Chapter 15:

Nominal and Ordinal Dependent Variables

Overview:

This chapter finishes our time examining various types of dependent variables. In this chapter, we examine dependent variables that are categorical — both nominal and ordinal response variables are covered in this chapter.

This could be variables with an ordering (like a Likert scale) or without (pet type). The type of regression used needs to take into consideration the characteristics of the dependent variable. Thus, this chapter starts with modeling nominal variables (no ordering) and proceeds to ordinal variables an ordering).

Forsberg, Ole J. (10 DEC 2024). "Nominal and Ordinal Dependent Variables." In *Linear Models and Řuritȁ Krȁlovstvì*. Version 0.704442$\eta(\alpha)$.

Chapter Contents

❧ ❧ ❧

One of the most pervasive research questions in Political Science is to predict a person's vote based on demographic information. In other words, if you know a person's age, gender, income, education, and religion, how well can you predict how that individual will vote in the upcoming parliamentary election?

At first glance, this question appears to be a binary dependent variable problem. After all, there are only two parties, right? Well, *even* if you ignore third parties, there is a third option: abstention. In each Ruritanian parliamentary election, a sizable number of registered voters decide not to vote. For instance, in the 2016 election, while Kuzněcov (of the royalist Král a Země party) received 48% of the vote cast and Ivanović (of the republican Republikánská Strana) received 46%, a full 45.3% of the eligible voters did not vote. Thus, the distribution of votes in this election is 26.3% Kuzněcov, 25.2% Ivanović, 3.2% other, and 45.3% none of the above. As such, conclusions based on those models that assume a binary outcome have definite issues with generalization to the voting public at large. They are ignoring important information.

A better alternative is to specifically add in 'abstention' and model the three possible outcomes at once (or 'abstention' and 'other' and model the four). Such a regression model is called a **nominal regression model** or a multinomial regression model, because there is no inherent ordering among the levels of the dependent variable.

There is a second type of dependent variable that is closely related to the nominal case — the ordinal dependent variable. The difference between the nominal and the ordinal is that the ordinal has more information contained in it. There is no ordering in the nominal case, whereas there *is* an implicit ordering in the ordinal case.[1] Examples of ordinal variables include ratings and indices.

If we just use our logistic regression methods (Chapter 12), we come up with some odd results. If we force a nominal variable into just two categories, we lose information in the data. If we treat ordinal dependent variables simply as nominal,

[1] Ordinal is actually a portmanteau for "ordered nominal."

information is also lost. If we treat them as continuous, our conclusions may not match reality.

Thus, both nominal and ordinal dependent variables need their own modeling methods. This chapter examines how to model both the nominal dependent variable and the ordinal dependent variable more properly.[2]

> **Note:** This chapter sits uneasily here. From the standpoint of the dependent variable type, this is its proper place. However, these are not generalized linear models (GLMs). They are particular expansions to the GLM paradigm. As such, if you are looking at the GLM modeling method as being the unifying theme to this part of the book, this chapter should not exist.

paradigm

But, it does.

[2] The study of statistics emphasizes both estimating the value (expected value) and the variance of that estimate (confidence interval).

15.1: Nominal Dependent Variable

A nominal variable is a categorical variable where there does not exist a meaningful ordering in the categories. Examples may include job type, presidential vote (and non-vote), and beer brand choice. These variables are categorical — not numeric — and the categories have no inherent ordering. White Collar is not 'greater than' Professional. Voting monarčista is not 'more than' voting republikán. Widmer is not 'more than' Coors.[3] How do we model such dependent variables?

There are a couple of ways of doing this. The first, easiest, and most understandable is to model the variable as a series of binary dependent variables. We already understand how this works, the testing of the model is already conceptually understood, and it works (*not really*, **but close**?).[4] There are just a couple things to clarify.

15.1.1 Mathematical model

As with the simply binary dependent variable case, let us layout the mathematical background to the nominal dependent variable case. As in the binary dependent variables case, we are actually modeling the underlying probabilities of each of the outcomes. Also, as in the binary case, there are five requirements for the random variable to follow a Multinomial distribution (*cf*, Section 13.1):

1. the number of trials, n, is known;
2. each trial has J possible outcomes;
3. the success probability for each trial, $\left\{\pi_1, \pi_2, \ldots, \pi_J\right\}$, is constant;
4. each trial is independent from the others; and
5. the random variable is the number of each type of outcome in those n trials.

Thus, if we let π_j be the probability that category j is selected, then the following two conditions must hold:

$$0 < \pi_j < 1 \qquad \text{for all } j \in \left\{1, 2, \ldots, J\right\} \tag{15.1}$$

$$\sum_{j=1}^{J} \pi_j = 1 \tag{15.2}$$

[3] Of course, there may be a time when you are predicting republikán vote by examining an underlying level of conservatism. In such a case, monarčista–republikán would be ordered. Thus, it really depends on what you are predicting (as always).

[4] Usually. Nothing in statistics *always* is best. As you have seen by now, there are always methods that work better, but with trade-offs. The science here is to be aware of the strengths with the weaknesses and balance them to get closer to the true process you are trying to model.

Condition (15.1) must hold because we are dealing with probabilities bounded by 0 and 1, and Condition (15.2) holds because one of the J possible outcomes *must* happen. In the binary case, our two probabilities were π and $1-\pi$, which satisfies the second condition by default and the first because it makes no sense to study phenomena that always or never occurs.

When we generalize the binary case, we need to select an appropriate probability distribution — one that can model J possible outcomes with J different probabilities. That distribution is called the multinomial distribution.[5] The probability density function for the multinomial distribution in the general case is

$$f_{\mathbf{X}}(\mathbf{X}) = \frac{n!}{x_1!x_2!\cdots x_J!}\ \pi_1^{x_1}\ \pi_2^{x_2}\ \cdots\ \pi_J^{x_J} \tag{15.3}$$

Here, x_i are non-negative integers and $\sum x_j = n$. The expected value of this distribution for a specified outcome is

$$\mathbb{E}\left[X_j\right] = n\pi_j \tag{15.4}$$

and the covariance between two outcomes is

$$\mathbb{C}ov\left[X_i, X_j\right] = -n\pi_i\pi_j \tag{15.5}$$

Be aware that $\mathbf{X}$ is a vector. So, if $n = 1$ and $J = 4$, the following could be outcomes from the Multinomial distribution:

$$x = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \qquad x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \qquad x = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

In the first example, a 2 came up; in the second, a 1; in the third, a 3. Note that in each case, the sum of the entries is n and the number of entries is J.

