

Logistic Regression in Stata

* Here's a description of how to do logistic regression, as well as ordinal & multinomial logit regression, in Stata. The examples use the UCLA-ATS data set **hsb2.dta**. For logistic regression, for the outcome variable use the dummy variable **hsci** (science achievement score>60); if necessary create it yourself. For ordinal logistic regression you'll create the outcome variable **sci3**, as described later on. For the multinomial logistic example we'll use **race** as the outcome variable. We'll begin with logistic regression, then we'll do brief examples of ordinal & multinomial logit. Rick Tardanico, March 2008.

* See 'Explanatory variables in logistic regression.doc'

* Open & examine the univariate characteristics the data set

u hsb2, clear

```
d
su
hist read, norm          [e.g.]
gr box read             [e.g.]
tab ses                 [e.g.]
```

* The dependent variable is **hsci** (science achievement score>=60). **Graphically & numerically examine the pertinent univariate, bivariate & (insofar as possible) multivariate distributions, consider possible transformations or other manipulations, & perhaps save a new data set consisting of listwise (i.e. 'complete') observations (using the commands 'mark' and 'markout' [see Long/Freese]).**

* Select the explanatory variables

* The procedure we'll use, as outlined in Hosmer & Lemeshow, **preliminarily** selects independent indvariables based on, first, substantive & theoretical relevance, and second, on pvalues<=.25 (which the procedure later modifies in view of more complete sets of variables and the modelling of nonlinearities). Begin by testing the potential explanatory variables with the dependent variable in 'mini' logit & logistic models.

* Note: 'logit' yields the logit coefficient, while 'logistic' yields odds ratios. Specifying the option 'or' after logit makes logit display odds ratios. The Stata manual emphasizes that the only distinction between 'logit' & 'logistic' is logit coefficients versus odds ratios. The conclusions reached by the two approaches are identical.

```
ci hsci, binomial          [options: agresti, jeffreys, wilson]
scatlog hsci math, ci      [download 'scatlog']
logit hsci math, or nolog
estimates store full
logit hsci, or
lrtest full
```

* Do the preceding graph & the tests--for each potential explanatory variable, & document the results for later reference. Regarding the tests, those variables that obtain pvalues $\leq .25$ will be included in the 'preliminary main-effects model'. Let's suppose that we've done so. The following is our preliminary model, which we'll test for significance. If our sample is unweighted we can use the likelihood ratio test to assess nested models. If the sample is weighted we use Wald tests (i.e. Stata's 'test' command in logit/logistic regression). But first, run all of the preliminary variables in an OLS regression in order to run a multicollinearity test (see Menard on the justification for doing so).

```
reg hsci read write math socst race ses prog
vif
```

* There appears to be no multicollinearity problem.

* Here, then, is our preliminary 'main effects' model. Note: 'listcoef' enables us to list the coefficients as odd ratios ('factor' option), percents ('per' option), or standardized ('std' option), as well as 'reverse' (which is helpful in interpreting negative odds ratios).

```
xi:logit hsci female i.race i.ses schtyp i.prog read write math socst, or nolog
listcoef, per help
est st f
test _lrace_2 _lrace_3 _lrace_4
test _lses_2 _lses_3
test _lprog_2 _lprog_3
```

* Compare the coefficients as well as their significance levels to those obtained in the preliminary, bivariate models.

* **'logistic...' is an alternative command to 'logit...', or'. The latter may work better than logistic, however, for many of the commands developed by Long/Freese.**

* Next we'll try a reduced model, eliminating the variables that tested insignificant. We'll use lrtest to test the nested models (or, if the sample is weighted, Wald-test [i.e. 'test']; recall that the #obs must be the same for each tested model.

```
xi:logit hsci female read write math, or nolog
listcoef, per help
estimates store 1
lrtest full 1
```

* The test supports the reduced model (1) versus the full model. Compare the coefficients & significance levels to those obtained in the preliminary, bivariate models.

