A SAS® Macro Procedure To Summarize the Output of Logistic Regression Models

Benjamin I. Broder, University of Southern California School of Medicine
Chih-Ping Chou, University of Southern California School of Medicine

Abstract

A SAS program, LOGRPT, has been developed to summarize the results obtained from logistic regression. The summary presents the results from several logistic regression models, all with the same set of covariates, but with differing dependent variables. For each covariate the following information is provided: regression weights, their standard errors, odds ratios, and a one character indication of the significance level of each regression weight. To assess the fit of the entire model, the -2 Log Likelihood statistic and its p-value are presented. This statistic has a chi-square distribution under the null hypothesis that all of the explanatory variables (covariates) in the model are zero. The LOGRPT program also allows users to specify a subset of covariates to be reported for comparison between models. The program uses the LOGISTIC procedure as well as SAS/IML and SAS macros to summarize the results. Making comparisons between several regression models is more efficient and error free when they are all presented on a single page, rather than scattered throughout many pages of SAS output.

Introduction

Binary and ordinal response categories arise in many fields of study. For example, presence or absence of disease in an epidemiologic study or severity of symptoms (mild, moderate, or severe) in a drug efficacy trial. Unlike standard linear regression, where we predict the level of the dependent variable given a set of independent variables, in studies such as these we often are interested in predicting the probability that the dependent variable has a certain value. Furthermore, the relationship of the covariates to this probability is in general multiplicative, rather than linear. We therefore use a logit transform of the risk of disease. In the following equation, \( P \) denotes the risk of disease, and \( y \) denotes the dependent variable for analysis:

\[
Y = \text{logit}(P) = \log \left( \frac{P}{1-P} \right)
\]

Equation 1 allows us to use linear regression by transforming a multiplicative relationship between the covariates into a linear one. Because \( P/(1-P) \) is known as the odds of disease, the logit transform is also known as the log odds transform. Finally, we have a standard linear regression equation:

\[
Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + e
\]

which can be solved using standard methods of linear regression.

Method

In epidemiologic research we often assess the effects of a given set of covariates on various outcome variables using equation (2). To assess the appropriateness of the entire model we can evaluate the -2 Log Likelihood statistic, which has a chi-square distribution under the null hypothesis that all the regression weights are zero. To investigate the impact specific covariates have on the given set of covariates on various outcome variables using equation (2). To assess the appropriateness of the entire model we can evaluate the -2 Log Likelihood statistic, which has a chi-square distribution under the null hypothesis that all the regression weights are zero. To investigate the impact specific covariates have on the

Discussion

The advantages of the LOGRPT program are that several logistic models can be compared on a single page of output. Previously this would have required retyping to summarize the results, an often error-prone step. Furthermore, as the names of the

Input Required for LOGRPT:

The LOGRPT program can be used on any system that supports SAS/IML and the SAS MACRO facility. The program will be attached, in Appendix 1. The following information must be provided by the user:

1. Input Data Set - including any necessary data transformations.
2. Output Destination - location of summarized output.
3. Model Specification - separate lists of covariates (both reported and unreported) and dependent variables for the models. Details are given in Appendix 1.

Example

An example will be provided in Appendix 1 analyzing the effects of program vs control group on weekly alcohol use at two follow-up times (FALC_WK1 and GALC_WK1) while controlling for baseline use (AALC_WK1). Only the covariates (group, race, sex, and baseline use) were compared in the summary. The summary output will also be demonstrated in Appendix 2.
dependent variables are included in the summary, the summary output can be included as a table in a report for publication with only minor modifications.

The authors may be contacted at:

University of Southern California School of Medicine
Department of Preventive Medicine
Institute for Health Promotion and Disease Prevention Research
1000 S. Fremont Ave. Suite 641
Alhamb,a, CA 91803·1358
(818) 457·4029 (B,odo,), (818) 457·4100 FAX

Appendix 1

% global depvar covrpt co"rp ndep nrpt nnrp debug printsum;

%macro dbugltxt debugvar=&debog;

%dbug{PRINT NV [COLNAME=CNAME]};

%if (&debugvar NE 0) %then %str1{txt};/*
%mend dbug;

%* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
* INIT -- Macro to do the needed initializations prior to running
* running the regressions or reporting the results.
*
* Arguments
* DEPVARs_used for the list of dependent variables. One regression
* will be run (and reported) for each dependent variable.
* COVRPTS_used for the list of independent variables (i.e.
* covariates) that will be reported.
* COVNRPS_used for the list of independent variables (i.e.
* covariates) that will NOT be reported in the summary
* output.
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
* OPTIONS MPRINT;
* OPTIONS PS = 58 LS = 132;
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
PROC IML;
START INIT _ARR;
* V1 = array of dependent variable names
* V2 = array of reported covariate variable names
* V3 = array of not reported covariate variable names
* V4 = array from 01 to ndep, i.e.,
* 01, 02, etc.
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
%dbug{PRINT NV [COLNAME=CNAME]};