Now, if $n = 4$ and $J = 3$, the following could be outcomes from a Multinomial distribution:

$$x = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix}; \qquad x = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}; \qquad x = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

In the first example, three 2s and a 3 came up; in the second, two 1s, a 2, and a 3 came up; in the last, four 3s came up.

> **Note:** Be aware that the sum of the entries in each outcome vector is n and that the number of entries is J.

[5] Recall that the distribution in the *bi*nary case was the *bi*nomial distribution.

If the random variable $\mathbf{X}$ follows a Multinomial distribution with $n = 3$ and $\boldsymbol{\pi} = [0.1, 0.5, 0.4]'$, then we could write it as

$$\mathbf{X} \sim Multi\left(n = 3,\ \boldsymbol{\pi} = \begin{bmatrix} 0.1 \\ 0.5 \\ 0.4 \end{bmatrix}\right) \tag{15.6}$$

and the expected value of $\mathbf{X}$ would be

$$\mathbb{E}[\mathbf{X}] = \begin{bmatrix} 0.3 \\ 1.5 \\ 1.2 \end{bmatrix} \tag{15.7}$$

The expected value of X_3 would be $\mathbb{E}[X_3] = 1.2$.

Note: Make sure you see that this is just an extension of the binomial distribution, where

$$f_X(x) = \frac{n!}{x!(n-x)!}\ \pi^x (1-\pi)^{n-x} \tag{15.8}$$

with

$$\mathbf{X} = \begin{bmatrix} x \\ n-x \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\pi} = \begin{bmatrix} \pi \\ 1-\pi \end{bmatrix} \tag{15.9}$$

Example 1

Let us illustrate the multinomial distribution with a typical "rolling a die example." Assuming that the die is fair, then the probability of rolling each of the six outcomes is $\frac{1}{6}$. If we roll a fair die 3 times, what is the probability the outcome is $[1,0,1,0,0,1]'$ (that is, a 1, a 3, and a 6 come up)? What is the expected value of $\mathbf{X}$?

Solution: This is a multinomial experiment. There are a fixed number of possible outcomes (six), the probabilities of each outcome are constant (they do not change as we roll the die), and the probabilities sum to one. As such, we know the probability mass function is

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{3!}{x_1!\, x_2!\, x_3!\, x_4!\, x_5!\, x_6!} \left(\frac{1}{6}\right)^{x_1} \left(\frac{1}{6}\right)^{x_2} \left(\frac{1}{6}\right)^{x_3} \left(\frac{1}{6}\right)^{x_4} \left(\frac{1}{6}\right)^{x_5} \left(\frac{1}{6}\right)^{x_6} \tag{15.10}$$

Thus,

$$\mathbb{P}\left[\mathbf{X} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right] = \frac{3!}{1!\, 0!\, 1!\, 0!\, 0!\, 1!} \left(\frac{1}{6}\right)^{1} \left(\frac{1}{6}\right)^{0} \left(\frac{1}{6}\right)^{1} \left(\frac{1}{6}\right)^{0} \left(\frac{1}{6}\right)^{0} \left(\frac{1}{6}\right)^{1} \tag{15.11}$$

$$= 6\left(\frac{1}{6}\right)^{3} \tag{15.12}$$

$$= \frac{1}{36} \tag{15.13}$$

Thinking through the problem should get us to the same point.

Finally, we know the expected value is

$$\mathbb{E}[\mathbf{X}] = n\pi = 3\begin{bmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \tag{15.14}$$

♦

As we have a formula for our expected value, we have our mechanism for estimating the several π_j: in an experiment (or set of data), count the number of times outcome j occurred and divide by the total number of trials (or records). This is actually the maximum likelihood estimator for π_j. Thus, our linear predictor is

$$\text{logit}(\pi_j) = \beta_{j,0} + \beta_{j,1}x_1 + \beta_{j,2}x_2 + \cdots + \beta_{j,k}x_k \tag{15.15}$$

Notice that this linear predictor has $k+1$ parameters to estimate for each of the j categories. Thus, you will need more than $j(k+1)$ pieces of data to fit it. There are ways to reduce the dimensionality of the problem (reduce the number of parameters in need of estimation); however, these are beyond the scope of this book.

We need the logit link (or something just like it) to force our linear predictions to be in the range $\pi_j \in (0,1)$. As any link that maps $g : \mathbb{R} \to (0,1)$ is acceptable, we could use the log-log link, the complementary log-log link, the probit link, or any of an infinite number of others… in theory. As before, the choice of the link function is largely a matter of tradition. If you deviate from tradition, the burden of proof is on you to justify the selection. Furthermore, the differences are usually slight. If the differences are large, then there is something wrong with your research model. Because of this, it would behoove you to fit your research model using a couple different (appropriate) link functions to help determine the stability (robustness) of your results.

robustness

Note: Thus, there are two things that you need to take away from this discussion: First, we are able to fit the entire model at once because we have a distribution that can produce the necessary nominal results. Second, we model the underlying probabilities (like in the binary case), not the actual outcomes, as usual. **two things**

To see this in action, let us look at an extended example.

Example 2

The General Social Survey (GSS) at the University of Chicago conducts an extensive survey of adult Americans every year. The data is freely available from NORC. In this small subset of the data, `gssocc`, I would like to predict a person's occupation category (`occ`) based on race (`white`), years of education (`ed`), and years of experience (`exper`).

Before getting started, let us examine the variables involved.[6] The race variable is binary, with a '1' representing the person identifying as 'white' and a '0' otherwise. As a side note, this is a race variable, not an ethnicity variable. Thus, Hispanics may self-identify as either white or non-white. Also note that this is a self-identification variable; that is, the individual being surveyed decided his or her reported race. Looking at a frequency count, a full 91.69% of the respondents stated they were white. This is significantly higher than the population at large, where approximately 80% of Americans were white when the survey was conducted. When we do the final analysis, we need to keep this in mind, as it is not necessarily representative of the nation as a whole.

