Paper PO-05

Using Macro to simplify to Calculate Multi-Rater Observation Agreement

Abbas S. Tavakoli, DrPH, MPH, ME, Richard Walker, PhD Candidate

ABSTRACT

This paper describes using several macros program to calculate multi-rater observation agreement using the SAS ® Kappa statistic. In the paper, we show an example of four raters observed a video to select certain tasks. Each rater could select up to ten tasks. Each rater could select different tasks numbers. Inter-rater reliability (IRR) between the four raters is examined using the Kappa statistic, calculated using the SAS ® PROC FREQ, MEANS, and PRINT procedures. The Kappa statistic and 95% CI for observers were calculated and the overall IRR was calculated by averaging pairwise Kappa agreements. This paper provides an example of how using macro to calculate percentage agreement with the Kappa statistic with a 95% CI using SAS ® PROC FREQ, MEANS, and PRINT for multiple raters with multiple observation categories. The program can be used for more raters and tasks. This paper expands the current functionality of the SAS ® PROC FREQ procedure to support application of the Kappa statistic for more than two raters and several categories.

Keywords:
Multi-rater Observation Agreement, Kappa Statistic, SAS

INTRODUCTION

Inter-rater reliability (IRR) among rater can be measured in any situation in which two or more raters are assessing the same thing. It was first time to introduce kappa statistics by Cohen (1960). The Kappa statistic estimates the percentage of agreement among raters after removing the percentage of agreement which would occur by chance. In observational studies it is often necessary to assess multi-rater agreement for multiple observation categories. The Kappa statistic is commonly used to measure agreement among raters for categorical data (Altman, 1991). The SAS ® PROC FREQ procedure supports application of the Kappa statistic for two raters and several categories. However, calculation is not straightforward for more than two raters as data must first be manipulated in a square form table in order to use the SAS ® PROC FREQ procedure.

PURPOSE

The purpose of this paper is to describe using macro program in SAS to calculate multi-rater observation agreement with the Kappa statistic.

BACKGROUND

In this paper, we show an example of four raters observed a video to select certain tasks. Each rater could select up to ten tasks for simplification. Each rater could select different tasks numbers. An example could be: four raters watch video of children with Autism to select specific behavioral characteristics (tasks) which is determined by researcher. Each observer can select different number of tasks.

DATA ANALYSIS

Analysis of this data was performed using **SAS/STAT** ® statistical software, version 9.2 (SAS, 2008). The SAS ® PROC FREQ procedure with the AGREE option was used for Kappa statistical calculations. The maximum number of tasks that could be selected by each rater was 10 for simplification. However, this can be extended to select more tasks. Some variability existed in the number and order of tasks recorded by each rater. Table 1 is part of the original data for the first three videos and shows the tasks that were selected by each rater. Table1 showed each rater selected different number of tasks. Also, there are missing for electing tasks.

The Macro requires that the dataset given has the columns ID, TASK1, and btask1 to the number of raters the user wishes to compare. The variable **numofRaters** determines how many raters the user wishes to compare, starting with 1 and going upwards. The dataset is loaded into SAS and formatted for easy use. In order to calculate Kappa in SAS the frequency table must be square. Uebersax (2002) used pseudo-observation to create square table. A weight of 1 for real observations and a weight of .0000000001 for pseudo-observations was assigned. Pseudo-

observations ensure responses for every task assigned by any other observers and the small weight does not have any effect on the Kappa statistic. A mapping of raters by ID is created, assigning an x to all tasks that are not equal to 1 (**Attachment: Data Mappings**). A weight of 1 is also assigned to all tasks present in the dataset, a dummy dataset is constructed to assign the task value to the each rater and assign a very small weight to force it to not affect results overtly (**Attachment: Data Dummy**). Both datasets are combined into a single dataset and used in the macro to calculate rater reliability **(Attachment: Data Wtid**).

