TN: 2632583
Pages: 143 RAPID:PUL
Exempt

UBY/UTUBYL ILL - Lending

Call #: QA 278.2 .L586 2023

Location: 2ND

Journal Title: Categorical data analysis and multilevel modeling using R /

Volume: **Issue:**
Month/Year: 2023 **Pages:** 143

Article Author: Xing Liu
Article Title: Chapter 4. Logistic Regression for Binary Data

Ref. Number / Item Barcode:

ILL Number: -21738745
Lending String:

Borrower: RAPID:PUL
NEW: Interlibrary Services,
Firestone

Odyssey: 128.112.202.139
Email: ilsborr@princeton.edu
12/8/2023 12:55 PM

ODYSSEY PROBLEM REPORT

If you have experienced a problem in the delivery of the requested item, please contact us within 5 Business Days. Please include the following information:

ILL #: ______________
Your OCLC Symbol: ______________
Date of Receipt: ______________

Please Specify if:

- □ Pages were missing pp.____ to ____
- □ Edges were cut off pp.____ to ____
- □ Illegible copy, resend entire copy.
- □ Wrong article sent.
- □ Other (explain)

FOR ODYSSEY RECEIPT PROBLEMS:

Phone:	801-422-6844
Email:	library_request@byu.edu

4

PROPORTIONAL ODDS MODELS FOR ORDINAL RESPONSE VARIABLES

OBJECTIVES OF THIS CHAPTER

This chapter introduces proportional odds models for ordinal response variables. It starts with an introduction to the model followed by a discussion of the odds and odds ratios in the model, goodness-of-fit statistics of the model, the proportional odds assumption, and how to interpret parameter estimates. After a description of the data, the proportional odds models with the `clm()` function in the `ordinal` package and the `vglm()` function in `VGAM` are illustrated with step-by-step instructions. R commands and output are explained in detail. The chapter focuses on fitting proportional odds models using R, as well as on interpreting and presenting the results. After reading this chapter, you should be able to:

- Identify when a proportional model is used.
- Conduct proportional odds models, and test the assumption using R.
- Interpret the output.
- Interpret the model in terms of odds ratios.
- Compute and plot the predicted probabilities.
- Compare models using the likelihood ratio test and other fit statistics.
- Present results in publication-quality tables using R.
- Write the results for publication.

4.1 PROPORTIONAL ODDS MODELS: AN INTRODUCTION

In the last chapter, we focused on binary logistic regression models when the outcome variable is dichotomous with values of 1 and 0. In your research, you may often encounter ordinal outcome variables, which are categorical variables with ranks or orders, for example, student's socioeconomic status ordered from low to high; children's proficiency in early reading scored from level 0–5; and a response scale of a survey instrument with five levels, ordered from strongly disagree to strongly agree.

Research examples for ordinal response variables in the literature include a three-level response scale for an item related to astrology in the 2006 General Social Survey (Agresti, 2010), the deprivation level with three categories (not deprived, mildly deprived, or severely deprived) (Borooah, 2002), student persistence through high school with three levels (i.e., dropped out, still in school but behind peers, and persisted) (Heck et al., 2012), four-category drug user type (Menard, 2010), children's literacy proficiency level (O'Connell, 2006), and four-category level of severity of illness (Rabe-Hesketh & Skrondal, 2012).

In the preceding examples, the outcome variables of interest are ordinal response variables with more than two categories. In this chapter and the following two chapters, we will focus on various logistic regression models for ordinal response variables, which are referred to as ordinal logistic regression models when they are broadly defined. The first model, which is introduced in this chapter, is the proportional odds (PO) model. The PO model, which is also called the cumulative odds model, is one of the most commonly used models for the analysis of ordinal response data. It is so popular that sometimes we just call it the ordinal regression model or the ordered logit model. The PO model is a generalization of a binary logistic regression model when the response variable has more than two ordinal categories. It is used to estimate the odds of being at or below a particular level of the response variable. For example, if there are J levels of ordinal outcomes, then the model makes $J-1$ predictions, each estimating the odds of being at or below the jth level of the outcome variable, which are referred to as the cumulative odds. The odd ratios for each predictor variable are assumed to be the same across all categories, which is referred to as the PO assumption, or the parallel lines assumption. This model can also estimate the odds of being above a particular level of the ordinal response variable as well, because below and above a particular category are just two opposite directions.

The PO model can be expressed in the logit form as follows:

$$\text{logit}\left[\pi_j(x)\right] = \ln\left(\frac{\pi_j(x)}{1-\pi_j(x)}\right) = \alpha_j + (-\beta_1 X_1 - \beta_2 X_2 - \ldots - \beta_p X_p) \quad (4.1)$$

where $\pi_j(x) = P(Y \leq j \mid x_1, x_2, \ldots, x_p)$, which is the probability of being at or below category j given a set of predictors. $j = 1, 2, \ldots, J-1$. α_j are the cut points, and β_1,

$\beta_2, \ldots, \beta_p$ are the logit coefficients. To estimate the ln(odds) of being at or below the j-th category, the PO model can be rewritten as:

$$\begin{aligned}\operatorname{logit}\left[P\left(Y \leq j|x_1,\, x_2, \ldots, x_p\right)\right] &= \left(\frac{P\left(Y \leq j|x_1,\, x_2, \ldots, x_p\right)}{P\left(Y > j|x_1,\, x_2, \ldots, x_p\right)}\right) \\ &= \alpha_j + \left(-\beta_1 X_1 - \beta_2 X_2 - \ldots - \beta_p X_p\right)\end{aligned} \tag{4.2}$$

To understand the PO model for an ordinal response variable, we can think of it as several binary logistic regression models that are estimated simultaneously. The outcome variables of these binary models are dichotomized from the ordinal outcome variable comparing outcomes at or below a category ($Y \leq$ cat. j) and above that category ($Y >$ cat. j). Therefore, each binary logistic regression estimates odds of being at or below a category (coded as 1) versus above that category (coded as 0). Although each logistic model has a different intercept, the estimated logit coefficients are constrained to be equal. In other words, the regression lines are parallel, or the odds are proportional across the categories. Therefore, for each predictor variable, we only need to estimate one regression coefficient rather than multiple coefficients. This constraint is the proportional odds assumption or the parallel lines assumption.

The `clm()` function from the `ordinal` package (Christensen, 2019) in R uses Equation 4.2 to express the PO model where there are negative signs before the logit coefficients in the linear predictor, whereas the `vglm()` function in the `VGAM` package (Yee, 2010) uses a different parameterization with positive signs before the logit coefficients.

Although this chapter focuses on the commonly used logit link function for ordinal logistic regression models, the probit link can be used to fit ordinal probit models, where the cumulative probability of being at or below a particular category can be expressed as the cumulative standard normal distribution function. See Chapter 3 on the discussion of the probit model. For more information on ordinal regression models, refer to Agresti (2010, 2013, 2015, 2019), Ananth and Kleinbaum (1997), Armstrong and Sloan (1989), Clogg and Shihadeh (1994), Liu (2009, 2016a, 2016b), Liu et al. (2018), Fullerton and Xu (2016), Long (1997), Long and Freese (2014), McCullagh (1980), McCullagh and Nelder (1989), Menard (2010), O'Connell (2000, 2006), Powers and Xie (2008), Smithson and Merkle (2014), and Tutz (2012).

4.1.1 Odds and Odds Ratios in PO Models

In binary logistic regression, the values of the outcome variable are either 1 or 0, and we model the odds of success or of having an event when the outcome variable takes the value of 1 ($Y = 1$). The odds of success are the probability of success (p) divided by the probability of failure ($1 - p$).

In proportional odds models, the outcome variable is ordered with multiple levels, and we estimate the odds of being at or below a particular category ($Y \leq j$). Similar to the odds in binary logistic regression, the odds of being at or below a category in ordinal logistic regression equals the probability of being at or below a category divided by the probability of being above that category:

$$\text{Odds}(Y \leq j) = \frac{P(Y \leq j)}{P(Y > j)}$$

where $P(Y \leq j)$ is the cumulative probability of being at or below a category j or the cumulative probability of the ordinal response variable Y less than or equal to a category j. Since the probability of being at or below a category and the probability of being above that category is complementary, $P(Y \leq j) + P(Y > j) = 1$, this equation can be rewritten as:

$$\text{Odds}(Y \leq j) = \frac{P(Y \leq j)}{1 - P(Y \leq j)}.$$

It reads as follows: The odds of being at or below a category j in ordinal logistic regression equal the probability of being at or below a category divided by its complimentary probability, 1 minus the probability of being at or below that category. The probability of being at or below a category $P(Y \leq j)$ is the cumulative probability since it equals the sum of the probabilities of all categories at or below that category:

$$P(Y \leq j) = P(Y = 1) + P(Y = 2) + \ldots + P(Y = j) \text{ When } j = 1, 2, \ldots, J$$

For example, an outcome variable, health status, is ordinal with four levels from 1 to 4, where 1 = poor, 2 = fair, 3 = good, and 4 = excellent:

$$\begin{aligned}
P(Y \leq 4) &= P(Y = 1) + P(Y = 2) + P(Y = 3) + P(Y = 4) = 1 \\
P(Y \leq 3) &= P(Y = 1) + P(Y = 2) + P(Y = 3) \\
P(Y \leq 2) &= P(Y = 1) + P(Y = 2) \\
P(Y \leq 1) &= P(Y = 1)
\end{aligned}$$

The probability of being at a category $P(Y = j)$ is equal to the difference between the cumulative probability $P(Y \leq j)$ and the cumulative probability $P(Y \leq j - 1)$. It is written as: $P(Y = j) = P(Y \leq j) - P(Y \leq j - 1)$. Therefore, the probability of being at each category in the above example can be computed as follows:

$$\begin{aligned}
P(Y = 4) &= P(Y \leq 4) - P(Y \leq 3) \\
P(Y = 3) &= P(Y \leq 3) - P(Y \leq 2) \\
P(Y = 2) &= P(Y \leq 2) - P(Y \leq 1) \\
P(Y = 1) &= P(Y \leq 1)
\end{aligned}$$

Since this outcome variable has four categories, we can estimate the following cumulative odds: the odds of being at or below category 1, the odds of being at or below category 2, and the odds of being at or below category 3. The odds of being at or below a category in ordinal logistic regression are also called the cumulative odds.

Odds ($Y \leq 1$) equal the ratio of probability of being at or below category 1 to the probability of being above this category. The probability, $P(Y > 1) = P(Y = 2) + P(Y = 3) + P(Y = 4)$, which is the sum of the probabilities when $Y = 2, 3,$ and 4. We define $P(Y = j)$ to be $P(j)$, so the equation can be written as:

$$\begin{aligned}\text{Odds}(Y \leq 1) &= \frac{P(Y \leq j)}{1 - P(Y \leq j)} = \frac{P(Y = 1)}{P(Y = 2) + P(Y = 3) + P(Y = 4)} \\ &= \frac{P(1)}{P(2) + P(3) + P(4)}\end{aligned}$$

Odds ($Y \leq 2$) equal the ratio of probability of being at or below category 2 to the probability of being above this category. Since $P(Y \leq 2) = P(1) + P(2)$, and $P(Y > 2) = P(3) + P(4)$, the odds of being at or below category 2, can be expressed as follows:

$$\text{Odds}(Y \leq 2) = \frac{P(Y \leq 2)}{1 - P(Y \leq 2)} = \frac{P(1) + P(2)}{P(3) + P(4)}$$

Odds ($Y \leq 3$) equal the ratio of probability of being at or below category 3 to the probability of being above this category. Using the same method, we get the following equation:

$$\text{Odds}(Y \leq 3) = \frac{P(1) + P(2) + P(3)}{P(4)}$$

The odds of being at or below category 1 are the probability comparisons between category 1 and categories 2, 3, and 4; the odds of being at or below category 2 compare the probabilities of categories 1 and 2 with the probabilities of categories 3 and 4; and the odds of being at or below category 3 compare the probabilities of categories 1, 2, and 3 with the probability of category 4. Therefore, the cumulative odds in ordinal logistic regression are basically comparisons between two complimentary probabilities [i.e., $P(Y \leq j)$ and $P(Y > j)$]. Table 4.1 presents the logits, odds, and category comparisons for the PO model for the health status with four levels.