The median number of years of education in the sample is 12 years, which corresponds to graduating from high school. The mean number of years is 13.09, which indicates the sample is right skewed (the Hildebrand ratio is +0.37). Furthermore, it is interesting to note that 51.0% of the sample only graduated from high school. Additionally, 23.4% of the sample received a bachelor's degree or more, which is close to the population (27% have received a bachelor's degree or higher). Finally, 18.7% of the sample did not graduate from high school, which is close to the 15% estimate of the population. From this, it appears as though the sample is representative of the population in terms of educational attainment.

The third independent variable is the years of experience in the job. There are no general statistics for the population, so we will have to make a large assump-

[6] The raw — and current — data can be accessed from `http://www.norc.uchicago.edu/GSS+Website/`.

	White	Education	Experience
White	1.0000	0.0243	-0.0794
Education	0.0243	1.0000	-0.2740
Experience	-0.0794	-0.2740	1.0000

Table 15.1: *Correlation matrix for the three independent variables in the example, from* `gssocc` *data.*

tion that the sample represents the population.[7] In the sample, the years of experience varies widely, from 2 to 66 years. The median is 17 years and the mean is 20.5 years. Thus, the sample is also right skewed. This makes sense as this is a count variable. Count variables tend to be right skewed as they cannot take on negative values. In fact, there is nothing in the distribution of the experience variable that looks wrong. With that said, however, one still needs to mention the caveat.

Looking at the correlations amongst the independent variables can help us avoid any unpleasantness and surprises due to collinearity and multicollinearity. The correlation matrix (Table 15.1) does not show any hint of multicollinearity. In fact, this correlation matrix suggests that these three variables are *effectively* independent of each other.[8]

Finally, let us note that there *may* be an inherent ordering in some of the jobs (White Collar greater than Blue Collar), but not for all five of the categories. As such, this is definitely a candidate for nominal regression.

[7] This was a safe assumption with respect to the education variable, but not with respect to the white variable. As such, it needs to be mentioned that you are unable to check the representativeness of the experience variable.

[8] Pearson's product-moment correlation test indicates that the correlation between education and experience is statistically significant at the $\alpha = 0.05$ level ($t = -5.2152, df = 335, p \ll 0.0001$). However, the coefficient of -0.2740 is a low level of correlation.

	Estimate	Std. Error	z-value	p-value
Intercept	3.1036	1.0110	3.07	0.0021
White	0.7090	0.6213	1.14	0.2538
Years of education	-0.3721	0.0640	-5.81	0.0000
Years of experience	-0.0259	0.0113	-2.30	0.0215

Table 15.2: *Results from the GLM (using the binomial family and the logit link) predicting whether or not a person is a blue collar worker. The AIC for this model is 304.75.*

NOMINAL REGRESSION: Now, let us model the outcome variable with the three independent variables. Actually, we need to step back and really think about what we mean by 'model the outcome'. Do I want to predict the probability that a person will be Blue Collar given the input variables? Or: Do I want to predict the job category given the input variables? These are different questions. They require slightly different methods.

The first question actually asks a binary question: What is the probability that a person will be Blue Collar (compared to *all* of the other job categories)? This is very much like the questions asked in Chapter 12. Here, the dependent variable takes on values 1 (Blue Collar) and 0 (not Blue Collar).

To answer *this* question, we need to create a variable called `bluecol` as an indicator variable for Blue Collared-ness. Thus, the model we fit will be

```
bluecol ~ white + ed + exper
```

We would fit it using a generalized linear model, a binomial family, and a logit link. The results of the regression are in Table 15.2. From this model, we can perform all of the goodness of fit measures from Chapter 12.

Looking at the results from running the model, we see that greater levels of education and greater levels of experience are associated with a lower probability of being a blue collar worker. For Bob, an individual who responded that he was white, had 20 years of education, and 10 years of experience in their current job, the probability of being a blue collar worker is approximately 2% (as compared to not being a blue collar worker).

Note: This last part is subtle, but extremely important. Here is why: What is the probability that Bob is a white collar worker? If we do the same steps above, we get that the probability that Bob is a white collar worker (as compared to not being a white collar worker) is 13.1%. Similarly, if we continue performing separate logistic regressions, the probability that Bob is a professional is 96.9%; menial, 2.3%; and craft, 7.9%. **important**

Note that all of these probabilities add up to **more than 100%**. There is something wrong here, since the probability that Bob holds one of these five job types cannot be greater than 100%.

moral of the story

The problem is that we kept changing the base category. In Chapter 12, we never mentioned the need to specify the base category since it always defaulted to the opposite of what we were modeling. In other words, we were actually measuring the probability of an event as compared to the probability of 'not the event' (a.k.a. the odds of the event). This ensured that the probabilities always added up to 100%. Within each of the above five regressions, if we added the probability of the event that Bob holds job type X with the probability that Bob holds job type not X, we always get 100%.

the lesson

The lesson: Comparing probabilities of events is not as easy as when we were only working in the binary realm. It is doable — easily so, with one small change. We need to select a base category that does not change throughout our analysis. The choice is up to you, as all choices are equally acceptable from a statistics standpoint.

Since we can select any job type as our base, let us select Blue Collar, since it is the first level according to the alphabet. (We will see again shortly how to switch between the bases.)

To perform this modeling, you will have to load the `nnet` package. Since this comes with your base distribution of `R`, there is no need to install it. Once loaded with the `library(nnet)` command, to fit the better model, use the `R` command

```
multinom(occ ~ white + ed + exper)
```

Because of the large amount of output, the regression table is structured slightly different. The coefficients (in logit units) and the standard errors are still presented. The statistical significance is not. However, a quick rule of thumb is that the variable is statistically significant (at the $\alpha = 0.05$ level) if the parameter estimate is more than twice the standard error. Table 15.3 presents the output from modeling the data in the form given in the output.

Note that one of the five job types is missing: Blue Collar. This is because all of the probabilities are measured *with respect to* Blue Collar. Thus, these percentages are directly comparable (after transforming from logit units).

