Journal Title: Categorical data analysis and multilevel modeling using R , Xing Liu

Article Title: Chapter 5. Partial Proportional Odds Models and Generalized Ordinal Logistic Regression Models
Article Author: Xing Liu

Volume/Issue: /
Date: 2023
Pages: 189- 227
NRC TN:1345925

12/14/2023 9:31:08 AM

RAPID

Borrower: RAPID:PUL
Library Name: RAPID:PUL
Email:
Odyssey: 128.112.202.139

5

GENERALIZED ORDINAL LOGISTIC REGRESSION MODELS AND PARTIAL PROPORTIONAL ODDS MODELS

OBJECTIVES OF THIS CHAPTER

This chapter presents partial proportional odds models and generalized ordinal logistic regression models when the proportional odds assumption is untenable. It begins with an introduction to the models, odds and odds ratio, model fit statistics, and interpretations of parameter estimates. After a description of the data, two examples on how to fit these two models using R are illustrated. The `vglm()` function in the `VGAM` package is used to fit the models. R functions are explained, and output is interpreted in detail. This chapter focuses on fitting partial proportional odds models and generalized ordinal logistic regression models using R and on interpreting and presenting the results. After reading this chapter, you should be able to:

- Identify when a partial proportional odds model or a generalized ordinal logistic regression model is used.
- Conduct analysis for both models using R.
- Interpret the output.
- Interpret the model in terms of odds ratios.
- Compute the estimated 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.

5.1 PARTIAL PROPORTIONAL ODDS MODELS AND GENERALIZED ORDINAL LOGISTIC REGRESSION MODELS: AN INTRODUCTION

In Chapter 4, we discussed the proportional odds (PO) model, which is used to estimate the cumulative probability of being at or below a particular level of the ordinal response variable or the complementary probability of being above that particular level. This model follows the assumption that the coefficient of each predictor is the same across the categories of the ordinal response variable. In other words, for each predictor, its effect on the log odds of being at or below any category remains the same. This restriction is referred to as the proportional odds assumption or the parallel lines assumption.

The PO assumption is strict and is often violated in real data analysis. An alternative solution is to fit the partial proportional odds (PPO) model or the generalized ordinal logit model (Fu, 1998; Fullerton & Xu, 2016; Liu, 2016a; Liu & Koirala, 2012; Peterson & Harrell, 1990; Williams, 2006). In the PPO model, not all predictor variables violate the PO assumption, so the effects of those predictors violating the assumption are allowed to vary across categories. The generalized ordinal logit model can be considered an extreme case of the PPO model, and it allows the effect of each explanatory variable to vary. The original PPO model proposed by Peterson and Harrell (1990) specifies an interaction between a predictor variable that violates the PO assumption and different categories of the ordinal outcome variable and requires data restructuring. Fu (1998) and William (2006) developed Stata add-on programs which made the analysis of generalized ordinal logit models and PPO models easier without data restructuring.

The generalized ordinal logistic regression model is an extension of the PO model by relaxing the PO assumption for all predictor variables. This model in the `vglm()` function from the `VGAM` package is expressed as follows:

$$\ln\left(\frac{\pi_j(x)}{1-\pi_j(x)}\right) = \alpha_j + \left(\beta_{1j}X_1 + \beta_{2j}X_2 + \cdots + \beta_{pj}X_p\right) \tag{5.1}$$

To estimate the odds of being at or below a certain category relative to being above that category, Equation 5.1 also can be expressed as follows:

$$\begin{aligned} \text{logit}\left[P\left(Y \leq j \middle| x_1, x_2, \ldots, x_p\right)\right] &= \ln\left(\frac{P\left(Y \leq j \middle| x_1, x_2, \ldots, x_p\right)}{P\left(Y > j \middle| x_1, x_2, \ldots, x_p\right)}\right) \\ &= \alpha_j + \left(\beta_{1j}X_1 + \beta_{2j}X_2 + \cdots + \beta_{pj}X_p\right) \end{aligned} \tag{5.2}$$

where in both equations α_j are the intercepts or cut points and $\beta_{1j}, \beta_{2j}, \ldots, \beta_{pj}$ are the logit coefficients. This model estimates the odds of being at or below a certain category relative to being above that category. A positive logit coefficient normally indicates that an individual is more likely to be in a lower category rather than in a higher category of the outcome variable. To estimate the odds of being above a particular category, however, the signs before both the intercepts and logit coefficients in Equation 5.2 need to be reversed as follows:

$$\begin{aligned}\text{logit}[P(Y>j|x_1,x_2,\ldots,x_p)] &= \ln\left(\frac{P(Y>j|x_1,x_2,\ldots,x_p)}{P(Y\leq j|x_1,x_2,\ldots,x_p)}\right)\\ &= -\alpha_j-\left(\beta_{1j}X_1+\beta_{2j}X_2+\cdots+\beta_{pj}X_p\right)\end{aligned} \tag{5.3}$$

The model in Equation 5.3 estimates the odds of being above a certain category relative to being at or below that category. A positive logit coefficient normally indicates that an individual is more likely to be in a higher category rather than in a lower category of the outcome variable.

Some software uses a modified form estimating the cumulative probability of being at or above a category as follows:

$$\begin{aligned}\text{logit}[P(Y\geq j|x_1,x_2,\ldots,x_p)] &= \ln\left(\frac{P(Y\geq j|x_1,x_2,\ldots,x_p)}{P(Y<j|x_1,x_2,\ldots,x_p)}\right)\\ &= -\alpha_j-\left(\beta_{1j}X_1+\beta_{2j}X_2+\cdots+\beta_{pj}X_p\right)\end{aligned} \tag{5.4}$$

Equations 5.3 and 5.4 are actually equivalent since Equation 5.3 estimates the probabilities of $J-1$ categories from $j=1$ to $J-1$, whereas Equation 5.4 estimates the probabilities of $J-1$ categories from $j=2$ to J. Therefore, $P(Y>1)$ in Equation 5.3 is equal to in $P(Y\geq 2)$ in Equation 5.4.

Compared to the generalized ordinal logistic regression model, the PPO model is more parsimonious since it only allows predictor variables violating the PO assumption to vary. Although the PPO model or the generalized ordinal logistic regression model provides a better fit than a PO model does, the former two models are not a panacea to deal with the violation of the PO assumption. We need to look at the underlying binary logistic models and examine the discrepancy of the coefficients. When the discrepancy is minor, the PO model may still be useful. Williams (2016) explained five possible reasons why the PO assumption was violated and suggested that researchers justify the use of the PPO model or the generalized ordinal logistic regression model.

5.1.1 Odds and Odds Ratios

As with the PO model, the partial proportional odds model and the generalized ordinal logit model estimate the odds of being at or below a certain category versus being above that category. The cumulative odds in partial proportional odds models and generalized ordinal logit models are comparing $P(Y\leq j)$ and $P(Y>j)$. In other words, the odds of being at or below a category are the probability of being at or below a category divided by the probability of being above that category:

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

Odds ($Y \leq 1$) equal the ratio of probability of being at or below category 1 to the probability of being above this category.

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

Similarly, 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.

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

In addition, the odds of being at or below category 3 compare the probabilities of categories 1, 2, and 3 with the probability of category 4.

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

The category comparisons for the odds of being at or below a certain category versus being above that category are also discussed in the previous chapter, so the table is omitted here.

The model in Equation 5.4 also estimates the odds of being at or above a certain category relative to being below that category. The `vglm()` function with the `reverse` = `TRUE` argument in the VGAM package (Yee, 2010) follows this model equation. In this example, since $P(Y \geq 1) = P(1) + P(2) + P(3) + P(4) = 1$, we estimate the odds ($Y \geq 2$), ($Y \geq 3$), and ($Y \geq 4$), respectively.

$$\text{Odds}(Y \geq j) = \frac{P(Y \geq j)}{P(Y < j)}$$

Odds ($Y \geq 2$) equal the ratio of the probability of being at or above category 2 [i.e., $P(Y \geq 2)$] to the probability of being below this category [i.e., $P(Y < 2)$], which also equals the ratio of the probability of being above category 1 [i.e., $P(Y > 1)$] to the probability of being at or below this category [i.e., $P(Y \leq 1)$]. The probability $P(Y \geq 2) = P(2) + P(3) + P(4)$ and the probability $P(Y < 2) = P(Y=1) = P(1)$:

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

Similarly, odds ($Y \geq 3$) = odds ($Y > 2$), which equal the ratio of probability of being at or above category 3 to the probability of being below this category. Since

$P(Y \geq 3) = P(Y > 2) = P(3) + P(4)$ and $P(Y < 3) = P(Y \leq 2) = P(1) + P(2)$, the odds of being at or above category 3, odds $(Y \geq 3)$, can be expressed as follows:

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

Finally, odds $(Y \geq 4)$ or odds $(Y > 3)$ equal the ratio of probability of being above category 3 to the probability of being at or below this category. The equation is as follows:

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

Table 5.1 presents the logits, odds, and category comparisons for the PPO model/generalized ordinal logistic model for the health status with four levels.

Odds Ratios in PPO Models/Generalized Ordinal Logistic Regression Models

Just like the odds ratios in the PO model, the odds ratios of being above a particular category in the PPO model and generalized ordinal logistic regression model are the exponentiated logit coefficients. An odds ratio is the change in the odds of being above a particular category for a one-unit increase from any value of x to the value of $(x + 1)$.

5.1.2 Goodness of Fit

Since partial proportional odds models and generalized ordinal logistic regression models are extensions of proportional odds models, measures of fit statistics for the latter, such as the deviance, likelihood ratio test, and pseudo R^2 measures, can be applied to the former models.

