

Sample Statistical Workflow: Binary Logistic Regression

Adam Davey

March 8, 2015

Logistic regression is a nonlinear model, estimated using maximum likelihood methods. Empty cells cause estimation difficulties. Look for them carefully before analysis and when interpreting results. Use predicted probabilities to interpret findings rather than odds ratios alone.

Adam

1 Preamble

Binary logistic regression models the association between one or more predictors and a *dichotomous* (binary) response variable. By convention, the class of interest (e.g., those with the disease) is coded as 1, and the other class is coded as 0. Logistic regression uses a logit “link function,” a *transformed* version of the response variable. This means that logistic regression assumes linearity in the *log odds* of the response variable. Other assumptions such as correct model specification, independence of observations, and that predictors are measured without error also apply. The objective of our analysis is to predict cases and non-cases.

2 Steps in a Binary Logistic Regression Workflow

Below, we present a set of sample steps that should be performed whenever you are performing a binary logistic regression analysis. It is not necessary for you to literally perform every one of these steps with every single analysis. However, this workflow should

essentially cover the range of requirements and assumptions that you are expected to be responsible for under most circumstances. Know, however, that it is in no way exhaustive and that research conventions in your area of research may differ in important ways from the expectations for biostatistics. Likewise, many areas of biostatistics are developing very quickly and so current conventions are subject to change.

2.1 Frequencies/Summary Statistics

Note. Stata treats 0 as non-cases and *all other values, positive or negative* as cases. So, when using Stata, always confirm that your response variable only takes on values of 0 and 1 prior to beginning analysis.

Look at the univariate distribution of your predictor across both levels of your outcome variable using frequencies and/or summary statistics (M, SD, Min, Max, etc.). In Stata, you would use the `tab x y, missing` command for categorical predictors and `by y, sort: summ x` or `tabstat x, by(y) columns(statistics) statistics(n mean sd min max)` for continuous predictors. If your predictor is categorical or ordinal, be sure to look for categories with very small numbers of observations. Logistic regression will fail if there are empty cells, or cells with little or no variance (e.g., if all of the “cases” have the same value for sex). If your variables have a very small number of observations in one or more categories you will need to revise your coding so as to omit small groups or simplify coding (e.g., White/non-White). This problem is much more acute for logistic regression than simple linear regression.

2.2 Univariate plots (Y and X)

Like frequencies, univariate plots help you to understand how your distributions differ across response categories. Use the same methods to plot univariate data as for simple linear regression. At minimum, you should consider histograms (`hist x, by(y)`), boxplots (`graph box x, over(y)`), and normal quantile plots run separately using `qnorm x if y==0` and `qnorm x if y==1`). In histograms, look at the shape of the distribution. Is the distribution symmetric or asymmetric (i.e., skewed)? Is it unimodal or multimodal? Are there any gaps or bumps in the distribution? What is the range of observed values? How much of the possible range of values is actually observed? Check to ensure that the data do not contain impossible values, extreme values, and that missing value codes (e.g., -9, -99) are not being treated as observed values. Boxplots provide a different kind of information about your distributions. Specifically, depending on the software package, they display points such as the median, 25th and 75th %iles (for the box) and other functions of the interquartile range (for the whiskers) and extreme values. Normal quantile plots show you how your distribution aligns with the quantiles expected if the data followed a normal distribution. Deviations away from the 45° line show the location of deviations from normality. From these plots you can determine whether the points tend to pile up in some parts of the distribution, or if the distribution is highly skewed. These plots are also a good

way to determine whether transformations to improve the shape of a distribution (such as a square root or log transformation) have been effective. The Stata commands `ladder`, `gladder`, and `qladder` might be useful in selecting an appropriate transformation.

2.3 Testing Normality of Distributions

Clearly, a binary response variable is not normally distributed so no tests are needed of normality.

2.4 Scatterplot

Scatterplots are also not useful for a binary response variable. However, sometimes it may be useful to plot the *mean values of Y* by X, but lowess plots discussed below are almost always a better approach.

2.5 Lowess

“Linearity” is important in logistic regression. But it is not linearity in y , but rather linearity in the log odds of y . Stata has built-in facilities for estimating lowess plots with binary response variables. With a binary response variable, the command is `lowess y x, logit`. In my experience, lowess plots can look a little bumpier with binary response variables (after all, we have considerably less information than with a continuous response variable). In all other regards, proceed to estimate, evaluate, and interpret linearity (in the log odds) for lowess plots with binary response variables using the same recommendations as for simple linear regression.