`R` is nice in that if you `predict` on a multinomial model, it will give you the category with the highest probability, by default. Thus, according to this model, Bob will most likely be a Professional (which was our conclusion above). If we want the probabilities for each of the possible job types for Bob, we need to add a `type="probs"` parameter to our function call:

```
predict(model.mn1, newdata=BOB, type="probs")
```

Such a call gives us the following probabilities (which sum to one, as they should):

Coefficients:				
	Constant Term	White	Education Level	Experience
Craft	-1.8328	-0.7642	0.1933	0.0230
Menial	-0.7412	-1.2365	0.0994	-0.0074
Prof	-12.2595	0.5376	0.8783	0.0309
WhiteCol	-6.9800	0.3349	0.4526	0.0299
Std. Errors:				
	Constant Term	White	Education Level	Experience
Craft	1.1861	0.6324	0.0775	0.0126
Menial	1.5195	0.1996	0.1023	0.0174
Prof	1.6681	0.7996	0.1005	0.0144
WhiteCol	1.7144	0.9340	0.1023	0.0153

Table 15.3: *Results of the multinomial regression. Note that the p-values are not provided. To determine which independent variables are statistically significant for predicting the dependent variable levels, divide the coefficient estimate by the standard error. If that ratio is greater than 2, then the variable is statistically significant at the* $\alpha = 0.05$ *level.*

BlueCol	Craft	Menial	Prof	WhiteCol
0.0020	0.0091	0.0020	0.9565	0.0304

Base switching: If you wish to switch your base category, there are two options. First, you can subtract the parameter estimates of the new base from all the other bases. Thus, if we want to change the base from Blue Collar to Professional, we would subtract the Professional parameter estimates from the other parameter estimates. So, for example, the White Color estimates with Professional as the base will be $-6.9800 - -12.2595 = 5.2795$. Unfortunately, the standard errors are not so easily calculated — or at all reasonably calculable by hand.

Also unfortunately, most statistical programs require you to physically reorder the data to select a different base; most programs use the level of the first data point as the base category. `R` does allow you to switch among the bases without having to physically alter the data. Unfortunately, the method is rather arcane. Fortunately, the `RFS` package has a function, `set.base` that allows you to change the bases much more easily.

Thus, to set `craft` as the base, we would use the command

```
occ = set.base(occ, base="craft", data=gssocc)
```

I leave it as an exercise to rerun the analysis with `craft` as the base. Check that the parameter estimates follow the above observation.

Interpretation: The interpretation of the coefficients (parameter estimates) is the same as for the binary dependent variable case. Just remember that the coefficients are in logit units. In `R`, however, this library does not require you to back-transform your predictions. To remember this, just look at the output — it is in proportions already (a quick check is that they sum to one).

Goodness of fit: The first check of the goodness of the model is the relative accuracy (see also Section 12.5). The accuracy is the number of correct predictions divided by the number of cases. The relative accuracy divides this number by the accuracy of always selecting the modal category (**the null model**). For this dataset, the modal category is Professional, with 140 out of 337 cases belonging to Professionals. Thus, the relative accuracy is $\frac{169}{337}/\frac{140}{337} = 1.207$. Thus, this model improves accuracy by 21% over the null model. Is this good? It depends on your other models.

As Maximum Likelihood Estimation is used, the Akaike Information Criteria score is also reported. For this model, $AIC = 885$. Is this good? Again, it depends on your other models. In other words, model comparison needs another model. I leave it as an exercise to see that the null model has $AIC = 1027$. Thus, our model is much better than the null model.

Now that we have looked at our model, let us look at the parameter estimates. According to our model, Whites have a higher probability of being Professionals and White Collar workers than they are to be Craft or Menial laborers. As for education, higher levels of education are associated with higher odds of being a Professional or a White Collar worker (both of these are statistically significant) than being a Blue

Collar worker. Finally, years of experience are not a statistically significant predictor of job type, as none of the coefficients are statistically significant (coefficient / standard error > 2).[9]

So, we have a picture of Professionals and White Collar workers, when compared to Blue Collar workers: they are White and well educated. Not an earth-shattering conclusion, but it is encouraging to see that our conclusions do seem to reflect reality.

[9]This rule of thumb comes from the fact that in a Normal distribution, the ratio needs to exceed 1.96 to be statistically significant at the $\alpha = 0.05$ level. These parameter estimates are not guaranteed to be Normally distributed. As such, the rule of thumb is to be more conservative. Even with the rule of thumb, do not bet the farm.

15.2: Ordinal Dependent Variable

Another variety of categorical dependent variables is ordinal. A variable is ordinal if it is categorical *and* the categories have an underlying order to them. Examples include movie ratings (number of stars), hurricane intensity, and so forth.

There are actually at least four ways of handling ordinal dependent variables:

1. Treat them as nominal. This allows us to fit ordinal data using previous techniques. Unfortunately, it is inefficient as it ignores important aspects of the data itself.
2. Treat their cumulative level as nominal. If the ordinal variable takes on values 1 – 5, then create nominal variables corresponding to Level 1, Levels 1 and 2, Levels 1–3, Levels 1–4, and Levels 1–5. This preserves much of the underlying information *and* allows us to fit it using a previous method.
3. Assume that there is an underlying continuous process that you wish to fit. The ordinal nature is just several threshold values along the possible values. This reduces to a pseudo-OLS, where you also need to fit the threshold values, not just the slopes and intercepts. Using Maximum Likelihood methods, this is trivial to solve.
4. Pretend that the ordinal values are continuous and fit it using ordinary least squares or one of its offsprings. This has the advantage of being easily fit.

Three of these ways have already been discussed, and you are quite adept at using them (Options 1, 2, and 4). Only the third option is completely new to you. This chapter focuses on how to fit Option Three.

15.2.1 Option Three

Let us assume that there is an underlying continuous process. We only experience (observe) this process through the ordinal variable. This is very similar to how we first looked at binary variables: underlying process exhibited only in the 0/1 outcomes (see Figure 12.2). Here, there is more than just the one threshold (which traditionally defaulted to 0.500). Thus, we have two sets of parameters to fit. The first is the parameters which describe the process (the βs). The second is the position of those threshold values (the τs).