TABLE 5.1 ● Category Comparisons for the Partial Proportional Odds Model/Generalized Ordinal Logistic Model with Four Levels of Health Status ($j = 1, 2, 3, 4$)

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

5.1.3 Interpretation of Model Parameter Estimates

The odds ratio in partial proportional odds models and generalized ordinal logistic regression models can be interpreted in the same way as that in the proportional odds regression. It can be interpreted as the change in the predicted logit or the log odds of being above a particular category relative to being at or below that category for a one-unit increase in the predictor variable.

To estimate the odds of being at or below a particular category, however, the signs before both the intercepts and logit coefficients in Equation 5.2 need to be reversed. Taking the multiplicative inverse of the odds of being above a particular category gives us the odds of being at or below that category.

5.2 RESEARCH EXAMPLE AND DESCRIPTION OF THE DATA AND SAMPLE

Research Problem and Questions: This chapter focuses on the same research problem as that in Chapter 4. We will still investigate the relationships between the ordinal response variable, health status, and the four predictor variables, including the highest education, marital status, gender, and working status. Unlike Chapter 3, however, here the research interest focuses on using the PPO model and the generalized ordinal logistic regression model when the PO assumption is violated. The research question is as follows: Do the four predictor variables predict the ordinal response variable, health status? Specifically, do the four predictor variables predict the cumulative odds and then the cumulative probabilities of being above a particular level of health status when the proportional odds assumption is violated?

Description of the Data and Sample: The data for the following analyses were 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)
- `educ`: the highest education completed
- `maritals`: the recoded variable of marital (marital status) with 1 = currently married and 0 = not currently married
- `female`: recoded variable of sex with 1 = female and 0 = male
- `wrkfull`: working full time or not

5.3 GENERALIZED ORDINAL LOGISTIC REGRESSION MODELS WITH R

5.3.1 The `vglm()` Function in the VGAM Package

Both the `ordinal` package (Christensen, 2019) and the `VGAM` package (Yee, 2010, 2015, 2021) were introduced to fit proportional odds models in the last chapter. They can also be used to fit both the partial proportional odds models and the generalized ordinal logistic models. We focus on the use of the `VGAM` package only in this chapter. You need to install the `VGAM` package first by typing `install.packages("VGAM")` if it is not already installed. Since the package has been installed in the preceding chapter, we only need to load the package by typing `library(VGAM)`. The syntax for partial proportional odds models and generalized ordinal logistic models is the same as that for proportional odds models using `vglm()` except that the different `parallel=` arguments are specified. We specify `parallel = TRUE` for proportional odds models and `parallel = FALSE` for nonproportional odds models or generalized ordinal logistic models. For example, the command `vglm(y ~ x, family = cumulative(parallel = FALSE), data = data1)` tells R to fit a generalized ordinal logistic model predicting the ordinal dependent variable *y* with an independent variable *x*. The `parallel = FALSE` argument tells us the predictor variable does not meet the proportional odds assumption and is allowed to vary across the ordinal categories. To fit a partial proportional odds model, we need to specify variables which violate the proportional odds assumption. For example, the command `vglm(y ~ x1 + x2, family = cumulative(parallel = FALSE ~ x2), data = data1)` tells R to fit a partial proportional odds model predicting the ordinal dependent variable *y* with two independent variables. The `parallel = FALSE ~ x2` argument tells R that the second predictor variable `x2` in the model violates the proportional odds assumption. When there are multiple predictor variables violate the assumption, they are connected by plus (+) symbols following the tilde (~). For more details on how to use this function, type `help(vglm)` and `help(cumulative)` in the command prompt after loading the `VGAM` package.

5.3.2 The Multiple-Predictor PO Model

We first fit a PO model including the four predictor variables with the following command: `mod.po <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel = TRUE, reverse = FALSE), data = chp5.gpo)`. The resulting output is displayed as follows.

```
> # Import the GSS 2016 data
> library(foreign)
> chp5.gpo <- read.dta("C:/CDA/gss2016.dta")
> chp5.gpo$healthre <- factor(chp5.gpo$healthre, ordered=TRUE)
> chp5.gpo$educ <- as.numeric(chp5.gpo$educ)
> chp5.gpo$wrkfull <- as.numeric(chp5.gpo$wrkfull)
> chp5.gpo$maritals <- as.numeric(chp5.gpo$maritals)
> attach(chp5.gpo)

> # PO model with vglm() in VGAM
> library(VGAM)
> mod.po <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
TRUE, reverse = FALSE), data = chp5.gpo)
> summary(mod.po)

Call:
vglm(formula = healthre ~ educ + maritals + female + wrkfull,
  family = cumulative(parallel = TRUE, reverse = FALSE), data = chp5.gpo)

Pearson residuals:
                    Min       1Q     Median      3Q     Max
logitlink(P[Y<=1]) -0.7759  -0.2336  -0.1641  -0.1267  7.038
logitlink(P[Y<=2]) -2.0262  -0.7009  -0.3364   0.4918  3.589
logitlink(P[Y<=3]) -6.1518   0.1530   0.3416   0.6290  1.163

Coefficients:
               Estimate  Std. Error  z value  Pr(>|z|)
(Intercept):1  -0.18286    0.22020   -0.830   0.4063
(Intercept):2   1.72231    0.21374    8.058   7.75e-16 ***
(Intercept):3   4.03153    0.23183   17.390   < 2e-16 ***
educ           -0.16610    0.01515  -10.966   < 2e-16 ***
maritals       -0.26248    0.08876   -2.957   0.0031**
female         -0.12648    0.08854   -1.429   0.1531
wrkfull        -0.41262    0.09002   -4.584   4.57e-06 ***
---
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: 4299.823 on 5612 degrees of freedom

Log-likelihood: -2149.912 on 5612 degrees of freedom

Number of Fisher scoring iterations: 4

No Hauck-Donner effect found in any of the estimates

Exponentiated coefficients:
     educ    maritals      female     wrkfull
0.8469646   0.7691443   0.8811878   0.6619166
```

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

We then fit a generalized proportional odds model, or a nonproportional odds model, mod.gpo, with the `parallel = FALSE` option and test the PO assumption by using the `lrtest()` function. A significant test indicates that the proportional odds assumption is violated. The output of the `lrtest(mod.po, mod.gpo)` command is shown as follows.

```
> # GPO model with vglm() in VGAM
> mod.gpo <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE, reverse = FALSE), data = chp5.gpo)
> # Using the lrtest() function to test the PO assumption
> lrtest(mod.po, mod.gpo)
Likelihood ratio test

Model 1: healthre ~ educ + maritals + female + wrkfull
Model 2: healthre ~ educ + maritals + female + wrkfull
  #Df   LogLik   Df   Chisq    Pr(>Chisq)
1 5612  -2149.9
2 5604  -2136.1  -8   27.617   0.0005528 ***

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

The likelihood ratio test yields $\chi^2_{(8)} = 27.617$, $p < .001$, which indicates that the proportional odds assumption for the overall model is violated.

5.3.4 The Multiple-Predictor Generalized Ordinal Logistic Regression Model

The generalized ordinal logistic regression model relaxes the proportionality assumption by allowing the logits coefficients of all four predictor variables to vary across the ordinal response variable. Similar to a series of underlying binary logistic regression models, where the data are dichotomized across different categories, the effects of the predictor variables estimated by the generalized ordinal logistic regression model can vary freely. After fitting the generalized ordinal logistic regression model named `mod.gpo` above, we can request the results by using the `summary(mod.gpo)` command.

```
> summary(mod.gpo)
Call:
vglm(formula = healthre ~ educ + maritals + female + wrkfull,
     family = cumulative(parallel = FALSE, reverse = FALSE), data = chp5.gpo)

Pearson residuals:
                       Min       1Q    Median       3Q     Max
logitlink(P[Y<=1])  -0.8896  -0.2524  -0.1560  -0.1116  9.009
logitlink(P[Y<=2])  -2.0847  -0.6751  -0.3579   0.4991  3.981
logitlink(P[Y<=3])  -6.0010   0.1532   0.3391   0.6317  1.000
```

```
Coefficients:
                Estimate   Std. Error   z value    Pr(>|z|)
(Intercept):1    0.19647     0.41350      0.475    0.634683
(Intercept):2    1.86004     0.25942      7.170    7.50e-13 ***
(Intercept):3    3.73087     0.30303     12.312     < 2e-16 ***
educ:1          -0.18037     0.03137     -5.749    8.98e-09 ***
educ:2          -0.16361     0.01857     -8.810     < 2e-16 ***
educ:3          -0.16597     0.02013     -8.246     < 2e-16 ***
maritals:1      -0.59747     0.21385     -2.794    0.005208 **
maritals:2      -0.38257     0.10878     -3.517    0.000437 ***
maritals:3      -0.08731     0.11480     -0.761    0.446910
female:1        -0.11250     0.19631     -0.573    0.566580
female:2        -0.20045     0.10708     -1.872     0.061211 .
female:3        -0.05112     0.11537     -0.443    0.657686
wrkfull:1       -0.75615     0.22072     -3.426    0.000613 ***
wrkfull:2       -0.65609     0.11068     -5.928    3.07e-09 ***
wrkfull:3       -0.10049     0.11562     -0.869    0.384772

---
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: 4272.206 on 5604 degrees of freedom

Log-likelihood: -2136.103 on 5604 degrees of freedom

Number of Fisher scoring iterations: 5

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):3'