TABLE 4.1 ● Category Comparisons for the Proportional Odds Model With Four Levels of Health Status ($j = 1, 2, 3, 4$)

Category	Logit $P(Y \leq j)$	Odds	Probability Comparisons
Level 1	logit $P(Y \leq 1)$	$\frac{P(Y \leq 1)}{P(Y > 1)}$	Category 1 vs. categories 2–4
Level 2	logit $P(Y \leq 2)$	$\frac{P(Y \leq 2)}{P(Y > 2)}$	Categories 1 and 2 vs. categories 3 and 4
Level 3	logit $P(Y \leq 3)$	$\frac{P(Y \leq 3)}{P(Y > 3)}$	Categories 1, 2, and 3 vs. category 4

Odds Ratios in PO Models

In binary logistic regression, the odds ratio is the ratio of two odds, the odds of success when the value of a predictor is $(x + 1)$ relative to the odds when the predictor has a value of x. In other words, it is the change in the odds for a one-unit increase in the predictor variable. Similar to binary logistic regression, the odds ratio in PO models is the change in the odds (i.e., the odds of being above a particular category versus being at or below that category) for a one-unit increase from any value of x to the value of $(x + 1)$, and it is the exponentiated logit coefficient, $\exp(\beta)$. In contrast, the odds ratio of being at or below a particular category is the multiplicative inverse or reciprocal of the odds of being above that category. It is the exponentiated logit coefficient with a negative sign before that [i.e., $\exp(-\beta)$].

4.1.2 The PO Assumption

PO Assumption

In the proportional odds models, we assume that each predictor has the same effects across the categories of the ordinal outcome variable. In other words, the logit regression coefficients for each predictor are the same across the ordinal categories. For example, if we predict the ordinal outcome variable, health status, from the predictor, marital status, we estimate the odds of being at or below a category of health status relative to above that category, given that predictor variable. The estimated logits and the corresponding odds ratios of being at or below category 1, category 2, and category 3 for the predictor, marital status, are assumed to be the same. Although we assume that they are equal, how can we know whether the assumption holds?

Likelihood Ratio Test

To test whether the PO assumption is met, we can use the likelihood ratio test to look at the logit coefficients of a series of underlying binary logistic regression models for the dichotomized ordinal outcome variable, comparing outcomes at or below a category versus above that category.

The likelihood ratio test of the PO assumption can be examined using the `nominal_test()` function in the `ordinal` package. It provides the likelihood ratio test result for each predictor. We can also use the `lrtest()` function in the `VGAM` package to test the PO assumption, which provides the omnibus test for the overall model.

4.1.3 Goodness-of-Fit Statistics

Since ordinal logistic regression is an extension of binary logistic regression, all measures-of-fit statistics in binary logistic regression models, such as pseudo R^2 statistics, the deviance, the likelihood ratio test, and Akaike's information criterion (AIC) and Bayesian information criterion (BIC), can also be applied to proportional odds models.

4.1.4 Interpretation of Model Parameter Estimates

The odds ratio in ordinal logistic regression can be interpreted in a similar way as that of the binary logistic regression. In the binary logistic regression, we estimate the odds of success when the outcome takes the value of 1 (i.e., $Y = 1$), whereas in ordinal logistic regression, the odds are the ones when the outcomes are at or below a particular category (i.e., $Y \leq j$).

Recall that the signs before the logit coefficients in the equation of the ordinal logistic regression (Equation 4.1) are negative. To get the odds ratio (OR) of being at or below a category, we need to exponentiate the logit coefficient with a negative sign before that. This odds ratio can be interpreted as the change in the predicted logit or the log odds of being at or below a particular category for a one-unit increase in the predictor variable. By removing the negative sign and then exponentiating the logit coefficient, we get the OR of being above a category. In contrast, taking the multiplicative inverse of the odds of being at or below a particular category also gives us the odds of being above that category. The odds ratio of being a particular category can be interpreted as the change in the predicted logit or the log odds of being above that particular category for a one-unit increase in the predictor variable.

When the logit coefficient itself is positive, it indicates the relationship between the predictor variable and the logit function of the probability is positive. In other words, a positive coefficient increases the probability of being above a category. By exponentiating the logit coefficient, you get the OR, which is greater than 1. This means that the odds of being above a particular category increases for a one-unit increase in the predictor variable.

When the logit coefficient itself is negative, it indicates that the relationship between the predictor variable and the logit function is negative. A negative coefficient decreases the probability of being above a category. The exponentiated coefficient, the OR, is less than 1. This means that the odds of being above a particular category decreases for a one-unit increase in the predictor variable.

When the logit coefficient equals 0, the OR equals 1. This indicates that there is no relationship between the predictor and the odds, so there is no change in the odds when the values of the predictor variable change.

4.2 RESEARCH EXAMPLE AND DESCRIPTION OF THE DATA AND SAMPLE

Research Problem and Questions: In this chapter, the purpose of the research example is to investigate the relationships between the ordinal response variable, health status, and the three predictor variables: marital status, the highest education completed, and gender. The research question is as follows: Do the three predictor variables predict the ordinal response variable, health status? Specifically, do the three predictor variables predict the cumulative odds and then the cumulative probabilities of being at or below a particular level of health status, or the cumulative odds and then the cumulative probabilities of being above that health status level?

Description of the Data and Sample: The data for the following analyses were from the General Social Survey 2016 (GSS 2016). The following are the variables used for data analysis in this chapter:

- `healthre`: the recoded variable of health (health status) with four ordinal categories (1 = poor health, 2 = fair health, 3 = good health, and 4 = excellent health)
- `maritals`: the recoded variable of marital (marital status) with 1 = currently married and 0 = not currently married
- `educ`: the highest education completed
- `female`: recoded variable of sex with 1 = female and 0 = male

4.3 FITTING A ONE-PREDICTOR PO MODEL USING THE `clm()` FUNCTION

4.3.1 Packages and Functions for Proportional Odds Models in R

Several packages in R can be used for fitting PO models. This chapter focuses on the `ordinal` package (Christensen, 2019) and the `VGAM` package (Yee, 2010, 2015, 2021). The `clm()` function in `ordinal` and the `vglm()` function in `VGAM` are both used. The `clm()` function is introduced first.

4.3.2 The `clm()` Function in the `Ordinal` Package

The `clm()` function in the `ordinal` package is used for the ordinal logistic regression analysis, where `clm` stands for the cumulative link model with the logit link as the default. Since `ordinal` is a user-written package, you need to install it first by typing `install.packages("ordinal")` and then load the package by typing `library(ordinal)`. The basic syntax is `clm()` with the model formula which is specified within the parentheses after the function name `clm`. Writing the model formula for ordinal logistic regression in `clm()` is similar to that for the linear regression in `lm()`. The ordinal response variable and the independent variable(s) in the model are separated by the tilde (~). For example, the command `clm(y ~ x)` tells R to run a simple ordinal logistic regression analysis predicting the ordinal dependent variable *y* with an independent variable *x*. When there are more than multiple predictor variables in the formula, they are connected by plus (+) signs. For example, the model formula in `clm(y ~ x1 + x2)` includes two predictor variables, *x1* and *x2*. The default link function is the logit function, which can be omitted. To fit an ordinal probit model, we can use the `link = "probit"` argument. For more details on how to use this command, type `help(clm)` in the command prompt after loading the `ordinal` package.

4.3.3 The PO Model: One-Predictor Model With the `clm()` Function

The command `PO.1 <- clm(healthre ~ educ, data = chp4.po)` tells R to conduct the ordinal logistic regression to estimate the ordinal outcome variable `healthre` using the predictor variable `educ` with the `clm()` function. In the function, the outcome variable `healthre` is estimated by the predictor variable `educ` with a tilde (~). The `data = chp4.po` argument specifies the data frame. The output of the fitted model is defined as an object named `PO.1`. The `summary(PO.1)` command prints out the output, which is displayed as follows.

```
> library(foreign)
> chp4.po <- read.dta("C:/CDA/gss2016.dta")
> chp4.po$healthre <- factor(chp4.po$healthre, ordered=TRUE)
> chp4.po$educ <- as.numeric(chp4.po$educ)
> chp4.po$wrkfull <- as.numeric(chp4.po$wrkfull)
> chp4.po$maritals <- as.numeric(chp4.po$maritals)
> attach(chp4.po)

> # One-predictor model with the clm() function in ordinal
> library(ordinal)
> PO.1 <- clm(healthre ~ educ, data = chp4.po)
> summary(PO.1)
formula: healthre ~ educ
data:    chp4.po

   link    threshold   nobs   logLik     AIC       niter   max.grad    cond.H
   logit   flexible    1873   -2166.16   4340.32   5(0)    5.85e-08    1.4e+04

Coefficients:

         Estimate    Std. Error    z value    Pr(>|z|)
educ       0.1790        0.0152      11.78      <2e-16 ***

---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:

       Estimate    Std. Error    z value
1|2     -0.3554        0.2162     -1.644
2|3      1.5258        0.2092      7.294
3|4      3.8056        0.2262     16.823
```

4.3.4 Interpreting R Output

The output at the beginning displays the model formula and the data. It then shows the link function, threshold type, number of observations, log likelihood, AIC statistic, number of Newton-Raphson iterations, maximum absolute gradient of the log-likelihood function, and condition number of the Hessian in sequence.

The link function of the PO model is logit. The estimated thresholds or intercepts are unconstrained. The number of observations for the analysis is 1,873. The maximum log likelihood value is −2,166.16, and the AIC statistic is 4,340.32.

The next part shows the coefficients table (labeled `Coefficients:`). It includes the parameter estimates for the predictor variables, their standard errors, the Wald z statistics, and the associated p values. The null hypothesis for the Wald test is that the coefficient of the predictor variable is 0, and the alternative hypothesis is that the coefficient of the predictor variable is significantly different from 0. The logit regression coefficient of the predictor variable `educ` $\beta = .179$ and its Wald $z = 11.78$. The associated p value `Pr(>|z|)` $<$ `.001`, so we rejected the null hypothesis. Therefore, the predictor variable `educ` is a significant predictor of the ordinal outcome variable, health status.

The final part of the output displays the intercepts or the threshold coefficients table (labeled `Threshold Coefficients:`). It includes the parameter estimates for the intercepts or the thresholds, their standard errors, and the Wald z values.

4.3.5 Interpreting the Coefficients and the Intercepts/Thresholds

The logit coefficients can also be obtained using `coef(PO.1)` and their confident intervals can be obtained with `confint(PO.1)`.

```
> coef(PO.1)
        1|2          2|3          3|4         educ
-0.3553797    1.5258035    3.8055757    0.1790195
> confint(PO.1)
           2.5 %          97.5 %
educ    0.1493574    0.2089509
```

$\beta = .179$. It can be interpreted as follows: for a one-unit increase in the years of education completed, the change in the logit or log odds of being above a category of health status (i.e., better health status) is .179. The 95% confidence interval of the regression coefficient is `[.149, .209]`. It does not contain 0, which indicates the coefficient is significantly different from 0.

The threshold coefficients table reports the three intercepts or thresholds: `1|2`, `2|3`, and `3|4`. These are the estimated thresholds on the latent variable Y^* used to differentiate the adjacent levels of health status. When the response category is 1, the latent variable falls at or below the first cut point α_1. When the response category is 2, the latent variable falls between the first cut point α_1 and the second cut point α_2; when the response category reaches 3, the latent variable falls between the second α_2 and the third cut point α_3; and when the response category reaches 4, the latent variable is at or above the third cut point α_3. These thresholds are also called intercepts or cut points.

They can be thought of as the intercepts for three underlying binary logistic regression models if we dichotomize the ordinal outcome variable.

4.3.6 Odds Ratios

We use the `exp(coef(PO.1))` command to get the odds ratios and the `exp(confint(PO.1))` command to produce the corresponding confidence intervals. The output is shown as follows.

```
> exp(coef(PO.1))
      1|2          2|3          3|4         educ
0.7009073    4.5988372   44.9511225    1.1960440
> exp(confint(PO.1))
          2.5 %      97.5 %
educ   1.161088    1.232384
```

The odds ratio for the predictor variable `educ` is `1.196`. It equals the exponentiated regression coefficient exp(.179). The 95% confidence interval of the odds ratio is `[1.161, 1.232]`.

4.3.7 Interpreting the Odds Ratio of Being at or Below a Particular Category

The estimated logit regression coefficient $\beta = .533$, $z = 5.11$, $p < .001$, which indicates that education is a significant predictor of the ordinal response variable, health status. By substituting the value of the coefficient into Equation 4.2, logit $[P(Y \leq j \mid \texttt{educ})] = \alpha_j + (-\beta_1 X_1)$, we calculated logit $[P(Y \leq j \mid \texttt{educ})] = \alpha_j - .179\ (\texttt{educ})$. OR $= e^{(-.179)} = .836$, which indicates that for a one-unit increase in the years of education the odds of being at or below any category of health status (i.e., less healthy) decrease by .836.

To estimate the cumulative odds being at or below a certain category *j* for `educ`, let us take a look at the logit form of proportional odds model, logit $[P(Y \leq j \mid \texttt{educ})] = \alpha_j$ –.179 (`educ`). For example, when $Y \leq 1$, $\alpha_{1,}$ $-.355$ is the first cut point for the model. By substituting it into Equation 4.2, we get logit $[P(Y \leq j \mid \texttt{educ})] = -.355$ $-.179(\texttt{educ})$. For `educ` ($x = 1$), logit $[P(Y \leq 1 \mid \texttt{educ})] = -.534$. By exponentiating the logit, we calculate the odds of being at or below the category 1 (poor health) when `educ` $= 1$, $e^{(-.534)} = .586$. For `educ` ($x = 2$), logit $[P(Y \leq 1 \mid \texttt{educ})] =$ $-.355\ -.179 \times 2 = -.713$, so the odds of being at or below the category 1 (poor health) when `educ` $= 2$, $e^{-.713} =.490$. The odds ratio of `educ` ($x = 2$) relative to `educ` ($x = 1$) $= .490/.586 = .836$.