Without going into the details, we will use Maximum Likelihood Estimation as our fitting method because it has many nice properties. Thus, our underlying process is

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{15.16}$$

Our thresholding process is illustrated in Figure 15.1. The line represents the underlying continuous process that you are trying to model. The A, B, C, and D represent the observed ordinal values. The threshold values, τ_1, τ_2, and τ_3 are the values of η that separate the observed ordinal values.

This model is very straight forward and understandable. Using R to obtain the fitting is also straight forward. The results presented are also relatively straight forward.

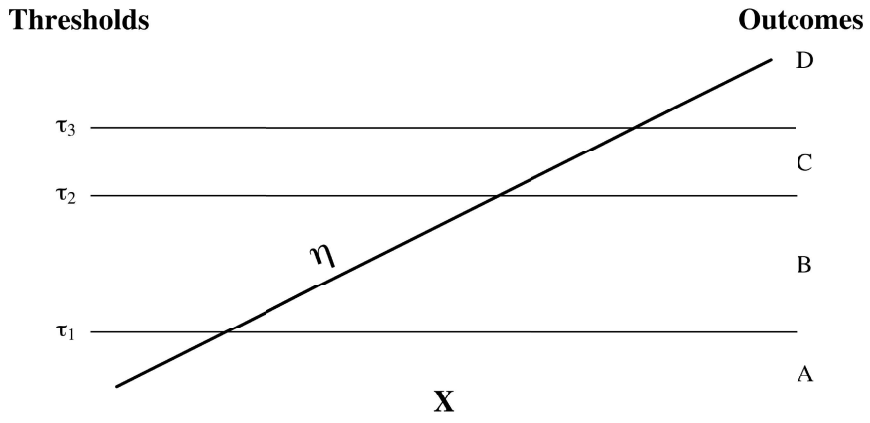

Figure 15.1: *Schematic diagram of the thresholding process. The line represents the linear continuous process. The τs represent the threshold values. A, B, C, and D represent the ordinal outcomes.*

Variables:		Value	Std. Error	t-value
	Woman	0.743	0.078	9.50
	White	-0.400	0.118	-3.39
	Age	-0.020	0.0024	-8.17
	Years of Education	0.098	0.013	7.52
Thresholds:				
	SD — D	-1.700	0.237	-7.18
	D — A	0.111	0.233	0.48
	A — SA	1.979	0.236	8.37

Table 15.4: *Result of ordinal regression in* `R`*. Note that the women tend to view President Obama in a more favorable light; whites, less; older, less; and higher educated, more. All of these agree with multiple surveys throughout his tenure as President.*

Example 3

Let us use some more data from the GSS. This data explores the 'warmth of feeling' the respondent has for President Obama. The demographic information is the gender (`male`), the race (`white`), the age, and the number of years of education (`ed`). The response variable has four ordered levels: Strongly Disagree (SD), Disagree (D), Agree (A), and Strongly Agree (SA). Our goal is to explain a person's feelings toward the president based solely on demographic information.

Solution: Let us fit this data with ordinal regression. The function in `R` is `polr`, which stands for "proportional odds logistic regression" (although the probit is an option as a link function). This function requires the `MASS` package. Thankfully, since `MASS` also comes with the base distribution of `R`, there is no need to install it, only to load it via the `library(MASS)` command.

The actual command to fit this model using ordinal regression is

```
polr( warm ~ male + white + age + ed )
```

This command will give the coefficients of the underlying linear regression and the threshold values separating the four categories. From Table 15.4, we see that the equation for the underlying linear process is

$$\eta = 0.743 \times \texttt{Woman} + -0.400 \times \texttt{white} + -0.020 \times \texttt{age} + 0.098 \times \texttt{ed}$$

The thresholds are also listed. The threshold between Strongly Disagree and Disagree is at $\tau_1 = -1.700$. The threshold between Disagree and Agree is $\tau_2 = 0.111$. The threshold between Agree and Strongly Agree is $\tau_3 = 1.979$. Thus, to calculate our prediction, we calculate the prediction based on the linear model, η, and compare that value to the intervals described by the thresholds. Thus, for Bob, who is Male, White, 40 years old and has 20 years of education, we have

$$\eta = 0.740 \times 0 + -0.400 \times 1 + -0.020 \times 40 + 0.098 \times 20 = 0.76$$

As $\eta = 0.76$, we have our prediction that Bob agrees with the president. If we actually want probabilities that Bob Strongly Disagrees, Disagrees, Agrees, or Strongly Agrees, we would have to back-transform using the inverse of the logit function and calculate each probability using integral calculus... or we could just ask the computer to do it for us:

back-transform

```
BOB = data.frame(male="Men", white="White", age=40, ed=20)
predict(model.ol1, newdata=BOB, type="probs")
```

This gives the probabilities as

SD	D	A	SA
0.0785	0.263	0.429	0.229

Thus, it is far from certain that Bob agrees (or strongly agrees) with the president, although that probability is rather high: $0.429 + 0.229 = 0.658$. ♦

Accuracy: Finally, let us look at the accuracy of the model. I leave it as an exercise to show that the relative accuracy is 1.105, which indicates that the model is about 10.5% better than the null model (the modal category is "Agree"). This is not a fantastic increase in accuracy, but we do know how certain demographics feel about the president: Whites tend to disagree, Males tend to disagree, older people tend to disagree, and lesser educated people tend to disagree.

Of course, we could have added in a quadratic education term to the model to see if both the more-educated and the less-educated both support the president. I also leave this as an exercise to show that there is no evidence of this. Thus, we have no evidence that the relationship between education and presidential support is anything other than linear.

15.3: Extended Example: Cattle Feed

Now that we have been introduced to these two new types of regression, let us deal with an example of each. This example tries to predict the feed type used for a cow. Such a question would arise if there is missing data in your data file and you wanted to estimate the missing value instead of throwing out the entire record.

Example 4

Previously, we attempted to model the weight of cattle based on a few factors. Let us try something different. Let us predict the brand of food used by the cattle based on the ranch, age, and weight.

Specifically, let's first model feed type. Then, let's say that the RUR ranch sent a 21-year-old cow to slaughter at 1197 pounds. Which food brand was most likely used? What are the probabilities of each brand being used?

