

Ordered Logit Models – Basic & Intermediate Topics

Richard Williams, University of Notre Dame, <https://academicweb.nd.edu/~rwilliam/>

Last revised September 29, 2024

This is adapted heavily from Menard's Applied Logistic Regression analysis; also, Borooah's Logit and Probit: Ordered and Multinomial Models; Also, Hamilton's Statistics with Stata, Updated for Version 7.

For a more detailed discussion with additional examples, see Williams, R. A., & Quiroz, C. (2019). *Ordinal Regression Models*. In P. Atkinson, S. Delamont, A. Cernat, J.W. Sakshaug, & R.A. Williams (Eds.), SAGE Research Methods Foundations. doi: 10.4135/9781526421036885901.
<https://methods.sagepub.com/Foundations/ordinal-regression-models>

We have talked about the analysis of dependent variables that have only two possible values, e.g. lives or dies, wins or loses, gets an A or doesn't get an A. Of course, many dependent variables of interest will have more than two possible categories. These categories might be unordered (doesn't move, moves South, moves East) or ordered (high, medium, low; favors more immigration, thinks the level of immigration is about right, favors less immigration). We will briefly discuss techniques for handling each of these.

Ordinal Regression

As Menard notes, when dependent variables are measured on an ordinal scale, there are many options for their analysis. These include

- Treating the variable as though it were continuous. In this case, just use OLS regression or the other techniques we have discussed for continuous variables. Certainly, this is widely done, particularly when the DV has 5 or more categories. Since this is probably the easiest approach for readers to understand, sometimes the other approaches are tried just to confirm that the use of OLS does not seriously distort the findings.
- Ignoring the ordinality of the variable and treating it as nominal. i.e. use multinomial logit techniques like those we will discuss later. The key problem here is a loss of efficiency. By ignoring the fact that the categories are ordered, you fail to use some of the information available to you, and you may estimate many more parameters than is necessary. This increases the risk of getting statistically insignificant results. But, your parameter estimates still should be unbiased.
- Treating the variable as though it were measured on an ordinal scale, but the ordinal scale represented crude measurement of an underlying interval/ratio scale. For example, the categories "High, Medium, Low" might be rough measures for Socio-economic status or intelligence. Ordered logit models can be used in such cases, and they are the primary focus of this handout.

Menard cautions that choosing the correct option requires careful judgment. In other words, don't just assume that because Stata has a routine called `ologit`, or that the SPSS pulldown menu for Ordinal Regression brings up PLUM, that these are necessarily the best way to go.

Ordered Logit/ Proportional Odds Models. Having made that caution, I'll now explain how the ordered logit models estimated by SPSS PLUM and `ologit` work. The ordered logit model fit by `ologit` is also known as the *proportional odds model*. The terms *parallel lines model* and *parallel regressions model* are also sometimes used, for reasons we will see in a moment.

1. In the ordered logit model, there is an observed ordinal variable, Y .
2. Y , in turn, is a function of another 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 points. (κ is the Greek small letter Kappa.) *Your value on the observed variable Y depends on whether or not you have crossed a particular threshold.* For example, when $M = 3$

$$Y_i = 1 \text{ if } Y^*_i \leq \kappa_1$$

$$Y_i = 2 \text{ if } \kappa_1 \leq Y^*_i \leq \kappa_2$$

$$Y_i = 3 \text{ if } Y^*_i \geq \kappa_2$$

For example, it might be that if your score on the unobserved latent variable Y^* was 37 or less, your score on Y would be 1; if your Y^* score was between 37 and 53, Y would equal 2; and if your Y^* score was above 53, Y would equal 3.

Put another way, you can think of Y as being a collapsed version of Y^* , e.g. Y^* can take on an infinite range of values which might then be collapsed into 5 categories of Y .

3. So, what does Y^* equal? How do you estimate this model?
 - a. In the population, the continuous latent variable Y^* is equal to

$$Y^*_i = \sum_{k=1}^K \beta_k X_{ki} + \varepsilon_i = Z_i + \varepsilon_i$$

Note that there is a random disturbance term, which, in this case, has a standard logistic distribution (mean of 0 and variance of 3.29; a $N(0, 1)$ distribution is also often used). This reflects the fact that relevant variables may be left out of the equation, or variables may not be perfectly measured.

