MA4128 2016

Overview of Logistic Regression Models

- Binary logistic regression, also called a logit model, is used to model **dichotomous outcome** variables. In the logit model the log odds of the outcome is modeled as a linear combination of the predictor variables.

- Multinomial Logistic Regression is used to model nominal outcome variables (categorical variables). Again the log odds of the outcomes are modeled as a linear combination of the predictor variables.
- Ordinal Logistic Regression is used to model nominal outcome variables, where a hierarchy within categories exists.

Multinomial Logistic Regression

- Multinomial Logistic Regression is a classification method that generalizes logistic regression to **multiclass problems**, i.e. with more than two possible discrete outcomes.
- That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables
- These predictor variables may be real-valued, binary-valued, categorical-valued, etc.

Multinomial Logistic Regression

- Multinomial logistic regression is used to model nominal outcome variables, in which the log odds of the outcomes are modeled as a linear combination of the predictor variables.

Multinomial Regression with R

The main package we will use is the **nnet** package. We will also use the **ggplot2** and **reshape2** package.

```
install.packages("nnet")
library(nnet)

library(ggplot2)
library(reshape2)
```

nnet

- Title : Feed-forward Neural Networks and Multinomial Log-Linear Models
- Description : Software for feed-forward neural networks with a single hidden layer, and for multinomial log-linear models.
- Authors : Brian Ripley ,William Venables
- URL : http://www.stats.ox.ac.uk/pub/MASS4/

Applications

- Which major will a college student choose, given their grades, stated likes and dislikes, etc.?
- Which blood type does a person have, given the results of various diagnostic tests?
- Which candidate will a person vote for, given particular demographic characteristics?
- Which country will a firm locate an office in, given the characteristics of the firm and of the various candidate countries?

Examples of Multinomial Logistic Regression

Example 1. Entering high school students make program choices among general program, vocational program and academic program.
Their choice might be modeled using their writing score and their social economic status.

Examples of Multinomial Logistic Regression

Example 2. People's occupational choices might be influenced by their parents' occupations and their own education level.
We can study the relationship of one's occupation choice with education level and father's occupation.
The occupational choices will be the outcome variable which consists of categories of occupations.

Examples of multinomial logistic regression

Example 3. A biologist may be interested in food choices that alligators make. Adult alligators might have different preferences from young ones.
The outcome variable here will be the types of food, and the predictor variables might be size of the alligators and other environmental variables.

Description of the Data

- The outcome variable is **prog**, the program type.
- The two predictor variables are

1. social economic status, **ses**, (a three-level categorical variable)
2. writing score, **write**, (a continuous variable).

- The data set contains variables on 200 students.

Multinomial Regression with R

```
table(ml$ses, ml$prog)

        general academic vocation
low          16       19       12
middle       20       44       31
high          9       42        7
```

	M	SD
general	51.33	9.398
academic	56.26	7.943
vocation	46.76	9.319

Multinomial Logistic Regression

- We use the `multinom` function from the **nnet** package to estimate a multinomial logistic regression model.
- Remark - There are other functions in other R packages capable of multinomial regression, such as the **mlogit** package.
- The `multinom` function does not require the data to be reshaped (as the **mlogit** package does)
- (Similar format to example code found in Hilbe's *Logistic Regression Models*).

Multinomial logistic regression

- We must choose the level of our outcome that we wish to use as our **baseline** and specify this in the `relevel` function.
  Let's choose "academic".
- Then, we run our model using *multinom*.
- The `multinom` command does not include p-value calculation for the regression coefficients.
- (We can calculate p-values using Wald tests or z-tests).

Multinomial Logistic Regression with R

```
ml$prog2 <- relevel(ml$prog, ref = "academic")
test <- multinom(prog2 ~ ses + write, data = ml)

 # weights:  15 (8 variable)
 initial  value 219.722458
 iter  10 value 179.982880
 final    value 179.981726
 converged
```

Multinomial Logistic Regression with R

```
summary(test)

 Call:
 multinom(formula = prog2 ~ ses + write,
             data = ml)

 Coefficients:
          (Intercept) sesmiddle seshigh     write
 general        2.852   -0.5333 -1.1628 -0.05793
 vocation       5.218    0.2914 -0.9827 -0.11360
```

Multinomial Logistic Regression with R

```
....
 Std. Errors:
          (Intercept) sesmiddle seshigh   write
 general        1.166    0.4437  0.5142 0.02141
 vocation       1.164    0.4764  0.5956 0.02222

 Residual Deviance: 360
 AIC: 376
```

