A SAS@ Macro Application on Confidence Intervals for Binominal Proportion

Kaijun Zhang Sheng Zhang

FMD K&L Inc., Fort Washington, Pennsylvanian

ABSTRACT:

Confidence Intervals (CI) are very important to present results in clinical trial. One of the most common methods for categorical data is used in situations in which the outcomes of interest can be categorized into two, i.e. either a success or a failure. Binomial proportion can either be the proportion of successes or the proportion of failures.

This topic is relevant to both statisticians and programmers because binomial proportion estimation is so commonly applied in a wide variety of situations. SAS procedure which is usually used to obtain these proportions and CIs is the SAS@ procedure PROC FREQ. The common methods for the computation of CI include Wald, Clopper Pearson and Willson score. But sometimes, the direct computation of CI based on mathematical formula may be better choice depending on data source. The right algorithms of CI is regularly chosen by statisticians.

This paper is for SAS@ programmers where three common methods to compute CI for single proportion are presented with executable SAS@ codes based on mathematical formulas and also can be verified by SAS PROC FREQ procedure.

INTRODUCTION

In this presentation, a brief review of the Wald, Wilson-Score, and exact Clopper Pearson methods of calculating confidence intervals for binomial proportions will be presented based on mathematical formulas.

- Wald method: It is the most common method, widely accepted and applied. The 100(1-α)% confidence limits are given by:

$$p = \hat{p} \pm z_{\alpha/2}\sqrt{\hat{p}\hat{q}/n}$$

- Clopper-Pearson method: Based on exact binominal distribution, and not a large sample normal approximation (as is the Wald method). It is given by:

$$\frac{1}{1+\frac{n-x+1}{x}F_{2(n-x+1),2x,\alpha/2}} \le p \le \frac{\frac{x+1}{n-x}F_{2(x+1),2(n-x),\alpha/2}}{1+\frac{x+1}{n-x}F_{2(x+1),2(n-x),\alpha/2}}$$

- Wilson score method: be accurate for most parameter values and does not suffer from being over-conservative, it is given by :

$$p = \frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1+\frac{z^2_{\alpha/2}}{n}}$$

The major way to calculate CIs is SAS procedure with "PROC FREQ", SAS programing according to mathematical formulas is another choice. SAS programs for these formulas are presented and applied to a worked out example (Case study) in **terms of data source**.

CASE STUDY

1. Brief Case Description

1.1 study test:

- N=80,

- 3 repeated evaluations (from visit 2,3,4 evaluation, visit 1 is screen,),
- each evaluation includes:
    - 4 grading Score questions for cushion say-on (grade 1-6)
    - 1 Rubbing Friction question (grade 1-4)
    - 3 subjective assessment questions (Grade 1-7)

1.2. Scope:

- Testing a cushion to adhere in place over a lesion area of the heel when used over a three-day period of activity (Attached Grading Scale (1, 2)).
- Required to have the lower limit of 95% confidence interval above or at 50% of threshold.
- Data source as table below:

Cushion 1: Left heel		Visit 2					Visit 3					Visit 4				
Total Responding:		40					40					40				
Q#	Objective	Base	n	%	Upper	Lower	Base	n	%	Uppr	Lower	Base	n	%	Upper	Lower
	Movement Grading Scale (Score 1,2)	40	39	97.5			40	38	95.0			40	37	92.5		
	Detachment Grading Scale (Score 6)	40	1	2.5			40	2	5.0			40	3	7.5		

Cushion 2: Right heel		Visit 2					Visit 3					Visit 4				
Total Responding:		40					40					40				
Q#	Objective	Base	n	%	Upper	Lower	Base	n	%	Upper	Lower	Base	n	%	Upper	Lower

	Movement Grading Scale (Score 1,2)	40	39	97.5			40	38	95.0			40	37	92.5		
	Detachment Grading Scale (Score 6)	40	1	2.5			40	2	5.0			40	3	7.5		