- b. The Ordered Logit Model estimates *part* of the above:

$$Z_i = \sum_{k=1}^K \beta_k X_{ki} = E(Y^*_i)$$

c. Note that, because of the random disturbance term, the unmeasured latent variable Y^* can be either *higher* or *lower* than Z . By way of analogy, the typical person with 12 years of education might make \$30,000 a year; but any specific person with 12 years of education may make more than that or less than that. Because of the disturbance term, i.e. because Z is not a perfect measure of Y^* , you will incorrectly classify some cases as falling within one range when they actually fall within another. But, because you know the distribution of the error term, you can also estimate what the probability of error is.

d. The K β s and the $M-1$ κ s are parameters that need to be estimated. Once you have done so, using the corresponding sample estimates for each case you compute

$$Z_i = \sum_{k=1}^K \beta_k X_{ik}$$

Note that there is no intercept term. You then use the estimated $M-1$ cutoff terms to estimate the probability that Y will take on a particular value.

OPTIONAL: The formulas are

$$P(Y_i > j) = \frac{\exp(X_i\beta - \kappa_j)}{1 + [\exp(X_i\beta - \kappa_j)]}, j = 1, 2, \dots, M-1, \text{ which implies}$$

$$P(Y_i = 1) = 1 - \frac{\exp(X_i\beta - \kappa_1)}{1 + [\exp(X_i\beta - \kappa_1)]}$$

$$P(Y_i = j) = \frac{\exp(X_i\beta - \kappa_{j-1})}{1 + [\exp(X_i\beta - \kappa_{j-1})]} - \frac{\exp(X_i\beta - \kappa_j)}{1 + [\exp(X_i\beta - \kappa_j)]} \quad j = 2, \dots, M-1$$

$$P(Y_i = M) = \frac{\exp(X_i\beta - \kappa_{M-1})}{1 + [\exp(X_i\beta - \kappa_{M-1})]}$$

In the case of $M = 3$, these equations simplify to

$$P(Y = 1) = \frac{1}{1 + \exp(Z_i - \kappa_1)}$$

$$P(Y = 2) = \frac{1}{1 + \exp(Z_i - \kappa_2)} - \frac{1}{1 + \exp(Z_i - \kappa_1)}$$

$$P(Y = 3) = 1 - \frac{1}{1 + \exp(Z_i - \kappa_2)}$$

4. Hence, using the estimated value of Z and the assumed logistic distribution of the disturbance term, the ordered logit model can be used to estimate the probability that the unobserved variable Y^* falls within the various threshold limits.

NOTE: As Long points out, you can also motivate the ordered logit model by thinking of it as a nonlinear probability model, i.e. you predict the probability of a 1, a 2, etc. You don't have to rely on the notion of an underlying y^* , and some prefer not to.

Example. In *Statistics With Stata, Updated for Version 7*, Hamilton presents a fascinating example that shows that the space shuttle Challenger disaster of January 28, 1986, might have been averted had NASA officials heeded the warning signs. Data cover the first 25 flights of the U.S. space shuttle. For each flight, the following variables are measured:

Distress	The number of "thermal distress incidents" in which hot gas damaged the joint seals of a flight's booster rockets. Damage to the joint seals helped lead to the Challenger disaster. This is the DV. It is coded 1 = None, 2 = 1 or 2, and 3 = 3 plus.
Temp	The calculated joint temperature at launch time. Temperature depends largely on weather. Colder temperatures cause the rubber o-rings sealing the booster rocket joints to become less flexible and hence more likely to have problems.
Date	Date, measured in days elapsed since January 1, 1960 (an arbitrary starting point). The rationale for this variable is that undesirable changes in the shuttle program and aging hardware may have caused launches to become more risky across time.

Here is the data:

flight	distress	temp	date	z^* (computed)
1	none	66	7772	14.09598
2	1 or 2	70	7986	14.10568
3	none	69	8116	14.70623
4	MISSING	80	8213	13.11785
5	none	68	8350	15.64853
6	1 or 2	67	8494	16.29509
7	none	72	8569	15.67466
8	none	73	8642	15.74117
9	none	70	8732	16.55703
10	1 or 2	57	8799	19.03107
11	3 plus	63	8862	18.19784
12	3 plus	70	9008	17.46397
13	none	78	9044	16.19526
14	none	67	9078	18.21411
15	3 plus	53	9155	20.89439
16	3 plus	67	9233	18.72344
17	3 plus	75	9250	17.3923
18	3 plus	70	9299	18.42019