* Can we improve the model? Recall that race=4 (whites) apparently tested significant in model 0 (though keep in mind the caveats concerning multiple significance testing). Substantively it could prove worthwhile to explore whether a white/nonwhite binary coding (which I'll call 'white') would improve the model. Let's find out.

```
xi:logit hsci female white read write math, or nolog
estimates store 2
lrtest 1 2
```

* The likelihood ratio test supports model 2 (which includes 'white') versus model 1. Thus we'll use model 2 as our new 'preliminary model'. Next: would a log-transformation or

squaring any of the independent quantitative variables improve the model? We'll use both 'sparl' & 'fracpoly' to explore these options. I'll display the commands for 'hsci' & 'read' only.

```
sparl hsci read
sparl hsci read, logx
sparl hsci read, quad
```

```
fracpoly logit hsci read, compare
fracplot read
```

* None of the tested transformations of 'read' looks at all promising (especially given that untransformed variables tested highly significant in the first place). I did the same tests for write & math, with the same results. I won't bother testing any of the squared terms in the model. Next: would interaction terms improve the model? I'll test these by incorporating them, one by one, into the preliminary main-effects model.

* Note: using pweight automatically gives robust standard errors, which preclude the use of diagnostic tools (& preclude using 'lrtest'; use 'test, which is a Wald-test, instead [see the Stata manual]).

* If you need to use pweight, a strategy is, first, to estimate the model without pweights and do the diagnostics; and second, to reestimate the model with pweight, making the diagnostic-based corrective changes after doing so (see Hosmer & Lemeshow).

```
xi:logit hsci female white read write math, or nolog
xi:logit hsci female white read write math femXread, or nolog
xi:logit hsci female white read write math femXwrite, or nolog
xi:logit hsci female white read write math femXmath, or nolog
xi:logit hsci female white read write math whiteXread, or nolog
xi:logit hsci female white read write math whiteXwrite, or nolog
xi:logit hsci female white read write math whiteXmath, or nolog
```

* We could have used Stata's interaction shortcut, e.g.:

```
xi: logit hsci ..... i.white*math, or nolog
```

* None of the interaction terms tests significant (according to the Wald tests, which are the basis of the reported pvalues in Stata's logistic regression). Next: I'll add all of the originally tested variables (but keeping 'white' instead of 'race') to see if any of them ends up testing significant in this configuration of explanatory variables.

```
xi:logit hsci female white read write math i.ses schtyp i.prog socst, or nolog
estimates store full
lrtest full 2
```

* Again, compare the coefficients & significant levels to those obtained in the preliminary, bivariate regressions. Then re-run model 2.

```
xi:logit hsci female white read write math, or nolog
```

* Compare model 2's coefficients & significance levels to those in the immediately preceding, "full" model, and of course assess the lrtest's evidence (which is only valid, however, if the #observations in the two models is the same). lrtest again supports model 2.

* The next two steps are very tedious but important: First, estimate model 2, then re-estimate it while sequentially removing one variable at a time, testing each reduced model against model 2 via lrtest (if in each case the #observations is the same). Second, repeat the procedure, but this time removing substantively/theoretically chosen sets of variables, again testing each reduced model against model 2 via lrtest (when the #observations is the same). The importance of these steps is that, even if there is no evidence of multicollinearity, any particular variable or set of variables may test significant or insignificant depending on the presence or absence of particular other variables. (For example, this occurred in key ways in my model building with data on household reported landslide & flood damage in San Salvador & Tegucigalpa.)

We'll now submit model 2 to the diagnostic tests. We'll begin by running 'model fit tests' (ldev & lfit, g(10)) & a model specification test 'linktest'). Then we'll run 'outlier/influence' tests. We'll first recall model 2 into Stata's memory by typing 'logit, or'.

```
logit, or
ldev
estat gof, g(10) table
```

* The 'fit' test looks fine (see Hosmer & Lemeshow). If lfit has cells with zero observations or with fewer than 5 observations, reduce the #groups, but the minimum should be six groups. Next: a model specification test ('linktest').

```
linktest
```

