` function instead.

To test the PO assumption, we need to run either the `clm()` function or the `vglm()` function to fit the PO model and then use the `nominal_test()` function in the `ordinal` package and the `lrtest()` function in the `VGAM` package to perform the test, respectively.

All four functions work with the `ggpredict()` function in the `ggeffects` package (Lüdecke, 2018), so we can use it to compute the predicted probabilities of being in an ordinal response category at any values of the predictor variables.

Of the four packages, only the `ordinal` package can fit multilevel models for ordinal response variables, while the other three packages lack this capability. We can use the `clmm()` function in the `ordinal` package, an extension of the `clm()` function, to fit multilevel models.

Conclusions

This study synthesized the `polr()`, `clm()`, `lrm()`, and `vglm()` functions in those `R` packages and compared the differences and similarities for model fitting. It illustrated the use of the `MASS`, `ordinal`, `rms`, and `VGAM` packages in `R` to fit the PO model. The `R` code and output were provided, and the results were interpreted and compared. In addition, this study compared the results from the `R` packages and those from SAS, SPSS Statistics, and Stata. Further, it compared the features such as model specification, parameter estimates, model fit statistics, and test of the PO assumption among those four functions.

Table 4. Feature Comparisons of Fitting the PO Model Using Multiple `R` Packages

Functions **Packages**	`polr` `MASS`	`clm` `ordinal`	`lrm` `rms`	`vglm` `VGAM`
Model Specification				
Cutpoints/Thresholds		✓		
Intercepts	✓		✓	✓
Negative Signs Before Coefficients	✓	✓		
Parameter Estimates				
Odds Ratio	✓	✓	✓	✓
T-statistic for Parameter Estimates	✓			
Z-statistic for Parameter Estimates		✓	✓	✓
Significance Tests		✓	✓	✓
Confidence Interval for Parameter Estimate	✓	✓		✓
Fit Statistics				
Log-likelihood		✓		✓
Goodness-of-Fit Test with `anova()`	✓	✓	✓	
Test of the PO Assumption				
Omnibus Test of Assumption of Proportional Odds				✓
Test of Assumption of Proportional Odds for Individual Variables		✓		
Predicted Probabilities with `ggeffects`	✓	✓	✓	✓
Extension to Multilevel Models		✓		

This study found that the `polr()`, `clm()`, `lrm()`, and `vglm()` functions in the four `R` packages parameterized the PO model differently by following different model equations. Thus, the signs of the intercepts or cut points and the logit coefficients were different in the resulting output. Ignoring the differences in parameterization will likely lead to erroneous interpretation of the results. Although not all researchers use multiple packages or programs, understanding the differences among different packages in `R` would help applied researchers and practitioners to clarify the confusion of different parameterizations of PO models and interpret the results correctly.

This study also compared the results of the PO model from those four `R` packages and those from SAS, SPSS Statistics, and Stata. We found that the results by SPSS Statistics and Stata were the same as those by the `polr()` and `clm()` functions. In addition, SAS `proc logistic` with the ascending option produced the same results as those by the `VGAM` package with the non-reversed ordinal categories. Finally, SAS `proc logistic` with the descending option, the `lrm()` function and the `vglm()` function with the reversed ordinal categories produced the same results. It is expected that this study will help general SAS, SPSS Statistics, and Stata users choose appropriate `R` packages for ordinal regression analysis.

In the end, we would like to note that this study only focused on the PO model with multiple `R` packages. For non-proportional odds models when the PO assumption is violated, only `clm()` and `vglm()` functions can be used, whereas the other two functions do not have this capacity. For future research, fitting non-proportional odds models or partial proportional odds models with those two functions will be conducted. In addition, when comparing the features for fitting the PO model with those four `R` packages, we focused on the current versions at the time of writing. With the development of those packages, new features may be added, so further research may be needed.

References

Agresti, A. (2010). *Analysis of ordinal categorical data* (2nd ed.). John Wiley & Sons.

