

PROC LOGISTIC: Traps for the unwary

Peter L. Flom, Independent statistical consultant, New York, NY

ABSTRACT

Keywords: Logistic.

INTRODUCTION

This paper covers some ‘gotchas’ in SAS[®] PROC LOGISTIC. A ‘gotcha’ is a mistake that isn’t obviously a mistake — the program runs, there may be a note or a warning, but no errors. Output appears. But it’s the wrong output. This is not a primer on PROC LOGISTIC, much less on logistic regression. There are many good books on logistic regression; one such is Hosmer and Lemeshow [2000].

Each section has several subsections. First, I identify the ‘gotcha’. Then I give an example. Third, I show what evidence you have that it occurs. Fourth, I show how to fix it — in some cases, referring to other resources. In some cases, I offer an ‘explanation’ between the evidence and the solution.

GOTCHA #1 CODING 0 AND 1

DESCRIPTION

PROC LOGISTIC can be used to run logistic regression on a dichotomous dependent variable. Often, these are coded 0 and 1, with 0 for ‘no’ or the equivalent, and 1 for ‘yes’ or the equivalent. In this case, we are usually interested in modeling the probability of a ‘yes’. However, by default, SAS models the probability of a 0 (which would be a ‘no’).

For example, we might be interested in modeling the presence of a disease, with 0 meaning the person is not infected, and 1 meaning he or she is infected. To keep it simple, I will use one independent variable: sex, code as 1 for female and 0 for male. So:

```
data today;
input disease female weight;
datalines;
0 0 100
1 0 200
0 1 200
1 1 100
;;;
```

we then run PROC LOGISTIC:

```
proc logistic data = today;
  model disease = female;
  weight weight;
run;
```

and get, among other output, an odds ratio estimate of 1.39 for female, while it’s clear that men are much more likely to be infected.

EVIDENCE

The evidence that this is happening is one line in the output: {Probability modeled is disease=0} and several lines in the log:

```
NOTE: PROC LOGISTIC is modeling the probability that disease=0.
      One way to change this to model the probability that disease=1
      is to specify the response variable option EVENT='1'
```

SOLUTION

There are several solutions. The simplest is not the one mentioned in the log, but rather the DESCENDING option.

```
proc logistic data = today descending;
model disease = female;
weight weight;
run;
```

Another method is the one mentioned in the log, which is more general:

```
proc logistic data = today;
model disease(event = '1') = female;
weight weight;
run;
```

GOTCHA #2 EFFECT CODING AND CLASS VARIABLES

DESCRIPTION

When you have a categorical independent variable with more than 2 levels, you need to define it with a CLASS statement. In PROC GLM the default coding for this is dummy coding. In PROC LOGISTIC, it's effect coding. To me, effect coding is quite unnatural.

EXAMPLE

Continuing with the same example of modeling probability of infection, suppose you now have race/ethnicity as an IV, with 6 categories, as defined by the Census Bureau: White, Black/African American, Hispanic/Latino, American Indian/Alaskan Native, Native Hawaiian or other Pacific Islander, and Asian.

```
proc logistic data = today2;
class race;
model disease(event = '1') = race;
weight weight;
run;
```

and get parameter estimates that include (among much else):

Parameter		DF	Estimate
Intercept		1	-0.8527
race	AIAN	1	-0.0636
race	AfrA	1	-0.7568
race	Asian	1	0.1595
race	Lat	1	0.5650
race	NHPI	1	0.1595

and OR estimates

Effect	Point Estimate
race AIAN vs White	1.000
race AfrA vs White	0.500
race Asian vs White	1.250
race Lat vs White	1.875
race NHPI vs White	1.250

but we know that the OR estimate should be e^{OR} , and, for example, $e^{-.06} = 0.94$ not 1.

EVIDENCE

The design matrix. With the default, the design matrix looks like this:

race					
AIAN	1	0	0	0	0
AfrA	0	1	0	0	0
Asian	0	0	1	0	0
Lat	0	0	0	1	0
NHPI	0	0	0	0	1
White	-1	-1	-1	-1	-1

and each parameter estimate estimates the difference between that level and the average of the other levels.

On the other hand, with dummy (or reference) coding, it looks like

race AIAN	1	0	0	0	0
AfrA	0	1	0	0	0
Asian	0	0	1	0	0
Lat	0	0	0	1	0
NHPI	0	0	0	0	1
White	0	0	0	0	0

and each parameter estimates the difference between that level and the reference group.

SOLUTION

Use the `param = reference` option on the class statement:

```
proc logistic data = today2;
  class race / param = ref;
  model disease(event = '1') = race;
  weight weight;
run;
```

For more information (and other possible parameterizations) see the SAS documentation for PROC LOGISTIC, in particular the section CLASS variable parameterization in DETAILS

GOTCHA #3 QUASI-COMPLETE AND COMPLETE SEPARATION

DESCRIPTION

If you picture the data as a 2x2 crosstab, then quasi-complete separation occurs when one of the cells is 0. Complete separation occurs when one cell in each row and column is 0.

