“What is the process for conducting Ordered Logistic Regression in Stata for data analysis?”

Authored by
stats writer

June 29, 2024

Ordered Logistic Regression is a statistical method used for analyzing data that has ordinal dependent variables. This process involves using Stata, a statistical software, to estimate the relationship between the dependent variable and one or more independent variables. The steps for conducting Ordered Logistic Regression in Stata typically include data preparation, model specification, model fitting, and interpretation of results. Data preparation involves organizing the data into the appropriate format and checking for any missing values. Model specification involves selecting the appropriate variables and functional form for the model. Model fitting is done through the use of Stata commands and the results are then interpreted to understand the relationship between the variables. This process allows researchers to analyze and make inferences about the impact of independent variables on the ordered dependent variable.

Ordered Logistic Regression | Stata Data Analysis Examples

Version info: Code for this page was tested in Stata 12.

Examples of ordered logistic regression

Example 1: 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.**

**Example 2: 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.**

**Example 3: 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".

Description of the data

For our data analysis below, we are going to expand on Example 3 about
applying to graduate school. We have simulated some data for this example
and it can be obtained from our website:

use https://stats.idre.ucla.edu/stat/data/ologit.dta, clear

This hypothetical data set has a three-level variable called apply
(coded 0, 1, 2), that we
will use as our outcome variable. We also have three variables that we will use as 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, and gpa, which is the student's grade point average.

Let's start with the descriptive statistics of these variables.

tab apply

apply | Freq. Percent Cum.
----------------+-----------------------------------
unlikely | 220 55.00 55.00
somewhat likely | 140 35.00 90.00
very likely | 40 10.00 100.00
----------------+-----------------------------------
Total | 400 100.00

tab apply, nolab

apply | Freq. Percent Cum.
------------+-----------------------------------
0 | 220 55.00 55.00

1 | 140 35.00 90.00

2 | 40 10.00 100.00

------------+-----------------------------------

Total | 400 100.00

tab apply pared

| pared

apply | 0 1 | Total

-----------------+----------------------+----------

unlikely | 200 20 | 220

somewhat likely | 110 30 | 140

very likely | 27 13 | 40

-----------------+----------------------+----------

Total | 337 63 | 400

tab apply public

| public

apply | 0 1 | Total

-----------------+----------------------+----------

unlikely | 189 31 | 220

somewhat likely | 124 16 | 140

very likely | 30 10 | 40

-----------------+----------------------+----------

Total | 343 57 | 400

summarize gpa

Variable | Obs Mean Std. Dev. Min Max
-------------+---
gpa | 400 2.998925 .3979409 1.9 4

table apply, cont(mean gpa sd gpa)

--
apply | mean(gpa) sd(gpa)
----------------+----------------------
unlikely | 2.952136 .403594
somewhat likely | 3.030071 .3893446
very likely | 3.14725 .3560322
--

Analysis methods you might consider

Below is a list of some analysis methods you may have encountered.
Some of the methods listed are quite reasonable while others have either
fallen out of favor or have limitations.

Ordered logistic regression

**Below we use the ologit command to estimate an ordered logistic regression
model. The i. before pared indicates that pared is a factor
variable (i.e.,
categorical variable), and that it should be included in the model as a series
of indicator variables. The same goes for i.public.**

ologit apply i.pared i.public gpa

Iteration 0: log likelihood = -370.60264
Iteration 1: log likelihood = -358.605
Iteration 2: log likelihood = -358.51248
Iteration 3: log likelihood = -358.51244
Iteration 4: log likelihood = -358.51244

Ordered logistic regression Number of obs = 400
LR chi2(3) = 24.18
Prob > chi2 = 0.0000
Log likelihood = -358.51244 Pseudo R2 = 0.0326

--

apply | Coef. Std. Err. z P>|z|

-------------+--

1.pared | 1.047664 .2657891 3.94 0.000 .5267266 1.568601

1.public | -.0586828 .2978588 -0.20 0.844 -.6424754 .5251098

gpa | .6157458 .2606311 2.36 0.018 .1049183 1.126573

-------------+--

/cut1 | 2.203323 .7795353 .6754621 3.731184

/cut2 | 4.298767 .8043147 2.72234 5.875195

In the output above, we first see the iteration log. At iteration 0,
Stata fits a null model, i.e. the intercept-only model. It then moves on to fit
the full model and stops the iteration process once the difference in log
likelihood between successive iterations become sufficiently small. The final log likelihood (-358.51244) is displayed again. It can be used
in comparisons of nested models.
Also at the top of the output we see that all 400 observations in our data set