Exponentiated coefficients:
    educ:1      educ:2      educ:3  maritals:1  maritals:2  maritals:3
 0.8349644   0.8490712   0.8470711   0.5502042   0.6821031   0.9163904
  female:1    female:2    female:3   wrkfull:1   wrkfull:2   wrkfull:3
 0.8935952   0.8183624   0.9501620   0.4694709   0.5188783   0.9043933
```

5.3.5 Interpreting R Output

As with the output for the PO model, the R output for the generalized ordinal logistic regression model also 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. Just like the PO model, the coefficients section displays the parameter estimates, standard errors, *z* statistics, and associated *p* values. Unlike the coefficients section in the output for the PO model, the output produced by the `vglm()` function displays the parameter estimates for the three underlying binary logistic models whose outcome variables are dichotomized from the ordinal response variable.

This generalized ordinal logistic regression model estimates the logit or the log odds of being at or below a particular category ($Y \leq j$). Since the ordinal outcome variable in the model has four levels, three underlying binary logistic regression models are fitted, including the models for `logit(P[Y<=1])`, `logit(P[Y<=2])`, and `logit(P[Y<=3])`. The estimated intercepts and logit coefficients for these three submodels are numbered 1, 2, and 3 in the output. As explained in Chapter 3, `logit(P[Y<=1])` is the log odds of being at or below category 1, which compares the probability of category 1 with the probabilities of categories 2, 3, and 4; `logit(P[Y<=2])` compares the probabilities of categories 1 and 2 with those of categories 3 and 4; and `logit(P[Y<=3])` compares the probabilities of categories 1, 2, and 3 with the probability of category 4.

5.3.6 Logit Coefficients of Being at or Below a Category

The `coef(mod.gpo, matrix = TRUE)` command produces the coefficients table for the three binary logistic regression models estimating `logit(P[Y<=1])`, `logit(P[Y<=2])`, and `logit(P[Y<=3])`.

```
> coef(mod.gpo, matrix = TRUE)
            logit(P[Y<=1])  logit(P[Y<=2])  logit(P[Y<=3])
(Intercept)      0.1964733       1.8600429      3.73086647
educ            -0.1803662      -0.1636122     -0.16597062
maritals        -0.5974657      -0.3825745     -0.08731283
female          -0.1125023      -0.2004501     -0.05112275
wrkfull         -0.7561489      -0.6560860     -0.10049090
```

The logit coefficients of all four variables are different across the three equations/models. For example, the regression coefficients for the first predictor variable, `educ`, are −.180, −.164, and −.166, respectively. A total of 12 logit coefficients are estimated, so it would be tedious to interpret each coefficient.

5.3.7 Odds Ratios of Being at or Below a Category

With `exp(coef(mod.gpo, matrix = TRUE))` and `exp(confint(mod.gpo, matrix = TRUE))`, we can get the odds ratios and the corresponding confidence intervals, respectively. The `cbind(exp(coef(mod.gpo)), exp(confint(mod.gpo)))` command combines the results.

```
> exp(coef(mod.gpo, matrix = TRUE))
            logit(P[Y<=1])  logit(P[Y<=2])  logit(P[Y<=3])
(Intercept)      1.2171028       6.4240125      41.7152375
educ             0.8349644       0.8490712       0.8470711
maritals         0.5502042       0.6821031       0.9163904
female           0.8935952       0.8183624       0.9501620
wrkfull          0.4694709       0.5188783       0.9043933
```

```
> exp(confint(mod.gpo, matrix = TRUE))
                     2.5 %       97.5 %
 (Intercept):1    0.5411958   2.7371596
 (Intercept):2    3.8635822  10.6812627
 (Intercept):3   23.0331168  75.5503937
 educ:1           0.7851678   0.8879192
 educ:2           0.8187227   0.8805446
 educ:3           0.8143041   0.8811566
 maritals:1       0.3618213   0.8366691
 maritals:2       0.5511299   0.8442013
 maritals:3       0.7317522   1.1476172
 female:1         0.6081972   1.3129169
 female:2         0.6634361   1.0094671
 female:3         0.7578662   1.1912498
 wrkfull:1        0.3046025   0.7235757
 wrkfull:2        0.4176918   0.6445774
 wrkfull:3        0.7210082   1.1344217
> cbind(exp(coef(mod.gpo)), exp(confint(mod.gpo)))
                                    2.5 %      97.5 %
 (Intercept):1    1.2171028     0.5411958   2.7371596
 (Intercept):2    6.4240125     3.8635822  10.6812627
 (Intercept):3   41.7152375    23.0331168  75.5503937
 educ:1           0.8349644     0.7851678   0.8879192
 educ:2           0.8490712     0.8187227   0.8805446
 educ:3           0.8470711     0.8143041   0.8811566
 maritals:1       0.5502042     0.3618213   0.8366691
 maritals:2       0.6821031     0.5511299   0.8442013
 maritals:3       0.9163904     0.7317522   1.1476172
 female:1         0.8935952     0.6081972   1.3129169
 female:2         0.8183624     0.6634361   1.0094671
 female:3         0.9501620     0.7578662   1.1912498
 wrkfull:1        0.4694709     0.3046025   0.7235757
 wrkfull:2        0.5188783     0.4176918   0.6445774
 wrkfull:3        0.9043933     0.7210082   1.1344217
```

Just like the logit coefficients, the corresponding odds ratios of all four predictor variables vary across the three binary models. The odds ratios can still be interpreted as the change in the odds of being at or below a category versus beyond that category for a one-unit change in the predictor variable when holding all other predictors constant. For example, the odds ratios for `wrkfull` across the three equations are .469, .519, and .904, respectively.

5.3.8 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 generalized ordinal logistic regression model with the null model using the `lrtest()` function. The command `gpomodel0 <- vglm(healthre ~ 1,` cumulative(parallel = TRUE, reverse = FALSE)) is used to fit the null model. The output is displayed below by the summary(gpomodel0) command.

```
> # Testing the overall model using the likelihood ratio test
> gpomodel0 <- vglm(healthre ~ 1, cumulative(parallel = TRUE, reverse = FALSE))
> summary(gpomodel0)

Call:
vglm(formula = healthre ~ 1, family = cumulative(parallel = TRUE,
   reverse = FALSE))

Pearson residuals:
                     Min       1Q      Median      3Q       Max
logitlink(P[Y<=1]) -0.5249   -0.1844   -0.1844   -0.1386   3.8138
logitlink(P[Y<=2]) -0.7477   -0.7477   -0.2980    0.5477   1.7380
logitlink(P[Y<=3]) -1.8483    0.1667    0.3002    0.6754   0.6754

Coefficients:
              Estimate   Std. Error   z value   Pr(>|z|)
(Intercept):1 -2.69954   0.09510     -28.39     <2e-16 ***
(Intercept):2 -0.89064   0.05087     -17.51     <2e-16 ***
(Intercept):3  1.25964   0.05569      22.62     <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: 4476.435 on 5616 degrees of freedom

Log-likelihood: -2238.218 on 5616 degrees of freedom

Number of Fisher scoring iterations: 1

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):1'
```

The lrtest(gpomodel0, mod.gpo) command compares the log-likelihood statistics of the fitted model mod.gpo and the null model gpomodel0 using the likelihood ratio test.

```
> lrtest(gpomodel0, mod.gpo)
Likelihood ratio test

Model 1: healthre ~ 1
Model 2: healthre ~ educ + maritals + female + wrkfull
```

```
  #Df    LogLik    Df    Chisq    Pr(>Chisq)
1  5616  -2238.2
2  5604  -2136.1   -12   204.23   < 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 generalized ordinal logistic regression model is better than the null model with no independent variables. The likelihood ratio test statistic $LR\ \chi^2_{(12)} = 204.23, p < .001$, which indicated that the overall model with the four predictors was significantly different from zero. Therefore, the generalized ordinal logistic regression model provides a better fit than the null model with no independent variables.

Pseudo R^2

The `nagelkerke(mod.gpo)` command produces the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the generalized ordinal logistic regression model. You need to load `rcompanion` first (Mangiafico, 2021) by typing `library(rcompanion)`.

```
> # PseudoR2
> library(rcompanion)
> nagelkerke(mod.gpo)
$`Models`

Model: "vglm, healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE, reverse = FALSE), chp5.gpo"
Null: "vglm, healthre ~ 1, cumulative(parallel = FALSE, reverse = FALSE), chp5.gpo"

$Pseudo.R.squared.for.model.vs.null
                              Pseudo.R.squared
McFadden                             0.0456232
Cox and Snell (ML)                   0.1033040
Nagelkerke (Cragg and Uhler)         0.1137250

$Likelihood.ratio.test
Df.diff  LogLik.diff  Chisq   p.value
   12      -102.11    204.23  4.3651e-37

$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 .046, Cox and Snell's R^2 is .103, and Nagelkerke's R^2 is .114.

AIC and BIC Statistics

The AIC and BIC statistics of the fitted model can be obtained using `AIC(mod.gpo)` and `BIC(mod.gpo)`, respectively.

```
> AIC(mod.gpo)
[1] 4302.206
> BIC(mod.gpo)
[1] 4385.235
```

5.3.9 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 the ordinal outcome variable. The `summary(mod.gpo2)` command produces the following output.