EXAMPLE — QUASI-COMPLETE SEPARATION

An example of quasi-complete separation is:

```
data today7a;
  input x $ y $ @@;
  datalines;
  A C A C A C A C A C
  B C B C B C B C B C B C B C B C B C
  B D B D B D B D B D
  ;;;;
```

EVIDENCE

It varies. Confidence intervals will be extremely wide. But sometimes there is a warning that there was complete or quasi-complete separation, sometimes a note. Sometimes, you get a warning that 'convergence was not attained'. With the weight option you sometimes get a note that 'observations with nonpositive frequencies or weights were excluded'. For example, if you run the data step creating data today7a, and then

```
proc logistic data = today7a;
  class x;
  model y = x;
run;
```

you get a warning in the log. But if you run this data step:

```
data today7;
input x $ y $ weight @@;
datalines;
A C 5 A D 0 B C 10 B D 5
;;;
```

and then

```
proc logistic data = today7;
class x;
model y = x;
weight weight;
run;
```

you get a note in the log.

SOLUTION

Sometimes, you can delete the offending variable. In the example, there was only one independent variable, but usually, there will be more, and one of them will be the problematic one. Alternatively, if there are multiple categories, it may be sensible to combine some of them. A third possibility is to leave the offending variable in the equation, and simply report the results for the other variables — these are still correct. You can then report the coefficients for the offending variables as inf. Finally, you can use exact inference with the EXACT option.

REFERENCES

Paul Allison’s paper at SGF 2008 has a great deal of lucid explanation of this problem [Allison, 2008], although his statement that PROC LOGISTIC gives clear diagnostic messages is, as we have seen, not always true.

GOTCHA #4 CONCORDANT AND DISCORDANT

DESCRIPTION

Part of the default output from PROC LOGISTIC is a table that has entries including ‘percent concordant’ and ‘percent discordant’. To me, this implies the percent that would correctly be assigned, based on the results of the logistic regression. But that is not what it is. It looks at all possible pairs of observations. A pair is concordant if the observation with the larger value of X also has the larger value of Y. A pair is discordant if the observation with the larger value of X has the smaller value of Y; here, X and Y are the predicted value and the actual value.

EXAMPLE

For our first example, the output looks like this: Association of Predicted Probabilities and Observed Responses

Percent Concordant	25.0	Somers’ D	0.000
Percent Discordant	25.0	Gamma	0.000
Percent Tied	50.0	Tau-a	0.000
Pairs	4	c	0.500

EVIDENCE

It is hard to find documentation of this. I couldn’t find it explained in the LOGISTIC documentation. I found a mention of ‘concordant’ and ‘discordant’ in the FREQ documentation, but it was not clear what X and Y were, until I searched SAS-L and found an explanation from David Cassell.

SOLUTION

For what I was thinking of, you need the CTABLE option on the MODEL statement, which gives the proportion correctly classified, the sensitivity, the specificity, and other measures for each of a number of cutpoints of the predicted probability level. By default, it gives probability levels from 0 to 1 at intervals of .02, but if you just want a few, you can get them:

```
proc logistic data = today3; class race sex/param = ref;
model disease(event = '1') = race sex /ctable pprob = (.25 .5 .75);
```

```
weight weight;
run;
```

which yields

Prob Level	Correct		Incorrect		Correct	Percentages			
	Event	Non-Event	Event	Non-Event		Sensitivity	Specificity	False POS	False NEG
0.250	1	0	12	11	4.2	8.3	0.0	92.3	100.0
0.500	0	6	6	12	25.0	0.0	50.0	100.0	66.7
0.750	0	11	1	12	45.8	0.0	91.7	100.0	52.2

GOTCHA #5 INTERACTIONS

DESCRIPTION

When you add interactions to a logistic model, no odds ratios are printed.

EXAMPLE

```
data today3;
input disease race $ sex $ weight @@;
datalines;
0 White F 500 0 White M 400
1 White F 200 1 White M 250
0 AfrA F 100 0 AfrA M 125
1 AfrA F 20 1 AfrA M 28
0 Lat F 100 0 Lat M 90
1 Lat F 75 1 Lat M 30
0 AIAN F 25 0 AIAN M 15
1 AIAN F 10 1 AIAN M 8
0 Asian F 100 0 Asian M 110
1 Asian F 50 1 Asian M 40
0 NHPI F 10 0 NHPI M 8
1 NHPI F 5 1 NHPI M 3
;;;
run;
```

and then

```
proc logistic data = today3; class race sex/param = ref;
model disease(event = '1') = race sex race*sex;
weight weight;
run;
```

EVIDENCE

There are no odds ratios printed.

