Logistic Regression With SAS©

Please read my introductory handout on logistic regression before reading this one. The introductory handout can be found at.

Run the program **LOGISTIC.SAS** from my SAS programs page, which is located at. Look at the program. The **NMISS** function is used to compute for each participant how many variables have missing data. The DATA CULL step deletes observations which are missing data on more than two variables or on the justify variable (which is not used in the presently presented analysis). IDEALISM and RELATVSM are computed as **mean**s on the appropriate items from Forysth's EPQ. The **dummy variables** coding scenario have their values initialized to 0 and then reset to 1 if appropriate given the value of the GROUP variable. All of the scenarios except the medical scenario are coded by having a 1 on their own dummy variable and 0's on the other dummy variables. The medical scenario is coded by having 0's on all dummy variables, and, accordingly, is the reference group to which all of the other scenarios are compared.

PROC LOGISTIC is used to predict CONTINUE (1 = support continuing the research, 2 = withdraw support for the research) from IDEALISM, RELATVSM, GENDER, and the scenario dummy variables. The **CTABLE** option is used to ask for a classification table. PROC LOGISTIC is invoked a second time on a reduced model (with the dummy variables for scenario removed) to determine if scenario has a significant omnibus effect. PROC TTEST and PROC FREQ are used to do some univariate analyses.

Look at the listing. SAS LOGISTIC predicts the probability of the event with the lower numeric code. For our data, that is CONTINUE = 1, supporting continuation of the research. Traditionally the criterion outcomes are coded 0,1, but SAS is not picky. I tried one run with the '2' codes changed to 0's and got exactly the same results except that each *B* was multiplied by minus one and each odds ratio inverted, since SAS predicts the outcome with the lower code. If you want to force SAS to predict an event that does not have the lowest numeric code, you can specify "EVENT='number'" – for example, suppose we have coded outcome variable Guilty with 0 = not guilty, 1 = guilty. We want to predict guilty. This would do the trick: `proc logistic; model Guilty(event='1') = Delib Plain Interaction; weight freq; run;`

Note that 128 of 315 participants supported continuation of the research. The **–2 LOG L** statistic measures how well the model predicts the probability of the event. The smaller this statistic the better the model. We are given this statistic for a model that contains only the intercept (–2 LOG L = 425.566, *df* = 1) and for the full model (with intercept and all 7 predictors, –2 LOG L = 338.060, *df* = 8). The difference between these two values of –2 LOG L is the **Chi-Square for Covariates**, which indicates that adding the 7 predictors significantly improves our model.

The most convenient way to test individual predictors' partial effects is with the Wald test. SAS gives us for each predictor its logistic regression coefficient (***b***, "parameter estimate"), the standard error thereof, and the **Wald** χ^2 (which equals the square of $b \div SE$, and is on 1 *df*). Note that we have significant partial effects for idealism, relativism, gender, and two of the scenario dummy variables, theory and meat (probability of approval for these two scenarios is significantly less than it is for the medical scenario).

Odds ratios provide a method of describing the strength of the partial relationship between an individual predictor and the predicted event. The odds ratio are computed quite simply as e^b. For example, for the gender variable, $e^{1.2551}$ (on my calculator enter 1.2551, hit shift ln) = 3.508. This means that the odds of approving the research if the respondent is male (which was coded 2, female was coded 1 -- some programs insist on coding 0,1, but SAS treats 1,2 just like 0,1) are 3.5 times as high as the odds for approving the research if the respondent is female. It might help to consider an univariate odds ratio. Look at the PROC FREQ output at the end of the listing. The odds of approval for a male respondent are 68 / 47 (approval 1.45 times more likely than nonapproval). For a female respondent the odds are 60 / 140 (approval only .43 as likely as nonapproval). The ratio of these odds, $\frac{68 \div 47}{60 \div 140} = 3.38$, shows that a man is 3.38 times as likely to approve the research as is a woman. This odds ratio differs from that given in the logistic analysis because that given in the logistic analysis is for a partial effect, that is, holding all other predictors constant.

The .496 odds ratio for idealism indicates that the odds of approval are more than cut in half for each one point increase in respondent's idealism score. Relativism's effect is smaller, and in the opposite direction. The odds ratios of the scenario dummy variables compare each scenario except medical to the medical scenario. For the theory dummy variable, the .314 odds ratio means that the odds of approval of theory-testing research are only .314 times those of medical research (or, inverting the ratio, the odds of approval for medical research are 3.18 times those for theoretical research).

The CTABLE command gives us extensive classification table output. To use our model to predict which outcome is obtained (approval or not), we need a decision rule of the form: If the probability of occurrence of the predicted event is P or higher, we predict that the event will occur; if less than P, we predict it will not. Some programs just use .5 as the P, but SAS lets you pick any value you want, or, if you don't give it a value, it shows you the statistics for many different values. There are three ways you can calculate your "success rate" in classifying observations. You could just count up the number of correct classifications and divide by the total number of predictions. This is the "Correct" percentage given by SAS. You could find the P(correct | event did occur), that is, the percentage of occurrences correctly predicted, known as the **Sensitivity**. You could find the P(correct | event did not occur), that is, the percentage of nonoccurrences correctly predicted, known as **Specificity**. Focusing on errors in prediction, you could compute the **False Positive** rate, the P(incorrect | occurrence was predicted), the percentage of predicted occurrences which are incorrect, or the **False Negative** rate, P(incorrect | nonoccurrence was predicted), the percentage of predicted nonoccurrences which are incorrect. Lower P values will be associated with greater

sensitivity and fewer false negatives, but less specificity and more false positives. Higher P values will be associated with greater specificity and fewer false positives, but lower sensitivity and more false negatives.