```
> # Logit coefficients of being at or above a category with reverse = TRUE
> mod.gpo2 <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE, reverse = TRUE), data = chp5.gpo)
> summary(mod.gpo2)

Call:
vglm(formula = healthre ~ educ + maritals + female + wrkfull,
  family = cumulative(parallel = FALSE, reverse = TRUE), data = chp5.gpo)

Pearson residuals:
                        Min       1Q  Median      3Q    Max
logitlink(P[Y>=2])   -9.009   0.1116  0.1560  0.2524 0.8896
logitlink(P[Y>=3])   -3.981  -0.4991  0.3579  0.6751 2.0847
logitlink(P[Y>=4])   -1.000  -0.6317 -0.3391 -0.1532 6.0010

Coefficients:
                 Estimate  Std. Error  z value  Pr(>|z|)
(Intercept):1    -0.19647    0.41350    -0.475  0.634683
(Intercept):2    -1.86004    0.25942    -7.170  7.50e-13 ***
(Intercept):3    -3.73087    0.30303   -12.312   < 2e-16 ***
educ:1            0.18037    0.03137     5.749  8.98e-09 ***
educ:2            0.16361    0.01857     8.810   < 2e-16 ***
```

```
educ:3          0.16597     0.02013     8.246     < 2e-16 ***
maritals:1      0.59747     0.21385     2.794    0.005208 **
maritals:2      0.38257     0.10878     3.517    0.000437 ***
maritals:3      0.08731     0.11480     0.761    0.446910
female:1        0.11250     0.19631     0.573    0.566580
female:2        0.20045     0.10708     1.872    0.061211 .
female:3        0.05112     0.11537     0.443    0.657686
wrkfull:1       0.75615     0.22072     3.426    0.000613 ***
wrkfull:2       0.65609     0.11068     5.928    3.07e-09 ***
wrkfull:3       0.10049     0.11562     0.869    0.384772

---
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: 4272.206 on 5604 degrees of freedom

Log-likelihood: -2136.103 on 5604 degrees of freedom

Number of Fisher scoring iterations: 5

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):3'

Exponentiated coefficients:
  educ:1     educ:2     educ:3  maritals:1  maritals:2  maritals:3
1.197656   1.177758   1.180538   1.817507    1.466054    1.091238
female:1   female:2   female:3   wrkfull:1   wrkfull:2   wrkfull:3
1.119075   1.221953   1.052452   2.130057    1.927234    1.105714
```

In the output, the three binary logistic regression models estimate `logit(P[Y>=2])`, `logit(P[Y>=3])`, and `logit(P[Y>=4])` and are numbered 1, 2, and 3, respectively. The logit coefficients can also be displayed as a matrix with `coef(mod.gpo2, matrix = TRUE)`. The output is omitted here.

5.3.10 Odds Ratios of Being at or Above a Category

The odds ratios of being at or above a category and the corresponding confidence intervals can be obtained by using the following functions.

```
> exp(coef(mod.gpo2, matrix = TRUE))
            logit(P[Y>=2]) logit(P[Y>=3]) logit(P[Y>=4])
(Intercept)      0.8216233      0.1556659     0.02397206
educ             1.1976558      1.1777575     1.18053842
maritals         1.8175069      1.4660541     1.09123800
female           1.1190749      1.2219526     1.05245207
wrkfull          2.1300574      1.9272343     1.10571358

> exp(confint(mod.gpo2, matrix = TRUE))
                   2.5 %      97.5 %
(Intercept):1 0.36534223  1.84776002
(Intercept):2 0.09362189  0.25882716
(Intercept):3 0.01323620  0.04341575
educ:1        1.12622857  1.27361311
educ:2        1.13566076  1.22141471
educ:3        1.13487202  1.22804239
maritals:1    1.19521569  2.76379514
maritals:2    1.18455160  1.81445415
maritals:3    0.87137073  1.36658293
female:1      0.76166281  1.64420343
female:2      0.99062171  1.50730406
female:3      0.83945453  1.31949418
wrkfull:1     1.38202537  3.28296763
wrkfull:2     1.55140405  2.39410997
wrkfull:3     0.88150640  1.38694685

> cbind(exp(coef(mod.gpo2)), exp(confint(mod.gpo2)))
                                 2.5 %     97.5 %
(Intercept):1 0.82162326 0.36534223 1.84776002
(Intercept):2 0.15566595 0.09362189 0.25882716
(Intercept):3 0.02397206 0.01323620 0.04341575
educ:1        1.19765582 1.12622857 1.27361311
educ:2        1.17775751 1.13566076 1.22141471
educ:3        1.18053842 1.13487202 1.22804239
maritals:1    1.81750690 1.19521569 2.76379514
maritals:2    1.46605408 1.18455160 1.81445415
maritals:3    1.09123800 0.87137073 1.36658293
female:1      1.11907489 0.76166281 1.64420343
female:2      1.22195259 0.99062171 1.50730406
female:3      1.05245207 0.83945453 1.31949418
wrkfull:1     2.13005741 1.38202537 3.28296763
wrkfull:2     1.92723427 1.55140405 2.39410997
wrkfull:3     1.10571358 0.88150640 1.38694685
```

In the output, the odd ratios of being at or above a category versus being below that category for all four predictor variables are different across the three binary models. They can be interpreted as the change in the odds of being at or above a particular category for a one-unit increase in the predictor variable when holding all other predictors constant. For example, the odds ratios for `educ` across three equations are 1.198, 1.178, and 1.181, respectively. They can be interpreted as the odds of being at or above a category increase by 19.8%, 17.8%, and 18.1% across three comparisons, respectively, for a one-unit increase in `educ`.

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

We 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 predictor variables. Since the package has been installed in earlier chapters, we only need to load the package by typing library (ggeffects). The command prob.e <- ggpredict(mod.gpo, terms = "educ[12, 14, 16]", ci = NA) tells R to compute the predicted probabilities for each category of the ordinal response variable using the ggpredict() function. In the function, mod.gpo 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 the ci = NA option specifies no confidence intervals. The terms option can specify up to four variables, including the second to fourth grouping variables. Please also note that the confidence intervals can only be obtained for the cumulative probabilities, so the ci = NA option is needed there. The output is assigned to the object named prob.e.

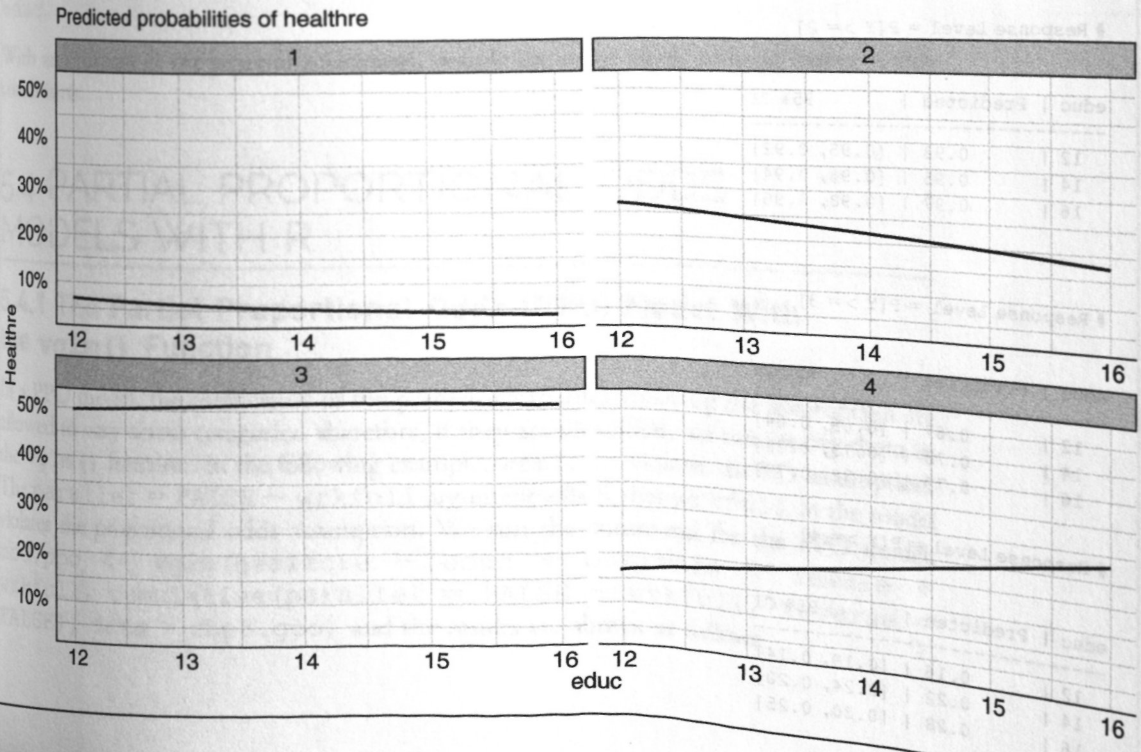
```
> # Predicted probabilities with ggpredict() in ggeffects
> library(ggeffects)
> prob.e <- ggpredict(mod.gpo, terms = "educ[12, 14, 16]", ci = NA)
> prob.e
# Predicted probabilities of healthre

# Response Level = 1

educ | Predicted
-----------------
  12 |      0.07
  14 |      0.05
  16 |      0.03

# Response Level = 2

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

# Response Level = 3

educ | Predicted
-----------------
  12 |      0.50
  14 |      0.52
  16 |      0.52
```

```
# Response Level = 4

educ | Predicted
----------------
  12 |    0.16
  14 |    0.22
  16 |    0.28


Adjusted for:
* maritals = 0.44
*   female = 0.56
*  wrkfull = 0.47

> plot(prob.e)
```

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

The predicted probabilities for all four response levels are plotted using plot (prob.e). Figure 5.1 shows the estimated probabilities of being in each category (i.e., $Y = 1, 2, 3,$ and 4) for educ at 12, 14, and 16.

FIGURE 5.1 ● Predicted Probabilities of Being in Categories 1, 2, 3, and 4 for educ



The graph shows that with an increase in the years of education, the probabilities of being in poor and fair health conditions (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 an 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.

5.3.12 Computing the Predicted Cumulative Probabilities With the `ggpredict()` Function

We can also compute the predicted cumulative probabilities of being at or above a particular category of the ordinal response variable at specified values of predictor variables. The command `cumprob.e <- ggpredict(mod.gpo, terms = "educ [12,14,16]")` tells R to compute the cumulative probabilities of being at or above a category of the ordinal response variable with the `ggpredict()` function by removing the `ci = NA` option. The output is assigned to the object named `cumprob.e`. The `as.data.frame()` function is used to request the standard errors.