2.6 Estimate Model

As with simple linear regression, it is usually a good idea to estimate models using robust standard errors. In Stata, type `logit y x, vce(robust)`. This syntax displays the intercept, and all coefficients are unexponentiated. To obtain results in the more familiar exponentiated (odds ratio) format, you can type `logistic y x, vce(robust)`. Once again, with one predictor, that’s almost always all there is to model estimation. In the multivariable framework, we will considerably elaborate the steps and tools available to us. At that point, we will also consider strategies for modeling nonlinear associations.

2.7 Evaluate Model

Like simple linear regression, logistic regression models also provide a variety of information that can be used to evaluate the results. The formal test that all regression coefficients are 0

is provided by a likelihood ratio χ^2 with regular standard errors and by a Wald χ^2 with robust standard errors. If the overall model χ^2 is not significant at our α , we stop. There is no association to interpret. With linear regression, we interpret the R^2 statistic. A pseudo- R^2 statistic is provided with logistic regression, as well, but I do not recommend using it. There are well-known problems with this index and it is rarely as informative as other approaches that we will consider below. Next, the model will provide you with estimated regression coefficients, standard errors, and the corresponding z-statistics and their p-values. Keep the following in mind when evaluating your model.

1. Did your model converge? Linear regression has a closed form solution, meaning that we can estimate parameters in a single step. The same is not true for logistic regression, which requires an iterative approach, so it is not guaranteed to converge. You can examine model convergence by looking at the iteration history along with any warning messages. If you see messages like “Not concave” or “Backing up” this is an indication that there are difficulties with estimation of your model and you should proceed carefully. It is usually a good idea to change the tolerance (a function of accuracy, or the change in parameter estimates from one iteration to the next) of estimation to a smaller value to make it more likely that estimation has converged at a global maximum. Stata has many estimation options that you can use to “tweak” estimation including the ability to change the number of iterations, the estimation procedure, and the tolerance. More important is to try and identify the source of estimation difficulties in the first place, such as empty cells.
2. Were any combinations/characteristics excluded from the analysis? This is an indication of empty cells that can cause problems for analyses and interpretation. When identified, they need to be diagnosed before proceeding.
3. Are the effects in the right (expected) direction and of a reasonable magnitude? If your predictor is participation in the treatment group, does the direction of the effect go in the right direction? Is participation in the treatment associated with better outcomes? If the effect is not in the expected direction, there may be something wrong, such as with the coding of your variables. This is the time to check. Likewise, if there were problems with estimation, Stata will usually flag these by providing coefficients (often very large), but no standard error. These are indications of problems with your analyses.
4. Do the standard errors look reasonable? If you have a large sample size, you should expect very precise estimates on the sampling distribution of the regression coefficients. If the standard errors are very large, there may be something wrong with the model and estimation. This is something we will revisit again when considering multivariable models. On the other hand, very small standard errors can also be a sign of problems with variable coding and estimation. Beware of small regression coefficients associated with large z-values. Although most statistics packages pay very careful attention to numerical accuracy, computers use something called floating point arithmetic which can sometimes introduce a great deal of imprecision.
5. Many people rely only on odds ratios for interpretation of logistic regression models.

However, I find it much more useful to think in terms of predicted probabilities. An odds ratio (OR) may be “large” (say 2.0), but it is still important to know whether that translates into differences in the outcome of 4% vs 2%, 40% vs 20%, or 80% vs 40%. Each of these pairs of predicted probabilities corresponds with the same OR. Similarly, it is possible to change the odds ratio just by the way the predictors are coded. You will get very different results when age is measured in years than decades, and when income is measured in dollars or thousands of dollars. There is also a psychological consideration. Even though odds ratios of 2.0 and 0.5 represent equivalent effects (just with opposite coding), people are still biased toward interpreting the first number as a larger effect than the second. One of the most useful commands for obtaining predicted values is `margins`. For example, to obtain predicted probabilities of being a genius for different ages observed in our sample, we would type `margins, at(age=(15(5)80))`. This tells us that while the predicted probability of being a genius at age 15 years in our sample is 9.9%, at 45 years it is 5.1%, and at 75 years it is 2.6%. (We show how to produce a plot of these values, with CIs using the sample syntax below.)