4.3.8 Interpreting the Odds Ratio of Being Above a Particular Category

The proportional odds model can also estimate the ln(odds) of being above a category j. Again, these ln(odds) can be transformed into the cumulative odds and cumulative probabilities. For example, we can estimate the cumulative probability of health status above category 3, $P(Y > 3)$; above category 2, $P(Y > 2)$; and above category 1, $P(Y > 1)$. The cumulative logit form can be expressed as logit $[P(Y > j \mid \text{educ})] = -\alpha_j + (\beta_1 X_1)$. When estimating the odds of being above category j, the sign of the cut points needs to be reversed and their magnitude remains unchanged since we estimate the cut points from the right to the left of the latent variable Y^*, that is, from the direction when $Y = 4$ approaches $Y = 1$. Therefore, three cut points from right to left turn to –3.806, –1.526, and .355.

When the predictor is dichotomous, a positive sign of the logit coefficient indicates that it is more likely for the group ($x = 1$) to be above a particular category than for the relative group ($x = 0$). When the predictor is continuous, a positive coefficient indicates that when the value of the predictor variable increases, the odds of being above a particular category increase.

The `exp(coef(PO.1))` command provides the odds ratios of being above a particular category: OR = 1.196. It can be interpreted that the odds of being above a particular category of health status (better health status) increase by a factor of 1.196 for each unit increase in years of education. In other words, the odds of being above a particular category of health status increase by 19.6% for each unit increase in education.

4.3.9 Model Fit Statistics

Testing the Overall Model Using the Likelihood Ratio Test

To test if the overall model is significant, we fit a null model with the intercept only and compare the one-predictor PO model with the null model using the `anova()` function. The command `PO.0 <- clm(healthre ~ 1, data = chp4.po)` is used to fit the null model. The output is displayed below by the `summary(PO.0)` command.

```
> # Null model with the intercept only
> PO.0 <- clm(healthre ~ 1, data = chp4.po)
> summary(PO.0)
formula: healthre ~ 1
data:    chp4.po

 link    threshold   nobs    logLik     AIC       niter    max.grad     cond.H
 logit   flexible    1873    -2238.22   4482.44   5(0)     4.56e-09     5.9e+00


Threshold coefficients:
        Estimate    Std. Error    z value
1|2     -2.69954      0.09510      -28.39
2|3     -0.89064      0.05087      -17.51
3|4      1.25964      0.05569       22.62
```

The `anova(PO.0, PO.1)` command compares the log-likelihood statistics of the fitted model `PO.1` and the null model `PO.0` using the likelihood ratio test.

```
> # Testing the overall model using the likelihood ratio test
> anova(PO.0, PO.1)
Likelihood ratio tests of cumulative link models:


        formula:              link: threshold:
PO.0    healthre ~ 1          logit flexible
PO.1    healthre ~ educ       logit flexible


        no.par      AIC      logLik    LR.stat    df   Pr(>Chisq)
PO.0         3   4482.4     -2238.2
PO.1         4   4340.3     -2166.2     144.11     1   < 2.2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis of the test for the overall model is that the predictor variable does not contribute to the model, and the alternative hypothesis is that the one-predictor PO model is better than the null model with no independent variables. The likelihood ratio test statistic $LR\ \chi^2_{(1)} = 144.11, p < .001$, which indicated that the overall model with one predictor was significantly different from zero. Therefore, the one-predictor PO model provides a better fit than the null model with no independent variables.

Pseudo R^2

The `nagelkerke(PO.1)` function in the `rcompanion` package (Mangiafico, 2021) produces the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the PO model. You need to install `rcompanion` first by typing `install.packages("rcompanion")` and then load it by typing `library(rcompanion)`.

```
> # Pseudo R2
> library(rcompanion)
> nagelkerke(PO.1)
$`Models`

Model: "clm, healthre ~ educ, chp4.po"
Null: "clm, healthre ~ 1, chp4.po"

$Pseudo.R.squared.for.model.vs.null
                                  Pseudo.R.squared
McFadden                                 0.0321935
Cox and Snell (ML)                       0.0740563
Nagelkerke (Cragg and Uhler)             0.0815267
```

```
$Likelihood.ratio.test
 Df.diff    LogLik.diff     Chisq       p.value
      -1        -72.056    144.11   3.358e-33

$Number.of.observations

Model:    1873
Null:     1873

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"
```

The McFadden R^2 is .032, the Cox and Snell R^2 is .074, and the Nagelkerke R^2 is .082. The same results can be computed using the equations introduced in the previous section. In the R command below, `LLM` is the log-likelihood value for the single-predictor model and `LL0` is the log-likelihood value for the null model. In addition, `McFadden` is the object name for the McFadden R^2, `CS` for the Cox and Snell R^2, and `NG` for the Nagelkerke R^2.

```
> LLM <- logLik(PO.1)
> LL0 <- logLik(PO.0)
> McFadden <- 1-(LLM/LL0)
'log Lik.' 0.03219349 (df=4)
> CS <- 1-exp(2*(LL0-LLM)/1873)
> CS
'log Lik.' 0.07405631 (df=3)
> NG <- CS/(1-exp(2*LL0/1873))
> NG
'log Lik.' 0.08152671 (df=3)
```

4.3.10 Using the Likelihood Ratio Test to Test the PO Assumption

The `nominal_test()` function in the `ordinal` package is used to test the PO assumption. It provides the likelihood ratio test result for each predictor. A nonsignificant test indicates that the proportional odds assumption is not violated for that predictor. The results of the `nominal_test(PO.1)` function are shown as follows.

```
> # PO assumption test
> nominal_test(PO.1)
Tests of nominal effects
```

```
 formula: healthre ~ educ
          Df     logLik      AIC        LRT    Pr(>Chi)
<none>          -2166.2    4340.3
educ       2    -2165.8    4343.5   0.82398      0.6623
```

The likelihood ratio test yields $\chi^2_{(2)} = .824$, $p = .662$, which indicates that the proportional odds assumptions for the model is upheld, suggesting that the effect of the explanatory variable educ is constant across the underlying binary models.

4.4 FITTING A MULTIPLE-PREDICTOR PO MODEL USING THE clm() FUNCTION

4.4.1 The PO Model: Multiple-Predictor Model With the clm() Function

The command PO.2 <- clm(healthre ~ maritals + educ + female, data = chp4.po) tells R to predict the ordinal response variable healthre from the three predictor variables maritals, educ, and female using ordinal logistic regression. The predictor variables are connected by plus signs in the model formula. The output is shown as follows after typing the summary(PO.2) command.

```
 > # Multiple-predictor model with the clm() function
 > PO.2 <- clm(healthre ~ maritals + educ + female, data = chp4.po)
 > summary(PO.2)
 formula: healthre ~ maritals + educ + female
 data:    chp4.po

  link   threshold   nobs   logLik     AIC      niter   max.grad   cond.H
  logit  flexible    1873   -2160.51   4333.01  5(0)    7.25e-08   1.5e+04


 Coefficients:
             Estimate   Std. Error   z value   Pr(>|z|)
 maritals     0.29157    0.08819      3.306    0.000946 ***
 educ         0.17502    0.01523     11.490     < 2e-16 ***
 female       0.06702    0.08760      0.765    0.444232
 ---
 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Threshold coefficients:
       Estimate   Std. Error   z value
 1|2    -0.2543     0.2222     -1.145
 2|3     1.6344     0.2158      7.574
 3|4     3.9243     0.2331     16.834
```

4.4.2 Interpreting R Output

The R output for the multiple-predictor PO model includes the model formula and the data. It then displays the link function, the threshold option, the number of observations, the log likelihood value, AIC, the number of Newton-Raphson iterations, the maximum absolute gradient of the log-likelihood function, and the condition number of the Hessian. The fourth and fifth parts show the coefficients table and the threshold coefficients table, respectively.

The number of observations for the analysis is 1,873. The maximum log likelihood value is −2,160.51, and the AIC statistic is 4,333.01.

The coefficients table (labeled `Coefficients:`) displays the parameter estimates for the three predictor variables, their standard errors, the Wald z statistics, and the associated p values.

For the predictor variable `maritals`, Wald $z = 3.306$. The associated p value, `Pr(>|z|) < .001`, so we rejected the null hypothesis.

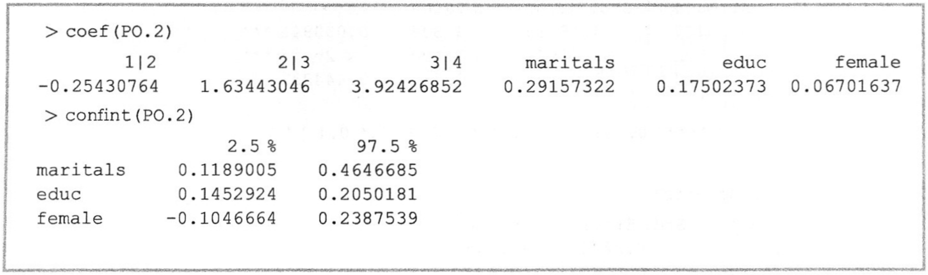
For the predictor variable `educ`, the Wald $z = 11.490$. The associated p value, `Pr(>|z|) < .001`, so we also reject the null hypothesis. Therefore, `maritals` and `educ` are significant predictors of the outcome variable.

For the predictor variable `female`, the Wald $z = .765$. The associated p value `Pr(>|z|) = .444`, so we fail to reject the null hypothesis and conclude that there is no significant effect of being female on the outcome variable.

The threshold coefficients table (labeled `Threshold Coefficients:`) includes the parameter estimates for the intercepts or the thresholds, their standard errors, and the Wald z values.

4.4.3 Interpreting the Coefficients and the Intercepts/Thresholds

The logit coefficients can also be obtained using the `coef(PO.2)` command and their confident intervals can be obtained with `confint(PO.2)`.



```
> coef(PO.2)
        1|2         2|3         3|4    maritals        educ      female
-0.25430764  1.63443046  3.92426852  0.29157322  0.17502373  0.06701637
> confint(PO.2)
              2.5 %     97.5 %
maritals  0.1189005  0.4646685
educ      0.1452924  0.2050181
female   -0.1046664  0.2387539
```

The logit coefficient for `maritals` $\beta = .292$. This means that the logit or log odds of being above a category of health status for the married is .292 points higher than that for the unmarried.

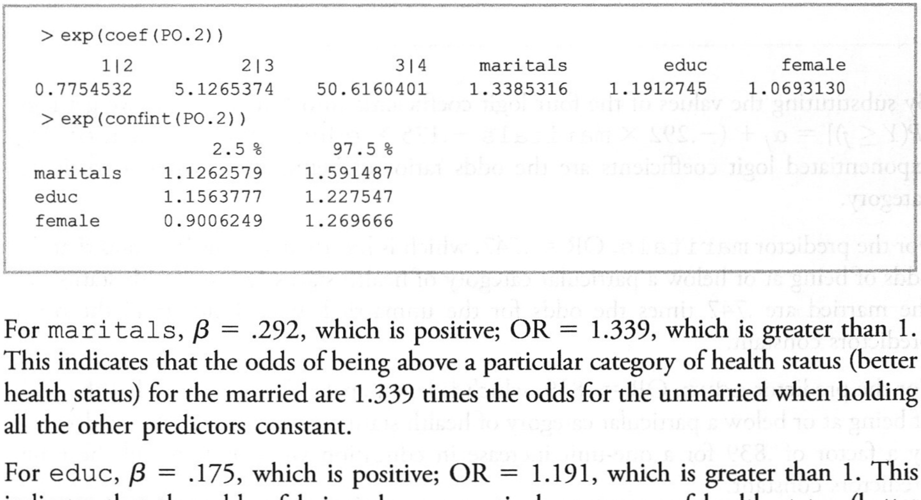
The logit coefficient for `educ` $\beta = .175$. This can be interpreted as the increase in the logit or log odds of being above a category of health status (i.e., better health status) is .179 for a one-unit increase in the years of education.

The logit coefficient for `female` $\beta = .067$. Since it is not significant ($p = .444$), being a female does not impact the logit or log odds of being above a category of health status.

The threshold coefficients table reports the three thresholds: `1|2`, `2|3`, and `3|4`. These are the estimated thresholds or intercepts to differentiate the adjacent categories of health status. The first intercept α_1 is $-.254$; the second intercept α_2 is 1.634; and the third intercept α_3 is 3.924.

4.4.4 Interpreting the Odds Ratios of Being Above a Particular Category

The `exp(coef(PO.2))` command provides the odds ratios of being above a category and the `exp(confint(PO.2))` command produces the corresponding confidence intervals. The following output is displayed.



