CHAPTER 7

Nominal and Ordinal Dependent Variables

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 presidential election?

At first glance, this 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 US presidential election, a sizable number of registered voters decide not to vote. For instance, while Obama received 53% of the vote cast and McCain received 46%, a full 37.8% of the eligible voters did not vote. Thus, the distribution of votes in the 2008 US Presidential election is 33.0% Obama, 28.6% McCain, 0.6% other, and 37.8% none of the above.

I'm not sure where I am going with this, so let me figure that out in the next draft. Start the next page anew.

7.1 Nominal Dependent Variable

A nominal variable is a categorical variable where there does not exist an inherent ordering in the categories. Examples include job type, presidential vote (and non-vote), and beer brand choice. These variables are categorical—not continuous—and the categories have no inherent ordering. White Color is not 'greater than' Professional. Democratic vote is not 'larger than' Republican vote. Budweiser is not 'more than' Coors.[1] How do we model such dependent variables?

There are a couple 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.[2] There are just a couple things to clarify. As such, let us look at an extended example.

7.1.1 Occupations

The General Social Survey (GSS) conducts an extensive survey of adult Americans every year. The data is freely available. In this small subset of the data, I would like to predict a person's occupation category (`occ`) based on race (`white`), years of education (`ed`), and years of experience (`exper`).[3]

The variables

Before getting started, let us examine the variables involved. 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 what his or her race was. Looking at a frequency count, a full 91.69% of the respondents stated they were white (and 8.31% stated they were non-white). This is significantly higher than the population at large, where approximately 80% of Americans are white. When we do the final analysis, we need to keep this in mind, as it is not representative of the nation as a whole.

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

[2] 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 balance the strengths with the weaknesses to get closer to the true process you are trying to model.

[3] The data is contained in the file `gssocc.csv`.

	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 7.1: *Correlation matrix for the three independent variables in the example.*

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 (mean is larger than the median). Furthermore, it is interesting to note that the first quartile is also 12 years. This indicates that at least 25% of the sample only graduated from high school. Digging a little, we find that 32.3% 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 assumption that the sample represents the population.[4] 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 unpleasantries and surprises due to collinearity and multicollinearity. The correlation matrix (Table 7.1) does not show any hint of multicollinearity. In fact, this correlation matrix suggests that these three variables are effectively independent of each other.

Finally, let us look at the dependent variable. The occupation variable is definitely categorical. There are five possible job types (levels): Blue Collar, Craft, Menial, Professional, and White Collar. Of those five types, 20.5% are Blue Collar, 24.9% are Craft, 9.2% are Menial, 33.2% are Professional,

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

	Estimate	Std. Error	z value	Pr(>|z|)
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 7.2: *Results from the GLM (family=binomial, link=logit) predicting whether or not a person is a blue collar worker. The AIC for this model is 304.75.*

and 12.2% are White Collar. I am not aware of any independent source to determine how close these proportions are to the population proportions. As such, I am not entirely certain that results from this analysis are generalizable to the United States.[5]

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 a candidate for nominal regression.