2.8 Regression Diagnostics

In contrast to linear regression, fewer options are available for evaluating models. First, let’s consider predicted values. As a reminder, our original response variable takes on values of 0 or 1 for individual cases. We can use this to consider the probability that a value is either 0 or 1 conditional on the value(s) of X. Probabilities are bounded between 0 and 1, which means that nonlinearity is likely if we try to model the association between Y and X directly. Instead, we consider transformations of Y that have more desirable properties. For example, if $Pr(Y = 1) = \theta$, then the odds are given by $\frac{\theta}{1 - \theta}$, which is bounded between 0 and ∞ . If instead, we consider the log-odds (i.e., logit), then $\ln\left(\frac{\theta}{1 - \theta}\right)$ is bounded between $-\infty$ and $+\infty$. Once Y can take on any value, linearity in predictors is more likely. This is what we model in logistic regression, but it is not what we wish to interpret. We need to transform the output from logistic regression back to the (original) probability scale. From simple linear regression, we know how to obtain \hat{y} s. We use the same approach here. Once we have the \hat{y} s, however, they can be converted to probabilities as $Pr(y_i = 1|x_i) = \frac{e^{\beta_0 + \beta_1 \times x_i}}{1 + e^{\beta_0 + \beta_1 \times x_i}}$. This looks a little grim, so remember that you already know how to calculate/obtain \hat{y} and once you have that, the predicted probabilities are just $\frac{e^{\hat{y}}}{1 + e^{\hat{y}}}$.

Stata will automatically calculate several of these variables for you and with greater precision than is available to you from the output. To obtain predicted *values* (in log-odds scale), type `predict yhat, xb`. To obtain predicted *probabilities*, type `predict phat, pr`.

Since our outcome is binary, we can use the predicted probabilities to assign observations to

the outcome for which they have the highest predicted probability. This means recoding all observations with predicted probabilities $\geq .5$ as 1 and all observations with predicted probabilities $< .5$ as 0. Now we can look at a cross-tabulation of observed and predicted class membership. This table is called a “classification table” by most statistics packages and a “confusion matrix” in machine learning. This table is obtained after estimating a logistic regression model by typing `estat class`, and gives us information about sensitivity, specificity, positive predicted value, negative predicted value, and overall correct classification rates. Sometimes, such as with rare events (such as “genius” in the simulated example), we can seem to be doing very well with our model (overall, 95% correct classification) even for a model that does not correctly predict a single “case,” the primary objective of our endeavor. We could actually do just as well by guessing that no patients were cases.

With this in mind, there is no firm reason why we need to use $p = .5$ as our cut-point to predict “cases,” particularly when dealing with very (un)common events. Typing `lsens` after a logistic regression model will plot sensitivity and specificity by predicted probabilities. This approach can often be useful for selecting a different cut-point. Useful cut-points might include the mean for y or the point where sensitivity and specificity values intersect. In this case, by choosing a much lower threshold (say .05 instead of .50), we can correctly predict about 54% of cases (but at a cost in specificity and overall accuracy).

Related to this, we can look at the receiver operating characteristic (ROC) curve by typing `lroc`. The ROC curve is a plot of *Sensitivity* on the y-axis against *(1 - Specificity)* on the x-axis. We are interested in how well our model performs in predicting the outcome above chance (the diagonal line), and we calculate this as the area under the curve (*AUC*, also call the *c*-statistic). A model that is *always* correct has $AUC = 1$, whereas a model that is *never* correct has $AUC = 0$. (It is perfectly predictive, just in a somewhat nefarious way.) A model that is useless for prediction has $AUC = .5$. Needless to say, we want the *AUC* to be as close to 1 as possible. Like the R^2 in linear regression, the “acceptable” range of *AUC*s is highly context-specific.

One of the most widely used measures of goodness-of-fit (i.e., congruence between observed and expected frequencies) is the Hosmer-Lemeshow test, which can be obtained with default parameters in Stata by typing `estat gof`. For decades, this test has been used with 10 groups, and in SPSS this remains the only choice. However, a recent article by Paul et al. (2013) examined power of the Hosmer-Lemeshow test as a function of the sample size (N) and number of groups (g). Here is a summary of their recommendations, which we will use in our implementation of the test. See the sample do-file for the full syntax.