Agresti, A. (2013). *Categorical data analysis* (3rd ed.). John Wiley & Sons.

Agresti, A. (2019). *An introduction to categorical data analysis* (3rd ed.). John Wiley & Sons.

Ananth, C. V., & Kleinbaum, D. G. (1997). Regression models for ordinal responses: A review of methods and applications. *International Journal of Epidemiology 26*, 1323-1333. https://doi.org/10.1093/ije/26.6.1323

Armstrong, B. B., & Sloan, M. (1989). Ordinal regression models for epidemiological data. *American Journal of Epidemiology, 129*(1), 191-204. https://doi.org/10.1093/oxfordjournals.aje.a115109

Christensen, R. H. B. (2015). *Analysis of ordinal data with cumulative link models-estimation with the R-package ordinal*. Retrieved from https://cran.r-project.org/web/packages/ordinal/vignettes/clm_article.pdf

Christensen, R. H. B. (2024). O*rdinal: Regression models for ordinal data.* [R package version 2023.12-4.1] [Computer Software] https://cran.r-project.org/web/packages/ordinal/index.html

Fullerton, A. S., Xu, J. (2016). *Ordered regression models: Parallel, partial, and non-parallel alternatives*. Chapman & Hall/CRC.

Harrell, F. E. (2001). *Regression modeling strategies: With applications to linear models, logistic regression, and survival analysis*. Springer.

Harrell, F. E. (2015). *Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis* (2nd ed.). Springer.

Hilbe, J. M. (2009). *Logistic regression models*. Chapman & Hall/CRC.

Ingels, S. J., Dalton, B., Holder, T. E., Lauff, E., & Burns, L.J. (2011). *High School Longitudinal Study of 2009 (HSLS:09): A First Look at Fall 2009 Ninth-Graders* (NCES 2011-327). U.S. Department of Education. Washington, DC: National Center for Education Statistics. https://files.eric.ed.gov/fulltext/ED523764.pdf

Liu, X. (2009). Ordinal regression analysis: Fitting the proportional odds model using Stata, SAS and SPSS. *Journal of Modern Applied Statistical Methods, 8*(2), 632-645. https://doi.org/10.22237/jmasm/1257035340

Liu, X. (2016). *Applied ordinal logistic regression using Stata: From single-level to multilevel modeling*. Sage.

Liu, X. (2023). *Categorical data analysis and multilevel modeling using R*. Sage.

Long, J. S. (1997). *Regression models for categorical and limited dependent variables*. Sage.

Long, J. S. & Freese, J. (2014). *Regression models for categorical dependent variables using Stata* (3rd ed.). Stata Press.

McCullagh, P. (1980). Regression models for ordinal data (with discussion). *Journal of the Royal Statistical Society Ser. B*, 42, 109-142. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x

McCullagh, P. & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). Chapman and Hall.

O'Connell, A.A., (2000). Methods for modeling ordinal outcome variables. *Measurement and Evaluation in Counseling and Development*, *33*(3), 170-193. https://doi.org/10.1080/07481756.2000.12069008

O'Connell, A. A. (2006). *Logistic* r*egression models for ordinal response variables*. Sage.

O'Connell, A.A., & Liu, X. (2011). Model diagnostics for proportional and partial proportional odds models. *Journal of Modern Applied Statistical Methods, 10*(1), 139-175. https://doi.org/10.22237/jmasm/1304223240

Powers D. A., & Xie, Y. (2008). *Statistical models for categorical data analysis* (2nd ed.). Emerald Press.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (4th ed.). Springer.

Yee, T. W. (2010). The VGAM package for categorical data analysis. *Journal of Statistical Software, 32*(*10*), 1-34. https://doi.org/10.18637/jss.v032.i10

Yee, T. W. (2024). *The VGAM package for R.* http://www.stat.auckland.ac.nz/~yee/VGAM

Send correspondence to: Xing Liu
Eastern Connecticut State University
Email: liux@easternct.edu