

Managing Overkill:

How to Use SAS PROC CATMOD to Estimate Odds Ratios

Sigurd W. Hermansen, Westat

ABSTRACT

While the CATMOD procedure in SAS/STAT[®] has powerful features, it can suffer from overkill when applied to relatively straightforward tasks. It offers too many choices of methods for estimating categorical models, but too little control over each of the methods. SAS Macro programs can help manage the standard exercise in many medical studies of estimating odds ratios from ordinary or summary SAS datasets. This presentation includes a review of programs of this type and a discussion of their use.

Categorical data modeling requires as a rule more observations and more computing power than comparable parametric modeling efforts. New methods of categorical analysis have emerged as a result of recent advances in computer technology. ...so much so, in fact, that some scientific investigators are finding it a challenge to manage the statistical programs used for categorical modeling. In medical research, for example, categorical modeling based on computer programs is progressively supplanting significance statistics computed directly from classification tables.

ODDS RATIO IN MEDICAL RESEARCH

Medical investigators have typically represented the results of studies of the association between risk factors and disease categories as four-fold classification tables with cell values $a = \#$ w/ disease and risk factor; $b = \#$ w/ disease and w/out risk factor; $c = \#$ w/out disease and w/ risk factor; and, $d = \#$ w/out disease and risk factor. (See Figure 1 for examples of four-fold classification and other equivalent data structures.). The odds ratio computed as $a/b / c/d = ad/bc$ provides a good measures of relative risk of disease when the prevalence of the disease has a small value in the population regardless of risk category and investigators can ignore other factors. More general methods represent results as a four-fold table per stratum of related factors, but for multifactor studies the computations for n four-fold tables become increasingly complex as n increases. Investigators may find it a long leap from simple four-fold tables to methods more suited to complex, multifactor studies. At this stage, categorical modeling programs such as the SAS PROC CATMOD take on a more critical role.

Figure 1. Data Structures Used in Categorical Modeling

Four-Fold Classification		
Outcome?	Risk factor?	
	Yes	No
Yes	a	c
No	b	d

Summary		
factor	outcome	freq
x_0	y_0	a
x_0	y_1	b
x_1	y_0	c
x_1	y_1	d

Contingency	
$a/(a+b)$ a	$c/(c+d) = 1 - a/(a+b)$ c
$b/(a+b)$ b	$d/(c+d) = 1 - b/(a+b)$ d

Events/Trials		
Events		
Factor	(Outcome)	Trials
x_0	a	a+b
x_1	c	c+d

The National Heart, Lung and Blood Institute Retrovirus Epidemiologic Donor Study (REDS) Medical Coordinating Center at Westat is developing categorical models of factors related to blood borne diseases in volunteer blood donors. Dr. David J. Wright is leading the statistical modeling work on several REDS studies, and he has specified the SAS PROC CATMOD programs used to estimate categorical model parameters. Dr. Robin McEntire prepared the datasets used to compute the estimates; each summarizes of around two million records of blood donations at the REDS centers. During the past two years we have developed SAS data step and macro language programs to support data summaries that range from simple classification tables to complex categorical models. We have striven to present balanced results, taking into account all of the factors that produce an observed effect, and at the same time present results as easily interpretable numbers. The process of developing a substantial number of categorical models and presenting their results as odds ratios, then repeating the process with more comprehensive data, has forced us to develop systematic methods.

THE CATMOD SOLUTION

Medical investigators grappling with specifying and interpreting categorical models seldom find enough time to understand automated procedures designed to feed data files to statistical procedures for estimating model parameters. SAS PROC CATMOD and its powerful language for specifying categorical models would seem at first glance a perfect choice for investigators interested in comparing risks across different groups where odds ratios serve as an appropriate measure of relative risks. CATMOD provides a number of useful services. It accepts a rich variety of options for specifying effects, including interactions and nested effects of qualitative and quantitative variables. For example, the statements

```
direct x;
model y=x a | b;
```

specify for a binary variable y the logistic regression of a quantitative variable x and two qualitative variables a and b , and all their interactions on y .

The CATMOD program detects the number of distinct categories found in the data for each variable in a model, and the number of interactions specified, and it prints the frequencies of each state of each variable. The investigator does not have to recode multiple

levels of a categorical variable into an array of binary variables. CATMOD automatically takes care of this tedious task.