Wald Test

- The Wald test in the context of logistic regression is used to determine whether a certain predictor variable X is significant or not.
- It rejects the null hypothesis of the corresponding coefficient being zero.
- The test consists of dividing the value of the coefficient by standard error

Multinomial Logistic Regression with R

```
# Coefficients Divided by Standard Errors
# Then Compute p-values.
# 2-tailed z test
# p.values

          (Intercept) sesmiddle seshigh      write
general     1.448e-02    0.2294 0.02374 6.819e-03
vocation    7.299e-06    0.5408 0.09895 3.176e-07
```

Multinomial Regression with R

- Remark: Some output is generated by running the model, even though we are assigning the model to a new R object.
- This model-running output includes some iteration history and includes the final **negative log-likelihood** (+ 179.981726).
- This value multiplied by two is then seen in the model summary as the **Residual Deviance** and it can be used in comparisons of nested models (360).

Multinomial Regression with R

- As with many summary outputs, the output contains a column of coefficients and a column of standard errors.
- Each of these blocks has one row of values corresponding to a model equation.
- Focusing on the block of coefficients, we can look at the first row comparing `prog = "general"` to our baseline `prog = "academic"` and the second row comparing `prog = "vocation"` to our baseline `prog = "academic"`.

Multinomial Regression with R

- If we consider our coefficients from the first row to be b_1 and our coefficients from the second row to be b_2, we can write our model equations:

$$ln\left(\frac{P(prog = gen.)}{P(prog = acad.)}\right) = b_{10} + b_{11}(ses = 2) + b_{12}(ses = 3) + b_{13}write$$

$$ln\left(\frac{P(prog = voc.)}{P(prog = acad.)}\right) = b_{20} + b_{21}(ses = 2) + b_{22}(ses = 3) + b_{23}write$$

Multinomial Regression with R

- A one-unit increase in the variable **write** is associated with the decrease in the log odds of being in general program vs. academic program in the amount of *0.058* (b_{13}).
- A one-unit increase in the variable **write** is associated with the decrease in the log odds of being in vocation program vs. academic program in the amount of *0.1136* (b_{23}).

Multinomial Regression with R

- The log odds of being in general program vs. in academic program will decrease by 1.163 if moving from *ses="low"* to *ses="high"* b_{12}.
- The log odds of being in general program vs. in academic program will decrease by 0.533 if moving from *ses="low"* to *ses="middle"* b_{11}, although this coefficient is not significant.

Multinomial Regression with R

- The log odds of being in vocation program vs. in academic program will decrease by 0.983 if moving from ses="low" to ses="high"(b_{22}).
- The log odds of being in vocation program vs. in academic program will increase by 0.291 if moving from ses="low" to ses="middle"(b_{21}), although this coefficient is not signficant.

Odds Ratios

- The ratio of the probability of choosing one outcome category over the probability of choosing the baseline category is often referred as **relative risk**
- It is also sometimes referred as **odds**.

Multinomial Logistic Regression with R

- The relative risk is the (right-hand side) linear equation exponentiated, leading to the fact that the exponentiated regression coefficients are relative risk ratios for a unit change in the predictor variable.
- We can exponentiate the coefficients from our model to see these odds ratios (next slide).

Multinomial Logistic Regression with R

Extract the coefficients from the model then and exponentiate

```
exp(coef(test))

          (Intercept) sesmiddle seshigh  write
 general        17.33    0.5867  0.3126 0.9437
 vocation      184.61    1.3383  0.3743 0.8926
```

Multinomial Logistic Regression with R

- The relative risk ratio for a one-unit increase in the variable **write** is 0.9437 for being in general program vs. academic program.
- The relative risk ratio switching from **ses** $= 1$ to 3 is 0.3126 for being in general program vs. academic program.