19	1 or 2	81	9341	16.65107
20	1 or 2	76	9370	17.61325
21	none	79	9407	17.2147
22	3 plus	75	9434	17.99692
23	1 or 2	76	9461	17.91227
24	3 plus	58	9508	21.18747
25 (Challenger)	MISSING	31	9524	25.92117

Here is what Stata's `ologit` gives you when Distress is regressed on Date and Temp. Challenger and one flight with MD are excluded, yielding an N of 23 cases.

```
. use https://academicweb.nd.edu/~rwilliam/statafiles/shuttle2.dta, clear
(First 25 space shuttle flights)

. * Date has been added to shuttle2.dta, but here is the command that created it.
. * gen date = mdy( month, day, year)

. ologit distress date temp
Iteration 0:   log likelihood = -24.955257
Iteration 1:   log likelihood = -18.871284
Iteration 2:   log likelihood = -18.79755
Iteration 3:   log likelihood = -18.79706
Iteration 4:   log likelihood = -18.79706

Ordered logistic regression               Number of obs   =           23
                                         LR chi2(2)      =          12.32
                                         Prob > chi2     =          0.0021
Log likelihood = -18.79706               Pseudo R2       =          0.2468
```

distress	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
date	.003286	.0012662	2.60	0.009	.0008043 .0057677
temp	-.1733752	.0834475	-2.08	0.038	-.3369293 -.0098212
/cut1	16.42813	9.554822			-2.298978 35.15524
/cut2	18.12227	9.722302			-.933092 37.17763

Here is how to interpret the results:

* The model chi-square is 12.32 with 2 d.f. This is highly significant, and tells us that date and/or temp has a significant effect on the number of thermal distress incidents.

* McFadden R^2 (aka pseudo R^2) is

$$\text{Pseudo } R^2 = \text{Model } L^2 / \text{DEV}_0 = 12.32/49.91 = .247$$

$$(\text{Remember, } \text{DEV}_0 = -2 * LL_0 = -2 * -24.955257 = 49.91)$$

* The positive coefficient for DATE means that the likelihood of distress incidents did increase with time. Similarly, the negative coefficient for TEMP implies that colder temps increased the likelihood of having distress incidents.

* The threshold parameters of 16.4281 and 18.1223 tell us the following. Since there are three possible values for Y (i.e. $M = 3$), the values for Y are

$$Y_i = 1 \text{ if } Y^*_i \leq 16.4281$$

$$Y_i = 2 \text{ if } 16.4281 \leq Y^*_i \leq 18.1223$$

$$Y_i = 3 \text{ if } Y^*_i \geq 18.1223$$

As usual, we can look at the sign and significance of coefficients when interpreting them, but it helps to plug in some hypothetical or real data values to get a better feel for the coefficients' meaning.

Shuttle flight 13: temperature was 78 on launch date and date equaled 9044. Hence, for Flight 13, we compute

$$Z_i = (.003286 * 9044) - .173375 * 78 = 16.195334$$

Note that this value is less than the lowest threshold estimate of 16.4281. For Flight 13, we can next compute

$$P(Y = 1) = \frac{1}{1 + \exp(Z_i - \kappa_1)} = \frac{1}{1 + \exp(16.195334 - 16.4281)} = .5579$$

$$\begin{aligned} P(Y = 2) &= \frac{1}{1 + \exp(Z_i - \kappa_2)} - \frac{1}{1 + \exp(Z_i - \kappa_1)} \\ &= \frac{1}{1 + \exp(16.195334 - 18.1223)} - \frac{1}{1 + \exp(16.195334 - 16.4281)} = .8729 - .5579 = .315 \end{aligned}$$

$$P(Y = 3) = 1 - \frac{1}{1 + \exp(Z_i - \kappa_2)} = 1 - \frac{1}{1 + \exp(16.195334 - 18.1223)} = .1271$$

Hence, for Flight 13, which occurred more than a year earlier than Challenger under much warmer conditions, the most likely outcome was that there would be no damage to the booster joints. In fact, Flight 13 did not have any problems.

That is, for Flight 13, our estimate of Z is 16.195334. This is our “best guess” for the value of Y^*_i , and this value places Flight 13 in the $Y = 1$ threshold. But, because of the random disturbance term, there is at least some chance that Y^*_i is larger than Z_i , i.e. because of other unmeasured influences there is some chance that Flight 13 was more at risk than our estimates indicate. If so, this could move Flight 13 into one of the higher threshold categories, e.g. the Y^* value for Flight 13 could actually be 17, in which case Y would equal 2; or it might be 18.5, in which case $Y = 3$. (Of course, it is also possible that the risk for Flight 13 was less than was estimated.)