```
> # Predicted cumulative probabilities with ggpredict() in ggeffects
> cumprob.e <- ggpredict(mod.gpo, terms = "educ[12, 14, 16]")
> cumprob.e
# Predicted probabilities of healthre

# Response Level = P[Y >= 2]

educ | Predicted |       95% CI
-------------------------------
  12 |      0.93 | [0.95, 0.92]
  14 |      0.95 | [0.96, 0.94]
  16 |      0.97 | [0.98, 0.96]

# Response Level = P[Y >= 3]

educ | Predicted |       95% CI
-------------------------------
  12 |      0.67 | [0.69, 0.64]
  14 |      0.73 | [0.75, 0.71]
  16 |      0.79 | [0.82, 0.77]

# Response Level = P[Y >= 4]

educ | Predicted |       95% CI
-------------------------------
  12 |      0.16 | [0.19, 0.14]
  14 |      0.22 | [0.24, 0.20]
  16 |      0.28 | [0.30, 0.25]
```

```
Adjusted for:
* maritals = 0.44
*   female = 0.56
* wrkfull = 0.47

> as.data.frame(cumprob.e)
   x   predicted   std.error    conf.low   conf.high  response.level  group
1 12   0.9337026  0.10979500   0.9458436   0.9190726       P[Y >= 2]      1
2 12   0.6656569  0.05886990   0.6908273   0.6395067       P[Y >= 3]      1
3 12   0.1644267  0.07658484   0.1861005   0.1448279       P[Y >= 4]      1
4 14   0.9528327  0.12049652   0.9623812   0.9410091       P[Y >= 2]      1
5 14   0.7341598  0.05587161   0.7549774   0.7122475       P[Y >= 3]      1
6 14   0.2152252  0.05808976   0.2350777   0.1966183       P[Y >= 4]      1
7 16   0.9666401  0.15766559   0.9752891   0.9551032       P[Y >= 2]      1
8 16   0.7929920  0.07440783   0.8159120   0.7680297       P[Y >= 3]      1
9 16   0.2765238  0.06422452   0.3024005   0.2520613       P[Y >= 4]      1

# Standard errors are on link-scale (untransformed).
> plot(cumprob.e)
```

The output provides the three cumulative probabilities with the confidence intervals for educ at 12, 14, and 16, and 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 with `plot(cumprob.e)`. Figure 5.2 shows the predicted cumulative probabilities of being at or above categories 2, 3, and 4 for educ.

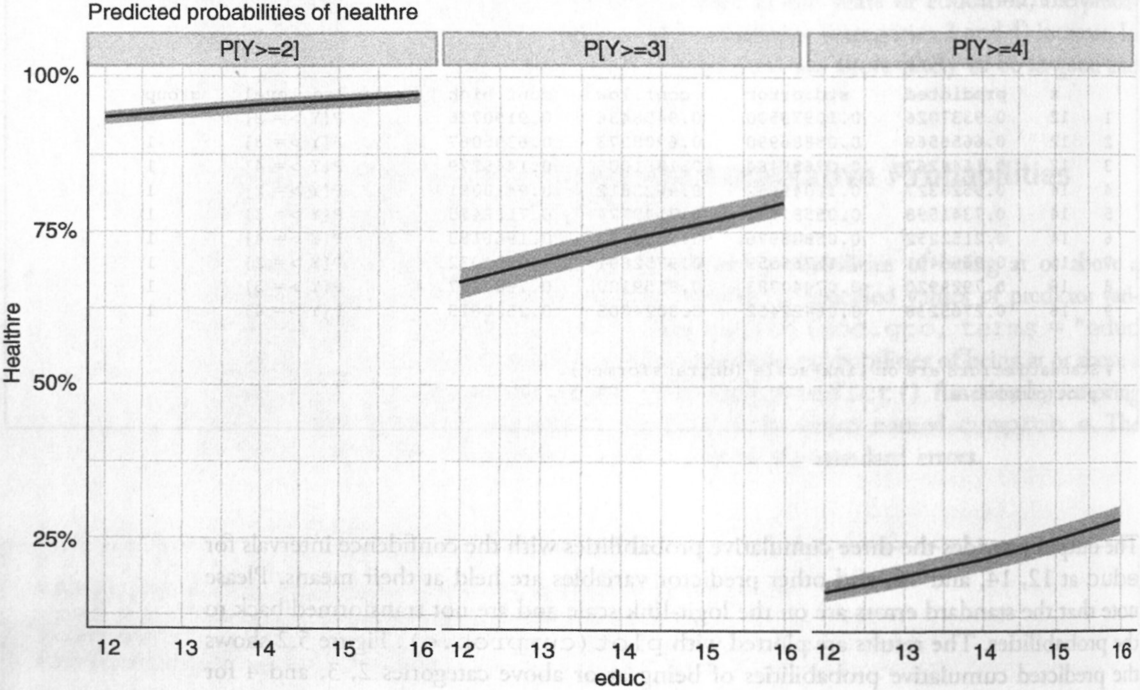
With an increase in the years of education, people are more likely to be in better health conditions.

5.4 PARTIAL PROPORTIONAL ODDS MODELS WITH R

5.4.1 The Partial Proportional Odds (PPO) Model With the `vglm()` Function

In a PPO model, the coefficients of the predictor variables violating the assumption are allowed to vary across categories, therefore, if they are identified, we can specify them in the `vglm()` function. In the following example, `wrkfull` violates the PO assumption. The `parallel = FALSE ~ wrkfull` argument tells R that `wrkfull` in the model violates the proportional odds assumption. We run the command for the PPO model, `mod.ppo <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel = FALSE ~ wrkfull, reverse = FALSE), data = chp5.gpo)` and the results are shown as follows.

FIGURE 5.2 ● Predicted Cumulative Probabilities of Being at or Above Categories 2, 3, and 4 for educ




```
> # PPO model with vglm() in VGAM
> mod.ppo <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE ~ wrkfull, reverse = FALSE), data = chp5.gpo)
> summary(mod.ppo)

Call:
vglm(formula = healthre ~ educ + maritals + female + wrkfull,
  family = cumulative(parallel = FALSE ~ wrkfull, reverse = FALSE),
  data = chp5.gpo)

Pearson residuals:
                        Min      1Q   Median      3Q   Max
logitlink(P[Y<=1]) -0.7817 -0.2403  -0.1562 -0.1189 8.028
logitlink(P[Y<=2]) -2.0978 -0.6728  -0.3658  0.4940 3.839
logitlink(P[Y<=3]) -5.5984  0.1583   0.3406  0.6397 1.077
```

```
Coefficients:
               Estimate  Std. Error  z value   Pr(>|z|)
(Intercept):1  -0.07075   0.22603    -0.313    0.754271
(Intercept):2   1.80233   0.21568     8.357    < 2e-16 ***
(Intercept):3   3.85156   0.23357    16.490    < 2e-16 ***
educ           -0.16541   0.01516   -10.908    < 2e-16 ***
maritals       -0.26358   0.08901    -2.961    0.003064 **
female         -0.12968   0.08881    -1.460    0.144237
wrkfull:1      -0.79925   0.21638    -3.694    0.000221 ***
wrkfull:2      -0.65348   0.10972    -5.956    2.59e-09 ***
wrkfull:3      -0.09649   0.11521    -0.837    0.402327

---
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: 4280.54 on 5610 degrees of freedom

Log-likelihood: -2140.27 on 5610 degrees of freedom

Number of Fisher scoring iterations: 4

No Hauck-Donner effect found in any of the estimates

Exponentiated coefficients:
     educ   maritals     female  wrkfull:1  wrkfull:2  wrkfull:3
0.8475485  0.7682963  0.8783731  0.4496664  0.5202321  0.9080234
```

5.4.2 Interpreting R Output

As with the generalized ordinal logistic regression model, the coefficients section displays the parameter estimates of the predictor variables for the three underlying binary logistic models estimating `logit(P[Y<=1])`, `logit(P[Y<=2])`, and `logit (P[Y<=3])`. Again, model 1 compares category 1 with categories 2, 3 and 4, model 2 compares categories 1 and 2 with categories 3 and 4, and model 3 compares categories 1 through 3 with category 4, respectively.

Let us look at the estimates for the predictor variables that meet the PO assumption and those violating the assumption separately. The coefficients of the former variables are constrained but the coefficients of those latter variables are free to vary across the three binary logistic models. In the output, `educ`, `maritals`, and `female` meet the PO assumption, so the equal-slope or the proportional odds constraints are placed on those variables. Each of these three variables has the same logit coefficient across all three binary models. For `educ`, $\beta = -.165$; for `maritals`, $\beta = -.264$; and for `female`, $\beta = -.130$.

The wrkfull predictor variable is the only one that violates the PO assumption, so its coefficients are allowed to vary across the three binary models. The estimated logit coefficients are $-.799$, $-.653$, and $-.096$ for each respective model. In model 1, the Wald z test for wrkfull $= -3.694$, $p < .001$; in model 2, Wald $z = -5.956$, $p < .001$; and in model 3, Wald $z = -.837$, $p > .05$. The results of the Wald z tests indicate that logit coefficients for wrkfull are significant across the first two models but not the third model.

The coef(mod.ppo, matrix= TRUE) command produces the coefficients table in the matrix form. The output is omitted.

Odds Ratios of Being at or Below a Category

With the exp(coef(mod.ppo, matrix = TRUE)) and the exp(confint (mod.ppo, matrix = TRUE)) commands, we can get the odds ratios and the corresponding confidence intervals, respectively. The results are combined using cbind(exp(coef(mod.ppo)), exp(confint(mod.ppo))). The following output displays the odds ratio and the corresponding confidence intervals.