Multinomial Regression with R

- You can also use predicted probabilities to help you understand the model.
- You can calculate predicted probabilities for each of our outcome levels using the `fitted` function.
- We can start by generating the predicted probabilities for the observations in our dataset and viewing the first few rows

Multinomial Regression with R

```
head(pp <- fitted(test))
```

	academic	general	vocation
1	0.1483	0.3382	0.5135
2	0.1202	0.1806	0.6992
3	0.4187	0.2368	0.3445
4	0.1727	0.3508	0.4765
5	0.1001	0.1689	0.7309
6	0.3534	0.2378	0.4088

Prediction

- Suppose we want to examine the changes in predicted probability associated with one of our two variables, we can create small datasets varying one variable while holding the other constant.
- We will first do this holding **write** at its mean and examining the predicted probabilities for each level of **ses**.
- *(i.e. Three Cases to predict for)*

Multinomial Logistic Regression with R

```
dses <- data.frame(ses = c("low", "middle", "h
                   write = mean(ml$write))
predict(test, newdata = dses, "probs")
```

```
   academic general vocation
1    0.4397  0.3582   0.2021
2    0.4777  0.2283   0.2939
3    0.7009  0.1785   0.1206
```

Multinomial Logistic Regression with R

- Another way to understand the model using the predicted probabilities is to look at the averaged predicted probabilities for different values of the continuous predictor variable **write** within each level of **ses**.

Multinomial Logistic Regression with R

Store the predicted probabilities for each value of ses and write

```
dwrite <- data.frame(
  ses = rep(c("low", "middle", "high"), each = 41),
  write = rep(c(30:70), 3))
```

```
pp.write <- cbind(dwrite,
     predict(test, newdata = dwrite,
     type = "probs", se = TRUE))
```

Multinomial Logistic Regression with R

Calculate the mean probabilities within each level of ses

```
by(pp.write[, 3:5], pp.write$ses, colMeans)
```

```
pp.write$ses: high
academic  general vocation
  0.6164   0.1808   0.2028
-------------------------------------
pp.write$ses: low
academic  general vocation
  0.3973   0.3278   0.2749
-------------------------------------
pp.write$ses: middle
academic  general vocation
  0.4256   0.2011   0.3733
```

Ordered logistic regression with R

Ordered logit model

- The Ordered (or Ordinal) logit model (also ordered logistic regression or proportional odds model), is a regression model for ordinal dependent variables.
- For example, questions on a survey answered by a choice among "poor", "fair", "good", "very good", and "excellent".
- The purpose of the analysis is to see how well that response can be predicted by the responses to other questions, some of which may be quantitative

Ordered logit model

- It can be thought of as an extension of the logistic regression model that applies to dichotomous dependent variables, allowing for more than two (ordered) response categories.
- The model only applies to data that meet the proportional odds assumption

Ordered logistic regression with `R`

`polr`

- In this section we will use the `polr` command (from the MASS package) to estimate an ordered logistic regression model.
- The command name comes from **proportional odds logistic regression**, due to the the **proportional odds assumption** in the model.

Ordered logistic regression with `R`

`polr`

- `polr` uses the standard formula interface in `R` for specifying a regression model with outcome followed by predictors.
- We will also specify `Hess=TRUE` to have the model return the observed information matrix from optimization (called the Hessian) which is used to get standard errors.

Ordered logistic regression with R

```
## fit ordered logit model and store results 'm'
m <- polr(apply ~ pared +
          public + gpa, data = dat, Hess=TRUE)

## view a summary of the model
summary(m)
## Call:
## polr(formula = apply ~ pared +
             public + gpa, data = dat,
             Hess = TRUE)
```

Ordered logistic regression with R

```
Coefficients:
         Value Std. Error t value
pared   1.0477      0.266   3.942
public -0.0588      0.298  -0.197
gpa     0.6159      0.261   2.363

Intercepts:
                            Value  Std. Error t value
unlikely|somewhat likely     2.204  0.780      2.827
somewhat likely|very likely  4.299  0.804      5.345
```

Ordered logistic regression with R

1. The “Call”, what type of model we ran, what options we specified, etc.
2. The usual regression output coefficient table including the value of each coefficient, standard errors, and $t-$value, which is simply the ratio of the coefficient to its standard error.
   (Remark: There is no significance test by default.)

Ordered logistic regression with R

3. We then have the estimates for the two intercepts (which are sometimes called **cutpoints**).
4. The intercepts indicate where the latent variable is cut to make the three groups that we observe in our data.

Ordered logistic regression with R

In the ordered logit model, there is an observed ordinal variable, Y. Y, in turn, is a function of another latent variable, Y*, that is not measured.

a. In the ordered logit model, there is a continuous, unmeasured latent variable Y*, whose values determine what the observed ordinal variable Y equals.
b. The continuous latent variable Y* has various threshold (or cutoff) points.