**were used in the analysis. The likelihood ratio chi-square of 24.18 with a p-value of 0.0000 tells us that our model as a whole is statistically
significant, as compared to the null model with no predictors. The pseudo-R-squared
of 0.0326 is also given.**

**In the table we see the coefficients, their standard errors, z-tests and
their associated p-values, and the 95% confidence interval of the coefficients.
Both pared and gpa are statistically significant; public is
not. So 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 log odds of being in a higher level of apply, given all of the other
variables in the model are held constant. For a one unit increase
in gpa, we would expect a 0.62 increase in the log odds of being in a
higher level of apply, given that all of the other variables in the model are**

held constant. The cutpoints shown at the bottom of the output indicate where the latent variable is cut to make the three
groups that we observe in our data. 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. For further information, please
see the Stata FAQ:
How can I
convert Stata's parameterization of ordered probit and logistic models to one in
which a constant is estimated?

We can obtain odds ratios using the or option after the ologit
command.

ologit apply i.pared i.public gpa, or

Iteration 0: log likelihood = -370.60264
Iteration 1: log likelihood = -358.605

Iteration 2: log likelihood = -358.51248

Iteration 3: log likelihood = -358.51244

Ordered logistic regression Number of obs = 400

LR chi2(3) = 24.18

Prob > chi2 = 0.0000

Log likelihood = -358.51244 Pseudo R2 = 0.0326

--

apply | Odds Ratio Std. Err. z P>|z|

-------------+--

pared | 2.850982 .75776 3.94 0.000 1.69338 4.799927

public | .9430059 .2808826 -0.20 0.844 .5259888 1.690644

gpa | 1.851037 .4824377 2.36 0.018 1.11062 3.085067

-------------+--

/cut1 | 2.203323 .7795353 .6754622 3.731184

/cut2 | 4.298767 .8043146 2.72234 5.875195

--

In the output above the results are displayed as proportional odds ratios.
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 pared, i.e., going from 0 to 1, the odds of high apply versus the combined middle and low categories are 2.85 greater, given that all of the other variables in the model are held constant. Likewise, the odds of the combined middle and high categories versus low apply is 2.85 times greater, given that all of the other variables in the model are held constant. For a one unit increase in gpa, the odds of the high category of apply versus the low and middle categories of apply are 1.85 times greater, given that the other variables in the model are held constant. Because of the proportional odds assumption (see below for more explanation), the same increase, 1.85 times, is found between low apply and the combined categories of middle and high apply.

You can also use the listcoef command to obtain the odds ratios, as

well as the change in the odds for a standard deviation of the variable.

We have used the help option to get the list at the bottom of the output

explaining each column. You can use the percent option to see the

percent change in the odds. The listcoeff command was written by Long and

Freese, and you will need to download it by typing search spost (see

How can I use the search command to search for programs and get additional

help? for more information about using search).

```
listcoef, help

ologit (N=400): Factor Change in Odds

Odds of: >m vs <=m

----------------------------------------------------------------------
apply | b z P>|z| e^b e^bStdX SDofX
-------------+--------------------------------------------------------
pared | 1.04766 3.942 0.000 2.8510 1.4654 0.3647
public | -0.05868 -0.197 0.844 0.9430 0.9797 0.3500
```

gpa | 0.61575 2.363 0.018 1.8510 1.2777 0.3979

--

b = raw coefficient

z = z-score for test of b=0

P>|z| = p-value for z-test

e^b = exp(b) = factor change in odds for unit increase in X

e^bStdX = exp(b*SD of X) = change in odds for SD increase in X

SDofX = standard deviation of X

listcoef, help percent

ologit (N=400): Percentage Change in Odds

Odds of: >m vs <=m

--

apply | b z P>|z| % %StdX SDofX

-------------+--

pared | 1.04766 3.942 0.000 185.1 46.5 0.3647

public | -0.05868 -0.197 0.844 -5.7 -2.0 0.3500

gpa | 0.61575 2.363 0.018 85.1 27.8 0.3979

--

b = raw coefficient

z = z-score for test of b=0
P>|z| = p-value for z-test
% = percent change in odds for unit increase in X
%StdX = percent change in odds for SD increase in X
SDofX = standard deviation of X

