

Paper 095-2010

A Program to Automatically Compute Agreement Statistics for an Asymmetric Table

Ying Feng, CTB/McGraw-Hill
Vincent Weng, Educational Testing Service

ABSTRACT

Cohen's Kappa coefficient is a method for assessing the degree of agreement between two raters. In order to correctly compute agreement statistics, the table must be square and row labels must match corresponding column labels. However, it may happen that one rater does not use one of the categories of a rating scale. That will result in a non-square table. Regular Kappa procedure will not compute the agreement statistics in this case. Beginning with SAS[®] 9, you can make the table square by adding an observation with zero weight in each missing category. However, what happens if you need to check the table symmetry for hundreds of samples and across different clusters within each sample? This paper presents several macro programs to automatically check and handle the non-square tables, then perform Kappa computation across clusters within a sample.

Keyword: SAS MACRO, Kappa statistics, zero option, PROC SQL

INTRODUCTION

The kappa coefficient is a standard measure of inter-rater agreement for categorical data (Cohen, 1960). Cohen (1968) also developed a weighted kappa coefficient to be used when the degree of disagreement between two raters can be quantified. The weighted kappa is a generalization of the simple kappa coefficient, using weights to effectively assign partial credit to near, but not exact, agreement.

The kappa coefficient is widely used for measuring the degree of reliability between raters. The rater responses are placed into a two way table. Kappa coefficient is calculated from data in the two way table. Based on the value of kappa, a general level of agreement can be assigned to indicate the amount of reliability between the ratings of the two raters.

In order to correctly compute agreement statistics, the table must be symmetric. However, it may happen that one rater does not use one of the categories of a rating scale. That will result in a non-square table. Default Kappa procedure will not compute the agreement statistics for a non-square table. Beginning with SAS 9, you can make the table square by adding an observation with zero weight in each missing category. If you have many samples and also needs to compute the Kappa statistics across clusters within each sample, automatically detecting and handling the non-square tables would be an efficient way to go. This paper will demonstrate the macros to accomplish the goal easily.

SAS PROCEDURES

SAS procedure PROC FREQ with the AGREE option can provide the kappa statistic when there are only two raters. PROC FREQ only works with complete or square data, i.e., each rater uses each possible choice on the rating scale at least once. For example, two raters (1 and 2) rated n subjects into four categories (1, 2, 3 and 4). Rater 2 did not give "3" to any of the n subjects. The frequency table would look like:

Table of rater1 by rater2				
rater1	rater2			Total
	1	2	4	
1	10	4	3	17
2	5	3	3	11
3	2	6	3	11
4	6	3	2	11
Total	23	16	11	50

The frequency table is not square. In the real world studies, it is not uncommon to see a non-square frequency table. PROC FREQ will not provide a kappa unless we make the table square and use zero option.

Beginning with SAS 9, you can make the table square by adding an observation with zero weight in each missing category. It is necessary to create an observation in only one cell of the category, not all cells.

Suppose we have a table a, containing 50 subjects and ratings from rater 1 and rater2.

```
data a;
  do subject=1 to 50;
    rater1=rantbl(12345, .25, .25, .25, .25);
    rater2=rantbl(12345, .50, .25, 0, .25);
    output;
  end;
run;
```

The following DATA steps create a single, zero-weight observation and add it to the original data. All observations in the original data set are given a weight of 1.

```
data b;
  rater1=1; rater2=3; wt=0; output;
run;
```

```
data c;
  set a b;
  if wt=. then wt=1;
run;
```

The analysis is done by using the WEIGHT statement and the ZEROS option, which tells PROC FREQ to allow zero-weighted cells to remain in the table. Notice that the table is now square and that the agreement statistics are computed.

```
proc freq data=c;
  weight wt / zeros;
  tables rater1*rater2 / agree nocol norow nopercent;
run;
```

Table of rater1 by rater2					
rater1	rater2				Total
	1	2	3	4	
1	10	4	0	3	17
2	5	3	0	3	11
3	2	6	0	3	11
4	6	3	0	2	11
Total	23	16	0	11	50

CAVEATS

One caveat we should be aware is that sometimes both raters might miss one category, but the categories missed are not the same one. For example, Rater 1 did not give "2" category to any of the n subjects while Rater 2 miss the "3" category. Since the columns are square, the agreement statistics will be computed. However, the statistics are wrong since the rows and columns do no match. The rows and columns match should be verified before the statistics are calculated.

AUTOMATION OF THE PROCEDURE

Since we have many samples, automatically detecting and handling non-square table will be desirable. Several macros are developed based on the above request.

Macro 1: KappaTweak

The %if %sysfunc(exist(&ExDs.))[^]=1 statement is to check whether the Kappa statistics has already been generated. If the statistics is already generated, then the macro will be skipped to save resources.