Ordered logistic regression with R

Your value on the observed ordinal variable Y depends on whether or not you have crossed a particular threshold. For example, when $M = 3$

- $Yi = 1$ if Y^*i is $\leq CP1$
- $Yi = 2$ if $CP1 \leq Y^*i \leq CP2$
- $Yi = 3$ id $Y^*i \geq CP2$

Ordered logistic regression with R

- Note that this latent variable is continuous. In general, these are not used in the interpretation of the results.
- The cutpoints are closely related to thresholds, which are reported by other statistical packages.

Model Diagnostics

- We see the residual deviance, -2 * Log Likelihood of the model as well as the AIC.
- Both the deviance and AIC are useful for model comparison.
- Of, course, some people are not satisfied without a $p-$value.
- One way to calculate a $p-$value in this case is by comparing the $t-$value against the standard normal distribution, like a $z-$test.

Ordered logistic regression R

- Of course this is only true with infinite degrees of freedom, but is reasonably approximated by large samples, becoming increasingly biased as sample size decreases.
- First we store the coefficient table, then calculate the $p-$values and combine back with the table.

Ordered logistic regression R

```
# store table
(ctable <- coef(summary(m)))
                              Value Std. Error t value
 pared                      1.04769     0.2658  3.9418
 public                    -0.05879     0.2979 -0.1974
 gpa                        0.61594     0.2606  2.3632
 unlikely|somewhat likely   2.20391     0.7795  2.8272
 somewhat likely|very likely 4.29936    0.8043  5.3453
```

Ordered Logistic regression R

```
# calculate and store p values
p <- pnorm(abs(ctable[, "t value"]),
      lower.tail = FALSE) * 2

# Combined table
(ctable <- cbind(ctable, "p value" = p))
                  Value Std. Error t value   p value
 pared          1.04769     0.2658  3.9418 8.087e-05
 public        -0.05879     0.2979 -0.1974 8.435e-01
 gpa            0.61594     0.2606  2.3632 1.812e-02
 unli..|some..  2.20391     0.7795  2.8272 4.696e-03
 some..|very..  4.29936     0.8043  5.3453 9.027e-08
```

Ordered Logistic Regression

Examples of ordered logistic regression

- A marketing research firm wants to investigate what factors influence the size of soda (small, medium, large or extra large) that people order at a fast-food chain.
- These factors may include what type of sandwich is ordered (burger or chicken), whether or not fries are also ordered, and age of the consumer.
- While the outcome variable, size of soda, is obviously ordered, the difference between the various sizes is not consistent.
- The difference between small and medium is 10 ounces, between medium and large 8, and between large and extra large 12.

Ordered Logistic Regression

Examples of ordered logistic regression

- A researcher is interested in what factors influence medaling in Olympic swimming.
- Relevant predictors include at training hours, diet, age, and popularity of swimming in the athlete's home country.
- The researcher believes that the distance between gold and silver is larger than the distance between silver and bronze.

Ordered Logistic Regression

- A study looks at factors that influence the decision of whether to apply to graduate school.
- College juniors are asked if they are unlikely, somewhat likely, or very likely to apply to graduate school.
- Hence, our outcome variable has three categories. Data on parental educational status, whether the undergraduate institution is public or private, and current GPA is also collected.
- The researchers have reason to believe that the "distances" between these three points are not equal.
- For example, the "distance" between *"unlikely"* and *"somewhat likely"* may be shorter than the distance between *"somewhat likely"* and *"very likely"*.

Ordered Logistic Regression

Data Set - Graduate School Entry (ologit.csv)

- This hypothetical data set has a three level variable called **apply**, with levels *"unlikely"*, *"somewhat likely"*, and *"very likely"*, coded 1, 2, and 3, respectively, that we will use as our outcome variable.

Ordered Logistic Regression

Predictors:

- pared , which is a 0/1 variable indicating whether at least one parent has a graduate degree;
- public , which is a 0/1 variable where 1 indicates that the undergraduate institution is public and 0 private,
- gpa , which is the student's grade point average.

Ordered Logistic Regression

```
            apply pared public  gpa
1     very likely     0      0 3.26
2 somewhat likely     1      0 3.21
3        unlikely     1      1 3.94
4 somewhat likely     0      0 2.81
5 somewhat likely     0      0 2.53
6        unlikely     0      1 2.59
```

Confidence Intervals

- We can also get confidence intervals for the parameter estimates.
- These can be obtained either by profiling the likelihood function or by using the standard errors and assuming a normal distribution.
- Note that profiled CIs are not symmetric (although they are usually close to symmetric).
- If the 95% CI does not cross 0, the parameter estimate is statistically significant.