Given that our estimate of Z for Flight 13 is very close to the Cutoff point for $Y = 1$, it is not surprising that we find $P(Y = 1) = .5579$, $P(Y = 2) = .315$, $P(Y = 3) = .1271$. That is, $Y = 1$ is the

most likely value for Flight 13, but $Y = 2$ and $Y = 3$ also have fairly high probabilities. If the Z value were much smaller, e.g. $Z = 2$, then it would be much less likely that Y^* actually fell into a higher threshold range, and we would find that $P(Y = 1)$ would be much higher.

Shuttle flight 25, Challenger: Remember, *Challenger's own data was not used when calculating these parameters*. Hence, it would have been possible for a NASA official to use these numbers on launch day to predict the likelihood of a problem. On Challenger's Launch Date, Date equaled 9524, and the temperature at launch time was 31 Fahrenheit (the previous coldest launch had been at 53 Fahrenheit). Hence, for Challenger,

$$Z_i = (.003286 * 9524) - .173375 * 31 = 25.9212$$

Note that this value is much higher than the upper threshold estimate of 18.1223 presented by `ologit`. Using the formulas presented earlier and the threshold estimates, we can now compute the probabilities of Challenger falling into each of the three different distress categories:

$$\begin{aligned} P(Y = 1) &= \frac{1}{1 + \exp(Z_i - \kappa_1)} = \frac{1}{1 + \exp(25.9212 - 16.4281)} = .0000754 \\ P(Y = 2) &= \frac{1}{1 + \exp(Z_i - \delta_2)} - \frac{1}{1 + \exp(Z_i - \kappa_1)} = \frac{1}{1 + \exp(25.9212 - 18.1223)} - \frac{1}{1 + \exp(25.9212 - 16.4281)} \\ &= .000410 - .0000754 = .0003346 \\ P(Y = 3) &= 1 - \frac{1}{1 + \exp(Z_i - \kappa_2)} = 1 - \frac{1}{1 + \exp(25.9212 - 18.1223)} = .99959 \end{aligned}$$

Hence, based on the experience from the previous 23 flights, there was virtually no chance that Challenger would experience no damage to its joint seals. Indeed, *it was a virtual certainty that Challenger would experience 3 or more damage incidents*.

In summary, in the case of Challenger, our estimate of Z is 25.9212, which is far above the upper threshold limit. It is possible that the actual risk faced by Challenger was less than this, e.g. maybe Challenger's score on Y^* was really only 23. But, it is very unlikely that the Y^* value for Challenger was actually ≤ 16.4281 . It is also possible, but still not very likely, for Challenger that $16.4281 \leq Y^* \leq 18.1223$. If, say, the estimate of Z for Challenger had been 18.5, there would have been a much better chance that the true value of Y^* fell into one of the lower threshold ranges.

Incidentally, if you run OLS regression instead, the predicted value for Challenger is 4.63 and the predicted value for Flight 13 is 1.65. The Challenger estimate of 4.63, of course, isn't a legitimate value for Y , but it is consistent with the finding that launching on that day was very risky.

Predicted Values. Stata also makes it easy to get the predicted Z values and the predicted probability of each of the three possible outcomes. Recall that, by default, Stata's `predict` command usually computes values for *all* cases in the data, not just those that were included in the analysis. Sometimes, this can be dangerous; but in this case it is nice because it gives us predicted values for Challenger, case 25.

```
. * Compute the Z value
. predict z, xb

. * Get the predicted probabilities for each of the three possible outcomes.
. * Specify one var for each outcome
. predict none onetwo threeplus, p
. list flight temp date distress z none onetwo threeplus
```