```
> exp(coef(PO.2))
       1|2         2|3          3|4     maritals         educ       female
 0.7754532   5.1265374   50.6160401    1.3385316    1.1912745    1.0693130
> exp(confint(PO.2))
              2.5 %      97.5 %
maritals  1.1262579    1.591487
educ      1.1563777    1.227547
female    0.9006249    1.269666
```

For `maritals`, $\beta = .292$, which is positive; OR = 1.339, which is greater than 1. This indicates that the odds of being above a particular category of health status (better health status) for the married are 1.339 times the odds for the unmarried when holding all the other predictors constant.

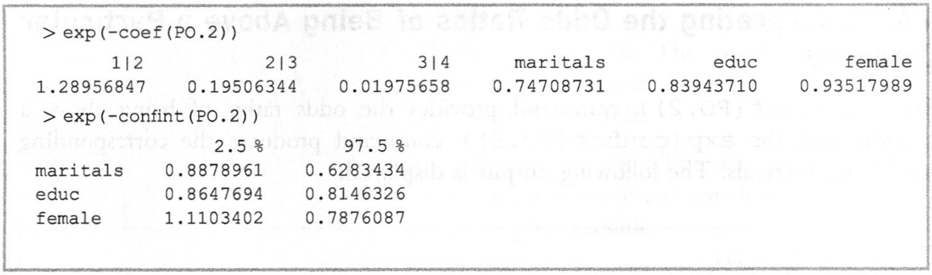
For `educ`, $\beta = .175$, which is positive; OR = 1.191, which is greater than 1. This indicates that the odds of being above a particular category of health status (better health status) increase by a factor of 1.191 for a one-unit increase in the predictor, education, when holding all the other predictors constant. In other words, for a one-unit increase in education, the odds of being healthier increase by 19.1%.

For `female`, $\beta = .067$, $p = .444$, which is not significantly different from 0; OR = 1.069, which almost equals 1. This indicates that there is no relationship between being a female and the cumulative odds of being in better health status. In other words, there is no significant difference between the males and females in better health status.

4.4.5 Interpreting the Odds Ratios of Being at or Below a Particular Category

In the preceding section, we interpreted the odds ratio of being above a category. We can also interpret how these predictor variables contribute to the odds of being at or below a particular category if we reverse the sign before the estimated logit coefficients and then compute the corresponding odds ratios.

The `exp(-coef(PO.2))` command tells R to reverse the odds of being above a category versus being at or below that category to the odds of being at or below a category versus above that category. The following is the output produced by the command.



```
> exp(-coef(PO.2))
        1|2         2|3         3|4     maritals        educ      female
 1.28956847  0.19506344  0.01975658   0.74708731  0.83943710  0.93517989
> exp(-confint(PO.2))
              2.5 %      97.5 %
maritals  0.8878961   0.6283434
educ      0.8647694   0.8146326
female    1.1103402   0.7876087
```

By substituting the values of the four logit coefficients into Equation 4.2, we get logit $[P(Y \leq j)] = \alpha_j + (-.292 \times$ `maritals` $-.175 \times$ `educ` $-.067 \times$ `female`). The exponentiated logit coefficients are the odds ratios of being at or below a particular category.

For the predictor `maritals`, OR = .747, which is less than 1. This indicates that the odds of being at or below a particular category of health status (worse health status) for the married are .747 times the odds for the unmarried when holding all the other predictors constant.

For the predictor `educ`, OR = .839, which is less than 1. This indicates that the odds of being at or below a particular category of health status (poorer health status) decrease by a factor of .839 for a one-unit increase in education when holding all the other predictors constant.

For the predictor `female`, OR = .935 ($p = .444$), which is close to 1. This indicates that there is no relationship between being a female and the cumulative odds of being in poorer health status.

4.4.6 Computing the Predicted Probabilities With the `ggpredict()` Function in the `ggeffects` Package

Since the `margins` package (Leeper, 2021) has not been fully developed for the ordinal regression models, introduction of the marginal effects is omitted. We use the `ggpredict()` function in the `ggeffects` package (Lüdecke, 2018b) compute the predicted probabilities of being in a particular category of the ordinal response variable at specified values of predictor variables. The command is as follows: `margins.e <- ggpredict(PO.2, terms = "educ[12, 14, 16]")`. In the `ggpredict()` function, `PO.2` is the fitted model; and the `terms = "educ[12, 14, 16]"` option specifies the predictor variable `educ` at the values of 12, 14, and 16 when holding the other predictor variables at their means. The `terms` option can specify up to four variables, including the second to fourth grouping variables. The output is assigned to an object named `margins.e`. To request the standard errors of the predicted probabilities, we can use either the `as.data.frame()` or the `sqrt(diag(vcov()))` function.

```
> # Predicted probabilities with ggpredict() in ggeffects
> library(ggeffects)
> margins.e <- ggpredict(PO.2, terms = "educ[12, 14, 16]")
> margins.e

# Predicted probabilities of healthre

# Response Level = 1

educ   |   Predicted   |        95% CI
--------------------------------------
  12   |        0.07   |  [0.06, 0.09]
  14   |        0.05   |  [0.04, 0.07]
  16   |        0.04   |  [0.03, 0.05]

# Response Level = 2

educ   |   Predicted   |        95% CI
--------------------------------------
  12   |        0.27   |  [0.25, 0.30]
  14   |        0.22   |  [0.20, 0.24]
  16   |        0.17   |  [0.15, 0.19]

# Response Level = 3

educ   |   Predicted   |        95% CI
--------------------------------------
  12   |        0.49   |  [0.47, 0.52]
  14   |        0.51   |  [0.49, 0.54]
  16   |        0.51   |  [0.49, 0.54]
```

```
# Response Level = 4

educ    |  Predicted    |         95% CI
----------------------------------------
  12    |       0.16    |   [0.14, 0.18]
  14    |       0.21    |   [0.19, 0.23]
  16    |       0.28    |   [0.25, 0.31]

 Adjusted for:
 * maritals = 0.44
 *   female = 0.56

 > plot(margins.e)
```

The predicted probabilities for each response level are listed in sequence. For each response level, the first column in the table of the output lists `educ` at the values of 12, 14, and 16. The remaining columns list the predicted probabilities and the lower and upper confidence intervals. The predicted probabilities of being in poor health (i.e., response level = 1) are .07, .05, and .04, respectively. The predicted probabilities of being in the other three categories are also listed in the output. The last section titled "`Adjusted for`" lists the means of the other variables.

The predicted probabilities for all four response levels are plotted using the `plot (margins.e)` command. Figure 4.1 shows the predicted probabilities of being in each category (i.e., $Y = 1, 2, 3,$ and 4) for `educ` at 12, 14, and 16.

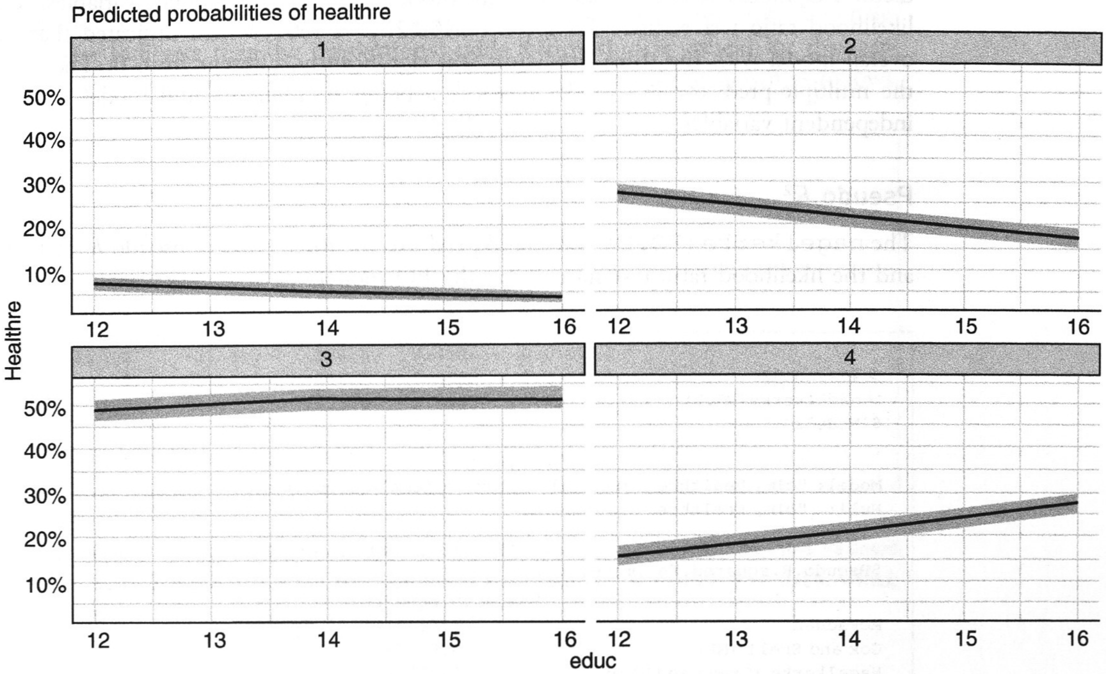
The graph shows that with the increase in the years of education, the probabilities of being in poor and fair health condition (categories 1 and 2) decrease. In other words, people with higher levels of education are less likely to be associated with poor and fair health conditions. In addition, with the increase in the years of education, the probabilities of being in good and excellent health conditions (categories 3 and 4) increase. In other words, people with a higher level of education are more likely to be in good and excellent health conditions.

4.4.7 Model Fit Statistics

Testing the Overall Model Using the Likelihood Ratio Test

To test if the overall model is significant, we fit a null model with the intercept only and compare it with the multiple-predictor PO model by using the `anova()` function. Since the null model is fitted in the previous section, the output is omitted here. The `anova(PO.0, PO.2)` command compares the log-likelihood statistics of the fitted model `PO.2` and the null model `PO.0` using the likelihood ratio test.

FIGURE 4.1 ⬢ Predicted Probabilities of Being in Categories 1, 2, 3, and 4 for educ



```
> # Testing the overall model using the likelihood ratio test
> anova(PO.0, PO.2)
Likelihood ratio tests of cumulative link models:

    formula:                                  link: threshold:
PO.0 healthre ~ 1                             logit flexible
PO.2 healthre ~ maritals + educ + female      logit flexible


       no.par    AIC    logLik  LR.stat  df  Pr(>Chisq)
PO.0        3  4482.4  -2238.2
PO.2        6  4333.0  -2160.5   155.42   3   < 2.2e-16 ***


---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis of the test for the overall model is that the predictor variables do not contribute to the model, and the alternative hypothesis is that the multiple-predictor PO model is better than the null model with no independent variables. The likelihood ratio test statistic $LR\ \chi^2_{(3)} = 155.42$, $p < .001$, which indicated that the overall model with the three predictors was significantly different from 0. Therefore, the multiple-predictor PO model provides a better fit than the null model with no independent variables.

Pseudo R^2

The nagelkerke(PO.2) command produces the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the PO model.

```
> #PseudoR2
> nagelkerke(PO.2)
$`Models`

Model: "clm, healthre ~ maritals + educ + female, chp4.po"
Null:  "clm, healthre ~ 1, chp4.po"

$Pseudo.R.squared.for.model.vs.null
                              Pseudo.R.squared
McFadden                             0.0347206
Cox and Snell (ML)                   0.0796320
Nagelkerke (Cragg and Uhler)         0.0876648

 $Likelihood.ratio.test
Df.diff    LogLik.diff     Chisq      p.value
     -3        -77.712    155.42   1.7801e-33

$Number.of.observations

Model:    1873
Null:     1873

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"
```

McFadden's R^2 is .035, Cox and Snell's R^2 is .080, and Nagelkerke's R^2 is .088.

4.4.8 Using the Likelihood Ratio Test to Test the PO Assumption

We use the nominal_test() function in the ordinal package to test the PO assumption. It provides the likelihood ratio test result for each predictor. A nonsignificant test indicates that the proportional odds assumption is upheld for that predictor. The results of the nominal_test(PO.2) command are shown as follows.

```
> # PO assumption test
> nominal_test(PO.2)
Tests of nominal effects

formula: healthre ~ maritals + educ + female
           Df     logLik       AIC        LRT      Pr(>Chi)
<none>            -2160.5     4333.0
maritals    2     -2156.3     4328.5     8.4986    0.01427   *
educ        2     -2160.1     4336.1     0.8962    0.63884
female      2     -2160.3     4336.6     0.3922    0.82194
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The proportional odds assumption is upheld for educ and female, whereas it is violated for maritals. For maritals, the likelihood ratio test $\chi^2_{(2)} = 8.499, p = .014$, which is significant.

4.4.9 Model Comparison Using the Likelihood Ratio Test

The likelihood ratio test or the deviance difference test is used to compare the full model and the one-predictor model. Recall that this test compares the reduced model, which contains less parameters, and the full model, which contains all parameters. The difference in deviance is often expressed as $G =$ Deviance for the reduced model $-$ Deviance for the full model or as $D_{\text{Reduced}} - D_{\text{Full}}$. The difference in deviance between nested models has a chi-square distribution with the degrees of freedom equal to the difference in the number of parameters between these two models.

The anova() function is used for the likelihood ratio test or the deviance difference test. Next, we compare the simple-predictor PO model and the multiple-predictor PO model with the anova(PO.1, PO.2) command.