The **macro 'CompRaters'** was created to use the constructed combined dataset and calculate the agreement among the raters. It uses two nested 'for' loops compare each rater and a third 'for' loop to calculate separate values for when task is taken into account and when task is ignored. Submacros exist to allow SAS to loop through the different sets of code without having duplication of code. Each submacro is used for taking the tasks into account when calculating the rater reliability. The Macro creates independent datasets for each rater comparison and calculates the frequency that raters agree. Only those tasks that agree are kept and are reweighted to either 0 for counts less than one and 1 for counts greater than one. The total number of agreements is summed up and the frequency of agreement is recalculated for each task. After all pairwise comparisons are calculated and the datasets are combined into a single file, all observations with no agreements are deleted. The total number of comparisons are summed and confidence intervals are calculated. Finally, the percent agreement is calculated from the means and presented in three tables. Table 2 includes all the rater's comparisons by tasks. Table 3 and Table 4 include the overall output ignoring tasks, with Table 3 averaging all the tasks and Table 4 ignoring tasks completely. The program automatically generates formats for the tables based on the dataset provided and automatically deletes all datasets after completion. The Macro works on any number of tasks and any number of raters, but for the purpose of example, only 10 tasks and 4 raters were analyzed for this paper.

RESULTS

Table 2 shows part of the output created for each video for the number of rater agreements, percentage of agreement, and Kappa with a 95% CI. For example, the Kappa agreement for video 1 between rater 1 and 2 was 0.86 (95% CI 0.61-1.0). It is shown that the agreement between two raters varies across different videos. Table 3 shows the complete output for the number of agreements, percentage of agreement, and Kappa with 95% CI ignoring individual videos. As indicated in Table 3, the Kappa between two raters ranged from 0.56 (raters 3 and 4) to 0.90 (raters 1 and 3) with an overall IRR between rater 1, 2, 3, and 4 of 0.69 (95% CI 0.41-0.94). According to Landis and Koch (1977), all the pairs had good to very good agreements except. Table 4 shows the complete output for the number of agreements, percentage of agreement, and Kappa with 95% CI by averaging individual videos. Table 4 reveals all pairwise comparisons had good to very good agreements.

CONCLUSION

This paper provides an example of how to use macro to calculate percentage agreement with the Kappa statistic with 95% CI using SAS ® PROC FREQ, MEANS, and PRINT for multiple raters with multiple observation categories. This paper expands the current functionality of the SAS ® PROC FREQ procedure to support application of the Kappa statistic for more than two raters and several categories.

REFERENCES

1. Altman, D. (1991). Practical Statistics for Medical Research. Chapman and Hall.
2. Cohen, J (1960). A Coefficient of Agreement for Nominal Scales. Education Psychological Measurement, 20, 37- 46.
3. Landis, J. & Koch, G. (1977). The Measurement of Observer Agreement for Categorical Data". Biometrics, 33:159-174.
4. Uebersax, J. (2002). Calculating Kappa with SAS: http://www.john-uebersax.com/stat/saskappa.htm

Contact Information

Abbas S. Tavakoli, DrPH, MPH, ME
College of Nursing
University of South Carolina
1601 Greene Street
Columbia, SC 29208-4001
Fax: (803) 777-5561
E-mail: abbas.tavakoli@sc.edu

Obs	ID	TASK1	btask1	btask2	btask3	btask4
1	1	task 1	1	1	1	.
2	1	task 2	1	1	1	.
3	1	task 3	1	.	1	.
4	1	task 4	1	1	1	.
5	1	task 5	1	1	1	.
6	1	task 6	1	1	1	1
7	1	task 7	1	1	1	1
8	1	task 8	.	.	.	1
9	1	task 9	.	.	.	.
10	1	task 10	1	1	1	1
11	2	task 1	1	1	1	.
12	2	task 2	1	1	1	.
13	2	task 3	1	.	.	.
14	2	task 4	1	1	1	1
15	2	task 5	1	1	1	.
16	2	task 6	1	1	1	1
17	2	task 7	1	1	1	1
18	2	task 8	.	.	.	1
19	2	task 9	.	.	.	.
20	2	task 10	1	1	1	1
21	3	task 1	1	1	1	.
22	3	task 2	1	1	1	.
23	3	task 3	1	.	1	.
24	3	task 4	1	1	1	1
25	3	task 5	1	1	1	.
26	3	task 6	1	1	1	1
27	3	task 7	1	1	1	1
28	3	task 8	.	.	.	1
29	3	task 9	.	.	.	.
30	3	task 10	1	1	1	1