```
>> exp(coef(mod.ppo, matrix = TRUE))
              logit(P[Y<=1])  logit(P[Y<=2])  logit(P[Y<=3])
(Intercept)        0.9316959       6.0637586      47.0665040
educ               0.8475485       0.8475485       0.8475485
maritals           0.7682963       0.7682963       0.7682963
female             0.8783731       0.8783731       0.8783731
wrkfull            0.4496664       0.5202321       0.9080234

> exp(confint(mod.ppo, matrix = TRUE))
                    2.5 %       97.5 %
(Intercept):1   0.5982474    1.4510005
(Intercept):2   3.9733536    9.2539381
(Intercept):3  29.7782630   74.3917064
educ            0.8227302    0.8731154
maritals        0.6453035    0.9147311
female          0.7380428    1.0453857
wrkfull:1       0.2942466    0.6871782
wrkfull:2       0.4195704    0.6450442
wrkfull:3       0.7244863    1.1380567

> cbind(exp(coef(mod.ppo)), exp(confint(mod.ppo)))
                                2.5 %       97.5 %
(Intercept):1   0.9316959   0.5982474    1.4510005
(Intercept):2   6.0637586   3.9733536    9.2539381
(Intercept):3  47.0665040  29.7782630   74.3917064
educ            0.8475485   0.8227302    0.8731154
maritals        0.7682963   0.6453035    0.9147311
female          0.8783731   0.7380428    1.0453857
wrkfull:1       0.4496664   0.2942466    0.6871782
wrkfull:2       0.5202321   0.4195704    0.6450442
wrkfull:3       0.9080234   0.7244863    1.1380567
```

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

With the `reverse = FALSE` option, the `vglm()` function estimates the odds ratios of being at or below a particular category versus being above that category in the PPO model and the generalized ordinal logistic model. The odds ratios in both models can be interpreted in a similar way as that in the PO model. It can be interpreted as the change in the odds of being at or below a particular category for a one-unit increase in the predictor variable when holding all the other predictors constant.

Unlike the PO model, the partial PO model allows the effects of some of the predictor variables to vary, so we need to interpret the odds ratios for the predictor variables that meet the PO assumption and those violating the assumption separately.

Three predictor variables, `educ`, `maritals`, and `female`, meet the PO assumption in the model. Let us interpret their odds ratios first. For the `educ` predictor variable, $\beta = -.165$, which is negative, so there is a negative relationship between age and the log odds of being at or below a category of health status; OR = .848, 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 .848 for a one-unit increase in the `educ` predictor when holding other variables constant. In other words, a one-unit increase in the number of years of education is associated with a decrease by 15.2% in the odds of being less healthy.

For the `maritals` predictor variable, $\beta = -.264$, and its corresponding OR = .768. This indicates that the odds of being at or below a particular category of health status (poorer health status) versus being beyond that category (better health status) for the married are .768 times the odds for the unmarried when holding the other predictors constant.

For the `female` predictor variable, $\beta = -.130$, $p = .114$, which is not significantly different from 0; OR = .878, which is close to 1. This indicates that there is no relationship between being female and the cumulative odds of being less healthy.

Next, let us interpret the predictor variables that violate the PO assumption. Since only one predictor variable `wrkfull` violates the assumption, in the output its odds ratios are different across the three binary models. The three odds ratios are .450, .520, and .908, respectively. Overall, working full time decreases the odds of being at or below a particular category of health status. In other words, working full time is associated with the odds of being in better health status.

5.4.4 Model Fit Statistics

Testing the Overall Model Using the Likelihood Ratio Test

To test if the overall PPO model is significant, we compare the PPO model with the null model using the `lrtest()` function. The `lrtest(gpomodel0, mod.ppo)` command compares the log-likelihood statistics of the fitted model `mod.ppo` and the null model `gpomodel0` using the likelihood ratio test.

```
> # Testing the overall model using the likelihood ratio test
> lrtest(gpomodel0, mod.ppo)
Likelihood ratio test

Model 1: healthre ~ 1
Model 2: healthre ~ educ + maritals + female + wrkfull
  #Df   LogLik   Df   Chisq   Pr(>Chisq)
1 5616  -2238.2
2 5610  -2140.3  -6   195.9   < 2.2e-16 ***

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

The likelihood ratio test statistic $LR\ \chi^2_{(6)} = 195.9, p < .001$, which indicated that the overall PPO model with the four predictors was significantly different from zero. Therefore, the PPO model provides a better fit than the null model with no independent variables.

Pseudo R^2

The `nagelkerke(mod.ppo)` command produces the three types of pseudo R^2 statistics and the likelihood ratio test statistic for the PPO model. The McFadden R^2 is .044, the Cox and Snell R^2 is .099, and the Nagelkerke R^2 is .109. The output is omitted here.

AIC and BIC Statistics

The AIC and BIC statistics of the fitted model can be obtained using `AIC(mod.ppo)` and `BIC(mod.ppo)`, respectively.

```
> AIC(mod.ppo)
[1] 4298.54
> BIC(mod.ppo)
[1] 4348.358
```

5.4.5 Logit Coefficients of Being at or Above a Category

Just like the generalized ordinal logistic regression model, the PPO model can also estimate the logit coefficients of being at or above a particular category of the ordinal outcome variable with the `reverse = TRUE` option. We fit the model named `mod.ppo2`. The `summary(mod.ppo2)` command produces the following output.

```
> # Logit coefficients of being at or above a category with reverse = TRUE
>mod.ppo2<-vglm(healthre~educ+maritals+female+wrkfull, cumulative(parallel=
FALSE~wrkfull, reverse = TRUE), data = chp5.gpo)
> summary(mod.ppo2)
```

```
Call:
vglm(formula = healthre ~ educ + maritals + female + wrkfull,
  family = cumulative(parallel = FALSE ~ wrkfull, reverse = TRUE),
  data = chp5.gpo)

Pearson residuals:
                     Min       1Q    Median      3Q     Max
logitlink(P[Y>=2])  -8.028   0.1189   0.1562   0.2403  0.7817
logitlink(P[Y>=3])  -3.839  -0.4940   0.3658   0.6728  2.0978
logitlink(P[Y>=4])  -1.077  -0.6397  -0.3406  -0.1583  5.5984

Coefficients:
               Estimate  Std. Error  z value   Pr(>|z|)
(Intercept):1   0.07075    0.22603     0.313   0.754271
(Intercept):2  -1.80233    0.21568    -8.357   < 2e-16 ***
(Intercept):3  -3.85156    0.23357   -16.490   < 2e-16 ***
educ            0.16541    0.01516    10.908   < 2e-16 ***
maritals        0.26358    0.08901     2.961   0.003064 **
female          0.12968    0.08881     1.460   0.144237
wrkfull:1       0.79925    0.21638     3.694   0.000221 ***
wrkfull:2       0.65348    0.10972     5.956   2.59e-09 ***
wrkfull:3       0.09649    0.11521     0.837   0.402327

---
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: 4280.54 on 5610 degrees of freedom

Log-likelihood: -2140.27 on 5610 degrees of freedom

Number of Fisher scoring iterations: 4

No Hauck-Donner effect found in any of the estimates


Exponentiated coefficients:
    educ   maritals     female   wrkfull:1   wrkfull:2   wrkfull:3
1.179873   1.301581   1.138468    2.223871    1.922219    1.101293
```

5.4.6 Interpreting R Output

The coefficients section displays the parameter estimates of predictor variables for the three underlying binary logistic models estimating `logit(P[Y>=2])`, `logit(P[Y>=3])`, and `logit(P[Y>=4])`. Specifically, model 1 compares categories

2, 3, and 4 with category 1, model 2 compares categories 3 and 4 with categories 1 and 2, and model 3 compares category 4 with categories 1 through 3, respectively.

In the coefficients section, the logit coefficients in the mod.ppo2 model are the same as those in the mod.ppo model in magnitude but are reversed in sign since the former model estimates the log odds of being at or above a category, logit$[P(Y \geq j + 1)]$, whereas the latter model estimates the log odds of being at or below a category, logit$[P(Y \leq j)]$.

The three predictor variables, educ, maritals, and female meet the PO assumption, so they have the same logit coefficients across all three binary models. For educ, $\beta = .165$; for maritals, $\beta = .264$; and for female, $\beta = .130$.

The wrkfull predictor variable violates the PO assumption, so its coefficients are allowed to vary across the three binary models. The estimated logit coefficients are .799, .653, and .096 for each respective model. In model 1, the Wald z test for wrkfull = 3.694, $p < .001$; in model 2, Wald $z = 5.956$, $p < .001$; and in model 3, Wald $z = .837$, $p > .05$.