CATMOD accepts input files in the form that the SAS PROC SUMMARY (or MEANS) uses to represent contingency tables. SUMMARY tables come in handy because for any contingency table with cell counts greater than five they occupy only a fraction of the storage space needed for the original data file. SUMMARY produces one row in the output data file for each unique combination of values in the set of variables named in the "by group". The program

```
proc summary data=<dataset name> nway;
  by <list of variable names including outcome>;
  output out=outfile;
run;
proc catmod;
  population <list of variable names excluding outcome>;
  model <binary outcome>=<list of variable names excluding outcome>;
  weight _freq_;
run;
```

computes logistic regression parameter estimates from summary data. SUMMARY creates the variable _freq_ to hold the number of instances of each unique row in the dataset prior to summary. The weight statement tells CATMOD to use the value of _freq_ to determine how many instances each row of the input table represents. Since the SUMMARY table contains exactly the same information as the original data file, CATMOD results (including parameter variance estimates) based on a SUMMARY and original table do not differ.¹

The rows of a SUMMARY table have an interesting meaning in CATMOD. It treats the explanatory variables in each set of rows that only differ in the outcome (response) as a set of arguments of a function that yields the observed proportion of each outcome in that set of rows. In this sense, the SUMMARY table contains the same information as a n-dimensional contingency table that has a cell for each matching set of values in the sample and contains in each cell the proportion of the count in one row (= _freq_) of the total count in the rows with the exact same values of explanatory variables. CATMOD captures this observed proportion, along with the cell values predicted by the categorical model and the residual difference between observed and predicted values, and as an option will store the values in output file.

The residual values serve as the basis for tests of significance reported by CATMOD. The significance statistics include χ^2 values, degrees of freedom and significance probabilities for main effects and for factor levels, interactions and models. These statistics, plus the difference in significance between a full model containing a risk factor and a reduced model excluding it, give an investigator a good look at the significance of a main effect or an interaction.² For large samples an insignificant probability for the model indicates a small sum of residual errors in the values (proportions) predicted by the model for each set of explanatory values after adjustment for degrees of freedom. In other words, the good fit of the model and the sample data has only that small chance of occurring in random samples taken from one population when the odds ratio equals one.

For calculations of model parameter estimates and significance statistics, CATMOD uses reasonably efficient numerical methods to compute estimates of categorical model parameters. Not, it appears, as efficient as the documentation claims (time required proportional to the number of parameters estimated), but generally acceptable for main effects models. Estimating interaction models with more than 100 parameters tend to take around six times as long per parameter as the smaller main effects models. Iterative searches may account for the longer run time per parameter. For all but the most complex analyses, the faster numeric processors provide acceptable turnaround times.

CATMOD OVERKILL

For all of the advantages of using CATMOD, it takes more than a little time and effort to understand how it works and to learn to use it effectively for estimating odds ratios. The abstract and general nature of the procedure introduces a few quirks. A quick review of some of the idiosyncrasies of the procedure should illustrate:

- To calculate the observed response function $[\log(p/(1-p))]$ where $p=n_{ij}/\text{Sum}(n_{i.})$ of the logistic model, CATMOD selects the smallest stored response value n_{ij} (the first value in the column containing the response variable after sorting it ascending order according to the collating sequence in effect). It ignores formats when selecting n_{ij} , yet it reports frequencies of levels of the response variable based on formatted values. For odds ratio estimates, the coding of the response variable determines the sign of parameter estimates;
- The coding of each risk factor of interest can also affect parameter estimates and corresponding odds ratios. Investigators not familiar with techniques for generating design matrices may find parameter estimates difficult to interpret;
- The investigator must rescale CATMOD parameter estimates for qualitative variables before they will match comparable estimates produced by PROC LOGISTIC or other programs that treat qualitative variables as indicator variables;
- Comparisons of odds ratio estimates require additional calculations, since CATMOD has no option for presenting parameter estimates and their confidence intervals as odds ratios and their confidence intervals.

These quirks have discouraged many potential users of CATMOD. It may take many hours of tinkering with the procedure before it produces meaningful results when applied to a specific task such as odds ratio estimation.

HOW TO ESTIMATE ODDS RATIOS USING CATMOD

Embedding CATMOD within a generic program helps the user gain control of the procedures. The generic program serves as a recipe for estimating the parameters of categorical models and converting them to understandable odds ratios. Once the investigator becomes familiar with the program he or she can rely on it to produce reliable and easily interpretable results.

To establish a baseline for a odds ratio based on model parameter estimates, the program first computes empirical odds ratios directly from one or more (depending on the number of levels of the variable) four-fold classification tables. This involves

- computing standard frequencies as an classification table and writing the table to a special list file;
- computing a SUMMARY table containing the same information as the classification table;
- for each classification table, computing the empirical odds ratio for the risk factor of interest and writing it to the special list file;
- creating a special dummy variable for each risk factor requiring odds ratio estimates;
- using CATMOD, computing parameter estimates for the risk factor and directing the estimates and standard errors to a SAS work dataset;
- reading the estimates and SE's, converting them to odds ratios and intervals and writing them to the special list file.