Ordered Logistic Regression with R

```
(ci <- confint(m))
# default method gives profiled CIs

 Waiting for profiling to be done...
           2.5 %   97.5 %
 pared   0.5282   1.5722
 public -0.6522   0.5191
 gpa     0.1076   1.1309
```

Ordered Logistic Regression with R

```
# CIs assuming normality

confint.default(m)

          2.5 %   97.5 %
 pared   0.5268    1.569
 public -0.6426    0.525
 gpa     0.1051    1.127
```

Confidence Intervals

- The CIs for both pared and gpa do not include 0; but the CI for public does.
- The estimates in the output are given in units of ordered logits, or ordered log odds.
- For pared, we would say that for a one unit increase in pared (i.e., going from 0 to 1), we expect a 1.05 increase in the expect value of apply on the log odds scale, given all of the other variables in the model are held constant.

Confidence Intervals

- For gpa, we would say that for a one unit increase in gpa, we would expect a 0.62 increase in the expected value of apply in the log odds scale, given that all of the other variables in the model are held constant.

Ordered Logistic Regression with R

- The coefficients from the model can be somewhat difficult to interpret because they are scaled in terms of logs.
- Another way to interpret logistic regression models is to convert the coefficients into odds ratios.
- To get the Odds Ratios and confidence intervals, we just exponentiate the estimates and confidence intervals.

Ordered Logistic Regression with R

Odds Ratios

```
exp(coef(m))
  pared public    gpa
 2.8511 0.9429 1.8514
```

```
 # Odds Ratios and CIs
exp(cbind(OR = coef(m), ci))
            OR  2.5 % 97.5 %
 pared  2.8511 1.6958  4.817
 public 0.9429 0.5209  1.681
 gpa    1.8514 1.1136  3.098
```

Ordered Logistic Regression with R

- These coefficients are called **proportional odds ratios** and we would interpret these pretty much as we would odds ratios from a binary logistic regression.
- For pared, we would say that for a one unit increase in parental education, i.e., going from 0 (*Low*) to 1 (*High*), the odds of "*very likely*" applying versus "*somewhat likely*" or "*unlikely*" applying combined are 2.85 greater, given that all of the other variables in the model are held constant.

Ordered Logistic Regression with R

- Similarly, the odds "*very likely*" or "*somewhat likely*" applying versus "*unlikely*" applying is 2.85 times greater, given that all of the other variables in the model are held constant.
- For gpa (and other continuous variables), the interpretation is that when a student's gpa moves 1 unit, the odds of moving from "*unlikely*" applying to "*somewhat likely*" or "*very likely*" applying (or from the lower and middle categories to the high category) are multiplied by 1.85.

Ordinal Logistic Regression with R

Assumption of Proportional Odds

- One of the assumptions underlying ordinal logistic regression is that the relationship between each pair of outcome groups is the same.
- In other words, ordinal logistic regression assumes that the coefficients that describe the relationship between, say, the lowest versus all higher categories of the response variable are the same as those that describe the relationship between the next lowest category and all higher categories, etc.

Testing the Assumption

- Because the relationship between all pairs of groups is the same, there is only one set of coefficients.
- If this was not the case, we would need different sets of coefficients in the model to describe the relationship between each pair of outcome groups.
- Thus, in order to asses the appropriateness of our model, we need to evaluate whether the proportional odds assumption is tenable.

Testing the Assumption

- Statistical tests to do this are available in some software packages.
- However, these tests have been criticized for having a tendency to reject the null hypothesis (that the sets of coefficients are the same), and hence, indicate that there the parallel slopes assumption does not hold, in cases where the assumption does hold *(Harrell 2001 p. 335)*.
- Currently R to perform any of the tests commonly used to test the parallel slopes assumption.

Ordinal Logistic Regression with R

łarge

- Harrell does recommend a graphical method for assessing the parallel slopes assumption.
- The values displayed in this graph are essentially (linear) predictions from a logit model, used to model the probability that y is greater than or equal to a given value (for each level of y), using one predictor (x) variable at a time.