* 'linktest' looks fine, too. Next we'll examine outliers/influence.

```
predict p if e(sample)          [predicted probabilities for the obs. used in the models]
predict db if e(sample), db
predict dd if e(sample), dd
predict dx if e(sample), dx
predict h if e(sample), h
predict n if e(sample), n
```

```
la var p "predicted probability"
```

```
histogram p, norm
su p-n, detail
```

* For the purposes of this exercise, we'll only examine 'db'. By the way, we'll examine 'n', which refers to 'covariate patterns,' by including it as our plotting symbol, 'ml(n)'. **Consider also using 'ml(id)' or 'ml()' for other explanatory categorical vars to explore their relationship to influence patterns.**

* In summarizing p-n, we see that 'dd' (basically, 'deviance' residuals) & 'dx' (basically, a model chi-sq measure) seem to include outliers. But they don't seem to cause any problems for 'db', which is the principal indicator of whether deleting an observation would change the model's fit. Roughly speaking, 'db' > 1 suggests that an outlier has undue influence. Recall that influence diagnostics do not include significance tests, so there's lots of subjective judgment.

```
[Begin by referring above to su db, d]
scatter db p [weight=dx], yline(1) ml(n)
```

sort db

list id hsci p db female white if db>1 [consider also listing: & db<.]

* We graphed 'db' versus p, weighting the plot by 'dx' & using 'n' as our plotting symbol.

* Let's take a look at covariate=18 (i.e. 'n'=18), which is an outlier.

sort id

list id p db dd dx h hsci science female race white ses read write math if n==18, compress

* n=18 is an outlier due to the disjuncture between low science/hsci scores versus relatively high read, write & math scores. Even so, it doesn't cause any notable problems for the model, so there's no need to exclude it. Note: the id#'s of 'n' are not necessarily stable from one round of model estimations to another. Thus if you do need to exclude one or more n-sets, its best to do so by specifying not the particular n-sets but rather the principal diagnostic statistic in question (typically 'db') or the case id#'s. E.g.: logistic ycat x1 x2 if db<10. Or: logistic ycat x1 x2 if id~ =64 & id~ =130.

* First, examine the pattern of predicted probabilities to gain insight into the model. Here's a partial example.

hist p, norm

* **Note: to use postgr3, begin the regression model with 'xi3' for categorical explanatory variables.**

xi3:logit hsci i.female i.white read write math, or nolog

postgr3 math, gr(s(pd))

postgr3 math, by(white) gr(s(pd))

hist p, by(white, total)

bys white: su p

sort p

list id science hsci n p dd h dx db female race in 1/25, compress

list id science hsci n p dd h dx db female race in -1/25, compress

* Here, again, is our final model.

logit hsci female white read write math, or nolog

listcoef, per help

* Note: Hosmer & Lemeshow are among those who argue that we shouldn't report 'pseudo R2' in published results because its metric is not comparable to that of OLS R2. They, Pampel, & Long/Freese argue, in addition, that we shouldn't report results for categorical data as standardized coefficients because doing so doesn't make substantive sense. Instead report Hosmer-Lemeshow chi2 (estat gof, group()).

* **Graph estimated probability & its confidence interval**

* Quickest way:

scatlog hsci math, ci

* Standard way:

```
twoway fpfitci hsci math
```

* Long-hand way:

```
predict p if e(sample) [already done above]
la var p "predicted probabilities"
predict se if e(sample), stdp
la var se "standard error"
display invttail(193, .05/2)
gen lo = p - 1.97*se
la var lo "low end"
gen hi = p + 1.97*se
la var hi "high end"

scatter p lo hi read if female==0, c(l l l) sort
scatter p lo hi read if female==1, c(l l l) sort
```

* Predicted probabilities:

Let's explore our final model from the standpoint of predicted probabilities (see Long/Freese 2001). Let's start by using 'postgr3' to graph predicted probabilities (for options type 'viewer help postgr3', & see UCLA-ATS for in-depth, downloadable examples).

```
xi3:logit hsci i.female i.white read write math, or nolog
```

```
postgr3 read, gr(s(pd))
postgr3 read, by(female) gr(s(pd))
postgr3 read, by(female) x(math=40) gr(s(pd))
postgr3 read, by(female) x(math=60) gr(s(pd))
```

* Of course, we could further apply 'postgr3' to examine the model, but for now let's apply the commands 'prchange', 'prtab' & 'prvalue' to display configurations of predicted probabilities. (See also the commands 'lincom' [which produces linear combinations of predicted values] & 'adjust' [which produces adjusted means or proportions, including probabilities, & whose output is similar to prvalue's & prtab's]).