1. If $N \leq 1000$ use the current default of 10 groups
2. If $1000 < N \leq 25000$ use the following formula where m is the number of successes

$$g = \max \left(10, \min \left[\frac{m}{2}, \frac{n - m}{2}, 2 + 8 \left(\frac{n}{1000} \right)^2 \right] \right)$$

3. If $N > 25000$ the Hosmer-Lemeshow test is not recommended

Finally, we can consider the hat test. We use our model to obtain \hat{y} (in log-odds metric). If our model is correctly specified, then nothing should predict our outcome beyond the predicted values. We test this by estimating a model which includes \hat{y} and \hat{y}^2 . If the \hat{y}^2 is significant then something has been omitted from our model. To perform the hat test in Stata, we type `linktest`.

Residuals have a somewhat different meaning in logistic regression compared with linear regression. Several different kinds of residuals are available in Stata, and we will use some of these for diagnostic plots. Standardized *Pearson residuals* are available by typing `predict spresid, rstandard`. Pearson residuals are obtained as the difference between observed and expected frequencies based on our model. *Deviance residuals* can be obtained by typing `predict devresid, deviance`. Deviance residuals reflect the difference between the observed and implied (fitted) likelihood functions, and are the actual quantity minimized in logistic regression. While we are at it, we can obtain Pregibon leverage values and influence values with `predict hhat, hat` and `predict dfhat, dbeta`.

Stata does not automatically generate any diagnostic plots for logistic regression models, but they are easy to access here. Some good choices for logistic regression models include:

- A plot of Pearson residuals against predicted probabilities `scatter dresid phat, mlab(id) yline(0)`.
- A plot of Deviance residuals against predicted probabilities `scatter dresid phat, mlab(id) yline(0)`.
- A plot of leverage values against id `scatter hhat id, mlab(id) yline(0)`.
- A plot of influence values against id `scatter dfhat id, mlab(id) yline(0)`.

When inspecting these plots, we generally look for values to be less than 2 (or 3 with larger sample sizes) in absolute magnitude. Anything outside of this range, we would inspect on a case-by-case basis to try and understand why the value is so high/low. For example, typing `list age genius if id==1 | id==247 | id==510 | id==925` tells us that we have a large number of 15 year old geniuses who are influential on the model parameter estimates. Alternatively, we see that a number of young individuals have large standardized Pearson residuals (`list age genius if spresid >= 3`). Further inspection reveals that in this case, the way we calculated “genius” status meant that everyone under age 20 was a genius, and no one over 75 (actually over 71) was a genius. This will cause their values to stand out from the others. Had we used real data (or simulated age differently, we would have obtained different results.

2.9 Tabling Results

Different disciplines have different expectations for how results from logistic regression models should be presented. In Public Health, critical information typically includes unexponentiated estimated regression coefficients, upper and lower confidence limits, and a p-value to 4 decimals. Many areas of medicine and the social sciences prefer presenting

exponentiated coefficients (odds ratios) along with their upper and lower confidence limits, z-values, and associated p-values. It will be a little more involved, but we will once again have Stata help with tabling logistic regression output to avoid inadvertently introducing errors with through typing and cut-and-paste. See the stata do-file for syntax to export results into an Excel file.

Table 1: Sample Logistic Regression Table Output in Stata (Robust SEs)

		(1)	(2)	(3)
		logitrob		
Outcome	Predictor	genius	ci	pval
genius	age	-0.02	[-0.041 - -0.007]	0.006
	Constant	-1.85	[-2.629 - -1.064]	0.001
Observations		1,000		

2.10 Writing Up Results

Prior to analysis, distributions for age were examined for genius and non-geniuses. Preliminary evidence for log-linearity of association between age and genius was examined by way of a lowess curve of logit-transformed genius. This plot showed some evidence of nonlinearity at the upper and lower extremes of age. Genius was regressed on age in logistic regression with robust standard errors. The overall model suggested a significant association between age and genius (Wald $\chi^2(1) = 7.51, p = 0.0061$) although age accounted for only a small proportion of the variance in genius (Pseudo- $R^2 = 0.02, AUC = 0.61$), and inspection of the classification table indicated that the model did not correctly predict any cases. Age was significantly negatively associated with the probability of being a genius, OR=0.976, 95%CI [0.960 – 0.993, $z = -2.74, p = .006$ suggesting that each additional year of age was associated with 2.4% lower odds of being a genius. (The 2.4% comes from $(1 - 0.976) \times 100\%$.) The regression equation was $\text{logit}(\textit{genius}) = -1.84 - 0.023 \times \textit{age}$. Model regression coefficients, confidence intervals, and p-values are shown in Table 1. A variety of regression diagnostic plots were evaluated for violations of assumptions. Inspection of leverage points and influential cases revealed that all individuals below age 20 were geniuses and that no individuals aged 72 years and older were geniuses. [Note that right here I am acknowledging the fact that my model really isn't linear in age, which was what we saw on the lowess plot. In fact, except at the extremes, the association is zero. Interpretation of the model is just for the purposes of this exercise. If we had real data, we would work on specifying the actual association between age and probability of being a genius.]