EXPLANATION

The reason no odds ratios are printed is a bit complex. Let's take a simpler case, with two independent variables, each with two levels, and, for simplicity, equal cell sizes.

If the dependent variable is continuous, then we might get something like this (with DV means in each cell).

	Male	Female	Mean
White	40	60	50
Other	90	50	70
Average	65	55	60

Now, we can see that White males are lower than White females, but Black males are higher than Black females. The average effect of being male is to add 5, and the average effect of being White is to subtract 10. If there were no interaction, the main effects would give

	Male	Female
White	$60 + 5 - 10 = 55$	$60 - 5 - 10 = 45$
Other	$60 + 5 + 10 = 75$	$60 - 5 + 10 = 65$

But if we put proportions in the table and estimate odds ratios, things are trickier. Start with proportions:

	Male	Female	Mean
White	.4	.6	.5
Black	.9	.5	.7
Average	.65	.55	.6

and it still works just like means. But change to odds:

	Male	Female	Mean
White	2 to 3	3 to 2	1 to 1
Black	9 to 1	1 to 1	7 to 3
Average	13 to 7	11 to 9	3 to 2

or, dividing through

	Male	Female	Mean
White	.67	1.5	1
Black	9	1	2.33
Average	1.86	1.22	1.5

Notice that the average odds is not the average of the odds....e.g. $1.86 \neq (.67 + 9)/2$. What would the main effects model predict?

SOLUTION

Some people suggest using the CONTRAST statement, but I (and many others) find these confusing and prone to error for interactions. I wasn't able to find a good reference for this method — that is, one that explained clearly how to do it.

But — SAS-L to the rescue — We can get what we want with a two step process: First, recode the race and sex variables as numbers

```
data today4;
  set today3;
  if race = 'White' then racenum = 0;
  else if race = 'AfrA' then racenum = 1;
  else if race = 'AIAN' then racenum = 2;
  else if race = 'Asian' then racenum = 3;
  else if race = 'NHPI' then racenum = 4;
  if sex = 'M' then sexnum = 0;
  else if sex = 'F' then sexnum = 1;
  racesex = 100*racenum + sexnum;
run;
```

Second, write a regular PROC LOGISTIC with the new variable racesex, defining the reference category:

```
proc logistic data = today4;
  class racesex (param = ref ref = '100');
  model disease(event = '1') = racesex;
  weight weight;
run;
```

which yields, among other things:

Effect	Odds Ratio Estimates		
	Point Estimate	95% Wald	Confidence Limits
racesex 0 vs 100	2.790	1.798	4.329
racesex 1 vs 100	1.786	1.148	2.777
racesex 101 vs 100	0.893	0.475	1.678
racesex 200 vs 100	2.381	0.920	6.162
racesex 201 vs 100	1.786	0.771	4.137
racesex 300 vs 100	1.623	0.940	2.804
racesex 301 vs 100	2.232	1.311	3.800
racesex 400 vs 100	1.674	0.418	6.713
racesex 401 vs 100	2.232	0.707	7.043

and we just need to remember the coding we used: 100 is African-American, 0 is male, so this is comparing each group to African-American males.

An alternate solution (also gotten through SAS-L) is to use a macro created by Paul Thompson, and presented at SUGI 31: www2.sas.com/proceedings/sugi31/203-31.pdf.

SUMMARY

PROC LOGISTIC offers many traps for the unwary. In addition to the above, one needs to be careful about Wald vs. likelihood ratio tests (SAS defaults to likelihood ratio tests for the confidence intervals and Wald tests for the parameter estimates. For more on these see Long [1997]. In addition to the ‘gotchas’ presented here, there are the usual issues of model building, variable selection, model interpretation and so on. For more on these issues see: Hosmer and Lemeshow [2000], Long [1997], Allison [1999, 2008]

REFERENCES

- D. W. Hosmer and S. Lemeshow. *Applied logistic regression*. John Wiley & Sons, New York, 2nd edition, 2000.
- Paul D. Allison. Convergence failures in logistic regression. In *SAS Global Forum*, number 360, 2008.
- J. S. Long. *Regression models of categorical and limited dependent variables*. Sage, Thousand Oaks, CA, 1997.
- P. D. Allison. *Logistic regression using the SAS system: Theory and application*. SAS Institute, Cary, NC, 1999.

ACKNOWLEDGMENTS

CONTACT INFORMATION

Peter L. Flom
515 West End Ave
New York, NY 10024
Phone: (917) 488-7176
peterflomconsulting@mindspring.com
Personal webpage: www.peterflom.com

SAS[®] and all other SAS Institute Inc., product or service names are registered trademarks or trademarks of SAS Institute Inc., in the USA and other countries. ® indicates USA registration. Other brand names and product names are registered trademarks or trademarks of their respective companies.