Solution: Since the food brand is a nominal variable, we will use multinomial regression. The data file is `cattleData`. Let's load it and look at some summary statistics on it.

```
library(nnet)
cowz = read.csv("http://rur.kvasaheim.com/data/cattleData.csv")
attach(cowz)

summary(cowz)

cor.test(weight,age)
table(ranch,feedType)
```

Note that there is (as expected) a strong correlation between age and weight. If we are doing model selection, we will need to keep this in mind as this multicollinearity will decrease the statistical significance of those two variables.

Note from the cross-tabulation that the EVA ranch only used Purina and the TCL ranch only used Rangeland (in this sample). That fact would make it really easy to predict the feed type for those ranches. The other ranches use a combination of all of the brands.

With this information to guide us, we fit the model

```
cowModel = multinom(feedType ~ weight + ranch + age)
summary(cowModel)
```

The first line fits the model. Note that the model did converge, so we can pay attention to the results. If it had not converged, we should first change the link function, then realize that the multicollinearity is a problem. Dropping one or more variables would be an appropriate action in that case.

The results of the `summary(cowModel)` command gives some insight into the relationships. First, note that the coefficient estimate for `ranchEVA` for estimating `Purina` is 20.37. This is extremely high, meaning it is almost guaranteed that a cow from the EVA ranch will used Purina.

But, from the cross-tabulation above, we already knew this.

Similarly, the coefficient estimate for `ranchTCL` for Rangeland is a huge 22.32. This indicates a cow from the TCL ranch will most certainly use Rangeland food. Again, we knew this from our cross-tabulation.

Note from the regression table that `Accuration` is missing. All feed measurements are taken with respect to that level. This is important to keep in mind if we do this by hand. It is just something to note if we are using the computer to do our calculations.

❧ ❧ ❧

So, let's estimate the food used by our mystery cow. First, let's define it:

```
mysteryMoo = data.frame(weight=1197, age=21, ranch="RUR")
```

Now, let's predict the probabilities it used each of the feed types:

```
predict(cowModel, mysteryMoo, type="prob")
```

The results tell us that the mystery cow most likely used Steakmaker. In fact, the probability it used Steakmaker was 79%. The second most likely feed type was Accuration (13%). ♦

15.3.1 Graphics

Let us talk about graphics for a bit. A two-dimensional scatter plot looks at two numeric variables. We can therefore easily plot a prediction curve when dealing with only a single dependent variable and single independent variable.

If there is a second independent variable, we can plot several curves, one for each level in that second independent variable.

Once we move beyond two independent variables, graphics are more difficult to do. A simple regression model like the one above may require dozens of graphics to illustrate each aspect.

However, we can simplify things by focusing on only a couple independent variables at a time. The choice depends on the story you are trying to learn (or tell).

Graphic: Feed Type versus Weight: For this first graphic, I am consciously making the decision to plot the predicted probability on the y-axis, the cattle weight on the x-axis, and have a prediction curve for each feed type. This will allow me to see the effect of weight on the predicted food type.

This means I need to select values for the other two independent variables. For the numeric age, I would typically use its mean or median, whichever was the "typical" age for these cattle.

For the selected value of the ranch, I would either select the ranch to which the mystery cow belonged (to continue that story) *or* the most popular ranch (to try to generalize the story). It is best to do separate graphics for all ranches so that you, the researcher, can better understand the effect of ranch on the probabilities. It is always better to do more to understand.

So, here is the code to create the predictions:

```
theWeights = seq(1019,1579, length=1e4)
theAge = median(age)
prRUR = predict(cowModel, newdata=data.frame(weight=theWeights
    , age=theAge, ranch="RUR"), type="probs")
```

The `prRUR` variable contains 10,000 rows (one for each weight) and 5 columns (one for each feed type). The entries are the probabilities.

Now, we just plot the data and these predictions:

```
par(family="serif", las=1)
par(xaxs="i", yaxs="i")
par(mar=c(4,4,0,0)+0.5)
par(cex.lab=1.2, font.lab=2)

plot.new()
plot.window( xlim=c(1000,1600), ylim=c(0,1))

axis(1); axis(2)
title(xlab="Weight [lb]")
title(ylab="Probability at RUR Ranch")

lines(theWeights,prRUR[,1], col=1) # Accuration
lines(theWeights,prRUR[,2], col=2) # Purina
lines(theWeights,prRUR[,3], col=3) # Rangeland
lines(theWeights,prRUR[,4], col=4) # Steakmaker
lines(theWeights,prRUR[,5], col=5) # Wind and Rain

legend("topright", bty="n", col=1:5, lwd=2,
  legend=c("Accuration","Purina","Rangeland","Steakmaker","Wind
     and Rain")
  )
```

Note that this graphic includes a legend that lets the reader know which probability curve belongs to which feed type. Legends are rather important to include on a graphic. Remember that graphics should be stand-alone with their caption. Because a legend contains so much information, it requires a large function. To see all a legend can do, run `?"legend"` in `R`.

Figure 15.2 is the resulting graphic. Note that the predicted feed type tends to be either Steakmaker, for light cows, or Purina, for heavy cows. When the weight of the cow is middling, there is great uncertainty in which feed type it used.

It is interesting that this analysis gives us additional insight on how we can create big cows for slaughter. This suggests we should use Purina brand. This conclusion, however, is based only on the RUR ranch and a middle-aged cow.

More importantly, this conclusion assumes that the data are representative of the population of interest. As this data was originally collected in conjunction with a dissertation in Animal Science, I tend to think it is representative.