Look at the classification output. Here are the computations for four values of P:

	Predict the occurrence of the event if P >			
	.02	.20	.40	.50
Correct	128/315 = .406	(120+72)/315 =.610	(94+130)/315 = .711	(71+147)/315 =.692
Sensitivity	128/128 = 1.00	120/128 =.938	94/128 = .734	71/128 =.555
Specificity	0/187 = 0.00	72/187 =.385	130/187 = .695	147/187 =.786
False Positives	187/(187+128) = .594	115/(120+115) =.489	57/(94+57) =.377	40/(71+40) =.360
False Negatives	none	8/(72+8) =.10	34/(130+34) =.207	57/(147+57) =.279

I reported the percentages for P = .4, which gave nearly equal values of sensitivity (73%) and specificity (70%). P = .42 would be even more nearly equal, but lowers the overall success rate a bit.

If you wish to evaluate the omnibus effect of a **categorical (*k* > 2) predictor**, you have to delete all of its dummy variables and see if the model performs significantly worse. Look at the results of my second invocation of PROC LOGISTIC. With the scenario dummy variables out, the –2 LOG L increased from 338.06 to 346.503, an increase of 8.443 on 4 *df* (one *df* for each dummy variable). From SAS's PROBCHI function I obtained the *p*, .0766, not quite statistically significant. I chose not to report this test, as the typical reader would not appreciate such a *p*.

You can include interaction terms in logistic regression. Here is a brief example:

```
data Trial;
input Delib Plain Guilty Freq;
Interaction = Delib*Plain;
 cards;
0 0 0 13
0 0 1 14
0 1 0 8
0 1 1 27
1 0 0 22
1 0 1 8
1 1 0 29
1 1 1 1
proc logistic; model Guilty(event='1') = Delib Plain Interaction; weight
freq; run;
```

The output shows a powerful interaction for the interaction term. Follow-up analysis shows that among jurors who did not deliberate, guilty verdicts were more likely

for plain defendants than for physically attractive defendants, but among jurors who did deliberate (and had been asked to discuss the case with a predisposition to changing their opinion based on the arguments of others) guilty verdicts were more likely for physically attractive defendants than for plain defendants.

Proc Genmod

Logistic regression can also be accomplished with Proc Genmod. Look at this program:

```
data genmod;
input Group Gender Continue N;
cards;
1 1 8 34
1 2 17 28
2 1 8 38
2 2 12 26
3 1 11 42
3 2 12 21
4 1 14 43
4 2 12 20
5 1 19 43
5 2 15 20
Proc Genmod; Class Group Gender;
Model Continue/N = Group Gender / dist=binomial link=logit; run;
```

Each row in the data stream represents one combination of level of group and level of gender. The first data row shows that for group 1 (cosmetic), gender 1 (female), 8 out of 34 participants voted to continue the research.

Proc Genmod creates the dummy variables for the categorical predictor variables. If continuous predictor variables were to be included in the model they would not be included in the Class statement.

Look at the output from PROC Genmod

Analysis Of Parameter Estimates

Parameter		DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
scenario	1	1	-0.7958	0.3844	4.29	0.0384
scenario	2	1	-1.1684	0.3919	8.89	0.0029
scenario	3	1	-0.8038	0.3819	4.43	0.0353
scenario	4	1	-0.5602	0.3766	2.21	0.1368
scenario	5	0	0.0000	0.0000	.	.
gender	1	1	-1.3163	0.2538	26.90	<.0001
gender	2	0	0.0000	0.0000	.	.

Now look at the output provided by Proc Logistic when testing the same model:

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
cosmetic	1	-0.7958	0.3844	4.2858	0.0384
theory	1	-1.1684	0.3919	8.8897	0.0029
meat	1	-0.8038	0.3819	4.4288	0.0353
veterin	1	-0.5602	0.3766	2.2133	0.1368
gender	1	1.3163	0.2538	26.9031	<.0001

Probit Regression

The generalized linear model is $g(p) = \beta X$, where *p* is the probability that some event will occur, X is the predictor variables, β is the regression coefficients, and *g* is some function (the link function) of *p* which is assumed to be related to *X* in a linear fashion. In a logistic regression the logit is the link function. That is, $\ln\left(\frac{p}{1-p}\right) = \beta X$. In a probit regression the link function is the cumulative standard normal distribution. That is, $p = \Phi BX$. "Probit" stands for "probability unit." The interpretation of the regression coefficients is not as easy as it is with logistic regression (in fact, it is mysterious to me - I like odds ratios). I have been told that the models constructed with probit regression are very similar to those constructed with logistic regression. Let us see if that is so for the example above.

```
proc logistic;
 model continue = cosmetic theory meat veterin gender
 / link=probit ctable; run;
```

If you just tell Proc Logistic to use the probit link function instead of the logit, it will. Here is the output:

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
cosmetic	1	-0.4885	0.2341	4.3527	0.0369
theory	1	-0.7090	0.2360	9.0244	0.0027
meat	1	-0.4916	0.2325	4.4714	0.0345
veterin	1	-0.3433	0.2305	2.2178	0.1364
gender	1	0.8103	0.1540	27.6873	<.0001

Looks a lot like the logistic regression to me. Do note that you can also get a classification table for the probit regression, just like we did for the logistic regression.

Links

- PowerLog – Macro for calculating sample size necessary for desired power, one or more quantitative predictors.
- Wuensch's Statistics Lessons