	flight	temp	date	distress	z	none	onetwo	threeplus
1.	STS-1	66	7772	none	14.09598	.9115049	.0709674	.0175277
2.	STS-2	70	7986	1 or 2	14.10568	.9107192	.0715853	.0176955
3.	STS-3	69	8116	none	14.70623	.8483726	.1198293	.031798
4.	STS-4	80	8213	.	13.11785	.9647798	.0285567	.0066635
5.	STS-5	68	8350	none	15.64853	.6855931	.236687	.0777198
6.	STS-6	67	8494	1 or 2	16.29509	.5332105	.3282149	.1385746
7.	STS-7	72	8569	none	15.67466	.6799332	.2404531	.0796137
8.	STS-8	73	8642	none	15.74117	.6652908	.2500842	.084625
9.	STS-9	70	8732	none	16.55703	.4678189	.359285	.1728961
10.	STS_41-B	57	8799	1 or 2	19.03107	.0689493	.2182961	.7127547
11.	STS_41-C	63	8862	3 plus	18.19784	.1455785	.3355387	.5188827
12.	STS_41-D	70	9008	3 plus	17.46397	.261954	.3969255	.3411205
13.	STS_41-G	78	9044	none	16.19526	.5579556	.3149626	.1270818
14.	STS_51-A	67	9078	none	18.21411	.143566	.33349	.522944
15.	STS_51-C	53	9155	3 plus	20.89439	.0113597	.04749	.9411502
16.	STS_51-D	67	9233	3 plus	18.72344	.091512	.2625642	.6459238
17.	STS_51-B	75	9250	3 plus	17.3923	.2760438	.398755	.3252012
18.	STS_51-G	70	9299	3 plus	18.42019	.1200389	.3060272	.5739338
19.	STS_51-F	81	9341	1 or 2	16.65107	.4444934	.3687458	.1867608
20.	STS_51-I	76	9370	1 or 2	17.61325	.2341337	.3904446	.3754217
21.	STS_51-J	79	9407	none	17.2147	.3129053	.3995974	.2874972
22.	STS_61-A	75	9434	3 plus	17.99692	.1723883	.3589076	.468704
23.	STS_61-B	76	9461	1 or 2	17.91227	.1848028	.3675054	.4476918
24.	STS_61-C	58	9508	3 plus	21.18747	.0084984	.0360675	.955434
25.	STS_51-L	31	9524	.	25.92117	.0000754	.0003346	.99959

Note that the values for flights 13 and 25 are the same as we previously computed.

Testing Model Assumptions. There are several ways to test the proportional odds/ parallel lines assumption of the ordered logit model. We will start with the Brant test, which also helps to clarify exactly what the assumption is. We will explain the assumption further when we discuss generalized ordered logit (GOLOGIT) models. Appendix A will show other methods.

Brant Test. If you have downloaded and installed `spost13`, you can use the `brant` command to do Brant's test of `ologit`'s parallel regression/ proportional odds assumption:


```
. brant, detail
```

Estimated coefficients from binary logits

Variable	y_gt_1	y_gt_2
date	0.003	0.006
	2.17	1.80
temp	-0.173	-0.234
	-1.48	-1.86
_cons	-13.644	-36.845
	-1.34	-1.52

legend: b/t

Brant test of parallel regression assumption

	chi2	p>chi2	df
All	0.83	0.662	2
date	0.81	0.369	1
temp	0.22	0.643	1

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

The insignificant overall chi-square value (given in the row labeled All) suggests that `ologit`'s assumptions are met. (Of course, this sample is very small; with larger samples it is not at all unusual to find that the proportional odds assumption is violated.) `brant` also gives tests for each individual independent variable.

The `detail` option for `brant` clarifies why the ordered logit model is also sometimes called the parallel lines or parallel regressions model. In the binary logits, the ordinal variable is dichotomized. First it is category 1 versus all higher categories; then categories 1 and 2 versus all higher categories; and so on. If the assumptions of the ordered logit model are met, then the coefficients (other than the constants) should be the same for each logistic regression, i.e. the regression lines will be parallel, differing only in their intercepts.

Appendix A: Ordered Logit Models – Intermediate Topics

You can go through most of this on your own. Most of this is just straightforward extensions of techniques we talked earlier, applying them to ordinal data. This appendix covers additional ways of testing model assumptions; adjusted predictions and marginal effects; combining multiple imputation and margins with the `mimrgns` command; and doing ordinal regression with `svyset` data.

gologit2 LR Test and/or BIC and AIC tests. Stata's `gologit2` command (downloadable from SSC) provides alternate tests that usually give very similar results, but which can also be used with probit and other links. Here is an LR test:

Likelihood-ratio test	LR chi2(2)	=	0.77
(Assumption: ologit nested in gologit)	Prob > chi2	=	0.6815