Table 1. Partial Data Set for First Three Video Observations by Four Observers.
Note:Btask1-btask4 are rate1-rater4

Obs	Rater	number of agreement	total tasks	percentage of agreement	Kappa agreement	lower 95% CI** kappa	Upper 95% CI** Kappa
1	Rater 1 vs. Rater 2	7	10	0.7	0.85965	0.61136	1.00000
2	Rater 1 vs. Rater 2	7	10	0.7	0.85965	0.61136	1.00000
3	Rater 1 vs. Rater 2	7	10	0.7	0.85965	0.61136	1.00000
4	.	.	.	.	.	.	.
11	Rater 1 vs. Rater 3	8	10	0.8	1.00000	1.00000	1.00000
12	Rater 1 vs. Rater 3	7	10	0.7	0.85965	0.61136	1.00000
13	Rater 1 vs. Rater 3	8	10	0.8	1.00000	1.00000	1.00000
14	.	.	.	.	.	.	.
21	Rater 1 vs. Rater 4	3	10	0.3	0.26027	-0.07991	0.60046
22	Rater 1 vs. Rater 4	4	10	0.4	0.38356	0.03041	0.73671
23	Rater 1 vs. Rater 4	4	10	0.4	0.38356	0.03041	0.73671
24	.	.	.	.	.	.	.

Table 2. Partial Output for Four Rater Task Selection for Individual Videos.

**CI = Confidence interval

Obs	Rater	number of agreement	total tasks	percentage of agreement	Kappa agreement	lower 95% CI** kappa	Upper 95% CI** Kappa
1	Rater 1 vs. Rater 2	6.5	10	0.65	0.76033	0.46835	0.98652
2	Rater 1 vs. Rater 3	7	10	0.70	0.89951	0.73952	1.00000
3	Rater 1 vs. Rater 4	5.3	10	0.53	0.58500	0.26246	0.87413
4	Rater 2 vs. Rater 3	6.4	10	0.64	0.79235	0.52642	0.99626
5	Rater 2 vs. Rater 4	5.2	10	0.52	0.58735	0.24779	0.90429
6	Rater 3 vs. Rater 4	5	10	0.50	0.55687	0.21316	0.88283
7	Overall	5.9	10	0.59	0.69690	0.40962	0.94067

Table 3. Output for Four Rater Ignoring Individual Videos.

**CI = Confidence interval

Obs	Rater	number of agreement	total tasks	percentage of agreement	Kappa agreement	lower 95% CI** kappa	Upper 95% CI** Kappa
1	Rater 1 vs. Rater 2	9	10	0.90000	0.75814	0.66147	0.85481
2	Rater 1 vs. Rater 3	8	10	0.80000	0.89710	0.82586	0.96834
3	Rater 1 vs. Rater 4	8	10	0.80000	0.58141	0.46791	0.69491
4	Rater 2 vs. Rater 3	7	10	0.70000	0.78656	0.69195	0.88118
5	Rater 2 vs. Rater 4	8	10	0.80000	0.59055	0.47384	0.70725
6	Rater 3 vs. Rater 4	6	10	0.60000	0.55325	0.43562	0.67088
7	Overall	7.666667	10	0.76667	0.69450	0.59277	0.79623

Table 4. Output for Averaged Four Rater of Individual Videos.