Checklist: Logistic Regression

- Frequencies and summary statistics for predictors by each level of the response variable.
- Univariate plots (histograms, box plots, Q-Q plots) for predictor by each level of the response variable.
- Lowess plot of logit response by predictor.
- Estimate model with normal and robust standard errors.
- Evaluate model. Proceed if χ^2 test significant; otherwise stop. Verify that model converged, coefficient is in the expected direction and that estimates and standard errors appear to have been estimated appropriately.
- Evaluate regression diagnostics. Calculate fitted values and residuals; evaluate variety of regression diagnostic plots (residuals by predicted probabilities, leverage and influence by id). Follow-up on influential cases and leverage points.
- Table results. Results should include unexponentiated regression coefficients, standard errors, z-values, and p-values, or else odds ratios, 95% CI, and p-values, plus number of observations used in analysis and model χ^2 . Format your results in accordance with the style and conventions for your area of research.
- Write up results; note any violations of assumptions and their effects, if any, on results. Refer to model table.

Stata Syntax: Logistic Regression

```
/******  
* Sample Logistic Regression Syntax  
* PH 8012, Spring 2015  
* Adam Davey  
* Requires outreg2  
* Type: findit outreg2 to install  
* Requires lrdrop1  
* Type: findit lrdrop1 to install  
*****/  
  
#delimit;  
clear all;  
capture log close;  
log using "mylogistic.log", replace;  
*log using "mylog.smcl", replace;  
  
* Below simulates some data for the workflow;  
set seed 8675309;  
set obs 1000;  
* As an aside, genius is usually 140+ for IQ;  
* This results in too few cases, so we will allow those Mensa morons;  
* Just imagine air quotes around the variable name;  
generate genius = _n <= 50;  
generate age = int(20 + 60*runiform() - 7*genius + 7*0.05);  
* The lines below are for in-class example;  
generate old = age>=65;  
generate older = age >=70;  
generate oldest = age>=75;  
gen id = _n;  
  
* Step 2.1 Frequencies / Summary Statistics;  
tab1 age genius, missing;  
by genius, sort: summarize age;  
* Alternatively, use syntax below for considerably more information;  
*summarize iq age, detail;  
  
* Step 2.2 Univariate Plots;  
histogram age, by(genius);  
graph export "hist_age.pdf", replace;  
graph box age, by(genius);  
graph export "box_age.pdf", replace;  
qnorm age if genius==0;  
graph export "qnorm_age0.pdf", replace;
```

```

qnorm age if genius==1;
graph export "qnorm_age1.pdf", replace;

* Step 2.3 Testing Normality of Distributions;
* NA;

* Step 2.4 Scatterplot;
* NA;

* Step 2.5 Lowess;
lowess genius age, logit;
graph export "lowess_genius_age.pdf", replace;

* Step 2.6 Estimate Model;
* First, estimate and store model with normal theory SEs;
logit genius age;
estimates store logitnorm;
logistic genius age;
estimates store logisticnorm;

* Then estimate and store model with robust SEs;
logit genius age, vce(robust);
estimates store logitrob;
logistic genius age, vce(robust);
estimates store logisticrob;

* Step 2.7 Evaluate Model;
* No additional analyses or syntax for this section, so here's a handy figure;
* Remember to check observed range for x-variable;
margins, at(age=(15(5)80));
marginsplot, xlabel(15(5)80) recast(line) recastci(rarea);
graph export "logit_margins_cis.pdf", replace;

* Step 2.8 Regression Diagnostics;
* Classification table (confusion matrix);
estat class;

* ROC curve with AUC statistic;
lroc;
graph export "roc.pdf", replace;

* Sensitivity-Specificity x phat;
lsens;
graph export "sensitivity_specificity.pdf", replace;