A SAS macro language procedure, %ordoc, takes care of the frequencies, summaries and empirical odds ratios. (See its definition in the appendix). To invoke it, specify the parameters of the procedure, as in

```

proc printto print=pre_hcv;
run;
title 'Delete this line';
%ordoc(&dsn,          /* input dataset */
  r&marker,           /* response-effect */
  riskfact,           /* explanatory variables */
  rf=1,               /* base position in above list */
  dvarl=%str("C" "N"), /* response-levels */
  dvaref=2,           /* base position in above list */
  rfi=%str(0 2),      /* levels of factor */
  rfref=1             /* base position in above list */
)

```

The print to statement preceding the macro call redirects the output produced by the procedure to a special list file. This list file holds both the empirical odds ratio estimates computed from the four-fold classification tables and the model odds ratio estimates computed at a later stage in the process.

Each risk factor for which we might need odds ratio estimates will need its own dummy variable. A SAS data step program creates one so that it has a value of 0 whenever the risk factor has the highest level (the reference value for the odds ratio). We can, if necessary, recode the risk factor to ensure correct interpretation of the odds ratio.

* Recode critical variables for CATMOD LR;

```

data &dsn (drop= hcv fact) missing;
  length r&marker $ 1 riskfact $ 1;
  set lib.&dsn;
  r&marker=put(hcv,$hcv.);
  riskfact=put(fact,$risk.);
  * Create dummy variable for risk factor;
  if riskfact=1 then dumvar=0;
  else dumvar=1;
  if riskfact not in ("0","1") or r&marker=""
  then output missing;
  else output &dsn;
run;

```

run;

As a first step in the process of computing comparable odds ratio estimates using a statistical model instead of a classification table, we call the CATMOD procedure to compute parameter estimates for a model reduced to the risk factor of interest.

```

proc catmod data=agehcv92;
title 'Unadjusted logistic regression';
  population riskfact dumvar;
  direct dumvar;
  model r&marker = dumvar*riskfact
  /ml nogls oneway noiter noprofile nodesign;
  response logits/outest=testparm;
  format r&marker $r.;
  weight _freq_;
run;

```

Naming the dummy variable (dumvar) in the direct statement makes it a quantitative variable. Its interaction with the risk factor of interest (dumvar*riskfact) has the same representation in a CATMOD design matrix as an indicator variable in PROC LOGISTIC and other logistic regression programs. This method forces CATMOD to produce both comparable parameter estimates and appropriate standard errors of estimates for use in computing confidence intervals for the odds ratios.

The format statements tell CATMOD to use the formatted values to identify variable values in printed output. The model statement options specify maximum likelihood estimates and suppress some of the standard printed output the procedure generates.

The response statement gives the standard OUTEST output dataset the name testparm. This dataset contains parameter estimates for each effect level and interaction in the model and the standard errors of each estimate.

To complete the baseline estimates of odds ratios, another SAS macrolanguage procedure (%ORcalc) captures the parameter estimates and SE's from the file, converts them to odds ratios and

writes the odds ratios to the special list file that holds the empirical odds ratio estimates.

```

proc printto print=pre_hcv;
run;
title 'Delete this line';
%ORcalc(testparm,          /* input dataset */
  %str("0","1"),          /* levels of factor */
  lbl=riskfact,            /* name of factor */
  cl=1.96,                 /* Type I confidence */
  orte=UNADJUSTED OR from LR: HCV I riskfact
)

```

A printed version of the special list file should show that the empirical odds ratio estimates match almost exactly the model odds ratio estimates. An investigator can use this baseline comparison to verify that odds ratio estimates derived from the statistical model have the same interpretation as odds ratios computed directly from the corresponding four-fold classification tables.

Simply appending variable names to the variable lists in the population and model statements of the CATMOD procedure and changing the label in %ORcalc will invoke the same estimation and reporting of odd ratio estimates for the same risk factor, only now adjusted for the effects of confounders and other risk factors. An investigator can easily repeat the process for a number of different model specifications, including models with interaction effects. The program directs the full output of the CATMOD procedures to another list file. It may contain CATMOD parameter estimates for model specifications that exclude the risk factor of interest. Comparing the significance statistics for the model including the risk factor of interest and for a model that only differs by the exclusion of the risk factor, an investigator can determine whether the risk factor contributes to the value of the model.