```
> # Model comparison using the likelihood ratio test
> anova(PO.1, PO.2)
Likelihood ratio tests of cumulative link models:

        formula:                                link: threshold:
PO.1    healthre ~ educ                         logit flexible
PO.2    healthre ~ maritals + educ + female     logit flexible
```

```
        no.par     AIC    logLik  LR.stat   df   Pr(>Chisq)
PO.1         4  4340.3   -2166.2
PO.2         6  4333.0   -2160.5   11.313    2    0.003496 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The likelihood ratio chi-square test $\chi^2_{(2)} = 11.313, p < .001$. This result indicates that the full model has a better fit than the one-predictor model. The same result can be obtained if we compute it using the following equation:

$$G = D_{\text{Reduced}} - D_{\text{Full}} = -2 \times [-2166.2 - (-2160.5)]$$
$$= 11.4, \text{ df} = 6 - 4 = 2$$

4.5 FITTING A SINGLE-PREDICTOR PO MODEL USING THE vglm() FUNCTION

4.5.1 The vglm() Function in the VGAM Package

The vglm() function in the VGAM package can also be used for the ordinal logistic regression analysis, where vglm stands for vector generalized linear models. You need to install the VGAM package first by typing install.packages("VGAM") since it is a user-written package. After installation, load the package by typing library (VGAM). The basic model formula command for vglm() is similar to that for either introduced in this chapter or glm() introduced in Chapter 2. In addition to the model formula, the family argument is needed for different types of models. For example, the command vglm(y ~ x, family = cumulative(parallel = TRUE), data = data1) tells R to fit a simple cumulative odds model predicting the ordinal dependent variable *y* with an independent variable *x*. The ordinal response variable and the independent variable in the model are separated by the tilde (~). The argument family = cumulative(parallel = TRUE) specifies the VGAM family function. It tells R to fit a cumulative odds model with the proportional odds assumption being specified. The data argument specifies the data frame used for the analysis. For more details on how to use this command, type help(vglm), help(propodds), and help(cumulative) in the command prompt after loading the VGAM package.

4.5.2 Using the vglm() Function to Fit a Single-Predictor PO Model

To fit the same single-predictor PO model introduced in the earlier section, we use the command model1 <- vglm(healthre ~ educ, cumulative(parallel = TRUE, reverse = FALSE), data = chp4.po). Following the model equation, `healthre ~ educ`, the argument `cumulative(parallel = TRUE, reverse = FALSE)` tells R to fit a cumulative odds model with the parallel odds or proportional odds assumption and nonreversed ordinal categories. The `data = chp4.po` argument specifies the data frame. The `summary(model1)` command produces the following output.

```
> # One-predictor model with the vglm() function in VGAM
> library(VGAM)
> model1 <- vglm(healthre ~ educ, cumulative(parallel = TRUE, reverse = FALSE),
data = chp4.po)
> summary(model1)


Call:
vglm(formula = healthre ~ educ, family = cumulative(parallel = TRUE,
     reverse = FALSE), data = chp4.po)


Pearson residuals:
                          Min        1Q    Median       3Q     Max
logitlink(P[Y<=1])    -0.9413   -0.2425   -0.1716  -0.1312   7.102
logitlink(P[Y<=2])    -2.2568   -0.7070   -0.3115   0.5197   3.031
logitlink(P[Y<=3])    -6.6746    0.1568    0.3433   0.6484   1.013


Coefficients:
                 Estimate   Std. Error   z value   Pr(>|z|)
(Intercept):1    -0.35539      0.21332    -1.666     0.0957 .
(Intercept):2     1.52580      0.20608     7.404   1.32e-13 ***
(Intercept):3     3.80557      0.22311    17.057    < 2e-16 ***
educ             -0.17902      0.01497   -11.955    < 2e-16***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


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


Residual deviance: 4332.323 on 5615 degrees of freedom


Log-likelihood: -2166.162 on 5615 degrees of freedom


Number of Fisher scoring iterations: 4


No Hauck-Donner effect found in any of the estimates


Exponentiated coefficients:
     educ
0.8360901
```

4.5.3 Interpreting R Output

The R output includes the call, the Pearson residuals, the coefficients, the number and names of the three linear predictors, the residual deviance, the log-likelihood value, the number of iterations, and the exponentiated coefficients.

The intercepts in the coefficients table are the same as those in the threshold coefficients table from the `clm ()` function. The logit coefficient of the `educ` predictor has the same magnitude as that in the coefficients table from the `clm ()` function, but has a negative sign. This is due to different parameterizations between the `clm ()` function and the `vglm ()` function. In the PO model equation for the `vglm ()` function, the signs before the coefficients are positive as follows.

$$\text{logit}\left[P\left(Y \leq j|x_1, x_2, \ldots, x_p\right)\right] = \alpha_j + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \tag{4.3}$$

In PO models, we estimate the logit or the log odds of being at or below a particular category ($Y \leq j$). The link functions for the three linear predictors in the model are `logit (P[Y<=1])`, `logit(P[Y<=2])`, and `logit(P[Y<=3])`. The log odds of being at or below category 1, `logit(P[Y<=1])`, compares the probability of category 1 to the probabilities of categories 2, 3, and 4; the log odds of being at or below category 2, `logit(P[Y<=2])`, compares the probabilities of categories1 and 2 to the probabilities of categories 3 and 4; and the log odds of being at or below category 3, `logit(P[Y<=3])`, compares the probabilities of categories 1, 2, and 3 to the probability of category 4.

4.5.4 Odds Ratios

The `exp(coef(model1, matrix= TRUE))` command provides the odds ratios of being at or below a category and the `exp(confint(model1, matrix = TRUE))` command produces the corresponding confidence intervals. We use the `cbind (exp(coef(model1)), exp(confint(model1)))` command to combine the odds ratios and the confidence intervals. The following output is displayed.

```
> exp(coef(model1, matrix = TRUE))
                logit(P[Y<=1])    logit(P[Y<=2])    logit(P[Y<=3])
(Intercept)          0.7009020         4.5988016        44.9507590
educ                 0.8360901         0.8360901         0.8360901
> exp(confint(model1, matrix = TRUE))
                    2.5 %         97.5 %
(Intercept):1    0.4614005     1.0647226
(Intercept):2    3.0706560     6.8874455
(Intercept):3   29.0288233    69.6056713
educ             0.8119083     0.8609921
> cbind(exp(coef(model1)), exp(confint(model1)))
                                   2.5 %         97.5 %
(Intercept):1    0.7009020     0.4614005      1.0647226
(Intercept):2    4.5988016     3.0706560      6.8874455
(Intercept):3   44.9507590    29.0288233     69.6056713
educ             0.8360901     0.8119083      0.8609921
```

4.5.5 AIC Statistic

We can get the AIC statistic using AIC(model1).

```
> AIC(model1)
[1] 4340.323
```

4.5.6 Logit Coefficients of Being at or Above a Category

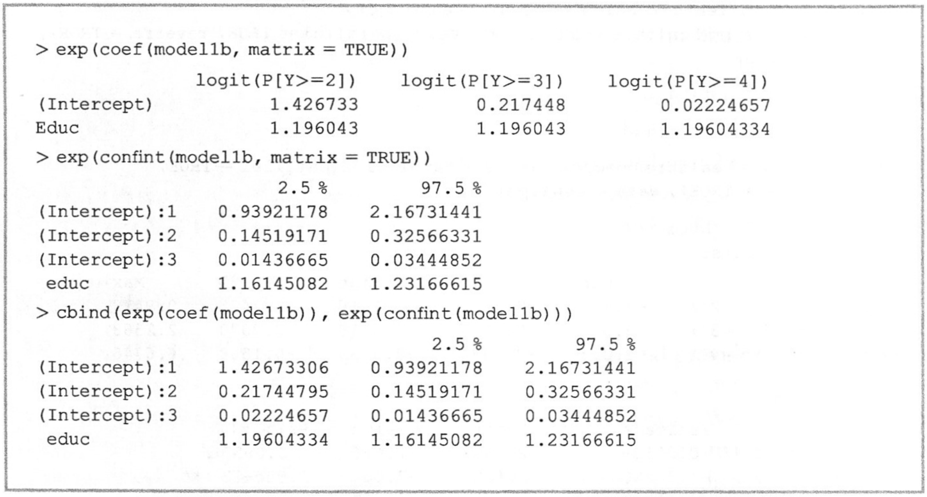
With the reverse = TRUE option, we can estimate the logit coefficients of being at or above a particular category of an ordinal outcome variable. The summary(model1b) command produces the following output.

```
> # Logit coefficients of being at or above a category
> model1b <- vglm(healthre ~ educ, cumulative(parallel = TRUE, reverse = TRUE),
data = chp4.po)
> summary(model1b)


Call:
vglm(formula = healthre ~ educ, family = cumulative(parallel = TRUE,
      reverse = TRUE), data = chp4.po)


Pearson residuals:
                            Min       1Q    Median       3Q     Max
logitlink(P[Y>=2])       -7.102   0.1312    0.1716   0.2425  0.9413
logitlink(P[Y>=3])       -3.031  -0.5197    0.3115   0.7070  2.2568
logitlink(P[Y>=4])       -1.013  -0.6484   -0.3433  -0.1568  6.6746

Coefficients:
                Estimate   Std. Error   z value    Pr(>|z|)
(Intercept):1    0.35539      0.21332     1.666      0.0957 .
(Intercept):2   -1.52580      0.20608    -7.404    1.32e-13 ***
(Intercept):3   -3.80557      0.22311   -17.057     < 2e-16 ***
educ             0.17902      0.01497    11.955     < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


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


Residual deviance: 4332.323 on 5615 degrees of freedom


Log-likelihood: -2166.162 on 5615 degrees of freedom


Number of Fisher scoring iterations: 4


No Hauck-Donner effect found in any of the estimates


Exponentiated coefficients:
    educ
1.196043
```

The intercepts and the coefficient in the output by the `summary(model1b)` command and in that by the `summary(model1)` command have opposite signs since the former model estimates of the log odds of being at or below a particular category ($Y \leq j$), whereas the latter model estimates the log odds of being at or above a particular category ($Y \geq j+1$). Please note that the log odds ($Y \geq j+1$) equal the log odds ($Y > j$).