* Note: the following set of commands works best **without** 'xi2: ..i.' 'xi3: ...i.' or 'xi:..i'.

```
logit hsci female white read write math, or nolog
```

```
ci hsci, binomial [options: agresti, jeffreys, wilson]
su p, d
hist p, norm
prvalue, r(mean) [CI options: delta, ept]
prvalue, r(median) [CI options: delta, ept]
```

* We've just displayed the model's predicted probabilities with 'all variables held constant at their means' & 'all held constant at their medians'. Let's see which particular explanatory variables exert the strong effects.

```
prchange, fromto brief
```

* Focusing on 'from' 'to' & 'diff' (i.e. on 'x=min', 'x=max' & 'min>max'), we see that, in this model, each of the explanatory variables seems to have a notable effect on the predicted probability of hsci=1. Let's take a further look.

```
prtab female, brief
```

```
prtab white, brief
```

```
prtab female white, brief
```

```
prtab female white, x(read=40 write=40 math=40) brief
```

```
prtab female white, x(read=60 write=60 math=60) brief
```

* Let's look at configurations for white males versus nonwhite males.

```
prvalue, x(white=1 female=0 read=40 write=40 math=40) delta save brief
```

```
prvalue, x(white=0 female=0 read=40 write=40 math=40) delta dif brief
```

* So, even at the low end of read-write-math scores, whites are considerably more likely to achieve hsci=1.

```
prvalue, x(white=1 female=0 read=60 write=60 math=60) delta save brief
```

```
prvalue, x(white=0 female=0 read=60 write=60 math=60) delta dif brief
```

* The same is true at the higher end of standardized scores. Let's do a graph, focusing solely on whites versus nonwhites & holding math scores constant at 40, 50 & 60. We'll use the command 'prgen' to format the data for graphing.

```
prgen white, x(math=40) gen(pm40) ncases(11) brief
```

```
prgen white, x(math=50) gen(pm50) ncases(11) brief
```

```
prgen white, x(math=60) gen(pm60) ncases(11) brief
```

```
la var pm40p1 "math 40"
```

```
la var pm50p1 "math 50"
```

```
la var pm60p1 "math 60"
```

```
gr pm40p1 pm50p1 pm60p1 pm60x, c(sss) s(OTS) xlabel(0 1) ylabel(0 .2 to .8)
```

```
b1("Predicted Science Achievement: Whites & non-Whites") b2("non-white=0 white=1")
```

```
l2(Pr(Hsci=1)) g(3)
```

* The graph charts the advantages of whites versus nonwhites in the probability of scoring relatively high in science at every graphed level of math scores. (see Long/Freese on techniques for incorporating curvilinearities in their set of commands for predicting & graphing probabilities, including the graph command 'praccum').

* Here's another--perhaps easier way--to compute and graph predicted probabilities for continuous independent variables in logistic regression.

```
findit predxcon
```

[download]

help predxcon

```
logit hsci female white read write math, or nolog
predxcon hsci, x(read) from(28) to(76) adjust(female white write math) graph
predxcon hsci, x(read) f(28) t(76) class(female) adj(white write math) graph
```

*** Ordinal logistic regression:**

* Let's take a brief look at ordinal logit regression in Stata. The model-building steps should be the same as for logistic regression. After estimating an ordinal logit model, the test 'brant' assesses the assumption of proportional odds (see Long/Freese for alternative forms of logistic regression if 'brant' rejects the assumption). The commands postgr, prvalue, prchange, prtab, prgen & praccum can be used in ordinal logit.