**CI = Confidence interval

Attachment

SAS Syntax

```
option nodate nocenter nonumber yearcutoff=1910 LS=140;

libname int 'c:\abbast\sasconf\dataset\';

%let numofRaters = 4;

data one;      set int.intmacb; run; quit;

/* create a table that contains the number of unique Tasks. These tasks are counted to be used in creating labels. */

proc sql noprint;    create table numberOfTasks as
  SELECT count(distinct task1) as numberOfTasks   from one;  run; quit;
%let numberOfTasksVar;

/* SAS Macro code to pull a single value out of a dataset was taken from */

%MACRO Get_data(myDataset=,myLine=,myColumn=,myMVar=);
  %GLOBAL &myMVar ;
  proc sql noprint ;   select &myColumn into :&myMVar
  from &myDataset    where monotonic() = &myLine;   Run; quit;
%MEND Get_data;

%Get_data(myDataset=numberOfTasks,myLine=1,myColumn=numberOfTasks,
        myMVar=numberOfTasksVar)  %put &numberOfTasksVar;

/* Create a dataset for each task and create that task as a format */

data taskf (keep=start label fmtname);
  do i = 1 to &numberOfTasksVar;     start = i;          label = 'task'||i;       fmtname='taskf';  output;   end;
run; quit;
proc format library=work cntlin=taskf; run;
```

```
/* Apply created format to the original dataset */

data one;   set one;     format task1 taskf.;  run; quit;

 *** CALCULATE KAPPA  FOR PICKING UP THE TASKS BY RATER  ***;

 data mappings (keep=id task1 btsk1-btsk&numofRaters wgt);     set one;     by id;
  length btsk1-btsk&numofRaters $4;

        *Create temporary arrays to allow looping though the list of raters;

        array tempBtask(&numofRaters) btask1-btask&numofRaters;
        array tempBtsk(&numofRaters) btsk1-btsk&numofRaters;

        *Loop through tall the raters that are part of the initial file;

        do i = 1 to &numofRaters by 1;
          if tempBtask(i) =1 then tempBtsk(i)=task1;       else tempBtsk(i)=' x ';     end;

        * ASSIGN WEIGHT OF '1' TO REAL RECORDS ;
        wgt = 1;    run; quit;

 * CREATE DUMMY DATA RECORDS TO ENSURE SQUARE TABLE FOR OBTAINING
   KAPPAS.  ASSIGN A TINY WEIGHT SO DUMMY OBSERVATIONS DO NOT EFFECT
   KAPPA VALUES. INCLUDE ROW OF MISSING ('x') VALUES FOR EACH id ;

 data dummy (keep=id task1 btsk1-btsk&numofRaters wgt);      set one;       by id;
  length btsk1-btsk&numofRaters $4;
        *Loop through tall the raters that are part of the initial file;

        array tempBtsk(&numofRaters) btsk1-btsk&numofRaters;
        do i = 1 to &numofRaters by 1;                  tempBtsk(i) = task1;      end;
  wgt = .0000000001;       output;
  if last.id        then do;                do i = 1 to &numofRaters by 1;                    tempBtsk(i) = ' x ';
        end;       output;      end;    run; quit;

 * CONCATENATE REAL & DUMMY DATA & SORT BY  id ;

 data wtId;     set mappings dummy;     run; quit;
 proc sort data=wtid; by id; run;

 /* Clean up unneeded datasets from SAS */