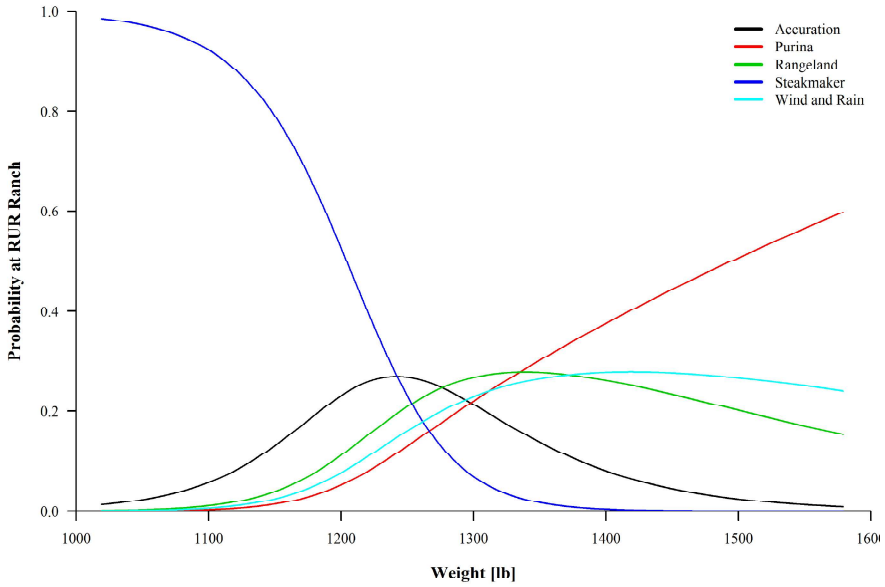

Figure 15.2: *Graphic of the probability for each feed type at the RUR ranch. The probabilities vary with the cow's weight. The age is held at the median, 21.*

From a strictly statistical standpoint, the additional insight is limited. However, if you are hired by RUR ranch to determine the best feed type, this graphic would be *very* persuasive for you, the decision-maker.

While we could create a similar graphic for feed type against age, I am not convinced it would be helpful. Age is not something one would like to optimize like weight. In other words, I am not sure what story I would tell about it.

Note: Don't make graphics just for fun. Make sure you create them knowing how to interpret them.

15.4: Extended Example: The State University of Ruritania

A second example will try to model the level of the student given some information about the student. Again, this may be interesting for imputation (filling in missing data).

impute

Example 5

Previously, we modeled the grade point average of students at the State University of Ruritania (*Státní Univerzita v Ruritánii*). Let us turn this around and model the student's class (Freshman, Sophomore, Junior, Senior) given only the gender and the current GPA of the student.

Let us also predict the class of Eliska, a female student with a 3.33 GPA.

Solution: As usual, the first step is to import the data and look at a summary, including a cross-tabulation of our categorical independent variable and the dependent variable:

```
library(MASS)

suvrData = read.csv("http://rur.kvasaheim.com/data/suvr.csv")
summary(suvrData)
```

Let us pause here. Note that the `class` variable is an ordinal variable. We need to let R know this:

```
suvrData$class = ordered(suvrData$class, levels=c("Non-
    Matriculated", "Freshman", "Sophomore", "Junior", "Senior")
     )
summary(suvrData)
```

There we go, the levels for the `class` variable are in the right order. Let's continue.

```
attach(suvrData)
table(gender, class)
```

Note that none of the non-matriculated students are female. This is just something to know and remember as we get results.

Now, we can fit our model and look at the summary results:

```
suvrModel = polr(class  ~  gender + gpa, data=suvrData)
```

```
summary(suvrModel)
```

check! A quick check that you have ordered the levels correctly is to look at the second table in the summary output. The rows should describe subsequent levels.

The AIC of this model is 1547. The AIC of the null model

```
suvrNullModel = polr(class ~ 1, data=suvrData)
summary(suvrNullModel)
```

is 1566. Thus, our model is an improvement.

The model predicts that Eliska is a Junior (44.7%) or a Senior (36.5%):

```
eliska = data.frame(gender="Female", gpa=3.33)
predict(suvrModel, eliska, type="prob")
```

Here are the (abbreviated) raw results

```
Non-Mat    Fresh      Soph       Junior     Senior
0.00211    0.02356    0.16293    0.44656    0.36483
```

Thus, we do have an estimate for Eliska's class level, but there is a second option which is rather close. I'm not sure I would bet any money on where to put Eliska.

Regardless, it is highly unlikely for Eliska to be either non-matriculated or a Freshman. Those probabilities, while non-zero, are *very* low. ♦

Warning: *Beware! Remember that the data are not representative of the population. The distribution of the classes is quite similar to the probabilities predicted for Eliska. This is not surprising. The effect of the independent variables on the dependent variable are not statistically significant. Thus, these probabilities are essentially the relative proportions of the classes in the sample.*

Graphic: Class against GPA: As for a graphic, we need our dependent variable to be the probability of each class. Since there is only one numeric independent variable, GPA, that will be the variable we graph along the x-axis.

The ultimate question is: What do we do with the gender variable?

One option is to plot the effect of gender on the same graphic. That means we will have 5×2 curves on the same plot (the number of levels by the number of genders recorded). That *may be* problematic as it may overwhelm the graphic. Figure 15.3 is this figure. Note that it does allow us to compare everything at once. However, you may find it overwhelming... or not.

For the higher GPA values, it is most likely that the student is a Junior, regardless of the gender. At no place is it likely the student is either a Freshman or non-matriculated. This is supported by the data, as the number of non-matriculated students is just 2 and the number of Freshman is just 22 — out of a sample size of $n = 661$.

We can also use this graphic to estimate the various probabilities for Eliska. Remember she has a GPA of 3.33. Since Eliska is female, we look at the dashed lines. Going to 3.33 on the x-axis and move vertically, we see that Eliska is most likely a member of the cyan level — Junior — with a close second being the magenta level — Senior. This conclusion agrees with our prediction above.