We again use coef(mod.ppo, matrix = TRUE) to obtain the coefficients table in the matrix form and use the exp(coef(mod.ppo2, matrix = TRUE)) and exp(confint(mod.ppo2, matrix = TRUE)) commands to request the odds ratios and the corresponding confidence intervals. The results are combined at the end.

```
> coef(mod.ppo2, matrix = TRUE)
             logit(P[Y>=2])  logit(P[Y>=3])  logit(P[Y>=4])
(Intercept)      0.07074881      -1.8023298     -3.85156158
educ             0.16540723       0.1654072      0.16540723
maritals         0.26357980       0.2635798      0.26357980
female           0.12968378       0.1296838      0.12968378
wrkfull          0.79924931       0.6534802      0.09648514

> exp(coef(mod.ppo2, matrix = TRUE))
             logit(P[Y>=2])  logit(P[Y>=3])  logit(P[Y>=4])
(Intercept)        1.073312       0.1649142      0.02124653
educ               1.179873       1.1798735      1.17987350
maritals           1.301581       1.3015812      1.30158116
female             1.138468       1.1384683      1.13846832
wrkfull            2.223871       1.9222190      1.10129321

> exp(confint(mod.ppo2, matrix = TRUE))
                    2.5 %     97.5 %
(Intercept):1  0.68917966 1.67154930
(Intercept):2  0.10806210 0.25167657
(Intercept):3  0.01344236 0.03358154
educ           1.14532398 1.21546523
maritals       1.09321745 1.54965832
female         0.95658471 1.35493503
wrkfull:1      1.45522669 3.39850942
wrkfull:2      1.55028134 2.38339049
wrkfull:3      0.87869084 1.38028835

> cbind(exp(coef(mod.ppo2)), exp(confint(mod.ppo2)))
                                2.5 %     97.5 %
(Intercept):1  1.07331159  0.68917966 1.67154930
(Intercept):2  0.16491422  0.10806210 0.25167657
```

```
(Intercept):3  0.02124653  0.01344236  0.03358154
educ           1.17987350  1.14532398  1.21546523
maritals       1.30158116  1.09321745  1.54965832
female         1.13846832  0.95658471  1.35493503
wrkfull:1      2.22387087  1.45522669  3.39850942
wrkfull:2      1.92221898  1.55028134  2.38339049
wrkfull:3      1.10129321  0.87869084  1.38028835
```

5.4.7 Interpreting the Odds Ratios of Being at or Above a Particular Category

We are interested in the odds ratios reported in the coefficients table in the second section in the output. First, let us take a look at the odds ratios for the predictor variables that meet the PO assumption. They are 1.180, 1.302, and 1.138, for `educ`, `maritals`, and `female`, respectively.

Second, we look at the predictor variables that violate the PO assumption. Only one predictor variable `wrkfull` violates the assumption. Its odds ratios are different across the three binary models. They are 2.224, 1.922, and 1.101, respectively.

The odds of being at or above a particular category versus being below that category are the multiplicative inverse of the odds of being below that category since $\exp(-\beta) = 1/\exp(\beta)$. The odds ratios in the partial PO model can also be interpreted as the change in the odds of being at or above a particular category for a one-unit increase in the predictor variable when holding all other predictors constant.

Three predictor variables `maritals`, `age`, and `male` meet the PO assumption in the model. For the `educ` predictor, OR = 1.180, which is greater than 1. This indicates that a one-unit increase in education is associated with an increase of 18% in the odds of being healthier.

For `maritals`, OR = 1.302. This indicates that the odds of being at or above a particular category of health status (better health status) versus being below that category (poorer health status) for the married are 1.302 times the odds for the unmarried when holding the other predictors constant.

For `female`, $\beta = .130$, $p = .114$, which is not significantly different from 0; OR = 1.138, which is close to 1. This indicates that there is no change in the odds for being female.

Next, let us interpret the predictor variables that violate the PO assumption. The odds ratios for `wrkfull` are different across the three binary models. The three odds ratios are 2.224, 1.922, and 1.101, respectively. Overall, working full time increases the odds of being at or above a particular category of health status. The odds of being at or above a particular category of health status for working full time are 122.4%, 92.2%, and 10.1% higher than the odds for not working full time in each of the binary logistic models, respectively, when holding other variables constant. The largest odds ratio is identified in the first binary model comparing categories 2, 3, and 4 with category 1, and the smallest odds ratio is found in the third binary model comparing category 4 with categories 1 through 3.

5.4.8 Computing the Predicted Probabilities With the ggpredict() Function for the PPO Model

To compute the predicted probabilities of being in a particular category of the ordinal response variable at specified values of predictor variables in the PPO model, we again use the ggpredict() function in the ggeffects package (Lüdecke, 2018b). The command prob.e2 <- ggpredict(mod.ppo, terms = "educ[12, 14, 16]", ci = NA) tells R to compute the predicted probabilities for each category of the ordinal response variable using the ggpredict() function. In the function, mod.ppo 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 the ci = NA option specifies no confidence intervals. The output is assigned to the object named prob.e2.

```
>> library(ggeffects)
> prob.e2 <- ggpredict(mod.ppo, terms = "educ[12, 14, 16]", ci = NA)
> prob.e2

# 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 |      0.50
  14 |      0.52
  16 |      0.52

# Response Level = 4
educ | Predicted
----------------
  12 |      0.16
  14 |      0.21
  16 |      0.27
```

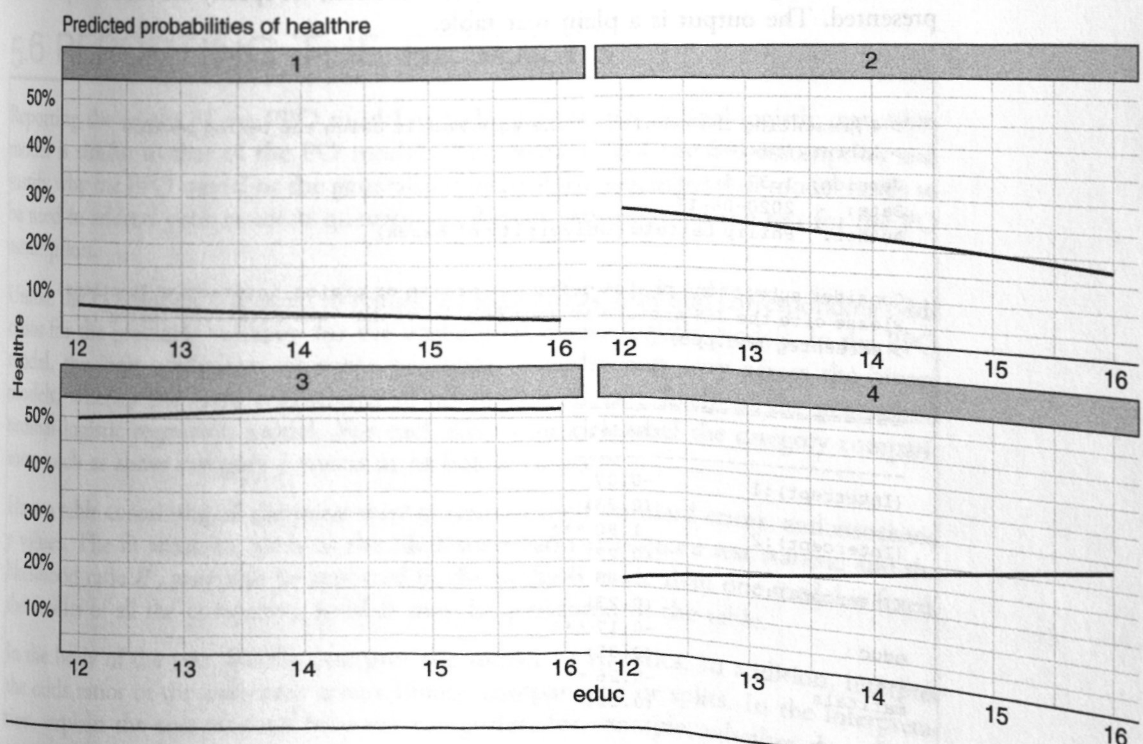
```
Adjusted for:
* maritals = 0.44
*   female = 0.56
* wrkfull = 0.47

> plot(prob.e2)
```

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 under the title "Adjusted for" lists the means of the other three variables.

The predicted probabilities for all four response levels are plotted using plot (prob.e2). Figure 5.3 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 for the PPO model looks similar to the one for the generalized ordinal logistic regression model introduced in the previous section.

FIGURE 5.3 ● Predicted Probabilities of Being in Categories 1, 2, 3, and 4 for educ



5.4.9 Computing the Predicted 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 predictor variables. By removing the ci = NA option, we use the command cumprob.e2 <- ggpredict(mod.ppo, terms = "educ[12, 14, 16]") to compute the predicted cumulative probabilities of being at or above a category. The output is assigned to the object named cumprob.e2. The as.data.frame() function can be used to request the standard errors. The output is omitted here.

5.5 MAKING PUBLICATION-QUALITY TABLES

Since the stargazer() function (Hlavac, 2018) currently cannot directly produce the results table from the vglm models, we use the screenreg() and htmlreg() functions from the texreg package (Leifeld, 2013). You need to install texreg first by typing install.packages("texreg") if you have not done so and then load the package by typing library(texreg). This package needs to be installed only once.