One of the assumptions underlying ordered logistic (and ordered probit) regression is that the relationship between each pair of outcome groups is the same. In other words, ordered 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. This is called the proportional odds assumption or the parallel regression assumption. Because the relationship between all pairs of groups is the same, there is only one set of coefficients (only one model). If this was not the case,

we would need different models to describe the relationship between each pair of outcome groups. We need to test the proportional odds assumption, and there are two tests that can be used to do so. First, we need to download a user-written command called omodel (type search omodel). The first test that we will show does a likelihood ratio test. The null hypothesis is that there is no difference in the coefficients between models, so we "hope" to get a non-significant result. Please note that the omodel command does not recognize factor variables, so the i. is ommited. The brant command performs a Brant test. As the note at the bottom of the output indicates, we also "hope" that these tests are non-significant. The brant command, like listcoeff, is part of the spost add-on and can be obtained by typing searchspost. We have used the detail option

here,which shows the estimated coefficients for the two equations. (We have two equations because we have three categories in our response variable.)
Also, you will note that the likelihood ratio chi-square value of 4.06 obtained from the omodel command is very close to the 4.34 obtained from the brant command.

omodel logit apply pared public gpa

Iteration 0: log likelihood = -370.60264
Iteration 1: log likelihood = -358.605
Iteration 2: log likelihood = -358.51248
Iteration 3: log likelihood = -358.51244

Ordered logit estimates Number of obs = 400
LR chi2(3) = 24.18
Prob > chi2 = 0.0000
Log likelihood = -358.51244 Pseudo R2 = 0.0326

--

apply | Coef. Std. Err. z P>|z|

-------------+--

pared | 1.047664 .2657891 3.94 0.000 .5267266 1.568601
public | -.0586828 .2978588 -0.20 0.844 -.6424754 .5251098
gpa | .6157458 .2606311 2.36 0.018 .1049183 1.126573
————-+—————————————————————-
_cut1 | 2.203323 .7795353 (Ancillary parameters)
_cut2 | 4.298767 .8043146
——————————————————————————

Approximate likelihood-ratio test of proportionality of odds
across response categories:
chi2(3) = 4.06
Prob > chi2 = 0.2553

brant, detail

Estimated coefficients from j-1 binary regressions

y>0 y>1
pared 1.0596117 .915596
public -.20055709 .53508208
gpa .54824568 .73632132
_cons -1.9829709 -4.7544684

Brant Test of Parallel Regression Assumption

Variable | chi2 p>chi2 df

-------------+-------------------------

All | 4.34 0.227 3

-------------+-------------------------

pared | 0.13 0.716 1

public | 3.44 0.064 1

gpa | 0.18 0.672 1

**A significant test statistic provides evidence that the parallel
regression assumption has been violated.**

**Both of the above tests indicate that we have not violated the proportional
odds assumption. If we had, we would want to run our model as a
generalized ordered logistic model using gologit2. You need to download
gologit2 by typing search gologit2.**

We can also obtain predicted probabilities, which are usually easier to

understand than the coefficients or the odds ratios. We will use the margins command. This can be used with either a categorical variable or a continuous variable and shows the predicted probability for each of the values of the variable specified. We will use pared as an example with a categorical predictor. Here we will see how the probabilities of membership to each category of apply change as we vary pared and hold the other variable at their means. As you can see, the predicted probability of being in the lowest category of apply is 0.59 if neither parent has a graduate level education and 0.34 otherwise. For the middle category of apply, the predicted probabilities are 0.33 and 0.47, and for the highest category of apply, 0.078 and 0.196. Hence, if neither of a respondent 's parents have a graduate level education, the predicted probability of applying to

graduate school decreases. We can see at values each variable is held at
the top of each output.

margins, at(pared=(0/1)) predict(outcome(0)) atmeans

Adjusted predictions Number of obs = 400
Model VCE : OIM

Expression : Pr(apply==0), predict(outcome(0))

1._at : pared = 0
public = .1425 (mean)
gpa = 2.998925 (mean)

2._at : pared = 1
public = .1425 (mean)
gpa = 2.998925 (mean)

--
| Delta-method
| Margin Std. Err. z P>|z|
-------------+--
_at |
1 | .5902769 .0268846 21.96 0.000 .5375841 .6429697
2 | .3356916 .0549943 6.10 0.000 .2279047 .4434784

----

margins, at(pared=(0/1)) predict(outcome(1)) atmeans

Adjusted predictions Number of obs = 400
Model VCE : OIM

Expression : Pr(apply==1), predict(outcome(1))

1._at : pared = 0
public = .1425 (mean)
gpa = 2.998925 (mean)