2. Macros [1] for CI calculation based on mathematical formulas:

```
%macro runprog;
        data propci;
        set parms;
        if sided = 1 then do;
        zalpha=probit(1-(alpha));
        end;
        else if sided = 2 then do;
        zalpha=probit(1-(alpha/2));
        end;
        ** Wald;
        q = 1 - p;
        stder = sqrt(p*q/n);
        Wald_lcl = p - zalpha * stder;
       Wald_ucl = p + zalpha * stder;
       ** Wilson Score;

        part1 = p + ((zalpha**2)/(2*n));
        part2 = sqrt( (p*q/n) + ((zalpha**2)/(4*n**2)) );
        part3 = 1 + (zalpha**2)/n;
```

```
Wilson_lcl = (part1 - (zalpha * part2))/ part3;
Wilson_ucl = (part1 + (zalpha * part2))/ part3;
** Exact Clopper Pearson;
x = round (n*p,0.1);
* Calculate the lower limit.;
v1 = 2*(n-x+1);
v2 = 2*x;
if sided = 1 then do;
a = 1-(alpha);
end;
else if sided = 2 then do;
a = 1-(alpha/2);
end;
coef = (n-x+1)/x;
fscore = finv(a,v1,v2);
exact_lcl = 1/(1+coef*fscore);
* Calculate the upper limit.;
v11 = 2*(x+1);
v22 = 2*(n-x);
fscore2 = finv(a,v11,v22);
coef2 = (x+1)/(n-x);
numer = coef2*fscore2;
exact_ucl = numer/(1+numer);
****eliminate negtive value and value greater than 1******;
if Wald_lcl le 0 then Wald_lcl=0;
  else if Wald_ucl gt 1 then Wald_ucl=1;
run;
```

```
        ods rtf file="&root.pg\raw\CI\ci.rtf";
        options nodate;
        title 'Confidence Intervals for a Single Proportion';
        proc print data = propci split = '_' noobs;
        var group visit n n2 p alpha sided Wald_lcl Wald_ucl  exact_lcl exact_ucl Wilson_lcl Wilson_ucl;
        label wald_lcl = 'LCL_(Wald)'
        wald_ucl = 'UCL_(Wald)'
        wilson_lcl = 'LCL_(Wilson_Score)'
        wilson_ucl = 'UCL_(Wilson_Score)'
        sided = 'Sides_on_CI'
        exact_lcl = 'LCL_(Exact)'
        exact_ucl = 'UCL_(Exact)';
        run;
        ods rtf close;
%mend runprog;

data parms;
  infile cards;
  input group $1-17 visit 18-19 n 20-22 p 23-28 alpha 29-33 sided 34-35 n2 36-38 loc $39-49;
        cards;
        left movement    2 41 0.732 0.05 2 30 left heel
        left detachment  2 41 0.537 0.05 2 22 left heel
        left movement    3 41 0.610 0.05 2 25 left heel
        left detachment  3 41 0.512 0.05 2 21 left heel
        left movement    4 41 0.390 0.05 2 16 left heel
        left detachment  4 41 0.293 0.05 2 12 left heel
        right movement   2 41 0.683 0.05 2 28 right heel
```

```
        right detachment 2 41 0.537 0.05 2 22 right heel
        right movement   3 41 0.585 0.05 2 24 right heel
        right detachment 3 41 0.415 0.05 2 17 right heel
        right movement   4 41 0.488 0.05 2 20 right heel
        right detachment 4 41 0.244 0.05 2 10 right heel
        run;
%runprog;
```

3. Verification through "Proc freq" procedure by creating dummy dataset:

```
 proc format;
  value loc    1="Left heel"
               2="right heel"
   ;
  value  vis    2="visit 2"
               3="visit 3"
               4="visit 4"
   ;
   value obj   0="movement"
              1="detachment"
    ;
run;


data dummy0;
     do Loc=1 to 2;
        do vis=2 to 4;
          do j=1 to 40;
```

```
                    obj=0;
                output;
            end;
          end;
        end;
run;


data dummy(drop=j);
   set dummy0;
        if loc=1 then do;
                              if vis=2 and j=1 then obj=1;
                              else if vis=3 and j in (1, 2) then obj=1;
                              else if vis=4 and j in (1, 2,3) then obj=1;
                        end;
        if loc=2 then do;
                             if vis=2 and j=1 then obj=1;
                              else if vis=3 and j in (1, 2) then obj=1;
                                  else if vis=4 and j in (1, 2,3) then obj=1;
                        end;

        location=put(loc,loc.);
        visit=put(vis, vis.);
        objective=put(obj, obj.);
run;

proc sort data=dummy1 out=dummy2;
   where loc=1;
  by  visit loc obj;
```

```
run;


****Take group= “Left movement” achieving score=1 or 2 as an example for calculation of CI;


ods output   Freq.ByGroup1.Table1.BinomialProp=visit2(drop=Table)
             Freq.ByGroup2.Table1.BinomialProp=visit3(drop=Table)
              Freq.ByGroup3.Table1.BinomialProp=visit4(drop=table);
proc freq data=dummy2 ;
      by visit;
          tables loc*obj/ nocum norow binomial;
          exact binomial;
run;

Data visit21(keep=visit label1 CI);
  set visit2(rename=(cValue1=CI));
    if _n_ in (6,7,8);
run;
Data visit31(keep=visit label1 CI);
  set visit3(rename=(cValue1=CI));
    if _n_ in (6,7,8);
run;
Data visit41(keep=visit label1 CI);
  set visit4(rename=(cValue1=CI));
    if _n_ in (6,7,8);
run;

data all;
```

```
    length group loc $20.;
  set visit21 visit31 visit41;
    group="left movement";
        loc="left heel";
run;
```

RESULTS AND DISCUSSIONS

The results of CI calculation are displayed on table 1 through the SAS@ macros based on mathematical formulas of three methods. The available data source is kind of summary data with total counts and percentage of success/failure in excel worksheet (seen in the section "Case Study"), the CIs can't be calculated directly by using PROC FREQ procedure. The SAS@ macros is a good choice for CI calculation although CIs could be achieved by creating dummy SAS dataset through PROC FREQ procedure. In the paper, we use PROC FREQ to verify the result of CIs as seen in the table 2 on Clopper Pearson method. The CIs highlighted in green are the same through in between SAS macros and PROC FREQ procedure.