4.5.7 Odds Ratios of Being at or Above a Category

We again use the `exp(coef(model1b, matrix = TRUE))` command to obtain the odds ratios of being at or above a category and use the `exp(confint(model1b, matrix = TRUE))` command to produce the corresponding confidence intervals. The results are combined using `cbind(exp(coef(model1b)), exp(confint(model1b)))`. The following output is created.



```
> exp(coef(model1b, matrix = TRUE))
                logit(P[Y>=2])      logit(P[Y>=3])      logit(P[Y>=4])
(Intercept)            1.426733            0.217448          0.02224657
Educ                   1.196043            1.196043          1.19604334
> exp(confint(model1b, matrix = TRUE))
                     2.5 %         97.5 %
(Intercept):1   0.93921178    2.16731441
(Intercept):2   0.14519171    0.32566331
(Intercept):3   0.01436665    0.03444852
 educ           1.16145082    1.23166615
> cbind(exp(coef(model1b)), exp(confint(model1b)))
                                   2.5 %          97.5 %
(Intercept):1   1.42673306    0.93921178      2.16731441
(Intercept):2   0.21744795    0.14519171      0.32566331
(Intercept):3   0.02224657    0.01436665      0.03444852
 educ           1.19604334    1.16145082      1.23166615
```

4.6 FITTING A MULTIPLE-PREDICTOR PO MODEL USING THE `vglm()` FUNCTION

4.6.1 Using the `vglm()` Function to Fit a Multiple-Predictor PO Model

To fit the same multiple-predictor PO model in the preceding section, we use the following command: `model2 <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = TRUE, reverse = FALSE), data = chp4.po)`. The resulting output is displayed as follows.

```
> # Multiple-predictor model with the vglm() function in VGAM
> model2 <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = TRUE,
reverse = FALSE), data = chp4.po)
> summary(model2)


Call:
vglm(formula = healthre ~ educ + maritals + female, family = cumulative(parallel =
TRUE,
  reverse = FALSE), data = chp4.po)


Pearson residuals:
                         Min        1Q     Median        3Q     Max
logitlink(P[Y<=1])   -0.8733   -0.2335   -0.1697   -0.1288   7.507
logitlink(P[Y<=2])   -2.3796   -0.7354   -0.3226    0.5107   3.293
logitlink(P[Y<=3])   -5.9188    0.1555    0.3464    0.6210   1.086

Coefficients:
                Estimate    Std. Error    z value    Pr(>|z|)
(Intercept):1   -0.25432      0.21950     -1.159    0.246613
(Intercept):2    1.63442      0.21261      7.687     1.5e-14 ***
(Intercept):3    3.92426      0.23005     17.058     < 2e-16 ***
educ            -0.17502      0.01503    -11.646     < 2e-16 ***
maritals        -0.29157      0.08842     -3.297    0.000976 ***
female          -0.06702      0.08754     -0.766    0.443957
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


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


Residual deviance: 4321.01 on 5613 degrees of freedom


Log-likelihood: -2160.505 on 5613 degrees of freedom


Number of Fisher scoring iterations: 4


No Hauck-Donner effect found in any of the estimates



Exponentiated coefficients:
     educ      maritals       female
0.8394376    0.7470883    0.9351809
```

The `coef(model2, matrix = TRUE)` command produces the coefficients table. Again, we can get the odds ratios and the corresponding confidence intervals using `exp(coef(model2, matrix = TRUE))` and `exp(confint(model2,` matrix = TRUE)), respectively. The cbind(exp(coef(model2)), exp(confint(model2))) command combines the results.

```
> coef(model2, matrix = TRUE)
              logit(P[Y<=1])    logit(P[Y<=2])    logit(P[Y<=3])
(Intercept)       -0.2543170         1.6344197         3.9242577
educ              -0.1750231        -0.1750231        -0.1750231
maritals          -0.2915719        -0.2915719        -0.2915719
female            -0.0670153        -0.0670153        -0.0670153

> exp(coef(model2, matrix = TRUE))
              logit(P[Y<=1])    logit(P[Y<=2])    logit(P[Y<=3])
(Intercept)        0.7754460         5.1264821        50.6154915
educ               0.8394376         0.8394376         0.8394376
maritals           0.7470883         0.7470883         0.7470883
female             0.9351809         0.9351809         0.9351809

> exp(confint(model2, matrix = TRUE))
                      2.5 %      97.5 %
(Intercept):1     0.5043268   1.1923151
(Intercept):2     3.3794230   7.7767178
(Intercept):3    32.2449950  79.4519578
educ              0.8150731   0.8645305
maritals          0.6282100   0.8884625
female            0.7877356   1.1102245

> cbind(exp(coef(model2)), exp(confint(model2)))
                                  2.5 %      97.5 %
(Intercept):1     0.7754460   0.5043268   1.1923151
(Intercept):2     5.1264821   3.3794230   7.7767178
(Intercept):3    50.6154915  32.2449950  79.4519578
educ              0.8394376   0.8150731   0.8645305
maritals          0.7470883   0.6282100   0.8884625
female            0.9351809   0.7877356   1.1102245
```

The AIC statistic of the fitted model can be obtained with AIC(model2).

```
> AIC(model2)
[1] 4333.01
```

We can also use the nagelkerke() function in the rcompanion package to obtain the three types of pseudo R^2 statistics for model2. The syntax is nagelkerke(model2). The results are omitted here.

4.6.2 Logit Coefficients of Being at or Above a Category in the Multiple-Predictor PO Model

We add the `reverse = TRUE` option to the multiple-predictor PO model so we can estimate the logit coefficients of being at or above a particular category of the ordinal outcome variable. The `summary(model2b)` command produces the following output.

```
> # Logit coefficients of being at or above a category with reverse = TRUE
> model2b <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = TRUE,
reverse = TRUE), data = chp4.po)
> summary(model2b)

Call:
vglm(formula = healthre ~ educ + maritals + female, family = cumulative(parallel =
TRUE,
  reverse = TRUE), data = chp4.po)

Pearson residuals:
                        Min        1Q     Median        3Q      Max
logitlink(P[Y>=2])   -7.507    0.1288     0.1697    0.2335   0.8733
logitlink(P[Y>=3])   -3.293   -0.5107     0.3226    0.7354   2.3796
logitlink(P[Y>=4])   -1.086   -0.6210    -0.3464   -0.1555   5.9188

Coefficients:
                 Estimate   Std. Error   z value   Pr(>|z|)
(Intercept):1     0.25432      0.21950     1.159   0.246613
(Intercept):2    -1.63442      0.21261    -7.687    1.5e-14 ***
(Intercept):3    -3.92426      0.23005   -17.058    < 2e-16 ***
educ              0.17502      0.01503    11.646    < 2e-16 ***
maritals          0.29157      0.08842     3.297   0.000976 ***
female            0.06702      0.08754     0.766   0.443957

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

Residual deviance: 4321.01 on 5613 degrees of freedom

Log-likelihood: -2160.505 on 5613 degrees of freedom

Number of Fisher scoring iterations: 4

No Hauck-Donner effect found in any of the estimates



Exponentiated coefficients:
    educ   maritals    female
1.191274   1.338530  1.069312
```

We can obtain the coefficients table, the odds ratios, and the corresponding confidence intervals with the `coef()`, `exp(coef())`, `exp(confint())`, and `cbind()` functions. The output is omitted here.

4.6.3 Computing the Predicted Probabilities With the `predict()` Function

We can use the `predict()` function to compute the predicted probabilities of being in a particular category of the ordinal response variable. For example, we would like to compute the predicted probabilities for `educ` at the specified values of 12, 14, and 16 when holding the other predictor variables at their means. We first create a data frame with the `data.frame()` function and then apply the `predict()` function. In the `data.frame()` function, `educ = c(12, 14, 16)` specifies the values of `educ`; `maritals = rep(mean(maritals), 3)` repeats the mean of `maritals` three times; and `female = rep(mean(female), 3)` repeats the mean of `female` three times. The created data frame is assigned to an object named `new1`.

```
> new1 <- data.frame(educ = c(12, 14, 16),
+                     maritals = rep(mean(maritals), 3),
+                     female = rep(mean(female), 3))
> new1
    educ    maritals     female
1     12   0.4372664  0.5563267
2     14   0.4372664  0.5563267
3     16   0.4372664  0.5563267
```

In the `predict()` function, we first specify the model object `model2` and then the `newdata = new1` argument, followed by the `type = "response"` argument for the predicted probabilities. The predicted probabilities labeled from `pred.prob.1` to `pred.prob.4` are provided in the data frame named `new1`.

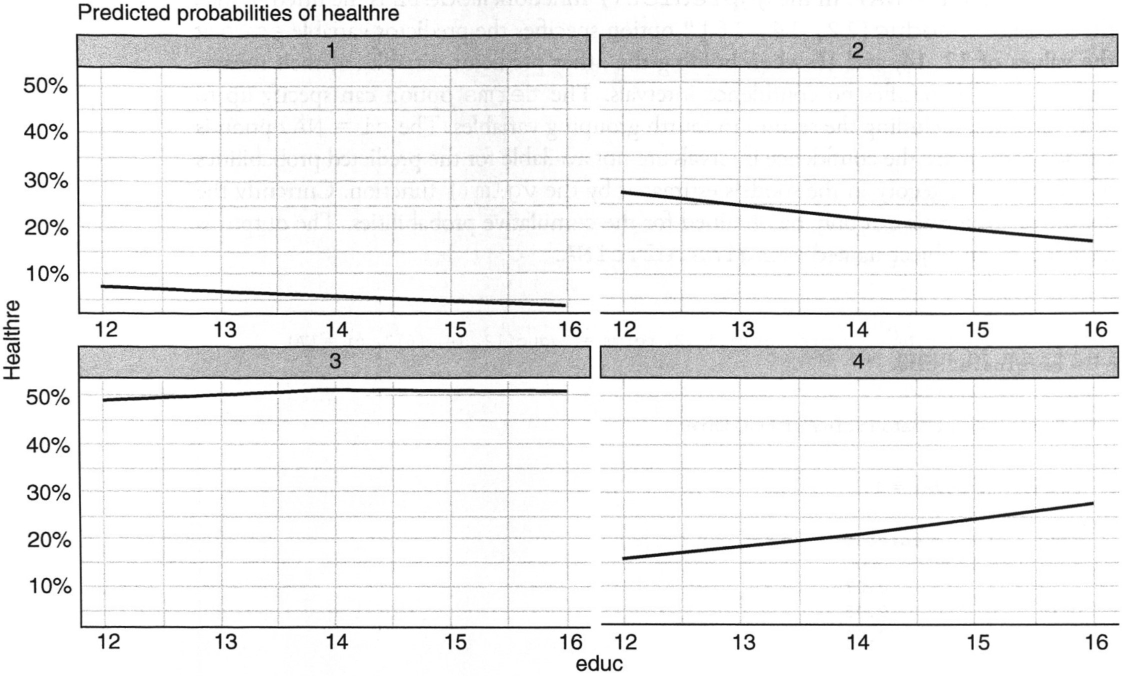
```
> new1[ , c('pred.prob')] <- predict(model2, newdata = new1, type = "response")
> new1
    educ    maritals     female  pred.prob.1  pred.prob.2  pred.prob.3  pred.prob.4
1     12   0.4372664  0.5563267   0.07451130   0.27285538   0.49276437   0.15986894
2     14   0.4372664  0.5563267   0.05368624   0.21907049   0.51461537   0.21262789
3     16   0.4372664  0.5563267   0.03843981   0.17059935   0.51390465   0.27705620
```

4.6.4 Computing the Predicted Probabilities With the `ggpredict()` Function in the `ggeffects` Package

We can also use the `ggpredict()` function in the `ggeffects` package (Lüdecke, 2018b) to compute the predicted probabilities of being in a particular category of the ordinal response variable at specified values of the predictor variables. The command is as follows: margins.e2.ciNA <- ggpredict(model2, terms = "educ[12, 14, 16]", ci = NA). In the ggpredict() function, model2 is the fitted model; the terms = "educ[12, 14, 16]" option specifies the predictor variable educ at the values of 12, 14, and 16 when holding the other predictor variables at their means; and ci = NA specifies no confidence intervals. The terms option can specify up to four variables, including the second to fourth grouping variables. The ci = NA option is needed there since the confidence intervals are not available for the predicted probabilities of a particular category in the models estimated by the vglm() function. Currently the confidence intervals can only be obtained for the cumulative probabilities. The output is assigned to an object named margins.e2.ciNA.

```
> margins.e2.ciNA <- ggpredict(model2, terms = "educ[12, 14, 16]", ci = NA)
> margins.e2.ciNA

# Predicted probabilities of healthre

# Response Level = 1

educ    |    Predicted
---------------------
  12    |         0.07
  14    |         0.05
  16    |         0.04

# Response Level = 2

educ    |    Predicted
---------------------
  12    |         0.27
  14    |         0.22
  16    |         0.17

# Response Level = 3

educ    |    Predicted
---------------------
  12    |         0.49
  14    |         0.51
  16    |         0.51

# Response Level = 4

educ    |    Predicted
---------------------
  12    |         0.16
  14    |         0.21
  16    |         0.28

Adjusted for:
* maritals = 0.44
*   female = 0.56
```

FIGURE 4.2 ● Estimated Probabilities of Being in Categories 1, 2, 3, and 4 for educ



When educ equals 12, 14, and 16, and other predictor variables are held at their means, the predicted probabilities of being in each category (i.e., $Y = 1, 2, 3,$ and 4) are displayed in the output. The last section titled "Adjusted for" lists the means of the other two variables.

The predicted probabilities for all four response levels are plotted using the plot (margins.e2.ciNA) function. Figure 4.2 shows the estimated probabilities of being in each category (i.e., $Y = 1, 2, 3,$ and 4) for educ at 12, 14, and 16.

The graph shows that people with higher levels of education are less likely associated with poor and fair health conditions (categories 1 and 2). In addition, with the increase in the years of education, the probabilities of being in good and excellent health conditions (categories 3 and 4) increase.