Note: N=Obs used in calculating BIC; see [R] BIC note

Also, note that you get BIC and AIC values, so you could, if you wished, use them for deciding on whether the parallel lines assumption was met. (Also, as we will see later, `gologit2` offers a means by which the parallel lines assumption is relaxed for some variables but not others).

```
chi2( 2) = 0.51
Prob > chi2 = 0.7764
```

Page 10

```
. use https://academicweb.nd.edu/~rwilliam/statafiles/shuttle2.dta, clear
. omodel logit distress date temp
```

```
Ordered logit estimates                                Number of obs   =          23
                                                        LR chi2(2)      =          12.32
                                                        Prob > chi2     =          0.0021
Log likelihood = -18.79706                                Pseudo R2      =          0.2468
```

distress	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
date	.003286	.0012662	2.60	0.009	.0008043	.0057677
temp	-.1733752	.0834473	-2.08	0.038	-.336929	-.0098215
-----+-----						
_cut1	16.42813	9.554813	(Ancillary parameters)			
_cut2	18.12227	9.722293				

```
Approximate likelihood-ratio test of proportionality of odds
across response categories:
      chi2(2) =          0.60
      Prob > chi2 =          0.7405
```

oparallel command. Maarten Buis's `oparallel` command (downloadable from SSC) provides most of these tests and more for non-svy data, and can also provide BIC and AIC tests. The BIC test can be especially good with large samples, since it is less likely to reject the ordered logit model than are other tests.

```
. quietly ologit distress date temp
. oparallel, ic
```

Tests of the parallel regression assumption

	Chi2	df	P>Chi2
Wolfe Gould	.6008	2	0.741
Brant	.8255	2	0.662
score	.535	2	0.765
likelihood ratio	.7668	2	0.682
Wald	.5063	2	0.776

Information criteria

	ologit	gologit	difference
AIC	45.59	48.83	-3.23
BIC	50.14	55.64	-5.50

All of the tests agree that the proportional odds assumption is not violated with the shuttle data, but again, this is a very small data set. It is quite common to find that at least one variable violates the assumption. We will talk more later about what to do when assumptions are violated.

Adjusted Predictions and Marginal Effects. The procedures for adjusted predictions and marginal effects are very similar for multiple outcome commands like `ologit`, `oprobit`, `mlogit`, `gologit2`, and `oglm`. They are therefore discussed in a separate handout (currently located at <https://academicweb.nd.edu/~rwilliam/xsoc73994/Margins05.pdf>).

Multiple Imputation and `mimrgns`. (Thanks to Daniel Klein, the author of `mimrgns`, for explaining this to me.) If you have given a command like

```
mi estimate: ologit y i.female x1 x2
```

and you want to use `mimrgns`, and you are using Stata 14 or later, you can do something like

```
mimrgns female, predict(default)
```

If using Stata 13 or earlier, I think you have to specify an `mimrgns` command for each outcome, e.g. something like

```
mimrgns female, predict(outcome(1) pr)
mimrgns female, predict(outcome(2) pr)
mimrgns female, predict(outcome(3) pr)
```

Here is a constructed example. Basically it shows that those who have previously had heart attacks tend to be in worse health today than those who have not, i.e. they are about 14 percentage points more likely to be in poor health and 9 percentage points less likely to be in excellent health.

```
. * Artificial example of mimrgns and ologit
. version 14.2
. webuse mheart1s20, clear
(Fictional heart attack data; bmi missing)
. generate health = 4 - irecode(age, 45, 55 , 65)
. label define health 1 "poor" 2 "fair" 3 "good" 4 "excellent"
. label values health health
. tab1 health
```

-> tabulation of health

health	Freq.	Percent	Cum.
poor	143	24.07	24.07
fair	177	29.80	53.87
good	148	24.92	78.79
excellent	126	21.21	100.00
Total	594	100.00	

```
. mi estimate: ologit health i.attack i.smokes bmi
```

Multiple-imputation estimates	Imputations	=	20
Ordered logistic regression	Number of obs	=	154
	Average RVI	=	0.0268
	Largest FMI	=	0.1314
DF adjustment: Large sample	DF: min	=	1,127.31
	avg	=	357,982.87
	max	=	1551474.12
Model F test: Equal FMI	F(3,20230.1)	=	2.41
Within VCE type: OIM	Prob > F	=	0.0649

health	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
1.attack	-.6983812	.320593	-2.18	0.029	-1.326733	-.0700292
1.smokes	-.1313008	.3147298	-0.42	0.677	-.7481603	.4855587
bmi	.0586439	.038692	1.52	0.130	-.0172725	.1345604
/cut1	.1345452	.9746209	0.14	0.890	-1.777505	2.046595
/cut2	1.200603	.9810356	1.22	0.221	-.7241132	3.125319

```

/cut3 | 2.767973 1.01049 2.74 0.006 .7854679 4.750479