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 x-variables? 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 is asking 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 **??**, the chapter on Binary Dependent Variables. Here, the dependent variable is 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
```

[5]With that said, there is no need to belabor the point. Merely stating the warning is sufficient unless you know the sample is not reflective of the population.

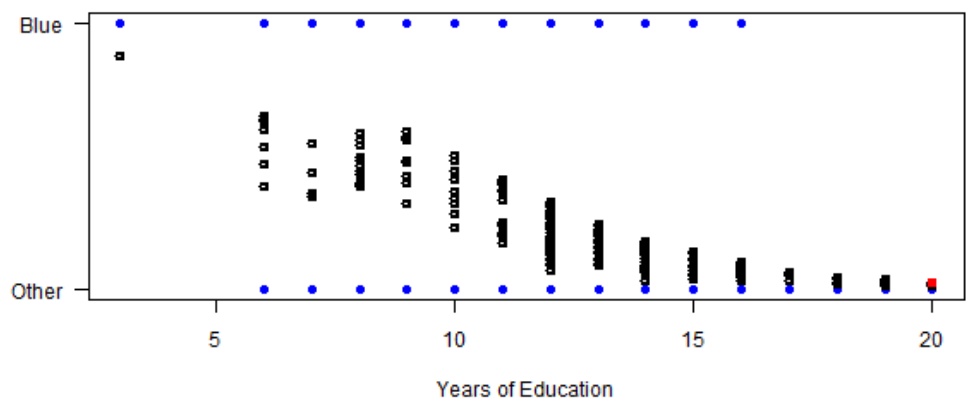

Figure 7.1: *A scatter plot of the actual Blue Collar values (solid) and the predicted probabilities of a person being a Blue Collar workers. The several predictions for each year of education is due to the two other variables. Crisp graphs only happen when the plot is bivariate. The red square represents Bob.*

We would fit it using a generalized linear model, a binomial family, and a logit link. The results of the regression are in Table 7.2. From this model, we can perform all of the goodness of fit measures from Chapter **??**.

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 (Also see Figure 7.1). 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). 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, the probability that Bob is a professional is 96.9%; menial, 2.3%; and craft, 7.9%.

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%. The problem is that we kept changing the base category. In Chapter **??**, we never mentioned the need

Coefficients:				
	(Intercept)	white	ed	exper
Craft	-1.8327912	-0.7641613	0.1932585	0.022963210
Menial	-0.7412468	-1.2365229	0.0994269	-0.007420163
Prof	-12.2594994	0.5376393	0.8782843	0.030930582
WhiteCol	-6.9799610	0.3348673	0.4525929	0.029876079
Std. Errors:				
	(Intercept)	white	ed	exper
Craft	1.186124	0.6324290	0.07749832	0.01255248
Menial	1.519538	0.1996015	0.10228157	0.01739840
Prof	1.668139	0.7996015	0.10054512	0.01440869
WhiteCol	1.714398	0.9340187	0.10226817	0.01529265

Table 7.3: *Results of the multinomial regression.*

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'. This ensured that the probabilities always added up to 100%. In 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 right way

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.

Since we can select any job type as our base, let us select Blue Collar, since it is first in our dataset, and since that is the default base for most programs. We will see shortly how to switch between the bases.

To perform this modeling, you will have to load the `nnet` library, which comes with your base distribution of `R`. Once loaded with the `library(nnet)` command, to fit the 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 7.3 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 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):

BlueCol	Craft	Menial	Prof	WhiteCol
0.002034233	0.009097206	0.001961453	0.956470566	0.030436542

Base switching I am not sure why you would ever need to change bases, since the computer does the predictions for you. However, I will put it here, in case someone can give me a good reason. To change among the bases, you will 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 $-6.9799610 - -12.259499 = 5.279538$.

Unfortunately, the standard errors are not so easily calculated—or at all calculable by hand.

Since the base is irrelevant, it does not matter in the predictions what the base actually is. Thus, allowing the statistical program to select the base makes sense.

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. To remember this, just look at the output—it is in proportions already (a quick check is that they sum to one).