%if (&.debugvar NE 0) %then %str1{txt};/*
%mend dbug;

%* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
* INIT -- Macro to do the needed initializations prior to running
* running the regressions or reporting the results.
*
* Arguments
* DEPVARs_used for the list of dependent variables. One regression
* will be run (and reported) for each dependent variable.
* COVRPTS_used for the list of independent variables (i.e.
* covariates) that will be reported.
* COVNRPS_used for the list of independent variables (i.e.
* covariates) that will NOT be reported in the summary
* output.
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
* OPTIONS MPRINT;
* OPTIONS PS = 58 LS = 132;
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
PROC IML;
START INIT _ARR;
* V1 = array of dependent variable names
* V2 = array of reported covariate variable names
* V3 = array of not reported covariate variable names
* V4 = array from 01 to ndep, i.e.,
* 01, 02, etc.
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
%dbug{PRINT NV [COLNAME=CNAME]};

%if (&debugvar NE 0) %then %str1{txt};/*
%mend dbug;

%* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
* INIT -- Macro to do the needed initializations prior to running
* running the regressions or reporting the results.
*
* Arguments
* DEPVARs_used for the list of dependent variables. One regression
* will be run (and reported) for each dependent variable.
* COVRPTS_used for the list of independent variables (i.e.
* covariates) that will be reported.
* COVNRPS_used for the list of independent variables (i.e.
* covariates) that will NOT be reported in the summary
* output.
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
* OPTIONS MPRINT;
* OPTIONS PS = 58 LS = 132;
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
PROC IML;
START INIT _ARR;
* V1 = array of dependent variable names
* V2 = array of reported covariate variable names
* V3 = array of not reported covariate variable names
* V4 = array from 01 to ndep, i.e.,
* 01, 02, etc.
* ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
%dbug{PRINT NV [COLNAME=CNAME]};

%if (&debugvar NE 0) %then %str1{txt};/*
%mend dbug;
%mend init;

%macro colnames_ColROOTS=EST Sl£ TRAnO P.N==I:
%MEND COLNAMES;

%dbuglproc print data =mnout;)

data chisqr&modeln;
output out=chisq sum=;
%do NUM = I %TO &N; %COLNAME_I&.COLROOTS) %END;
%if (%quote(&'colname) NEI "then
%&colname&NUM: &stop: &Ilongthl&syspbutf):

%set REGOUT;
%keep INTERCEP %SlR(&COVRPT):
%data O&MODELN;
%if &DVAR = 1 THEN PNHAT = 1-PNHAT;
%if &OVAR=1 THEN PHAT=1-PHAT;

var &dvar;
output out = mnout mean = mdvar;
%dbug(proc print data = mnout;)

%** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* The actual call on PROB LOGISTIC to perform the regressions, REGOUT*
* dataset, created by the OUTTEST statement, contains parameter*
* estimates as well as the estimated covariance matrix of the parameter*
* estimates (because of the COVOUT option to PROC LOGISTIC). The*
* covariance matrix contains the variances, which are used to generate*
* standard errors for the parameter estimates.
* REGOUT2 dataset, created by the output statement, is used to calculate*
* a Chi-Square value for model fit, i.e. how much the addition of the*
* covariates improves the predictive power of the model.
* ** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
PROC LOGISTIC COVOUT DATA = DATA1 OUTTEST = REGOUT
%if (%ddebug = 0) %then noprint ;
MODEL &DVAR = %STR(&COVRPT)
%STR(&COVNRPT) / MAXITER = 100;
output out = regout2 prob = phat;
%dbug(proc print data = regout;
%&&dbug(proc print data = regout;

title "OUTTEST from &dvar = &covrp c&covrpt");

%** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* This is one of the dependencies on binary dependent*
* VARIABLES - WE ASSUME ONLY 1 INTERCEPT VARIABLE*
* AND OBSERVATION (so we only keep 1 obs for the*
* parameter estimates and NRPT + 1 observations for the est. cov.*
* matrix). The part of the covariance matrix that represents*
* covariates that have been specified to be UNREPORTED is deleted. *
* ** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* DATA &MODEL:
* SET REGOUT;
* IF _N_ LE (&NRPT + 2);
* KEEP INTERCEP %STR(&COVRPT);

%** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* The following calculations are used to calculate the equivalent of *
* the -2 Log L statistic in the PROC LOGISTIC output. Unfortunately *
* PROC LOGISTIC does not allow this statistic to be output in an *
* output data set. *
* ** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* DATA LOGISTIC
* SET REGOUT2 (KEEP = ADVAR PHAT); *
* IF _N_ = 1 THEN SET MNOUT;

G = 1-MDVAR;
T = 1/(0-1);
ALPHA = LOGIT; *
IF &DVAR = OR PHAT =, THENDELETE;
PHAT = 1/(1+EXP-ALPHA));
IF &DVAR = 1 THEN PHAT = 1-PHAT;
IF &DVAR = 1 THEN PHAT = 1-PHAT;
TPI = LOG(PHAT);
TPI = -TPI; 
TPH = LOG(1-PHAT);
TPh = -2*TPI;
%dbug(proc print data = mnout; title "mnout - used to calculate -2 Log L");

PROC MEANS SUM N/O PRINT %dbug(PRINT):
var tph trph;
output out = chiysum;
%dbug(proc print data = chiys; title "CHISQUARE");

%** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* Each CHISQ1, CHISQ2, etc. data set contains the -2 Log L value in *
* the variable CHISQ, and the associated p-value in the variable P.*
* ** +++++++++++++++++++++++++++++++++++++++++++++++++++++**
* data chiys&model;
* set chiys (keep = tph trph);
* chiys = tph - tph;
p = 1 - probchiysq, &NRPT + &NNRP;

815
%MEND REG;
%dbug!proc print data=chisqr&'modeln; title "CHISQUARE&modeln";
%macro headerCdvs

%macro header;

%macro headerCdvs /parmsbuff:
%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;

%macro headerCdvs /parmsbuff:

%local b;
%let first = 1;
%let stop = %length&parmsbuff;
%do i = 1 %to %let stop;
%let fromvar = %scan&parmsbuff, &i;
%let to = %length&fromvar;
%let varname = %upcase&fromvar;
%let width = 8;
%let offset1 = 15;
%let offset2 = 8;
%let header_str=' beta SE OR ';
%let hdr_str=' beta SE OR ';
%let header(&offs) = ;
%end;
%&header;
%set(prefix=VAR,number=\$ndep); 
READ ALL INTO TVN;

**Matrices:**
- **RNAME** - Row vector containing labels for each of the summary rows.
- **N** - Column vector containing the N for each regression on each dependent variable.
- **V** - Column vector containing the variances for each dependent variable.
- **Colname** - Specifies the summary variable names to be printed in the summary report.
- **EST** - Parameter Estimates.
- **STE** - Standard errors of parameter estimates.
- **OR** - Odds Ratio = Odds of dependent variable equal to 1 if parameter=0. By dividing the odds if parameter=0 by the odds if parameter=1 divided by odds if parameter=0.
- **LCI** - Lower limit of 95% confidence interval of OR.
- **UCI** - Upper limit of 95% confidence interval of OR.
- **TRATIO** - T-ratio for parameter estimate.
- **P** - Probability that a particular parameter estimate is actually zero and the estimated value has occurred by chance.

**Scalers:**
- **I** - Specifies which dependent variable and regression we are calculating summary statistics for (i.e. ranges from 1 to \$ndep - the number of dependent variables.)
- **Est_Row** - Start row in REG matrix for parameter estimates of the regression results currently being summarized.
- **Cov_Row** - Start row in REG matrix for estimated covariance matrix of the reported covariates for the regression currently being summarized.
- **End_Row** - End row in REG matrix for estimated covariance matrix of the reported covariates for the regression currently being summarized.
- **P** - Probability that a particular parameter estimate is actually zero and the estimated value has occurred by chance.

**Matrices:**
- **E** - Vector containing parameter estimates from a single regression (i.e. for a single dependent variable.)
- **T** - Matrix containing estimated covariance matrix for the parameter estimates from a single regression (i.e. for a single dependent variable.)
- **S** - Vector from the diagonal of the T matrix above - first it contains the variances of the parameter estimates, then after taking the square root of each element it contains standard errors of the parameter estimates.
- **T1** - Matrix of values to put in the summary table - it initially contains just the estimates, standard errors of the estimates, t-ratios, and probabilities for a single regression, but then the summary values for the Total N for this regression, the number of covariates, and the -2 Log L model fit statistic.
- **TT** - Matrix of values to put in the summary table - it contains the values for Total N, number of covariates, and the -2 Log L model fit statistic.
- **RST** - Matrix of values to put in the summary table - made up of successive versions of T1 (one for each regression) concatenated together horizontally.