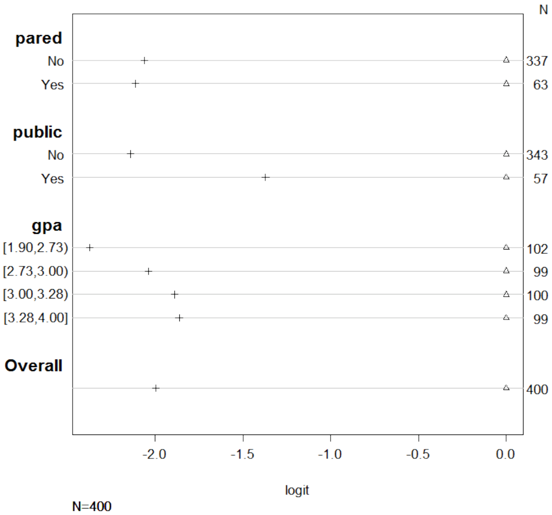

N
pared
No
Yes
337
63
public
No
Yes
343
57
gpa
[1.90,2.73)
[2.73,3.00)
[3.00,3.28)
[3.28,4.00]
102
99
100
99
Overall
400
-2.0
-1.5
-1.0
-0.5
0.0
logit
N=400

Figure:

Ordered Logistic regression R

- Turning our attention to the predictions with public as a predictor variable, we see that when public is set to "*No*" the difference in predictions for apply greater than or equal to two, versus apply greater than or equal to three is about 2.14 $(-0.204 - (-2.345) = 2.141)$.
- When public is set to "yes" the difference between the coefficients is about 1.37. $(-0.175 - (-1.547) = 1.372)$.

Ordered Logistic regression R

- The differences in the distance between the two sets of coefficients (2.14 vs. 1.37) may suggest that the parallel slopes assumption does not hold for the predictor public.
- That would indicate that the effect of attending a public versus private school is different for the transition from "*unlikely*" to "*somewhat likely*" and "*somewhat likely*" to "*very likely*."

Ordered logistic regression R

- If the proportional odds assumption holds, for each predictor variable, distance between the symbols for each set of categories of the dependent variable, should remain similar.
- To help demonstrate this, we normalized all the first set of coefficients to be zero so there is a common reference point.

Ordered logistic regression R

- Looking at the coefficients for the variable pared we see that the distance between the two sets of coefficients is similar.
- In contrast, the distances between the estimates for public are different (i.e. the markers are much further apart on the second line than on the first), suggesting that the proportional odds assumption may not hold.

Ordered logistic regression R

```
plot(s, which=1:3, pch=1:3,
    xlab='logit', main=' ',
    xlim=range(s[,3:4]))
```

Ordered logistic regression R

- Once we are done assessing whether the assumptions of our model hold, we can obtain predicted probabilities, which are usually easier to understand than either the coefficients or the odds ratios.
- For example, we can vary gpa for each level of pared and public and calculate the probability of being in each category of apply.
- We do this by creating a new dataset of all the values to use for prediction.

Ordered logistic regression R

```
newdat <- data.frame(
  pared = rep(0:1, 200),
  public = rep(0:1, each = 200),
  gpa = rep(seq(from = 1.9, to = 4,
      length.out = 100), 4))

newdat <- cbind(newdat, predict(m,
   newdat, type = "probs"))
```

Ordered logistic regression R

```
# Show first few rows
head(newdat)
  pared public   gpa unlikely somewhat likely very likely
1     0      0 1.900   0.7376          0.2205     0.04192
2     1      0 1.921   0.4932          0.3946     0.11221
3     0      0 1.942   0.7325          0.2245     0.04299
4     1      0 1.964   0.4867          0.3985     0.11484
5     0      0 1.985   0.7274          0.2285     0.04407
6     1      0 2.006   0.4802          0.4023     0.11753
```

Ordered logistic regression R

- Now we can reshape the data long with the reshape2 package and plot all of the predicted probabilities for the different conditions.
- We plot the predicted probilities, connected with a line, coloured by level of the outcome, apply, and facetted by level of pared and public.
- We also use a custom label function, to add clearer labels showing what each column and row of the plot represent.