Estimating the proportion of successes in a population is simple and involves only calculating the ratio of successes to the sample size. The most common method for CI calculation is Wald method, it is flawed and inaccurate for a large range of n and p [1] or when the sample size is pretty small. The exact Clopper-Pearson method, which is based on the exact binomial distribution, and not a large sample normal approximation (as is the Wald method). This confidence interval is very conservative, having coverage levels as high as 99% for a 95% CI. Wilson Score method is often suggested as a compromise. It has been shown to be accurate for most parameter values and does not suffer from being over-conservative, having coverage levels closer to the nominal level of 95% for a 95% CI. From Table1 below, it shows that CI intervals are the narrowest on Wilson Score method and the greatest range on

Wald method. The binomial distribution is a discrete distribution and as such its cumulative probabilities will have discrete jumps,

Table 1. Confidence Intervals by cushion adherence/detachment and visit for three methods

group	visit	n	p	alpha	Sides on CI	LCL (Wald)	UCL (Wald)	LCL (Wilson Score)	UCL (Wilson Score)	LCL (Exact)	UCL (Exact)
left movement	2	40	0.975	0.05	2	0.92662	1.00000	0.87119	0.99557	0.86841	0.99937
left detachment	2	40	0.025	0.05	2	-0.02338	0.07338	0.00443	0.12881	0.00063	0.13159
left movement	3	40	0.950	0.05	2	0.88246	1.00000	0.83496	0.98618	0.83080	0.99389
left detachment	3	40	0.050	0.05	2	-0.01754	0.11754	0.01382	0.16504	0.00611	0.16920
left movement	4	40	0.925	0.05	2	0.84338	1.00000	0.80136	0.97416	0.79614	0.98426
left detachment	4	40	0.075	0.05	2	-0.00662	0.15662	0.02584	0.19864	0.01574	0.20386
Right movement	2	40	0.975	0.05	2	0.92662	1.00000	0.87119	0.99557	0.86841	0.99937
Right detachment	2	40	0.025	0.05	2	-0.02338	0.07338	0.00443	0.12881	0.00063	0.13159
Right movement	3	40	0.950	0.05	2	0.88246	1.01754	0.83496	0.98618	0.83080	0.99389
Right detachment	3	40	0.050	0.05	2	-0.01754	0.11754	0.01382	0.16504	0.00611	0.16920
Right movement	4	40	0.925	0.05	2	0.84338	1.00662	0.80136	0.97416	0.79614	0.98426

Right detachment	4	40	0.075	0.05	2	-0.00662	0.15662	0.02584	0.19864	0.01574	0.20386

Table2. Confidence Intervals for the left heel archiving grading score=1 and 2 on Clopper Pearson method

Group	**Loc**	**Visit**	**Label1**	**CI**
left movement	left heel	visit 2	Exact Conf Limits	
left movement	left heel	visit 2	95% Lower Conf Limit	0.8684
left movement	left heel	visit 2	95% Upper Conf Limit	0.9994
left movement	left heel	visit 3	Exact Conf Limits	
left movement	left heel	visit 3	95% Lower Conf Limit	0.8308
left movement	left heel	visit 3	95% Upper Conf Limit	0.9939
left movement	left heel	visit 4	Exact Conf Limits	
left movement	left heel	visit 4	95% Lower Conf Limit	0.7961
left movement	left heel	visit 4	95% Upper Conf Limit	0.9843

and thus you'll be hard pressed to get (say) exactly 95% coverage. What Clopper-Pearson does do is guarantee that the coverage is AT LEAST 95% (or whatever level you specify) and so it is desirable in that sense. It is able to accomplish this goal by using the exact binomial distribution in its calculations [2].

Conclusion

SAS “Proc freq” procedure and SAS programing according to mathematical formula provide us a flexible way to calculate confidence interval for one sample proportion with binominal distribution.

Reference

1. Keith Dunnigan. Confidence Interval Calculation for Binomial Proportions. Pharmsug, P08, 2008.
2. Agresti, A. and Coull, B. A. (1998). Approximate is Better than “Exact” for Interval Estimation of Binomial Proportions. The American Statistician 52, 119-126.