Figure 2. shows the contents (after some formatting) of the special list file that holds the baseline results. It also includes an odds ratio adjusted for other main effects and, possibly, their interactions.

Odds Ratio Summary					1
10:02 Tuesday, May 3, 1994					
TABLE OF Outcome BY riskfact					
Outcome		Riskfact			
		Frequency			
		Percent			
		Row Pct			
Col Pct	Low	Between	High	Total	
Conf Pos	219	89	285	593	
	0.08	0.03	0.10	0.21	
	38.93	15.01	48.06		
	0.15	0.19	0.31		
Negative	146321	48010	90902	285233	
	51.19	16.80	31.80	99.79	
	51.30	16.83	31.87		
	99.85	99.81	99.69		
Total	146540	48099	91187	285826	
	51.27	16.83	31.90	100.00	

STATISTICS FOR TABLE OF Outcome BY Riskfact			
Statistic	DF	Value	Prob
Chi-Square	2	73.625	0.000
Likelihood Ratio Chi-Square	2	69.669	0.000
Mantel-Haenszel Chi-Square	1	73.599	0.000
Fisher's Exact Test (2-Tail)			
Phi Coefficient	0.016		
Contingency Coefficient	0.016		
Cramer's V	0.016		
Sample Size = 285826			

Odds Ratio Summary				2
10:02 Tuesday, May 3, 1994				
Odds Ratio estimated from frequencies				
Level	Odds Ratio	+/- Std. Err	ln(OR)	
Low	1	ref		
Between	1.23857083	0.15583283	0.21395816	
High	2.09475853	0.18844466	0.73943829	
ODDS Ratio estimated using statistical model				
UNADJUSTED OR from LR: Outcome riskfact ODDS Ratio				
Level	Odds Ratio	Conf. interval		
Low	1 (ref.)			
Between	1.2385708	0.9678920	1.5849471	
High	2.0947585	1.7561494	2.4986560	
ODDS Ratio estimated using statistical model				
ADJUSTED OR from LR: Outcome riskfact ODDS Ratio				
Level	Odds Ratio	Conf. interval		
Low	1 (ref.)			
Between	1.1985201	0.9340180	1.5379259	
High	1.4830860	1.2248184	1.7958125	

The odds ratio estimates for the different levels of the risk factor closely match the empirical odds ratios calculated from the frequencies. This checkpoint helps investigators accustomed to four-fold classification tables understand the correspondence between classification tables and categorical models. The next step, using categorical models to represent more complex causal systems, merely extends the analysis to multiple classification tables that isolate the joint effects of the risk factor and other factors on medical outcomes.

RECOMMENDATIONS

SAS documentation makes the newer PROC LOGISTIC the procedure of choice for binary logistic regression. LOGISTIC and others like it offer step-wise model specification options. CATMOD does not. LOGISTIC (beginning with release 6.07 TS301) also has a option for printing odds ratio estimates and their confidence intervals that CATMOD does not have. Except for data in SUMMARY form, as described earlier, this feature proves very useful.

LOGISTIC has a few quirks of its own. For data in SUMMARY form, the user has to transform pairs of rows representing two outcomes into a single events/trials row (described in SAS/STAT documentation of PROC LOGISTIC) to obtain accurate estimates of the odds ratio confidence intervals³. It sets the default response function value differently than CATMOD, but in an equally arbitrary way. Schlotzhauer's detailed discussion of PROC LOGISTIC and its application to odds ratio estimation provides the details.⁴

Having invested the time required to understand at least some of the features of CATMOD, we now appreciate its power and generality. Its ability to estimate multilevel effects directly from the data, and the fact that it produces both effect and level significance statistics, distinguishes it from PROC LOGISTIC. One would hope that the SAS Institute will upgrade the PROC CATMOD options to allow users to use variable formats to determine the argument for a logistic function or set a reference value for contrasts, and to produce odds ratios and their confidence intervals. Until SAS adds these options to CATMOD, calling CATMOD repeatedly within a generic program makes it possible to test alternative model specifications without reading multi-level variables into binary arrays and resorting to step-wise selection from a list of binary factors.

FOOTNOTES:

1. For a quick test CATMOD's handling of SUMMARY datasets, first estimate parameters using a dataset in SUMMARY form and then expand the SUMMARY to one row per observation

and estimate parameters again. You should, as we did, find identical results.

2. Hosmer, D.W. and Lemeshow, S. (1989), Applied Logistic Regression, New York: John Wiley & Sons, Inc.
3. See SAS Institute Inc. (1990), SAS/STAT User's Guide, Version 6, Fourth Edition, Volume 2, p. 1086, Cary, NC: SAS Institute Inc.
4. Schlotzhauer, D. (1993), "Some Issues in Using the LOGISTIC Procedure for Binary Logistic Regression", Observations, Vol. 2., No. 4, pp. 42-49.