```

```

* Calculate number of groups based on Paul et al 2013 criteria;
quietly {;
local n = _N;
if 'n' < 1000 {;
local numgps = 10;
noisily: estat gof, group('numgps');
};
else if 'n' >= 1000 & 'n' <=25000 {;
quietly sum genius;
local m = r(sum);
local numgps = int(max(10,min('m'/2,('n'-'m')/2,(2 + 8*('n'/1000)^2))));
noisily: estat gof, group('numgps');
};
else if 'n' > 25000 {;
local numgps = .;
noisily: di "Hosmer-Lemeshow not recommended when N = 'n'";
};
};
lsens;
linktest;

* Various predicted values;
predict yhat, xb;
predict phat, pr;
predict spresid, rstandard;
predict devresid, deviance;
predict hhat, hat;
predict dfhat, dbeta;

* Various diagnostic plots;
scatter spresid phat, mlab(id) yline(0);
graph export "spresid_x_phat.pdf", replace;
scatter devresid phat, mlab(id) yline(0);
graph export "devresid_x_phat.pdf", replace;
scatter hhat id, mlab(id) yline(0);
graph export "hhat_x_id.pdf", replace;
scatter dfhat id, mlab(id) yline(0);
graph export "dfhat_x_id.pdf", replace;

* Step 2.9 Tabling Results;
* Example 1 -- direct to Excel;
quietly: estimates replay logitrob;
putexcel set "logistic.xls", sheet("Logit Robust SEs") replace;
putexcel F1=("Number of obs")          G1=(e(N));
local chi2type = e(chi2type);

```

```

putexcel F2=("‘chi2type’ Chi-square") G2=(e(chi2));
putexcel F3=("df") G3=(e(df_m));
putexcel F4=("Prob > Chi-square") G4=(e(p));
putexcel F5=("Pseudo R-squared") G5=(e(r2_p));
matrix a = r(table)';
matrix a = a[.,1..6];
putexcel A6=matrix(a, names);
quietly: estimates replay logitnorm;
quietly: estimates replay logisticrob;
putexcel set "logistic.xls", sheet("Logistic Robust SEs") modify;
putexcel F1=("Number of obs") G1=(e(N));
local chi2type = e(chi2type);
putexcel F2=("‘chi2type’ Chi-square") G2=(e(chi2));
putexcel F3=("df") G3=(e(df_m));
putexcel F4=("Prob > Chi-square") G4=(e(p));
putexcel F5=("Pseudo R-squared") G5=(e(r2_p));
matrix a = r(table)';
matrix a = a[.,1..6];
putexcel A6=matrix(a, names);
quietly: estimates replay logitnorm;
putexcel set "logistic.xls", sheet("Logit Normal SEs") modify;
putexcel F1=("Number of obs") G1=(e(N));
local chi2type = e(chi2type);
putexcel F2=("‘chi2type’ Chi-square") G2=(e(chi2));
putexcel F3=("df") G3=(e(df_m));
putexcel F4=("Prob > Chi-square") G4=(e(p));
putexcel F5=("Pseudo R-squared") G5=(e(r2_p));
matrix a = r(table)';
matrix a = a[.,1..6];
putexcel A6=matrix(a, names);
quietly: estimates replay logisticnorm;
putexcel set "logistic.xls", sheet("Logistic Normal SEs") modify;
putexcel F1=("Number of obs") G1=(e(N));
local chi2type = e(chi2type);
putexcel F2=("‘chi2type’ Chi-square") G2=(e(chi2));
putexcel F3=("df") G3=(e(df_m));
putexcel F4=("Prob > Chi-square") G4=(e(p));
putexcel F5=("Pseudo R-squared") G5=(e(r2_p));
matrix a = r(table)';
matrix a = a[.,1..6];
putexcel A6=matrix(a, names);

* Example 2 -- also direct to Excel;
* Nicer table, but labels need to be changed;
outreg2 [logitrob logitnorm] using "logistic2",

```

```
replace excel stats(coef ci pval)
bdec(2) cdec(2) pdec(3) bracket(ci) noaster sideways;

log close;

* Convert text output to PDF;
translate mylogistic.log mylogistic.ps, replace;
!ps2pdf mylogistic.ps;
```