P-Value Calculation Made Easy with the SAS® Call Execute Routine

Huanhong Xia, AstraZeneca, Wayne, PA 19087
Richard Birkenmaier, AstraZeneca, Wayne, PA 19087

Abstract

In many situations we need to compare the counts of an event in two populations to see if the populations differ statistically in terms of the event. The Chi-square procedure or Fisher's exact method are usually used to generate p-values for assessing the difference. It's an easy task to perform when there are only a couple of events under consideration. However, if the number of events increase to hundreds, like adverse events in clinical trials, we want the work to be done in an automatic way. This paper presents a simple SAS® macro %p_comp to perform the p-value calculations.

Key Words: SAS macro, P-value, call execute, Chi-square test, Fisher's Exact test

Introduction

Macro %p_comp was first used in clinical studies. In the clinical trial practice, it is of great interest to see if one treatment has a better safety record over the other one. Therefore, a statistical test needs to be performed for all the adverse events (AE). Suppose you have two treatments: an active drug vs. a placebo, and the AE data has been summarized into a SAS® dataset, say table1.sd2, as shown in Table 1, in which variables n_a and all_a stand for count of event and total number of patients, respectively, in the active treatment group. Similarly n_p and all_p are for count of event and total number of patients in the placebo group. Your task is to add a column of p-values, from either Chi-square test or Fisher's exact test, to the dataset for comparing number of occurrences for every AE. Your final dataset would look like:

Table 1. Partial print-out of a SAS dataset

<table>
<thead>
<tr>
<th>Event</th>
<th>n_a</th>
<th>all_a</th>
<th>n_p</th>
<th>all_p</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSCESS</td>
<td>1</td>
<td>168</td>
<td>0</td>
<td>173</td>
</tr>
<tr>
<td>ALOPECIA</td>
<td>2</td>
<td>168</td>
<td>0</td>
<td>173</td>
</tr>
<tr>
<td>AMBLYOPIA</td>
<td>2</td>
<td>168</td>
<td>0</td>
<td>173</td>
</tr>
<tr>
<td>ANEMIA</td>
<td>0</td>
<td>168</td>
<td>1</td>
<td>173</td>
</tr>
<tr>
<td>ANOREXIA</td>
<td>2</td>
<td>168</td>
<td>0</td>
<td>173</td>
</tr>
<tr>
<td>ANXIETY</td>
<td>2</td>
<td>168</td>
<td>4</td>
<td>173</td>
</tr>
<tr>
<td>ASTHENIA</td>
<td>7</td>
<td>168</td>
<td>2</td>
<td>173</td>
</tr>
<tr>
<td>ASTHMA</td>
<td>3</td>
<td>168</td>
<td>2</td>
<td>173</td>
</tr>
<tr>
<td>BACK PAIN</td>
<td>6</td>
<td>168</td>
<td>6</td>
<td>173</td>
</tr>
<tr>
<td>CHILLS</td>
<td>1</td>
<td>168</td>
<td>1</td>
<td>173</td>
</tr>
<tr>
<td>COLITIS</td>
<td>0</td>
<td>168</td>
<td>1</td>
<td>173</td>
</tr>
</tbody>
</table>

Table 2. The macro %p_comp was developed to handle this situation.

<table>
<thead>
<tr>
<th>Event</th>
<th>n_a</th>
<th>all_a</th>
<th>n_p</th>
<th>all_p</th>
<th>p_exact2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSCESS</td>
<td>1</td>
<td>168</td>
<td>0</td>
<td>173</td>
<td>0.4927</td>
</tr>
<tr>
<td>ALOPECIA</td>
<td>2</td>
<td>168</td>
<td>0</td>
<td>173</td>
<td>0.2420</td>
</tr>
<tr>
<td>AMBLYOPIA</td>
<td>2</td>
<td>168</td>
<td>0</td>
<td>173</td>
<td>0.2420</td>
</tr>
<tr>
<td>ANEMIA</td>
<td>0</td>
<td>168</td>
<td>1</td>
<td>173</td>
<td>1.0000</td>
</tr>
<tr>
<td>ANOREXIA</td>
<td>2</td>
<td>168</td>
<td>0</td>
<td>173</td>
<td>0.2420</td>
</tr>
<tr>
<td>ANXIETY</td>
<td>2</td>
<td>168</td>
<td>4</td>
<td>173</td>
<td>0.6849</td>
</tr>
<tr>
<td>ASTHENIA</td>
<td>7</td>
<td>168</td>
<td>2</td>
<td>173</td>
<td>0.1004</td>
</tr>
<tr>
<td>ASTHMA</td>
<td>3</td>
<td>168</td>
<td>2</td>
<td>173</td>
<td>0.6811</td>
</tr>
<tr>
<td>BACK PAIN</td>
<td>6</td>
<td>168</td>
<td>6</td>
<td>173</td>
<td>1.0000</td>
</tr>
<tr>
<td>CHILLS</td>
<td>1</td>
<td>168</td>
<td>1</td>
<td>173</td>
<td>1.0000</td>
</tr>
<tr>
<td>COLITIS</td>
<td>0</td>
<td>168</td>
<td>1</td>
<td>173</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Algorithm and the Macro Details

The macro %p_comp was developed based on the following algorithm consisting of three parts.