 proc datasets nodetails;      delete mappings dummy Taskf NumberOfTasks;     run; quit;

/* Beginning creation of final Format table for use in final output   */

data FRater;    input start label $ 7 - 26 fmtname $;
 datalines;
 100  Overall           FRater  ; run; quit;

/* Now calculate the Percent agreements                */

%Macro doby; by id; %mend;
```

```
%Macro doby2;    by ID ;
 if (last.ID);   if (count = 0) then Frequency = 0;
%mend;
%Macro doby3; if (count = 1); %mend;
%Macro keepdo; keep ID; %mend;
%Macro renameit; REName ID = Video; %mend;
%Macro dosort; proc sort data=test&testCompNum; by id; %mend;
%Macro raterclass; class Rater; %mend;
%Macro raterSum;

 /* Delete all extra information that comes from Proc Means. Also calculate       the percent agreement. */

 data combineVids;      set combineVids&indv;    run; quit;

 data combineVids&indv;     set means;
          if (_STAT_ = 'MEAN');      if ( Rater = . ) then delete;
  PercentAgree = NumAgree/totCap;
  Keep Rater NumAgree _Kappa_ L_Kappa U_Kappa totCap PercentAgree;   run; quit;

 /* Sum up all the rater values. This will sum up the raters into one    overall score. */

 proc means data=combineVids&indv noprint;
  var NumAgree totCap _Kappa_ L_Kappa U_Kappa;           output out=means;    run; quit;
%mend;

/* Perform a Macro that calculates the raters. */

%Macro CompRaters;
 %do indv = 1 %to 2;
  %let testCompNum = 0;
  %do I = 1 %to &numofRaters;
   %let K = &I+1;
   %do J = &K %to &numofRaters;

     /* Calculate total number of tests for use in deleting datasets */

     %let testCompNum = %eval(&testCompNum+1);

     /* CREATE DATA SET FOR COMPARING CODERS I & J */
     data test&testCompNum;          set wtId;

      /* DELETE RECORDS WHERE task CODE WAS NOT USED BY EITHER CODER */

      if btsk&I=' x ' and btsk&J=' x ' and wgt=1 then delete;       run; quit;

     /* PRODUCE FREQ TABLE COMPARING CODERS I & J. USE WEIGHT STATEMENT TO
      * OBTAIN SQUARE TABLES, BUT USE NON-WEIGHTED KAPPA RESULTS. Also
      * Obtain kappa statistics and number of agreements and output into  their own dataset */

     proc sort data=test&testCompNum; by id;
     proc freq data=test&testCompNum noprint;          weight wgt;
      tables btsk&I*btsk&J / out=Store norow nocol nopercent agree;
      %if &indv = 1 %then %doby;
          output out=kappafile agree;
```

```
 title 'Kappa calculation / RATER &I AND &J';       run; quit;

/* Keep only the most important columns from the kappaoutput */
data kappafile;          set kappafile;
    %if &indv = 1 %then %keepdo;
 keep _Kappa_ U_Kappa L_Kappa;         run; quit;

/* Keep only the diagonal columns and adjust the count so the  numbers are more pure. */
data store;          set store;
      if (btsk&I = ' x ') then delete;
      if (btsk&J = ' x ') then delete;
      if (COUNT < .9) then COUNT = 0;
      if (COUNT >= 1) then COUNT = 1;         run; quit;
```