**DO**
- **I = 1 TO \$ndep:**
  - **Est_Row = (I-1)*\$nrpt1+1:**
  - **Cov_Row = Est_Row+1:**
  - **E = REG(Est_Row,);**
  - **E=E(-1);**
  - **End_Row = Est_Row + \$nrpt1:**
  - **T = REG(Cov_Row,End_Row,);**
  - **S = VECdiag(T);**
  - **B = SQRT(S);**
  - **OR = EXP(E);**
  - **LCI = EXP(B-(1.96*S));**
  - **UCI = EXP(E+(1.96*S));**

**%**
- **DO**
  - **I = 1 TO \$ndep:**
    - ** NYT IS USED FOR N and N_COV and the -2 Log L model fit statistic.**
      - **TT = J(3,2,99);**
      - **TT(I,1) = \$nucov1;**
      - **TT(I,2) = \$chisqsl;**
      - **T(I) = TT(I,1);**
      - **TT = RST;**

**END;**

**DO**
- **I = 1 TO \$ndep:**
  - **TT1 = J(1,2,99);**
  - **TT1(I,1) = \$colnames1;**
  - **TT1(I,2) = \$colnames2;**

**END;**

**DO**
- **I = 1 TO \$ndep:**
  - **TT2 = J(1,2,99);**
  - **TT2(I,1) = \$colnames1;**
  - **TT2(I,2) = \$colnames2;**

**END;**

**DO**
- **I = 1 TO \$ndep:**
  - **TT3 = J(1,2,99);**
  - **TT3(I,1) = \$colnames1;**
  - **TT3(I,2) = \$colnames2;**

**END;**

**FINISH;**

RUN IML_RPT;
DATA REPORT;
SET SUMMARY;
FILE &OUTPUT;

ARRAY EST %OSET(prefix=EST,NUMBER=&NOEP);
ARRAY STE %OSET(prefix=STE,NUMBER=&NDEP);
ARRAY LCI %OSET(prefix=LCI,NUMBER=&NDEP);
ARRAY UCI %OSET(prefix=UCI,NUMBER=&NDEP);
ARRAY OR %OSET(prefix=OR,NUMBER=&NDEP);

%do i=1 %to &NDEP;
  STE&I = E10. pal E10.;
%END;

ARRAY P %OSET(prefix=P,NUMBER=&NDEP);
ARRAY SG $ %OSET(prefix=SG,NUMBER=&NDEP);

%QSCAN&'COVRPT,&'I)' %END;

--++++++--

%do,i=1 %to &NDEP;
  P-values for EST&I
%END;

- Convert -99 (place-holders' in the summary data to missing values.

IF P=-99 THEN P =.;
IF P=1 THEN SG=' ';
IF P<.001 THEN SG='$';
IF P<.01 THEN SG='!';
IF P<.05 THEN SG='=';
IF P>.1 THEN P=1-P;

END;

SG=' '; %if CN_I=1 then do;
  parameter estimates";
  KEEP RNAME EST&1-EST&NDEP STE1-STE&NDEP OR1-OR&NOEP
  -2LogUP = Chi-Square model fit statistic
  
- Calculate values for

-2 Log L; put;

%LOG1DEPVARs = falc_wk1 galc_wk1 falc_wk2,
  COVRPTS = group grade aalc wk1 sex1,
  LIB = /cb/broder/sugi93-logrpt,
  OUTFILE = /cb/broder/sugi93/logrpt.out;
  DATA = txt, debug = 1, printsum = 1); 
RUN;

Appendix 2

LOGRPTI. OUT at 16APR93:10:17:58
Logistic Regression of Drug Use adjusting for Standard Covariates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Std Error</th>
<th>Odds Ratio</th>
<th>Lower 95% CI</th>
<th>Upper 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERCEP</td>
<td>-1.930</td>
<td>0.62</td>
<td>0.15</td>
<td>0.10</td>
<td>0.25</td>
</tr>
<tr>
<td>GROUP</td>
<td>-0.250</td>
<td>0.05</td>
<td>0.78</td>
<td>0.68</td>
<td>0.88</td>
</tr>
<tr>
<td>GRADE</td>
<td>0.250</td>
<td>0.15</td>
<td>1.28</td>
<td>1.06</td>
<td>1.54</td>
</tr>
<tr>
<td>AALC WK1</td>
<td>1.580</td>
<td>0.80</td>
<td>4.84</td>
<td>2.84</td>
<td>8.44</td>
</tr>
<tr>
<td>SEX1</td>
<td>0.410</td>
<td>0.15</td>
<td>1.51</td>
<td>0.91</td>
<td>2.51</td>
</tr>
<tr>
<td>NCOV/DF</td>
<td>5.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>1105.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2LogL/P</td>
<td>39.59</td>
<td>1.86</td>
<td>36.42</td>
<td>37.89</td>
<td>39.59</td>
</tr>
</tbody>
</table>