The first check of the goodness of the model is the relative accuracy. The accuracy is the number of correct predictions divided by the number of cases. The relative frequency divides this number by the accuracy of always selecting the modal category. For this dataset, the modal category is Professional, with 112 out of 337 cases belonging to Professionals. Thus, the relative accuracy is $\frac{169}{337}/\frac{112}{337} = 1.509$. Thus, this model improves accuracy by 50% over the null model. Is this good? Well, it depends on your other models.

The Akaike Information Criteria score is also reported. For this model, $AIC = 885$. Is this good? It depends on the other models available. In other words, model comparison needs another model. We can compare it to the null model, which has an $AIC = 1027$. 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 $\div$ standard error > 2).[6]

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 nice to see that our conclusions do seem to reflect reality.

7.2 Ordinal dependent variable

Another choice for 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.

[6] This rule of thumb comes from the fact that in a Normal distribution, the ration 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.

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, but allows us to fit it with 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 quasi-ordinary least squares, 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. Assume that the ordinal values are essentially 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.

7.2.1 Option Three

Let us assume that there is an underlying continuous process. We only experience 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. Here, there is more than just the one threshold (which defaulted to $\lambda = 0.500$). Thus, we have two sets of parameters to fit. The first is the parameters which describe the process. The second is the position of those threshold values. Without going into the details, we will use Maximum Likelihood as our fitting method because it has many nice properties and because we can fit all of our parameters at once.

Thus, our underlying process is

$$\tilde{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Our thresholding process is shown in Figure 7.2. 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 that separate the observed ordinal values.

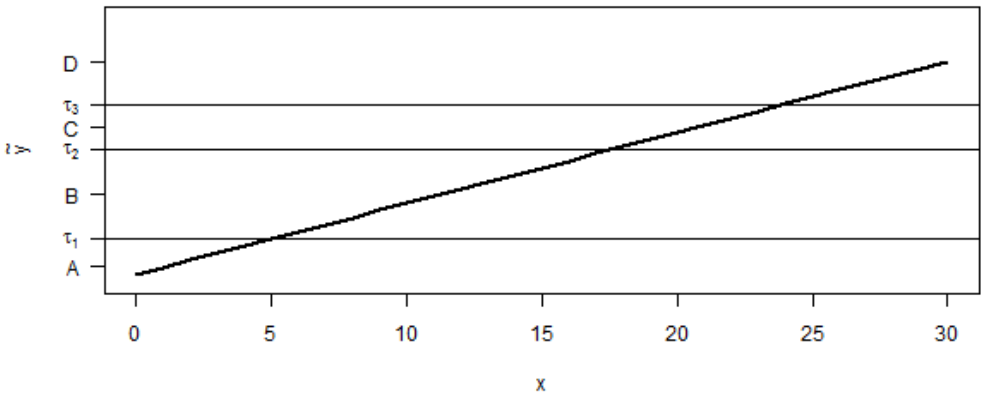

Figure 7.2: *Schematic diagram of the thresholding process. The line represents the linear continuous process. The τs represent the threshold values. A, B, C, and D represent the ordinal outcomes.*

This model is very straight forward and understandable. The fitting is also straight forward. The results are also straight forward.

Example 7.1. *Let us use some more data from the GSS. This data explores the 'warmth of feeling' the respondent has for the president. 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.*

In this section, I will give a description of the variables as I did in the nominal case.

Now, let us fit this data with ordinal regression. The command in `R` is `polr()`, which requires the `MASS` library. The actual command I use 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 (see Table 7.4).

Coefficients:			
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
Intercepts:			
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 7.4: *Result of ordinal regression in* R.

From Table 7.4, we see that the equation for the underlying linear process is

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

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, $\tilde{y}$, 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

$$\tilde{y} = 0.740 \times 0 + -0.400 \times 1 + -0.020 \times 40 + 0.098 \times 20 = 0.76$$

As $\tilde{y} = 0.76$, we have our prediction that Bob approves of the president. If we actually want probabilities the Bob Strongly Disagrees, Disagrees, Agrees, or Strongly Agrees, we would have to back-transform using the inverse of the logit function. Or, we could just ask the computer to do it for us:

```
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 supports the president (although he probably does).

Accuracy

Finally, let us look at the accuracy of the model. The relative accuracy is 1.105, which indicates that the model is about 10% better than the null model (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 highly educated and the lesser educated both support the president. If we do this, we find that there is no evidence of this. Thus, we can conclude that the relationship between education and presidential support is linear.

⋆ ⋆ ⋆

Thus, there should be something I need to add, but everything is really just this straight forward. In the next draft, I will work on a little conclusion here.

7.3 Conclusion

We are at the end of another chapter that is not entirely finished. This one, however, is a bit better than the ANOVA chapter (which never gets to ANOVA).

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.

7.4 Libraries used

- nnet
- MASS

7.5 R functions

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` library.

7.6 Extensions

Coming soon?

7.7 R scripts

The script for the nominal section:

```
###
# Get to know the data
library(xtable)

data <- read.csv("gssocc.csv", header=TRUE)
names(data)
attach(data)

summary(white)

summary(ed)
length(which(ed==12))/length(ed)
length(which(ed>=16))/length(ed)
length(which(ed<12))/length(ed)

summary(exper)
```

```
cor(data)
cor( cbind(white, ed, exper) ) # this is how you would do it with more than one variable

summary(data$occ)/length(data$occ)



# Let us define Bob, to make things a bit easier
BOB <- data.frame(ed=20, exper=10, white=1)



# The obvious, yet wrong, way
bluecol <- 1*(occ=="BlueCol") + 0
model.1a <- glm(bluecol ~ white + ed + exper, binomial(link=logit) )
summary(model.1a)
p.1a <- predict(model.1a, newdata=BOB )
pr.1a <- 1 / (1 + exp( -p.1a))


# Graph our results:

#png("blue-educ.png", width=600, height=300)
plot(ed,bluecol, pch=16, las=1, xlab="Years of Education", yaxt="n", ylab="", col='blue')
axis(2, label=c("Blue", "Other"), at=c(1,0), las=1 )
points(ed, 1/(1+exp(-predict(model.1a))), lwd=2)
points(20,pr.1a, pch=15, lwd=2, col='red')
#dev.off()


whitecol <- 1*(occ=="WhiteCol") + 0
model.1b <- glm(whitecol ~ white + ed + exper, binomial(link=logit) )
summary(model.1b)
p.1b <- predict(model.1b, newdata=BOB )
pr.1b <- 1 / (1 + exp( -p.1b))

prof <- 1*(occ=="Prof") + 0
model.1c <- glm(prof ~ white + ed + exper, binomial(link=logit) )
summary(model.1c)
p.1c <- predict(model.1c, newdata=BOB )
pr.1c <- 1 / (1 + exp( -p.1c))

meni <- 1*(occ=="Menial") + 0
model.1d <- glm(meni ~ white + ed + exper, binomial(link=logit) )
summary(model.1d)
p.1d <- predict(model.1d, newdata=BOB )
pr.1d <- 1 / (1 + exp( -p.1d))

craft <- 1*(occ=="Craft") + 0
model.1e <- glm(craft ~ white + ed + exper, binomial(link=logit) )
summary(model.1e)
p.1e <- predict(model.1e, newdata=BOB )
pr.1e <- 1 / (1 + exp( -p.1e))
```

```
#####
# The right way:
# Select a base and measure all probabilities off that base.

pr.1a/pr.1c
pr.1b/pr.1c
log( pr.1d/pr.1c )
pr.1e/pr.1c

11.51833 -1.774306*1 -0.7788519*20 -0.0356509*10

library(nnet)

model.mn1 <- multinom(occ ~ white + ed + exper)
summary(model.mn1)
predict(model.mn1, newdata=BOB, type="class")
predict(model.mn1, newdata=BOB, type="probs")

sum( predict(model.mn1) == occ )/length(occ)

summary(occ)

model.mn.null <- multinom(occ ~ 1)
summary(model.mn.null)
```

The script for the ordinal section:

```
### Ordinal

library(MASS)

detach(data)
data <- read.csv("ordwarm.csv", header=TRUE)
names(data)
data$warm <- factor(data$warm, levels=c("SD", "D", "A", "SA") )
attach(data)

model.ol1 <- polr(warm ~ male + white + age + ed )
summary(model.ol1)

sum( predict(model.ol1)==warm )
summary(warm)
946/856

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

model.ol2 <- polr(warm ~ male + white + age + I(ed^2) + ed )
summary(model.ol2)
```