```

```

. * Adjusted predictions for attack
. mimrgns attack, predict(default)

```

```

Multiple-imputation estimates      Imputations      =      20
Predictive margins                Number of obs      =     154
                                   Average RVI          =     0.0199
                                   Largest FMI          =     0.0022
DF adjustment: Large sample        DF: min         = 3781017.30
                                   avg                  = 1.43e+08
Within VCE type: Delta-method      max             = 1.04e+09

Expression : predict()

```

-----+-----							
		Margin	Std. Err.	t	P> t	[95% Conf. Interval]	
-----+-----							
_predict#attack							
1 0		.2185914	.0397999	5.49	0.000	.140585	.2965979
1 1		.3580002	.0564998	6.34	0.000	.2472626	.4687379
2 0		.226214	.0342753	6.60	0.000	.1590356	.2933925
2 1		.2564113	.0366576	6.99	0.000	.1845637	.3282588
3 0		.3449389	.0418305	8.25	0.000	.2629527	.4269251
3 1		.2679858	.0410432	6.53	0.000	.1875426	.348429
4 0		.2102556	.0410699	5.12	0.000	.1297601	.2907511
4 1		.1176027	.0301815	3.90	0.000	.058448	.1767573
-----+-----							

```

. * AMES for attack
. mimrgns, predict(default) dydx(attack)

```

```

Multiple-imputation estimates      Imputations      =      20
Average marginal effects          Number of obs      =     154
                                   Average RVI          =     0.0018
                                   Largest FMI          =     0.0046
DF adjustment: Large sample        DF: min         = 896,139.85
                                   avg                  = 1988287.78
Within VCE type: Delta-method      max             = 4600516.71

Expression : predict()
dy/dx w.r.t. : 1.attack

```

		dy/dx	Std. Err.	t	P> t	[95% Conf. Interval]	
-----+-----							
1.attack							
_predict							
1		.1394088	.0646855	2.16	0.031	.0126273	.2661903
2		.0301972	.0166543	1.81	0.070	-.0024447	.0628391
3		-.0769531	.0375799	-2.05	0.041	-.1506084	-.0032977
4		-.0926529	.0426154	-2.17	0.030	-.1761777	-.0091282
-----+-----							

Note: dy/dx for factor levels is the discrete change from the base level.

Ordered Logit: Svysset Data. Most/all of the points made before about using svyset data continue to hold when using `ologit`, e.g. you can't do Likelihood Ratio tests; many post-estimation commands that you can use when the data are not svyset will not work when they are.

Also, *the only test of the parallel lines assumption that I know works with svyset data is the gologit2 Wald test*. In the following example, the variable health has 5 values so we test the equality of the coefficients in equations 1 through 4.

```
. webuse nhanes2f, clear
. quietly svy: gologit2 health weight i.female age i.race, npl
. test [#1 = #2 = #3 = #4]
```

Adjusted Wald test

```
( 1) [poor]weight - [fair]weight = 0
( 2) [poor]0b.female - [fair]0b.female = 0
[Output deleted]
```

```
      F( 15,      17) =      21.04
      Prob > F =      0.0000
```

It looks to me like the margins command and some, but not all, spost13 commands continue to work fine with svyset data, e.g. mtable works but fitstat does not.

Appendix B (Optional): Ordered Logit Models Using SPSS PLUM

Here is what SPSS PLUM gives you when Distress is regressed on Date and Temp. (At least this is what SPSS used to give. I haven't tried this in years.) Challenger and one flight with MD are excluded, yielding an N of 23 cases. (Note that, in keeping with SPSS's policy of internal inconsistency, PLUM graciously reports things like DEV_0 and McFadden's R^2 , numbers which it prefers to keep secret in its LOGISTIC REGRESSION routine.)

GET

```
FILE='D:\SOC593\SpssFiles\shuttle2.sav'.
```

COMPUTE Date = Yrmoda(year,month,day) - yrmoda(1960,1,1) .