Acknowledgements

Dr. George Schreiber, Dr. Dave Wright, and Mike Rhoads suggested changes that significantly improved the contents of the article; the author retains full responsibility for remaining defects.

Author Contact

For questions, comments or suggestions for Macro functions, contact the author:

Sig Hermansen
Westat, Inc.
1650 Research Boulevard
Rockville, MD 20850-1520
(301) 251-4268

APPENDIX

```
%macro ordoc(dsn,
  dvar,
  vlist,
  dvar1=0,
  dvaref=1,
  dvarfmt=,
  dvref=1,
  rf=1,
  rf1=0,
  rfref=1,
  rfmt=
);
  %if (&dvaref=1) %then %let __m=2;
  %else %let __m=1;
  %let __d12=%scan(&dvar1,&__m,%str( ));
  %let __d11=%scan(&dvar1,&dvaref,%str( ));
  %let __r11=%scan(&rf1,&rfref,%str( ));
  %if (&rfref=1) %then %let __n=2;
  %else %let __n=1;
  %let __r12=%scan(&rf1,&__n,%str( ));
  %let __rf=%scan(&vlist,&rf,%str( ));
  * Compute nXm table from numeric data;
  title "Cross-frequencies";
  proc freq data=&dsn;
    tables &dvar * &__rf /missing exact;
    weight _freq_;
    format &dvar &dvarfmt &__rf &rfmt ;
  run;
  options center;
  data _null_;
  file print;
  _tmp1_=&__r11;
  put @1 "Odds Ratio estimated from frequencies" ;
  put @1 "-----" /;
  put @10 "Level"
    @25 "Odds Ratio"
    @40 "+/- Std. Err"
    @55 "ln(OR)" ;
  put @10 _tmp1_ &rfmt
    @25 "1"
    @40 "***** ref *****" ;
  run;
  %do %while(&__r12 ne);
  proc summary data=&dsn nway;
```

```

class &dvar &__rf;
freq __freq_;
output out=tempout (keep=&dvar &__rf __freq_);
where (&__rf = &__r1 or &__rf = &__r2)
and (&dvar = &__d1 or &dvar = &__d2);
run;
proc IML;
use tempout;
read all into frq_m;
file print;
__tmp2_=&__r12;
d=ncol(frq_m);
if nrow(frq_m)=4 then
do;
__OR__ = (frq_m[2,d]/frq_m[1,d])/(frq_m[4,d]/frq_m
[3,d]);
%if (&__r1 gt &__r2)
%then __OR__=1/__OR__ %str( );
__SE__ = __OR__*(1/frq_m[4,d]+1/frq_m[3,d]+1
/frq_m[2,d]+1/frq_m[1,d])
**(1/2);
lnOR = log(__OR__);
put @10 __tmp2_ &rffmt
@25 __OR__ 10.8
@40 __SE__ 10.8
@55 lnOR 10.8 ;
end;
else put @10 __tmp2_ &rffmt
@20 "not defined";
%let __n=%eval(&__n + 1);
%if (&rref=&__n) %then %do;
%let __n=%eval(&__n + 1);
%end;
%let __r2=%scan(&r1,&__n,%str( ));
%end;
options nocenter;
%mend ordoc;

```

```

%macro ORcalc(dsn,vib1,bl=,cf=1.96,ortype=);
%count(__i)=&vib1
* Yields macrovariable __n=number of elements in
vib1;
%put &__n;
proc iml;
iblmtrx={&vib1};
iblmtrx=iblmtrx';
use &dsn;
read all where(_type_="PARMS") into bin;
read all where(_type_="COV") into covin;
__OR__ = exp(bin');
Min_CI = exp(bin'-(&cf *sqrt(vecdiag(covin))));
Max_CI = exp(bin'+(&cf *sqrt(vecdiag(covin))));
file print;
put @10 "ODDS Ratio estimated using statistical
model" /
@10 "_____"/
@10 " &ortype ODDS Ratio for &bl" /
@10 "Level Odds RatioConf. interval" /
@10 "_____";
%do __i=1 %to &__n;
a=iblmtrx[&__i,1];
b=__OR__[&__i,1];
c=Min_CI[&__i,1];
d=Max_CI[&__i,1];
%if &__i=1 %then put @10 a $char10.
@20 "1 (ref.)" %str( );
%else put @10 a $char10.
@20 b 10.7
@35 c 10.7 " - "
@49 d 10.7
%str( );

```