/* Sum up and obtain the total number of agreements. This is obtained by summing up the diagonals of the matrix. The more 1's means the more agreements. So the higher the number, the more the one rater agreed at that video. This code also puts the frequencies into a table. */

```
ODS Listing Close;        ODS Output OneWayFreqs = tableconst&testCompNum;
proc freq data = store ;
 %if &indv = 1 %then %doby;
 tables COUNT / norow nocol nopercent;       run; quit;         ODS Listing;
```

/* Get all the agreement frequencies as well as constructing a rater variable to tell us which raters we are comparing and the total number of sequences within the dataset (also known as the total number of comparisons. */

```
%if &indv = 1 %then %dosort;
data tableconst&testCompNum;
 set tableconst&testCompNum;
    %if &indv = 1 %then %doby2;
    %else %doby3;
 Rater1 = &I&J;
    totCap = 10;
 %if &indv = 1 %then %keepdo;
 keep Rater1 Frequency totcap;       run; quit;
```

/* Merge the frequencies and the kappa statistics into 1 file. this combines the one datasets so all the information in one place. */

```
data tableconst&testCompNum;
 merge tableconst&testCompNum kappafile;
    %if &indv = 1 %then %doby;
run; quit;
```

/* Create the format dataset containing the values needed for the final output. This format will be dynamically created with each table creation to help speed up the time it takes to finish rater calculations. This section will be used at the end of the program for formatting the final results. */

```
data FRater2;
 start = &I&J;
 label = "Rater &I vs. Rater &J";
 fmtname = "FRater";       run; quit;

data Frater;
 set Frater Frater2;       run; quit;
```

```
            proc sort data=Frater NODUB; by start; run; quit;       %end;     %end;

  /* Combine all the raters into one table.  */

  data combineVids&indv;
    set tableconst1 - tableconst&testCompNum;
    %if &indv = 1 %then %renameit;
    REName Frequency = NumAgree Rater1 = Rater;   run; quit;

  /* Delete datasets */

  proc datasets nodetails;
    delete Clean Kappafile Store one
          Tableconst1 - Tableconst&testCompNum
          Test1 - Test&testCompNum;   run; quit;

  /* Delete any ob with no agreements. */

  data combineVids&indv;
    set combineVids&indv;
    if NumAgree = 0 then delete;
            PercentAgree = NumAgree/totCap;   run; quit;

  /* Sum up the number of agreements, the total number of comparisons, the  Kappa statistic and its one CI intervals.
This is combining all the  information corresponding to the observer. Then output all the means */

  proc means data=combineVids&indv noprint;
    var NumAgree totCap _Kappa_ L_Kappa U_Kappa;
            %if &indv = 1 %then %raterclass;
            output out = means;   run; quit;

  %if &indv = 1 %then %raterSum;

  /* Adjust the output and delete all extra information that comes from  Proc Means. Also calculate the Percent
Agreement. */

  data means;     set means;
            if (_STAT_ = 'MEAN');     Rater = 100;
    PercentAgree = NumAgree/totCap;
    Keep Rater NumAgree _Kappa_ L_Kappa U_Kappa totCap PercentAgree;   run; quit;

  /* Combine the dataset that holds all the raters separately with the  overall calculation. */

  data OverallSummery&indv;
    set means combineVids&indv;   run; quit;

  /* Delete datasets */

  proc datasets nodetails;
    delete Means combineVids&indv Frater2;   run; quit;
  %end; %mend;

%CompRaters;
```

```
/* Print out calculated datasets for viewing          */

/* Create a format so it is easier to read the output. */

proc format library=work cntlin=Frater; run;

ods rtf;  ods listing close;  title;

  /* Print out our final sets of output  These one outputs correspond to taking into account the individual. */

  proc sort data=combineVids; by Rater;
  proc print data=combineVids split = '*';
    var video rater numagree totcap percentagree _kappa_ l_kappa u_kappa;
              label
                 numagree = ' number of *agreement'
                            totcap   = ' total tasks'
                            percentagree='percentage of * agreement'
                            _kappa_   = 'Kappa agreement'
                            l_kappa = ' lower 95%*CI kappa'
                            u_kappa='Upper 95%*CI Kappa'
                            ;
    title 'Raters along individual ';     format Rater FRater.;     run; quit;

  proc sort data=OverallSummery1; by Rater;
  proc print data=OverallSummery1 split='*';
var  rater numagree totcap percentagree _kappa_ l_kappa u_kappa;
              label
                 numagree = ' number of *agreement'
                            totcap   = ' total tasks'
                            percentagree='percentage of * agreement'
                            _kappa_   = 'Kappa agreement'
                            l_kappa = ' lower 95%*CI kappa'
                            u_kappa='Upper 95%*CI Kappa'
                            ;

    title 'Raters along individual  - Summed';      format Rater FRater.;    run; quit;

  /* Print out our final sets of output  This output corresponds to ignoring the separate individual. */
  proc sort data=OverallSummery2; by Rater;
  proc print data=OverallSummery2 split='*';
var  rater numagree totcap percentagree _kappa_ l_kappa u_kappa;
              label
                 numagree = ' number of *agreement'
                            totcap   = ' total tasks'
                            percentagree='percentage of * agreement'
                            _kappa_   = 'Kappa agreement'
                            l_kappa = ' lower 95%*CI kappa'
                            u_kappa='Upper 95%*CI Kappa'
                            ;
    title 'Raters ignoring individual - Summed';     format Rater FRater.;    run; quit;
ods rtf close;
ods listing;    quit; run; quit;

/* Final Data delete to remove all SAS datasets created by the program */
```

```
 proc datasets nodetails;
   delete combinevids Overallsummery1 Overallsummery2 Wtid Frater;   run; quit;

/* Clear code was taken from the website  http://listserv.uga.edu/cgi-bin/wa?A2=ind0310c&L=sas-l&P=35375
 */
%macro cls();       dm  'clear output';      dm  'clear log';  %mend  cls;  %cls;  * clear the log and output windows ;
```