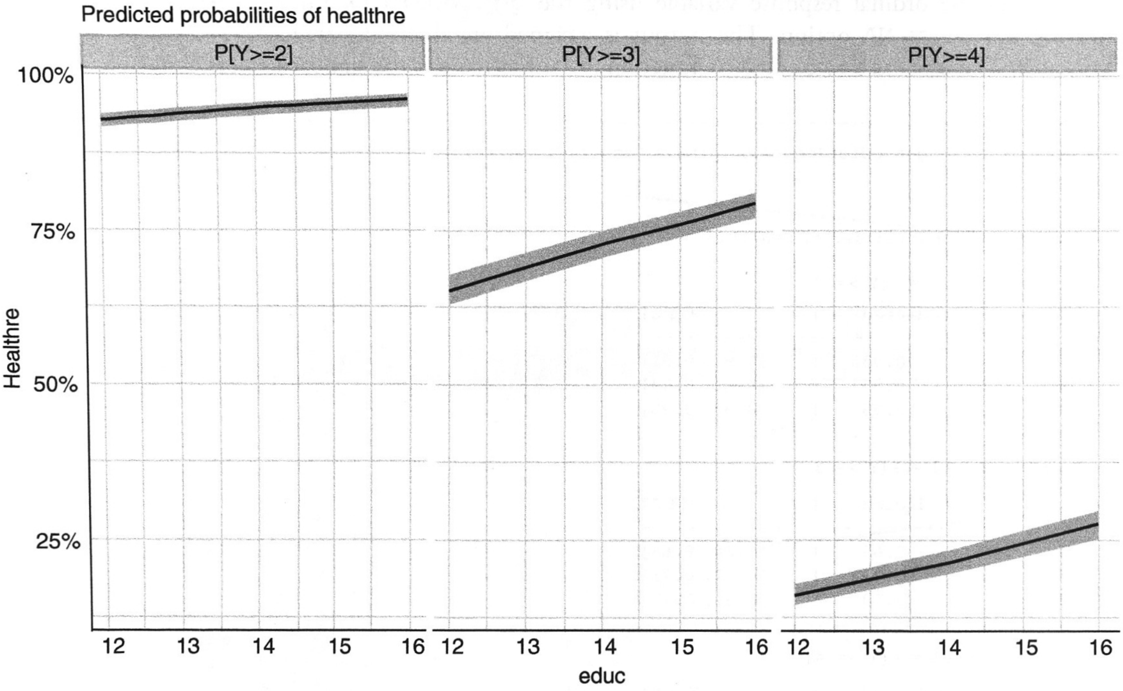
4.6.5 Computing the Cumulative Probabilities With the ggpredict() Function

We can also compute the cumulative probabilities of being at or above a particular category of the ordinal response variable at specified values of the predictor variables.

The command `margins.e2 <- ggpredict(model2, terms = "educ [12, 14, 16]")` tells R to compute the cumulative probabilities of being at or above a category of the ordinal response variable using the `ggpredict()` function by removing the `ci = NA` option. The output is assigned an objected named `margins.e2`. The `as.data.frame()` function is used to request the standard errors.

```
> margins.e2 <- ggpredict(model2, terms = "educ[12, 14, 16]")
> margins.e2

# Predicted probabilities of healthre

# Response Level = P[Y >= 2]
educ    |   Predicted    |          95% CI
-----------------------------------------
  12    |        0.93    |    [0.94, 0.91]
  14    |        0.95    |    [0.96, 0.94]
  16    |        0.96    |    [0.97, 0.95]

# Response Level = P[Y >= 3]
educ    |   Predicted    |          95% CI
-----------------------------------------
  12    |        0.65    |    [0.68, 0.63]
  14    |        0.73    |    [0.75, 0.71]
  16    |        0.79    |    [0.81, 0.77]

# Response Level = P[Y >= 4]
educ    |   Predicted    |          95% CI
-----------------------------------------
  12    |        0.16    |    [0.18, 0.14]
  14    |        0.21    |    [0.23, 0.19]
  16    |        0.28    |    [0.30, 0.25]

Adjusted for:
* maritals = 0.44
*   female = 0.56

> as.data.frame(margins.e2)
     x   predicted   std.error    conf.low   conf.high   response.level  group
1   12   0.9254887  0.09662647   0.9375413   0.9113303        P[Y >= 2]      1
2   12   0.6526333  0.05581345   0.6770017   0.6274382        P[Y >= 3]      1
3   12   0.1598689  0.06749109   0.1784443   0.1428909        P[Y >= 4]      1
4   14   0.9463138  0.09773966   0.9552544   0.9357070        P[Y >= 2]      1
5   14   0.7272433  0.05309915   0.7473924   0.7061189        P[Y >= 3]      1
6   14   0.2126279  0.05739799   0.2320700   0.1944022        P[Y >= 4]      1
7   16   0.9615602  0.10759248   0.9686394   0.9529605        P[Y >= 2]      1
8   16   0.7909608  0.06580753   0.8114879   0.7688353        P[Y >= 3]      1
9   16   0.2770562  0.06197374   0.3020310   0.2533968        P[Y >= 4]      1

> plot(margins.e2)
```

FIGURE 4.3 ● Cumulative Probabilities of Being at or Above Categories 2, 3, and 4 for educ



The output provides the three cumulative probabilities with the confidence intervals for `educ` at 12, 14, and 16, while other predictor variables are held at their means. Please note that the standard errors are on the logit-link scale and are not transformed back to the probabilities. The results are plotted by using the `plot(margins.e2)` function. Figure 4.3 shows the cumulative probabilities of being at or above categories 2, 3, and 4 for `educ`.

With the increase in the years of education, people are more likely to be in better health conditions.

4.6.6 Using the `lrtest()` Function to Test the PO Assumption

The `lrtest()` function is used to test the PO assumption. We fit a cumulative odds model, `model2c`, with the `parallel = FALSE` option and then compare it with the PO model, `model2`, with the `parallel = TRUE` option. A nonsignificant test indicates that the proportional odds assumption is upheld for the PO model. The results of the `lrtest(model2, model2c)` command are shown as follows.

```
> # PO assumption test
> model2c <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = FALSE,
reverse = FALSE),data = chp4.po)
> lrtest(model2, model2c)
Likelihood ratio test

Model 1: healthre ~ educ + maritals + female
Model 2: healthre ~ educ + maritals + female
     #Df    LogLik    Df    Chisq    Pr(>Chisq)
1   5613   -2160.5
2   5607   -2155.7    -6   9.6479        0.1403
```

The likelihood ratio test yields $\chi^2_{(6)} = 9.648$, $p = .140$, which indicates that the proportional odds assumption for the overall model is met.

4.6.7 Model Comparison Using the Likelihood Ratio Test With the `lrtest()` Function

Since the `anova()` function for model comparisons does not work with the `vglm()` function, the `lrtest()` function is used. The `lrtest(model1, model2)` command compares the simple-predictor PO model and the multiple-predictor PO model using the likelihood ratio test. The resulting output is as follows.

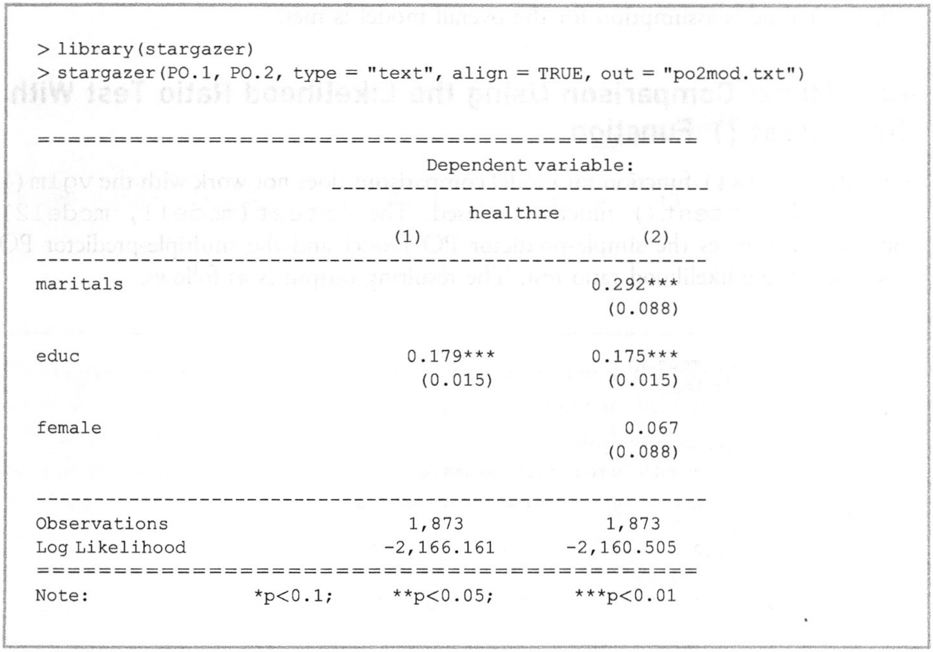
```
> lrtest(model1, model2)
Likelihood ratio test

Model 1: healthre ~ educ
Model 2: healthre ~ educ + maritals + female
     #Df    LogLik    Df    Chisq    Pr(>Chisq)
1   5615   -2166.2
2   5613   -2160.5    -2   11.313      0.003496  **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The difference in deviance, $G = D_{\text{Reduced}} - D_{\text{Full}} = 2 \times (2{,}166.2 - 2{,}160.5) = 11.4$. The likelihood ratio test $\chi^2_{(2)} = 11.313$, $p < .001$. This result indicates that the full model has a better fit than the one-predictor model.

4.7 MAKING PUBLICATION-QUALITY TABLES

4.7.1 Presenting the Results of the clm Models Using the stargazer Package

We can use the stargazer package (Hlavac, 2018) to make a table containing the results of the fitted models with the clm() function. Since the package has been installed in earlier chapters, we only need to load the package by typing library(stargazer). After fitting the single-predictor model PO.1 and the multiple-predictor model PO.2, we load the stargazer package and then use the command as follows: stargazer(PO.1, PO.2, type = "text", align = TRUE, out = "po2mod.txt"). In the stargazer() function, we first specify the two model objects to be presented and then the type of table. The option type = "text" specifies the table type and the align = TRUE option aligns the results of the two models. The out = "po2mod.txt" argument saves the output named po2mod.txt.



```
> library(stargazer)
> stargazer(PO.1, PO.2, type = "text", align = TRUE, out = "po2mod.txt")

==========================================================
                          Dependent variable:
                 -----------------------------------------
                              healthre
                         (1)                (2)
----------------------------------------------------------
maritals                                 0.292***
                                         (0.088)

educ                   0.179***          0.175***
                       (0.015)           (0.015)

female                                    0.067
                                         (0.088)

----------------------------------------------------------
Observations            1,873             1,873
Log Likelihood       -2,166.161        -2,160.505
==========================================================
Note:             *p<0.1;     **p<0.05;       ***p<0.01
```

We can also create the table in the HTML format and copy it into Microsoft Word. The command is as follows: stargazer(PO.1, PO.2, type = "html", align = TRUE, out = "po2mod.htm"). The resulting table is omitted here.

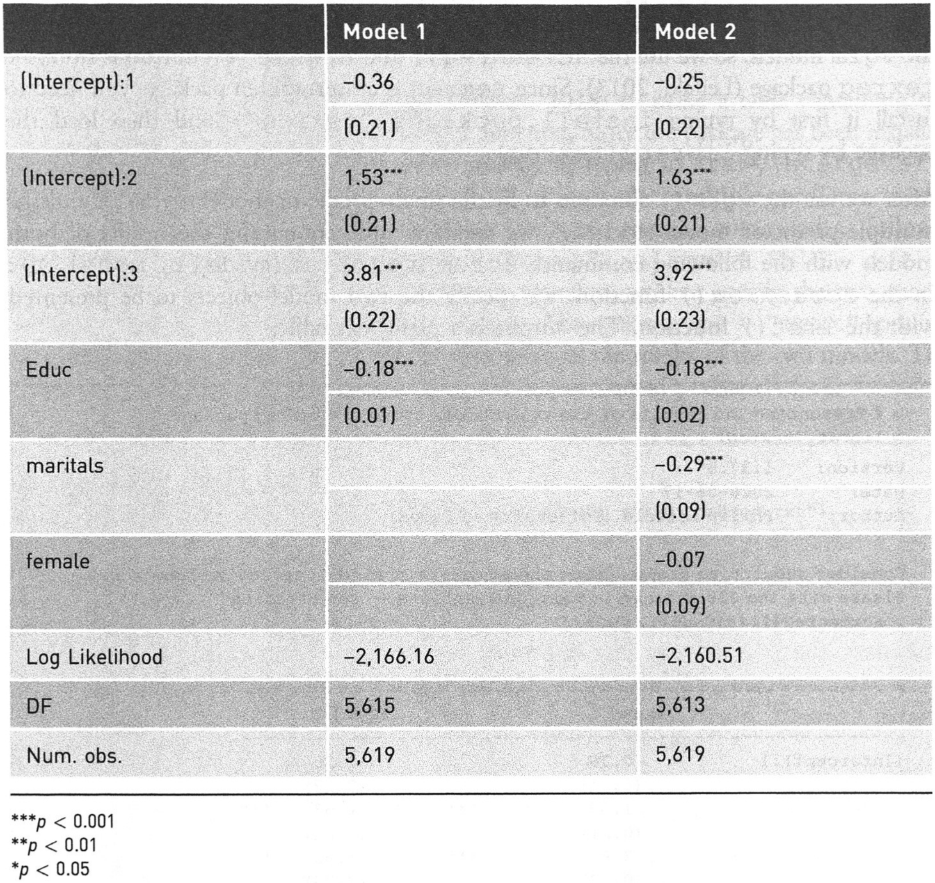
4.7.2 Presenting the Results of the vglm Models Using the texreg Package

The stargazer() function currently cannot directly produce the results table from the vglm models, so we use the screenreg() and htmlreg() functions from the texreg package (Leifeld, 2013). Since texreg is a user-written package, you need to install it first by typing install.packages("texreg") and then load the package by typing library(texreg).

After we use the vglm() function to fit the single-predictor model model1 and the multiple-predictor model model2, we create a table containing the results of both models with the following command: screenreg(list(model1, model2)). In the screenreg() function, we specify the two model objects to be presented with the list() function. The output is a plain text table.