* The dependent variable is **sci3**, which you need to create.

* Create sci3:

```
. su science, d
. gen sci3=science
. replace sci3=1 if science<=44
. replace sci3=2 if science>=45 & science<=57
. replace sci3=3 if science>=58 & science<.
. bys sci3: su science
. la def s 1 "min-44" 2 "45-57" 3 "58-max"
. la val sci3 s
. la var sci3 "science scores: levels 1, 2 & 3"
. note sci3: TS - I created sci3 as follows: replace sci3=1 if science<=44; replace sci3=2 if
science>=45 & science<=57; replace sci3=3 if science>=58 & science<. ; bys sci3: su
science.
. compress
. sa, replace
```

* Next, run an OLS regression & test for multicollinearity, using the quantitative explanatory variables.

```
reg sci3 read write math
vif
```

* There's no multicollinearity problem.

* Now estimate the ordinal logit.

```
ologit sci3 female white acad read write math, table
```

```
listcoef, per help
est st f
brant
linktest
```

```
predict p1 p2 p3 if e(sample)
su p*, d
```

dotplot p*

* See Hosmer & Lemeshow on how to carry out diagnostics in ordinal (& multinomial) logit. The UCLA-ATS Hosmer & Lemeshow text module demonstrates how to do the diagnostics in ordinal (& multinomial) logit. Here's a brief example. **Use logit, not ologit, to do the approximation.**

```
logit sci3 female white acad read write math if sci3~=3, table nolog
predict p1 if e(sample)
su p1, d
hist p1, norm
predict db if e(sample), db
predict dd if e(sample), dd
predict dx if e(sample), dx
predict h if e(sample), h
predict n if e(sample), n
```

* Proceed with the diagnostics as demonstrated under logistic regression. Then repeat the same set of procedures by estimating ologit if sci3~=0.

* **Multinomial logit regression:**

* Finally, let's do an example of multinomial logit, for which the model-building steps should be the same as those we've previously discussed. In estimating a multinomial logit, use the option 'base' to specify which category will serve as the reference category. The dependent variable is **race**.

* In this case I've chosen 'White'(4) (versus Hispanic, Black & Asian) as the base (i.e. comparison or reference) category. The likelihood ratio test is carried out for all of the categories via 'mlogtest', which also tests (via 'lcombine') whether each of the dependent variable's categories can be considered statistically independent of the others & tests (via both 'hausmann' & 'sm') 'the independence of irrelevant alternatives'. I will not discuss building this model. See Long & Freese, as well as Hosmer & Lemeshow, on the procedures. The tools postgr, pvalue, prchange, & prtab can be used for multinomial logit (using the option 'outcome' to specify the dependent-variable level), but graphing predicted probabilities is done via mlogview & mlogplot (see Long & Freese).

* Begin by running an OLS regression & testing for multicollinearity, using the quantitative explanatory variables.

```
reg race read write math science socst
vif
```

* There's no multicollinearity problem.

* Estimate the multinomial logit. The 'rr' option yields relative risk ratios, while 'nolog' suppresses display of the log-likelihood iterations. Using 'listcoef' with 'factor' or 'per' makes the 'rr' option unnecessary.

```
xi:mlogit race i.ses schtyp i.prog acad read write math science socst, base(4) rr nolog
```

```
listcoef, per pv(.1) help
```

mlogtest, lr

xi:mlogit race i.ses write math science, base(4) rr nolog

listcoef, per pv(.1) help

mlogtest, lr

mlogtest, lrc sm

predict prAfr prHisp prAsian prWh if e(sample)

su prAfr-prWh, d

dotplot prAfr-prWh

* See the comments under ordinal logit concerning how to do the 'approximated' diagnostics in multinomial logit (Hosmer & Lemeshow).

* Graph the predicted probabilities using 'mlogview' & 'mlogplot'. See Long & Freese for details.

* **Some other useful commands:**

* Here are some other useful commands for logistic regression: probpred, relrisk, logsub.

* After estimating a model: estat ic [to obtain AIC and BIC]