PLUM

```
distress WITH date temp
/CRITERIA = CIN(95) DELTA(0) LCONVERGE(0) MXITER(100) MXSTEP(5)
PCONVERGE(1.0E-6) SINGULAR(1.0E-8)
/LINK = LOGIT
/PRINT = FIT PARAMETER SUMMARY TPARALLEL
/SAVE = ESTPROB (Plum) .
```

PLUM - Ordinal Regression

Warnings

There are 46 (66.7%) cells (i.e., dependent variable levels by combinations of predictor variable values) with zero frequencies.

Case Processing Summary

	N	Marginal Percentage
DISTRESS thermal 1 none	9	39.1%
distress incidents 2 1 or 2	6	26.1%
3 3 plus	8	34.8%
Valid	23	100.0%
Missing	2	
Total	25	

Model Fitting Information

Model	-2 Log Likelihood	Chi-Square	df	Sig.
Intercept Only	49.911			
Final	37.594	12.316	2	.002

Link function: Logit.

Goodness-of-Fit

	Chi-Square	df	Sig.
Pearson	42.858	42	.434
Deviance	37.594	42	.665

Link function: Logit.

Pseudo R-Square

Cox and Snell	.415
Nagelkerke	.468
McFadden	.247

Link function: Logit.

Parameter Estimates

		Estimate	Std. Error	Wald	df	Sig.	95% Confidence Interval	
							Lower Bound	Upper Bound
Threshold	[DISTRESS = 1]	16.4281	9.865	2.773	1	.096	-2.908	35.764
	[DISTRESS = 2]	18.1223	10.009	3.278	1	.070	-1.495	37.739
Location	DATE	.003286	.001	6.697	1	.010	.001	.006
	TEMP	-.173375	.085	4.157	1	.041	-.340	-.007

Link function: Logit.

Other Output Available from PLUM. Notice that we specified /SAVE=ESTPROB (Plum) on the Plum command. This causes SPSS to save the predicted probabilities as part of the .sav file, with each variable name starting with the arbitrarily chosen prefix of PLUM. Thus,

```
Formats plum1_1 plum2_1 plum3_1 (f8.4).
List flight temp date distress plum1_1 plum2_1 plum3_1 .
```

List

FLIGHT	TEMP	DATE	DISTRESS	PLUM1_1	PLUM2_1	PLUM3_1
1	66	7772	1	.9115	.0710	.0175
2	70	7986	2	.9107	.0716	.0177
3	69	8116	1	.8484	.1198	.0318
4	80	8213
5	68	8350	1	.6856	.2367	.0777

6	67	8494	2	.5332	.3282	.1386
7	72	8569	1	.6799	.2405	.0796
8	73	8642	1	.6653	.2501	.0846
9	70	8732	1	.4678	.3593	.1729
10	57	8799	2	.0689	.2183	.7128
11	63	8862	3	.1456	.3355	.5189
12	70	9008	3	.2620	.3969	.3411
13	78	9044	1	.5580	.3150	.1271
14	67	9078	1	.1436	.3335	.5229
15	53	9155	3	.0114	.0475	.9412
16	67	9233	3	.0915	.2626	.6459
17	75	9250	3	.2760	.3988	.3252
18	70	9299	3	.1200	.3060	.5739
19	81	9341	2	.4445	.3687	.1868
20	76	9370	2	.2341	.3904	.3754
21	79	9407	1	.3129	.3996	.2875
22	75	9434	3	.1724	.3589	.4687
23	76	9461	2	.1848	.3675	.4477
24	58	9508	3	.0085	.0361	.9554
25	31	9524

Number of cases read: 25 Number of cases listed: 25

Note that PLUM does NOT provide predicted probabilities for Case 25, which was missing on distress (and which, of course, is the case we most wanted the predicted probabilities for!) Stata handles this better.

Plum also provides a means of testing whether the assumptions required for its use are reasonable:

Test of Parallel Lines^a

Model	-2 Log Likelihood	Chi-Square	df	Sig.
Null Hypothesis	37.594			
General	36.827	.767	2	.682

The null hypothesis states that the location parameters (slope coefficients) are the same across response categories.

a. Link function: Logit.

If this chi-square value is insignificant (which it is) then the use of Plum is justified. If the value is significant, then you may want to consider another approach, such as the multinomial or generalized ordered logit models presented later. SPSS's Test of Parallel Lines is identical to the Likelihood Ratio test using `gologit2` that was presented earlier.