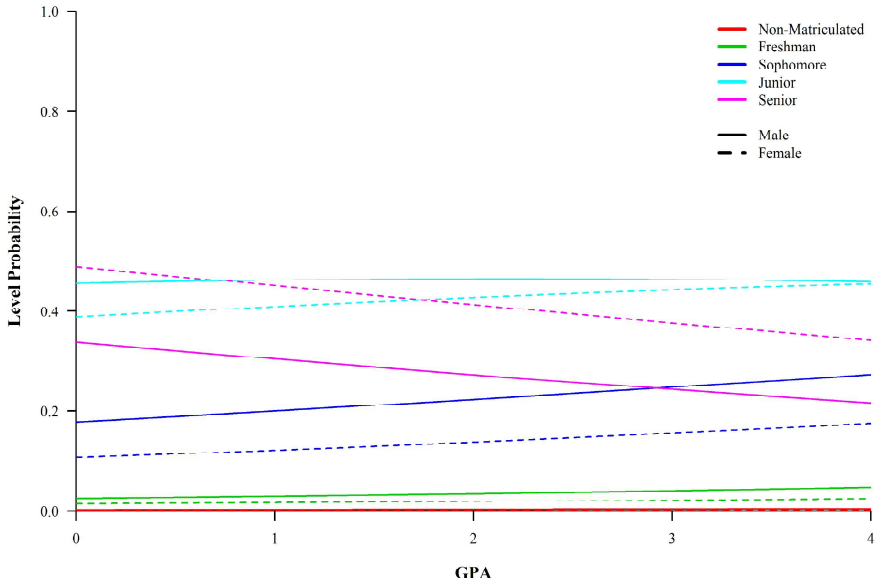

Figure 15.3: *Graphic of the probability for each class level for each gender. Note that the non-matriculated and the Freshman levels uniformly have low probability. This is due to the nature of the data; only 2 non-matriculated and 22 Freshmen are in the sample of size* $n = 661$. *This limits what we can say about the population, unless the level distributions are similar to the population.*

15.5: Conclusion

In this chapter, we examined the special issues behind fitting dependent variables that are either nominal or ordinal. Nominal dependent variables are still basically fit with a series of logistic (or other link) regressions. The alteration comes about because we need to keep the same base category throughout in order to make our results comparable.

The ordinal dependent variable can be fit using a technique similar to the previous chapter: fit an underlying linear function, then create thresholds to divide a constant function into an ordinal response.

In both cases, predictions in R follow the typical structure, with the addition of being able to just predict the outcome category or being able to predict the probabilities associated with the case fitting in each bin.

15.6: End-of-Chapter Materials

15.6.1 R Functions In this chapter, we were introduced to several R functions that will be useful in the future. These are listed here.

Packages:

RFS This is a "book package," that is not yet complete. In lieu of installing this package and loading it with `library(RFS)`, you will activiate all of its important parts by running `source("http://rfs.kvasaheim.com/rfs.R")`.

MASS This package is also a "book package," a package created for a specific book. Here, that book is "Modern Applied Statistics with S."

nnet This package contains many functions dealing with neural networks. For this chapter, we use it to fit multinomial models.

Statistics:

multinom() This modeling function allows you to fit nominal dependent variables. Its structure is standard in that its main argument is the formula. In order to use the `multinom` function, you must load the `nnet` library.

polr() This modeling function allows you to fit ordinal dependent variables when there is an underlying linear function that drives the process. In order to use the `polr` function, you must load the `MASS` package.

predict(model, newdata) As with almost all statistical packages, R has a predict function. It takes two parameters, the model, and a dataframe of the independent values from which you want to predict. If you omit `newdata`, then it will predict based on the independent variables of the data itself, which can be used to calculate residuals. The dataframe must list all independent variables with their associate new values. You can specify multiple new values for a single independent variable.

set.base() This allows one to change the base category from which all other levels are estimated. It is a member of the `RFS` package.

15.6.2 Exercises

This section offers suggestions on things you can practice from this chapter.

1. In Section 15.1.1, we fit a multinomial model to the `gssocc` data. The base used was 'Blue Collar.' Refit the model using 'Craft' as the base category.
2. Determine the AIC of the null model in Section 15.1.1.
3. As mentioned in Section 15.2.1, calculate the relative accuracy of the model of Example 15.2.1.
4. As mentioned in Section 15.2.1, add a quadratic education term to the model of Example 15.2.1 to see if both the highly educated and the lesser educated both support the president.

15.6.3 Applied Readings

- Paul D. Allison and Nicholas A. Christakis. (1994) "Logit Models for Sets of Ranked Items." *Sociological Methodology* 24: 199–228.
- John Fox and Robert Andersen. (2006) "Effect Displays for Multinomial and Proportional-Odds Logit Models." *Sociological Methodology* 36: 225–55.
- Daniel Carson Johnson. (1997) "Formal Education vs. Religious Belief: Soliciting New Evidence with Multinomial Logit Modeling." *Journal for the Scientific Study of Religion* 36(2): 231–46.
- Mark R. Killingsworth and Cordelia W. Reimers. (1983) "Race, Ranking, Promotions, and Pay at a Federal Facility: A Logit Analysis." *Industrial and Labor Relations Review* 37(1): 92–107.
- Alan B. Lowther and John R. Skalski. (1998) "A Multinomial Likelihood Model for Estimating Survival Probabilities and Overwintering for Fall Chinook Salmon Using Release: Recapture Methods." *Journal of Agricultural, Biological, and Environmental Statistics* 3(2): 223–36.
- Christopher Winship and Robert D. Mare. (1984) "Regression Models with Ordinal Variables." *American Sociological Review* 49(4): 512–25.
- Judith E. Zeh, Daijin Ko, Bruce D. Krogman and Ronald Sonntag. (1986) "A Multinomial Model for Estimating the Size of a Whale Population from Incomplete Census Data." *Biometrics* 42(1): 1–14.

15.6.4 Theory Readings

- B. R. Bhat and N. V. Kulkarni. (1966) "On Efficient Multinomial Estimation." *Journal of the Royal Statistical Society. Series B (Methodological)* 28(1): 45–52.
- Zhen Chen and Lynn Kuo. (2001) "A Note on the Estimation of the Multinomial Logit Model with Random Effects." *The American Statistician* 55(2): 89–95.
- Jean-Yves Dauxois and Syed N. U. A. Kirmani. (2003) "Testing the Proportional Odds Model under Random Censoring." *Biometrika* 90(4): 913–22.
- Byung Soo Kim and Barry H. Margolin. (1992) "Testing Goodness of Fit of a Multinomial Model Against Overdispersed Alternatives." *Biometrics* 48(3): 711–19.
- Bercedis Peterson and Frank E. Harrell, Jr. (1990) "Partial Proportional Odds Models for Ordinal Response Variables." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 39(2): 205–17.
- G. A. F. Seber and S. O. Nyangoma. (1974) "Residuals for Multinomial Models." *Biometrika* 87(1): 183–91.
- M. Stone. (1974) "Cross-Validation and Multinomial Prediction." *Biometrika* 61(3): 509–15.
- Y. K. Tse. (1987) "A Diagnostic Test for the Multinomial Logit Model." *Journal of Business & Economic Statistics* 5(2): 283–86.