```
> # Presenting the results of the vglm Models using the texreg package
> library(texreg)
Version:     1.37.5
Date:        2020-06-17
Author:      Philip Leifeld (University of Essex)

Consider submitting praise using the praise or praise_interactive functions.
Please cite the JSS article in your publications -- see citation("texreg").
> screenreg(list(model1, model2))

=============================================
                   Model 1           Model 2
---------------------------------------------
(Intercept):1       -0.36             -0.25
                    (0.21)            (0.22)
(Intercept):2        1.53      ***     1.63    ***
                    (0.21)            (0.21)
(Intercept):3        3.81      ***     3.92    ***
                    (0.22)            (0.23)
educ                -0.18      ***    -0.18    ***
                    (0.01)            (0.02)
maritals                              -0.29    ***
                                      (0.09)
female                                -0.07
                                      (0.09)
---------------------------------------------
Log Likelihood   -2166.16          -2160.51
DF                5615              5613
Num. obs.         5619              5619
=============================================
*** p < 0.001;    ** p < 0.01;    * p < 0.05

> htmlreg(list(model1, model2), file="chap4po.doc", doctype=TRUE, html.tag=TRUE,
head.tag=TRUE)
The table was written to the file 'chap4po.doc'.
```

TABLE 4.2 ● Results of the Proportional Odds Models: Single-Predictor Model and Multiple-Predictor PO Model (Shown in Original Format Generated by R)



	Model 1	Model 2
(Intercept):1	−0.36	−0.25
	(0.21)	(0.22)
(Intercept):2	1.53***	1.63***
	(0.21)	(0.21)
(Intercept):3	3.81***	3.92***
	(0.22)	(0.23)
Educ	−0.18***	−0.18***
	(0.01)	(0.02)
maritals		−0.29***
		(0.09)
female		−0.07
		(0.09)
Log Likelihood	−2,166.16	−2,160.51
DF	5,615	5,613
Num. obs.	5,619	5,619

***$p < 0.001$
**$p < 0.01$
*$p < 0.05$

We can also use the `htmlreg()` function to create a regression table for the estimated results and save it to a Microsoft Word file named `chap4po.doc` with the following command: `htmlreg(list(model1, model2), file = "chap4po.doc", doctype = TRUE, html.tag = TRUE, head.tag = TRUE)`. It automatically produces Table 4.2, as shown here in its original format, presenting the results of both the single-predictor and the multiple-predictor PO models.

4.8 REPORTING THE RESULTS

Writing the results of ordinal logistic regression models is similar to that of binary logistic regression models.

First, describe the statistical method you used for data analysis, the dependent variable and the independent variables in the models, and your research hypothesis, or the purpose of your study.

Second, report the model fit statistics, including but not limited to the likelihood ratio statistic and the associated p value, and the pseudo R^2, followed by a concise statement of interpretation on whether the fitted model is better than the null model. If more fit statistics, such as various pseudo R^2 values, deviance statistic, and AIC and BIC statistics are computed, then include them in a table.

Third, report the parameter estimates for the predictor variables, their standard errors, the associated p values, and odds ratios either in a table or in the text. A table is preferable for models with multiple predictors. The odds ratios for each predictor should be interpreted.

If more than one model is fitted, then the results of all the competing models from the simple model to the full model should be presented in a table. The following is an example of summarizing the results from the ordinal logistic regression model.

The proportional odds model was fitted to estimate the ordinal outcome variable, health status, from a set of predictor variables, such as marital status, years of education, and gender. A single-predictor model with marital status as the predictor was fitted first, and then the full model with all the predictors was fitted. The likelihood ratio test is used to compare the two models, $\chi^2_{(2)} = 11.313$, $p < .001$. The result indicated that the full model fitted data better than the single-predictor model.

For the `maritals` predictor, OR = 1.339, which was greater than 1. This indicated that the odds of being above a particular category of health status (better health status) for the married were 1.339 times the odds for the unmarried when holding all the other predictors constant.

For the `educ` predictor, OR = 1.191, which was greater than 1. This indicated that the odds of being above a particular category of health status (better health status) increased by a factor of 1.191 for a one-unit increase in the predictor, education, when holding other predictors constant. In other words, for a one-unit increase in education, the odds of being healthier increased by 19.1%.

For `female`, $\beta = .067$, $p = .444$, which was not significantly different from 0; OR = 1.069, which almost equaled 1. This indicated that there was no relationship between being a female and the cumulative odds of being in better health status.

4.9 SUMMARY OF R COMMANDS IN THIS CHAPTER

```
# Chap 4 R Script

# Remove all objects
rm(list = ls(all = TRUE))

# The following user-written packages need to be installed first by using
install.packages(“ ”) and then by loading it with library()
# library(ordinal)
# library(rcompanion)                  # It is already installed for Chapter 3
# library(ggeffects)                   # It is already installed for Chapter 2
# library(stargazer)                   # It is already installed for Chapter 2
# library(VGAM)
# library(texreg)

# Import GSS 2016 Stata data file
chp4.po <- read.dta("C:/CDA/gss2016.dta")
chp4.po$healthre <- factor(chp4.po$healthre, ordered=TRUE)
chp4.po$educ <- as.numeric(chp4.po$educ)
chp4.po$wrkfull <- as.numeric(chp4.po$wrkfull)
chp4.po$maritals <- as.numeric(chp4.po$maritals)
attach(chp4.po)
str(healthre)

# One-predictor model with the clm() function in ordinal
library(ordinal)
PO.1 <- clm(healthre ~ educ, data = chp4.po)
summary(PO.1)
coef(PO.1)
confint(PO.1)
exp(coef(PO.1))
exp(confint(PO.1))

# Null model with the intercept only
PO.0 <- clm(healthre ~ 1, data = chp4.po)
summary(PO.0)

# Testing the overall model using the likelihood ratio test
anova(PO.0, PO.1)

# Pseudo R2
library(rcompanion)
nagelkerke(PO.1)
LLM <- logLik(PO.1)
LL0 <- logLik(PO.0)
McFadden <- 1-(LLM/LL0)
```

```
McFadden
CS <- 1-exp(2*(LL0-LLM)/1873)
CS
NG <- CS/(1-exp(2*LL0/1873))
NG


# PO assumption test
nominal_test(PO.1)


# Multiple-predictor model with the clm() function
PO.2 <- clm(healthre ~ maritals + educ + female, data = chp4.po)
summary(PO.2)
coef(PO.2)
confint(PO.2)
exp(coef(PO.2))
exp(confint(PO.2))


exp(-coef(PO.2))
exp(-confint(PO.2))


# Predicted probabilities with ggpredict() in ggeffects
library(ggeffects)
margins.e <- ggpredict(PO.2, terms = "educ[12, 14, 16]")
margins.e
plot(margins.e)


# Predicted probabilities with predict(): Omitted in the chapter
New <- data.frame(educ=c(12,14,16),
                  maritals=rep(mean(maritals), 3),
                  female=rep(mean(female), 3))
new
new[,c(’pred.prob’)] <- predict(PO.2, newdata=new, type="prob", se.fit=TRUE,
interval=TRUE)
new


# Testing the overall model using the likelihood ratio test
anova(PO.0, PO.2)


# Pseudo R2
nagelkerke(PO.2)


# PO assumption test
nominal_test(PO.2)


# Model comparison using the likelihood ratio test
anova(PO.1, PO.2)


# Presenting the results of the clm models using the stargazer package
library(stargazer)
stargazer(PO.1, PO.2, type="text", align=TRUE, out="po2mod.txt")
stargazer(PO.1, PO.2, type="html", align=TRUE, out="po2mod.htm")


# One-predictor model with the vglm() function in VGAM
library(VGAM)
```

```
model1 <- vglm(healthre ~ educ, cumulative(parallel = TRUE, reverse = FALSE),
data = chp4.po)
summary(model1)
exp(coef(model1, matrix = TRUE))
exp(confint(model1, matrix = TRUE))
cbind(exp(coef(model1)), exp(confint(model1)))
nagelkerke(model1)
AIC(model1)


# Logit coefficients of being at or above a category
model1b <- vglm(healthre ~ educ, cumulative(parallel = TRUE, reverse = TRUE),
data = chp4.po)
summary(model1b)
exp(coef(model1b, matrix = TRUE))
exp(confint(model1b, matrix = TRUE))
cbind(exp(coef(model1b)), exp(confint(model1b)))


# Multiple-predictor model with the vglm() function in VGAM
model2 <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = TRUE,
reverse = FALSE),data = chp4.po)
summary(model2)
coef(model2, matrix = TRUE)
confint(model2, matrix = TRUE)
exp(coef(model2, matrix = TRUE))
exp(confint(model2, matrix = TRUE))
cbind(exp(coef(model2)), exp(confint(model2)))
AIC(model2)
# nagelkerke(model2)


# Predicted probabilities with predict()
new1 <- data.frame(educ=c(12,14,16),
                   maritals=rep(mean(maritals), 3),
                   female=rep(mean(female), 3))
new1
new1[,c('pred.prob')] <- predict(model2, newdata=new1, type="response")
new1


# Predicted probabilities with ggpredict() in ggeffects
library(ggeffects)
margins.e2.ciNA <- ggpredict(model2, terms="educ[12, 14, 16]", ci=NA)
margins.e2.ciNA
plot(margins.e2.ciNA)


margins.e2 <- ggpredict(model2, terms="educ[12, 14, 16]")
margins.e2
as.data.frame(margins.e2)
plot(margins.e2)


# Logit coefficients of being at or above a category with reverse = TRUE
model2b <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = TRUE,
reverse = TRUE), data = chp4.po)
summary(model2b)
coef(model2b, matrix = TRUE)
confint(model2b, matrix = TRUE)
exp(coef(model2b, matrix = TRUE))
exp(confint(model2b, matrix = TRUE))
cbind(exp(coef(model2b)), exp(confint(model2b)))
# AIC(model2b)
# Testing the Overall Model Using the Likelihood Ratio Test: Omitted in the chapter
model0 <- vglm(healthre ~ 1, cumulative(parallel = TRUE, reverse = FALSE), data =
chp4.po)
summary(model0)
lrtest(model0, model1)
lrtest(model0, model2)

# PO assumption test
model2c <- vglm(healthre ~ educ + maritals + female, cumulative(parallel = FALSE,
reverse = FALSE),data = chp4.po)
lrtest(model2, model2c)

# Model comparison with the likelihood ratio test
lrtest(model1, model2)

# Presenting the results of the vglm Models using the texreg package
library(texreg)
screenreg(list(model1, model2))
htmlreg(list(model1, model2), file="chap4po.doc", doctype=TRUE, html.tag=TRUE,
head.tag=TRUE)

detach(chp4.po)
```

Glossary

An ordinal probit regression model is a regression model for an ordinal response variable with the probit link.

Ordinal logistic regression models are regression models for ordinal response variables with the logistic function or the logit link.

The cumulative probability of being at or below a category $P(Y \leq j)$ equals the sum of the probabilities of all categories at or below that category.

The odds of being at or below a category in ordinal logistic regression equals the probability of being at or below a category divided by the probability of being above that category.

The proportional odds (PO) model is one of the most commonly used models for the analysis of ordinal response variables. The odds ratio of any predictor is assumed to be constant across all categories, so it is referred to as the proportional odds assumption or the parallel lines assumption.

Exercises

Use the GSS 2016 data available at **https://edge.sagepub.com/liu1e** for the following problems.

1. Conduct an analysis for a proportional odds model to estimate the ordinal response variable `fechld` from the three predictor variables `sex`, `educ`, and `age`.
2. Identify the likelihood ratio test of the model and interpret it.
3. Compute the deviance statistic for the model.
4. List three measures of pseudo R^2 and the AIC statistic.
5. Identify the logit coefficient, the Wald z test, and the 95% confidence interval for the predictor variables `sex` and `educ`. Are they statistically significant?
6. Compute the odds ratios for `sex` and `educ`.
7. Test the proportional odds assumption and interpret the results.
8. What are the important criteria you may use for model comparisons?
9. Make a publication-quality table containing the estimated logit coefficients.
10. Write a report to summarize the results from the output.