If the Kappa statistics does not exist yet, PROC SQL procedure is used to generate the categories of rater1 and rater2, and then a full merge is performed to generate the category two way table. The full merge and the subsequent zero weight assignment not only catches the missing categories, but ensures the rows and columns labels match.

```
*Kappa/Weighted Kappa Computation regardless table symmetry;
*Macro variable ExDs is the name of SAS output dataset;
*InDs is the name of SAS input dataset;
*Grp1 is the first rater; and catgGrp2 is the second rater.
*grpNm is the name for the Kappa stats.

%macro KappaTweak(ExDs=, InDs=, catg=, grp1=,grp2=,grpNm=);

%if %sysfunc(exist(&ExDs.))^=1 %then %do;

proc sql;
  create table catg&grp1. as
  select unique &catg.&grp1.
  from &InDs.
  group by &catg.&grp1.;
```

```

create table catg&grp2. as
select unique &catg.&grp2.
from &InDs.
group by &catg.&grp2.;

create table catgMrg as
select *
from catg&grp1. a full join catg&grp2. b
on &catg.&grp1.=&catg.&grp2.;

quit;

data catgMrg2;
set catgMrg;
if &catg.&grp1.=. then do; &catg.&grp1.=&catg.&grp2.; wt=0; output;
end;
if &catg.&grp2.=. then do &catg.&grp2.=&catg.&grp1.; wt=0; output;
end;
run;

data WtTweak;
set &InDs. catgMrg2;
if wt=. then wt=1;
run;

proc freq data=WtTweak;
weight wt/zeros;
table &catg.&grp1.*&catg.&grp2./norow nocol agree(wt=fc);
output out=&ExDs.(rename=( _kappa_=Kappa_&grpNm.
_wtkap_=wtkappa_&grpNm.)) kappa wtkap ;
run;

%end;

%mend KappaTweak;

```

Scenario:

Given two raters Rater 1 and Rater2, the input dataset is sampleData, and the output data is called Kappas_stats, here is the sample macro call:

```
%KappaTweak(ExDs=kappas_stats, InDs=sampleData, catg=Rater, grp1=1,grp2=2,
grpNm=essay1);
```

Macro 2: KappaTweakBy

In many cases we would also like to know how the agreement statistics are across different groups. The macro below is designed to produce Kappa statistics across cluster within each sample.

```

%macro
KappaTweakby(ExDs=, InDs=, catg=, grp1=, grp2=, grpNm=, sqlstr=, byvar=, byvar2=, byvar3=);

%if %sysfunc(exist(&ExDs.))<math>^</math>=1 %then %do;

proc sql;
create table catg&grp1. as
select unique &byvar., &catg.&grp1.
from &InDs.
group by &byvar., &catg.&grp1. ;

create table catg&grp2. as
select unique &byvar., &catg.&grp2.
from &InDs.
group by &byvar., &catg.&grp2.;

proc sql;
create table catgMrg as
select &sqlstr., &catg.&grp1., &catg.&grp2.
from catg&grp1. a full join catg&grp2. b
on &byvar2. and
&catg.&grp1.=&catg.&grp2.;

data catgMrg;
set catgMrg;
if &catg.&grp1.=. and &catg.&grp2.=. then delete;
run;

data catgMrg2;
set catgMrg;
if &catg.&grp1.=. then do; &catg.&grp1.=&catg.&grp2.; wt=0; output; end;
if &catg.&grp2.=. then do &catg.&grp2.=&catg.&grp1.; wt=0; output; end;
run;

data WtTweak;
set &InDs. catgMrg2;
if wt=. then wt=1;
run;

proc sort data=wtTweak; by &byvar3.;run;

proc freq data=WtTweak;
weight wt/zeros;
table &catg.&grp1.*&catg.&grp2./norow nocol agree(wt=fc);
by &byvar3.;
output out = &ExDs.(rename=(_kappa_=Kappa_&grpNm. _wtkap_=wtkappa_&grpNm.))
kappa wtkap ;
run;

%end;
%mend KappaTweakby;

```

Scenario:

Given two raters Rater 1 and Rater2, the input dataset is sampleData, and the output data is called Kappas_stats, here is the sample macro call if we would like to run Kappa statistics across gender and grade:

```
%KAPPATWEAKBY(ExDs=KAPPAS_STATS,INDS=SampleData,catg=Rater, grp1=1,grp2=2,grpNm=HH,  
sqlstr=%str(coalesce(a.grade,b.grade) as grade, coalesce(a. gender,b.gender)  
as gender), byvar=%str(grade, gender), byvar2=%str(a.grade=b.grade and  
a.gender=b.gender), byvar3=%str(grade gender));
```

CONCLUSION

User-friendly macros are demonstrated here to automatically detect and handle the non-square table issue for Kappa statistics. The programs provide flexibility to define different samples and clusters for agreement computation.

REFERENCES

1. Cohen, J. (1960), A Coefficient of Agreement for Nominal Scales. *Educ. Psychol. Meas.*, 20, 37-46.
Ebel, R.L. (1951), Estimation
2. <http://support.sas.com/kb/22/883.html>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Ying Feng
CTB/McGraw-Hill
148 Princeton-Hightstown Road
East Windsor, NJ 08520
Email: ying_feng@ctb.com

Vincent Weng
Educational Testing Services
Rosedale Road, Princeton, NJ 08541
E-mail: vweng@ets.org

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 and product names are trademarks of their respective companies.