1> Create a macro to compute p-value for a simple 2X2 table.

2> Use a DATA step interface to interact with the macro in step 1 to have p-values calculated, and append all the p_values to a dataset.

3> Merge p-values to the original dataset.
CALL EXECUTE routine plays a crucial role in step 2 of the algorithm. It allows massive information stored in a dataset to be passed to other SAS procedures and macros. It is efficient and very convenient, requiring little coding. CALL SYMPUT was also used in step 2 to help identify a p-value to its corresponding event.

The code for macro %p_comp is listed in Figure 1 at the end of article. The following is an example showing how to invoke %p_comp, given dataset table1.sd2 with its variables:

```sas
%p_comp(
  datain = table1, /*original dataset */
  eventvar = event, /*event variable */
  dataout = table2, /*new dataset */
  pvalue=p_exact2, /*p_pchi or p_exact2*/
  n_a = n_a, /*# of event in treat A */
  all_a = all_a, /*# of patients in treat A*/
  n_p = n_p, /*# of event in treat P */
  all_p = all_p /*# of patients in treat P*/
);
```

The value for parameter `pvalue` can be `p_pchi` or `p_exact2`, meaning the p-value from the Chi-square test or from the Fisher’s exact test, respectively. The execution of the above program gives you a new dataset table2.sd2 with p-values on (see Table 2).

**Conclusion**

The macro `%p_comp` saves tremendous work if a large amount of 2X2 tables need to be analyzed. It is very easy to use. However, before applying the macro, you need to manipulate data to form a dataset like Table 1.

**References**


**Contact Information**

Huanhong Xia  
Programming Consultant at AstraZeneca  
Wayne, PA 19087  
Phone: (610)578-8414  
E-mail: huanhong.xia@astrazeneca.com or Huanhongxia@yahoo.com

**Figure 1.** Macro %P_COMP takes every record in a dataset, analyzes it as a 2 by 2 contingency table and produces a new dataset with a column of p-values.

```sas
/*--------- Comments on parameters ---------
(1) original dataset
(2) event variable
(3) new dataset to be created
(4) p_pchi: p-value from Chi-square test;
    p_exact2: p-value from Fisher’s Exact test
(5) variable for # of event in treat A
(6) variable for total # of patients in treat A
(7) variable for # of event in treat P
(8) variable for total # of patients in treat P.
--------------------------------------------------*/
proc datasets lib=work;
  delete _pvalue_ _p_;  
run;
/*--- Macro x2test calculates p-value for a 2X2 table---*/

%macro x2test(n11=&n_a,total1=&all_a,
n21=&n_p; total2=&all_p;
data x2test;
treat="A"; event="Y"; count = &n11;
output;
treat="A"; event="N";
count = &total1-&n11;
output;
treat="B"; event="Y"; count = &n21; output;
treat="B"; event="N";
count = &total2-&n21;
output;
run;
/*initiate _p_ for proc append*/
data _p_;&pvalue = .;
run;
proc freq data=x2test noprint;
tables treat*event/chisq exact;
weight count;
output out=_p_ exact chisq;
run;
/*Identify the event for the p-value using macro variable name, which’s created by call symput routine later.*/
data _p_;&eventvar = "&name";
keep &pvalue &eventvar;
run;
proc append base=_pvalue_ data=_p_;
by &eventvar;
run;
%MEND x2test; /*End macro x2test */
proc sort data=&datain out=temp nodupkey;
by &eventvar;
run;
data _null_;set temp;
call symput("name",&eventvar);
call execute('%x2test( n11='||&n_a. ||', total1='||&all_a. ||', n21='||&n_p. ||', total2='||&all_p||'));
run;
if &dataout = %then %do;
data &datain;
merge &datain _pvalue_;by &eventvar;
run;
%end;
%else %do;
data &dataout;
merge &datain _pvalue_;by &eventvar;
run;
%end;
proc datasets lib=work;
delete _pvalue_ _p_ temp x2test;
run;
%MEND p_comp;