2._at : pared = 1
public = .1425 (mean)
gpa = 2.998925 (mean)

| Delta-method
| Margin Std. Err. z P>|z|
-------------+--
_at |
1 | .331053 .0242226 13.67 0.000 .2835775 .3785285
2 | .4685299 .0344096 13.62 0.000 .4010883 .5359714

----

margins, at(pared=(0/1)) predict(outcome(2)) atmeans

Adjusted predictions Number of obs = 400

Model VCE : OIM

Expression : Pr(apply==2), predict(outcome(2))

1._at : pared = 0

public = .1425 (mean)

gpa = 2.998925 (mean)

2._at : pared = 1

public = .1425 (mean)

gpa = 2.998925 (mean)

--

| Delta-method

| Margin Std. Err. z P>|z|

-------------+--

_at |

1 | .0786702 .0132973 5.92 0.000 .052608 .1047323

2 | .1957785 .040827 4.80 0.000 .1157591 .275798

--

We can also use the margins command to select values

of
a continuous variable and see what the predicted probabilities are at each
point. Below, we see the predicted probabilities for gpa at 2, 3
and 4. As you can see, almost for each value of gpa, the highest predicted
probability is for the lowest category of apply and only when gpa is 4, the predicted probability is slightly higher for somewhat likely than unlikely, which makes sense
because most respondents are in that category. You can also see that the
predicted probability increases for both the middle and highest categories of
apply as gpa increases.

margins, at(gpa=(2/4)) predict(outcome(0)) atmeans

Adjusted predictions Number of obs = 400
Model VCE : OIM

Expression : Pr(apply==0), predict(outcome(0))

1._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 2

2._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 3

3._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 4

--

| Delta-method

| Margin Std. Err. z P>|z|

-------------+--

_at |

1 | .6932137 .060112 11.53 0.000 .5753963 .811031

2 | .5496956 .0255013 21.56 0.000 .499714 .5996773

3 | .3974013 .0665332 5.97 0.000 .2669986 .5278041

--

margins, at(gpa=(2/4)) predict(outcome(1)) atmeans

Adjusted predictions Number of obs = 400

Model VCE : OIM

Expression : Pr(apply==1), predict(outcome(1))

1._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 2

2._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 3

3._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 4

--

| Delta-method

| Margin Std. Err. z P>|z|

-------------+--

_at |

1 | .2551558 .0472683 5.40 0.000 .1625116 .3477999

2 | .3587569 .0246482 14.56 0.000 .3104474 .4070664

3 | .4453892 .0399212 11.16 0.000 .367145 .5236334

--

margins, at(gpa=(2/4)) predict(outcome(2)) atmeans

Adjusted predictions Number of obs = 400

Model VCE : OIM

Expression : Pr(apply==2), predict(outcome(2))

1._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 2

2._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 3

3._at : pared = .1575 (mean)

public = .1425 (mean)

gpa = 4

--

| Delta-method

| Margin Std. Err. z P>|z|

-------------+--

_at |

1 | .0516305 .0158556 3.26 0.001 .0205541 .0827069

2 | .0915475 .0142998 6.40 0.000 .0635204 .1195745

3 | .1572095 .0397767 3.95 0.000 .0792486 .2351703

Here we loop through the values of apply (0, 1, and 2) and calculate
predicted probabilities when gpa = 3.5, pared = 1, and public
= 1.

```
forvalues i = 0/2 {
margins, at(gpa = 3.5 pared = 1 public = 1) predict(outcome(`i'))
}
```

Adjusted predictions Number of obs = 400
Model VCE : OIM

Expression : Pr(apply==0), predict(outcome(0))
at : pared = 1
public = 1
gpa = 3.5

| Delta-method
| Margin Std. Err. z P>|z|

-------------+--

_cons | .2807452 .0695883 4.03 0.000 .1443547 .4171357

Adjusted predictions Number of obs = 400

Model VCE : OIM

Expression : Pr(apply==1), predict(outcome(1))

at : pared = 1

public = 1

gpa = 3.5

| Delta-method

| Margin Std. Err. z P>|z|

-------------+--

_cons | .4796188 .0326872 14.67 0.000 .4155531 .5436844

Adjusted predictions Number of obs = 400

Model VCE : OIM

Expression : Pr(apply==2), predict(outcome(2))

at : pared = 1

public = 1

gpa = 3.5

| Delta-method

| Margin Std. Err. z P>|z|

-------------+--

_cons | .239636 .063819 3.75 0.000 .114553 .364719

Things to consider

See also

References