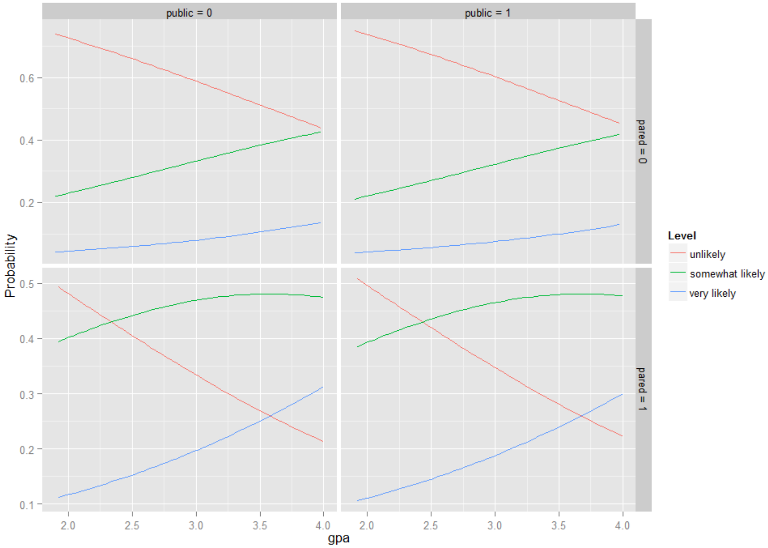
Ordered logistic regression R

```
library(reshape2)

lnewdat <- melt(newdat,
  id.vars = c("pared", "public", "gpa"),
  variable.name = "Level",
  value.name="Probability")
```

Ordered logistic regression R

```
%## view first few rows
%head(lnewdat)
%##   pared public   gpa    Level Probability
%## 1     0      0 1.900 unlikely      0.7376
%## 2     1      0 1.921 unlikely      0.4932
%## 3     0      0 1.942 unlikely      0.7325
%## 4     1      0 1.964 unlikely      0.4867
%## 5     0      0 1.985 unlikely      0.7274
%## 6     1      0 2.006 unlikely      0.4802
```


public = 0
public = 1
pared = 0
pared = 1
Probability
0.6
0.4
0.2
0.5
0.4
0.3
0.2
0.1
2.0
2.5
3.0
3.5
4.0
gpa
Level
unlikely
somewhat likely
very likely

Things to consider

Perfect prediction: Perfect prediction means that one value of a predictor variable is associated with only one value of the response variable.

Pseudo-R-squared:

There is no exact analog of the R-squared found in OLS. There are many versions of pseudo-R-squares.

Diagnostics:

Doing diagnostics for non-linear models is difficult, and ordered logit/probit models are even more difficult than binary models.

Ordered logistic regression R

Sample size:
Both ordered logistic and ordered probit, using maximum likelihood estimates, require sufficient sample size.

Empty cells or small cells:
You should check for empty or small cells by doing a crosstab between categorical predictors and the outcome variable. If a cell has very few cases, the model may become unstable or it might not run at all.

Multinomial Logistic Regression with R

```
## melt data set to long for ggplot2
lpp <- melt(pp.write, id.vars = c("ses", "write"), value.
head(lpp)  # view first few rows

##   ses write variable probability
## 1 low    30 academic     0.09844
## 2 low    31 academic     0.10717
## 3 low    32 academic     0.11650
## 4 low    33 academic     0.12646
## 5 low    34 academic     0.13705
## 6 low    35 academic     0.14828
```

Multinomial Logistic Regression with R

```
## plot predicted probabilities across write values for e
## facetted by program type
ggplot(lpp, aes(x = write, y = probability, colour = ses)
    ., scales = "free")
```

Ordinal Logistic Regression with R

```
ggplot(dat, aes(x = apply, y = gpa)) +
  geom_boxplot(size = .75) +
  geom_jitter(alpha = .5) +
  facet_grid(pared ~ public, margins = TRUE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1,
```

Poisson Regression with R

```
with(p, tapply(num_awards, prog, function(x) {
  sprintf("M (SD) = %1.2f (%1.2f)", mean(x), sd(x))
}))
```

Poisson Regression with R

```
##                      General                     Academic
## "M (SD) = 0.20 (0.40)" "M (SD) = 1.00 (1.28)" "M (SD)

ggplot(p, aes(num_awards, fill = prog)) +
  geom_histogram(binwidth=.5, position="dodge")
```

Negative Binomial Regression with R

```
ggplot(dat, aes(daysabs, fill = prog)) + geom_histogram(b
    ., margins = TRUE, scales = "free")
```

```
with(dat, tapply(daysabs, prog, function(x) {
    sprintf("M (SD) = %1.2f (%1.2f)", mean(x), sd(x))
}))

##                      General                       Academic
## "M (SD) = 10.65 (8.20)"  "M (SD) = 6.93 (7.45)"  "M (S
```