After we use the vglm() function to fit the PPO model mod.ppo, we create a table containing the results of the model with the following command: screenreg(mod.ppo). In the screenreg() function, we specify the model object to be presented. 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(mod.ppo)

=========================
               Model 1
-------------------------
(Intercept):1   -0.07
                (0.23)
(Intercept):2    1.80 ***
                (0.22)
(Intercept):3    3.85 ***
                (0.23)
educ            -0.17 ***
                (0.02)
maritals        -0.26 **
                (0.09)
```

```
female          -0.13
                (0.09)
wrkfull:1       -0.80 ***
                (0.22)
wrkfull:2       -0.65 ***
                (0.11)
wrkfull:3       -0.10
                (0.12)
--------------------------
Log Likelihood  -2140.27
DF               5610
Num. obs.        5619
====================
***p<0.001; **p<0.01; *p<0.05

> htmlreg(list(mod.ppo), file="chap5ppo.doc", doctype=TRUE, html.tag=TRUE,
head.tag=TRUE)
The table was written to the file 'chap5ppo.doc'.
```

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 `chap5ppo.doc` with the following command: `htmlreg(list(mod.ppo), file = "chap5ppo.doc", doctype = TRUE, html.tag = TRUE, head.tag = TRUE)`. It automatically produces Table 5.2, as shown here in its original format, presenting the results of the PPO model.

5.6 REPORTING THE RESULTS

Reporting the results of the PPO model and the generalized ordinal logistic regression model is similar to that of the PO models. You need to test the PO assumption and justify why the PPO model or the generalized ordinal logistic regression model needs to be used to address your research question(s). Report the results of the test for the PO assumption.

Unlike the PO models, you need to report the logit coefficients and corresponding odds ratios for the predictor variables for the underlying binary logistic models. In the PPO model, the logit coefficients of some predictor variables may vary across the binary models, whereas the logit coefficients of all predictors vary freely in the generalized ordinal logistic regression model. For each binary model, label the category comparisons, such as above category *j* versus at or below category *j*.

Have a table containing all the parameter estimates, their standard errors, and associated *p* values. The fit statistics, such as the likelihood ratio chi-square test statistic and the likelihood ratio R^2, may also be reported in the table. If more than one model is fitted, the results of all the competing models may be presented in the table.

In the body of the text, briefly interpret the model fit statistics. In addition, interpret the odds ratios of the estimates across binary comparisons or splits. In the interpretation, explain the comparisons between categories, for example, whether the odds of

TABLE 5.2 ● Results of the Partial Proportional Odds Model (Shown in Original Format Generated by R)

	Model 1
(Intercept):1	−0.07
	(0.23)
(Intercept):2	1.80***
	(0.22)
(Intercept):3	3.85***
	(0.23)
Educ	−0.17***
	(0.02)
maritals	−0.26**
	(0.09)
female	−0.13
	(0.09)
wrkfull:1	−0.80***
	(0.22)
wrkfull:2	−0.65***
	(0.11)
wrkfull:3	−0.10
	(0.12)
Log Likelihood	−2140.27
DF	5610
Num. obs.	5619

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

being above a category versus at or below that category, or the reciprocal odds of being at or below a category. The following is an example of summarizing the results for the partial PO model.

The partial proportional odds model was fitted to estimate the ordinal outcome variable, health status, from a set of predictor variables, such as years of education, marital status, gender, and working status. This model was used as it allows the effects of some predictor variables to vary when the proportional odds assumption (PO) does not hold.

The likelihood ratio statistic $LR\ \chi^2_{(6)} = 195.9, p < .001$, which indicated that the overall PPO model with the four predictors provided a better fit than the null model with no independent variables in predicting the ordinal response variable.

Table 5.2 presents the coefficients and standard errors of the predictor variables. Three predictor variables, `educ`, `maritals`, and `female`, meeting the PO assumption had the same logit regression coefficients across all three binary models. For `educ`, $\beta = -.165, p < .001$; for `maritals`, $\beta = -.264, p < .001$; and for `female`, $\beta = -.130, p > .05$.

Three predictor variables, `edu`, `maritals`, and `female`, meet the PO assumption in the model. For the `educ` predictor, OR = 1.180, which is greater than 1. This indicates that the odds of being at or above a particular category of health status (better health status) increase by a factor of 1.180 for a one-unit increase in the number of years of education when holding other variables constant. In other words, a one-unit increase in education is associated with an increase of 18% in the odds of being healthier.

For `maritals`, OR = 1.302. This indicates that the odds of being at or above a particular category of health status (better health status) versus being below that category (poorer health status) for the married are 1.302 times the odds for the unmarried when holding other predictors constant.

For `female`, $\beta = .130, p = .114$, which is not significantly different from 0; OR = 1.138, which is close to 1. This indicates that there is no change in the odds for being female.

The odds ratios for `wrkfull` are different across the three binary models since this predictor violates the PO assumption. The three odds ratios are 2.224, 1.922, and 1.101, respectively. Overall, working full time increases the odds of being at or above a particular category of health status. The odds of being at or above a particular category of health status for working full time are 122.4%, 92.2%, and 10.1% higher than the odds for not working full time in each of the binary logistic models, respectively, when holding other variables constant. The largest odds ratio is identified in the first binary model comparing categories 2, 3, and 4 with category 1, and the smallest odds ratio is found in the third binary model comparing category 4 with categories 1 through 3.

5.7 SUMMARY OF R COMMANDS IN THIS CHAPTER

```
# Chap 5 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(VGAM)          # It is already installed for Chapter 4
# library(rcompanion)    # It is already installed for Chapter 3
# library(ggeffects)     # It is already installed for Chapter 2
# library(texreg)        # It is already installed for Chapter 4

# Import the GSS 2016 data
library(foreign)
chp5.gpo <- read.dta("C:/CDA/gss2016.dta")
chp5.gpo$healthre <- factor(chp5.gpo$healthre, ordered=TRUE)
chp5.gpo$educ <- as.numeric(chp5.gpo$educ)
chp5.gpo$wrkfull <- as.numeric(chp5.gpo$wrkfull)
chp5.gpo$maritals <- as.numeric(chp5.gpo$maritals)
attach(chp5.gpo)

# PO model with vglm() in VGAM
library(VGAM)
mod.po <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
TRUE, reverse = FALSE), data = chp5.gpo)
summary(mod.po)

# GPO model with vglm() in VGAM
mod.gpo <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE, reverse = FALSE), data = chp5.gpo)

# Using the lrtest() function to test the PO assumption
lrtest(mod.po, mod.gpo)


summary(mod.gpo)
coef(mod.gpo, matrix = TRUE)
confint(mod.gpo, matrix = TRUE)
exp(coef(mod.gpo, matrix = TRUE))
exp(confint(mod.gpo, matrix = TRUE))
cbind(exp(coef(mod.gpo)), exp(confint(mod.gpo)))

# Testing the overall model using the likelihood ratio test
gpomodel0 <- vglm(healthre ~ 1, cumulative(parallel = TRUE, reverse = FALSE))
summary(gpomodel0)
lrtest(gpomodel0, mod.gpo)
```

```
# Pseudo R2
library(rcompanion)
nagelkerke(mod.gpo)
AIC(mod.gpo)
BIC(mod.gpo)

# Predicted probabilities with ggpredict() in ggeffects
library(ggeffects)
prob.e <- ggpredict(mod.gpo, terms = "educ[12, 14, 16]", ci = NA)
prob.e
plot(prob.e)

# Predicted cumulative probabilities with ggpredict() in ggeffects
cumprob.e <- ggpredict(mod.gpo, terms = "educ[12, 14, 16]")
cumprob.e
as.data.frame(cumprob.e)
plot(cumprob.e)

# Logit coefficients of being at or above a category with reverse = TRUE
mod.gpo2<-vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE, reverse = TRUE), data = chp5.gpo)
summary(mod.gpo2)
coef(mod.gpo2, matrix = TRUE)
confint(mod.gpo2, matrix = TRUE)
exp(coef(mod.gpo2, matrix = TRUE))
exp(confint(mod.gpo2, matrix = TRUE))
cbind(exp(coef(mod.gpo2)), exp(confint(mod.gpo2)))
AIC(mod.gpo2)
BIC(mod.gpo2)
nagelkerke(mod.gpo2)

# PPO model with vglm() in VGAM
mod.ppo<-vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE ~ wrkfull, reverse = FALSE), data = chp5.gpo)
summary(mod.ppo)
coef(mod.ppo, matrix = TRUE)
confint(mod.ppo, matrix = TRUE)
exp(coef(mod.ppo, matrix = TRUE))
exp(confint(mod.ppo, matrix = TRUE))
cbind(exp(coef(mod.ppo)), exp(confint(mod.ppo)))

# Testing the overall model using the likelihood ratio test
lrtest(gpomodel0, mod.ppo)

# Pseudo R2
nagelkerke(mod.ppo)
AIC(mod.ppo)
BIC(mod.ppo)

# Predicted probabilities with ggpredict() in ggeffects
prob.e2 <- ggpredict(mod.ppo, terms = "educ[12, 14, 16]", ci = NA)
prob.e2
plot(prob.e2)

# Predicted cumulative probabilities with ggpredict() in ggeffects
cumprob.e2 <- ggpredict(mod.ppo, terms = "educ[12, 14, 16]")
cumprob.e2
as.data.frame(cumprob.e2)
plot(cumprob.e2)
```

```
# Logit coefficients of being at or above a category with reverse = TRUE
mod.ppo2 <- vglm(healthre ~ educ + maritals + female + wrkfull, cumulative(parallel =
FALSE ~ wrkfull, reverse = TRUE), data = chp5.gpo)
summary(mod.ppo2)
coef(mod.ppo2, matrix = TRUE)
exp(coef(mod.ppo2, matrix = TRUE))
exp(confint(mod.ppo2, matrix = TRUE))
cbind(exp(coef(mod.ppo2)), exp(confint(mod.ppo2)))
AIC(mod.ppo2)
BIC(mod.ppo)

# Presenting the results of the vglm Models using the texreg package
library(texreg)
screenreg(mod.ppo)
htmlreg(list(mod.ppo),   file="chap5ppo.doc",   doctype=TRUE,   html.tag=TRUE,
head.tag=TRUE)

detach(chp5.gpo)
```