Paper SD103

Confidence Intervals for Binomial Proportion Using SAS®: The All You Need to Know and No More…

Jiangtang Hu
d-Wise, Morrisville, NC

ABSTRACT

Confidence Intervals (CI) are extremely important in presenting clinical results. The choosing of right algorithms of CI is the plate of statisticians, and this paper is for SAS programmers where more than 14 methods to compute CI for single proportion is presented with executable SAS codes, by SAS procedures and customized codes from the scratch.

These codes is currently hosted in my Github page:

https://raw.githubusercontent.com/Jiangtang/Programming-SAS/master/CI_Single_Proportion.sas

Some commentaries from A SAS programmer's point of view will also be presented.

INTRODUCTION

Suppose n is the same size, r the number of count of interested outcome, and p = r / n is so called binomial proportion (sample proportion). A confidence interval (CI) is a range of values, computed from the sample, which is with probability of 95% to cover the population proportion (well, you may use any pre-specified probabilities, but 95% is the most common one). From statistical point of view, confidence intervals are generally more informative than p-value. In clinical studies, the size of difference of the outcome between groups (measured by confidence intervals) is much more useful for researchers than the single significant indicator, namely, p-value [1].

For SAS programmers, it's nice to know that the confidence interval is much preferred to p-value for presenting clinical outcomes. Computing varieties of confidence intervals (there are A LOT!) is part of SAS programmers' daily job. The selection of the right type of CI is decided by statisticians, and this paper is primary for SAS programmers: you might not be very familiar with the story behind the different CIs, but with this paper and SAS codes attached, you can always get the right results given the type of CIs is articulated in the Statistical Analysis Plan(SAP).

I will leave the calculation of confidence interval for the difference between independent proportions [2] to another paper and this one is only for single binomial proportion. Particularly, 14 methods will be presented with fully executable SAS programs:

1. Simple asymptotic, without Continuity Correction (CC), mostly know as **Wald**
2. Simple asymptotic, with CC
3. Score method, without CC, also known as **Wilson**
4. Score method, with CC
5. Binomial-based, 'Exact' or **Clopper-Pearson**
6. Binomial-based, Mid-p
7. Likelihood-based
8. Jeffreys
9. Agresti-Coull, pseudo frequency, z^2/2 successes| psi = z^2/2
10. Agresti-Coull, pseudo frequency, 2 successes and 2 failures| psi = 2
11. Agresti-Coull, pseudo frequency, psi = 1
12. Agresti-Coull, pseudo frequency, psi = 3
13. Logit
14. Blaker

To get a quick impression, just run the following piece of code:

```
filename CI url 'https://raw.github.com/Jiangtang/Programming-
SAS/master/CI_Single_Proportion.sas';
%include CI;
%CI_Single_Proportion(r=81,n=263);
```

The output (note: CC is short for Continuity Correction):

r	n	p	method	p_CI
81	263	0.30798	1. Simple asymptotic, Without CC | Wald	[0.2522,0.3638]
81	263	0.30798	2. Simple asymptotic, With CC	[0.2503,0.3657]
81	263	0.30798	3. Score method, Without CC | Wilson	[0.2553,0.3662]
81	263	0.30798	4. Score method, With CC	[0.2535,0.3682]
81	263	0.30798	5. Binomial-based, 'Exact' | Clopper-Pearson	[0.2527,0.3676]
81	263	0.30798	6. Binomial-based, Mid-p	[0.2544,0.3658]
81	263	0.30798	7. Likelihood-based	[0.2542,0.3655]
81	263	0.30798	8. Jeffreys	[0.2545,0.3656]
81	263	0.30798	9. Agresti-Coull, pseudo frequency, z^2/2 successes| psi = z^2/2	[0.2552,0.3663]
81	263	0.30798	10. Agresti-Coull, pseudo frequency, 2 successes and 2 failures| psi = 2	[0.2553,0.3664]
81	263	0.30798	11. Agresti-Coull, pseudo frequency, psi = 1	[0.2538,0.3651]
81	263	0.30798	12. Agresti-Coull, pseudo frequency, psi = 3	[0.2569,0.3676]
81	263	0.30798	13. Logit	[0.2551,0.3664]
81	263	0.30798	14. Blaker	[0.2539,0.3665]

Method #1-7 was well documented in a famous paper by Robert Newcombe, *Two-sided confidence intervals for the single proportion: comparison of seven methods* [3] where the corresponding output is:

Method	n r	263 81
Simple asymptotic		
1 Without CC		0·2522, 0·3638
2 With CC		0·2503, 0·3657
Score method		
3 Without CC		0·2553, 0·3662
4 With CC		0·2535, 0·3682
Binomial-based		
5 'Exact'		0·2527, 0·3676
6 Mid-p		0·2544, 0·3658
Likelihood-based		
7		0·2542, 0·3655

CC: continuity correction

A. BASIC ALGORITHMS AND COMMENTS

A1 Method #1 and #2

The most popular method in introductory textbooks (note not in practice) is method #1 ('Wald') due to its simplest form:

$$p \pm z \times \mathrm{SE}$$

where

p is the empirical estimate of the proportion, r / n,

SE, standard error = √ (p(1-p)/n)

z, the quantile of the standard normal distribution (1.960 for the usual two-sided 95% interval).

Despite its popularity, the Wald method is very deficient. For example, it is not boundary-respecting and it can extend beyond 0 or 1.

When p = 0 or 1, method #1 ('Wald') will get a zero width interval [0, 0]. To avoid this degeneracy issue, method #2 ('Wald with CC') introduces a continuity correction (CC), namely, $1 / (2n)$. But be aware of the trade-off, the problem of adding the continuity correction is that it would lead to more instances of overshoot. A step-by-step approach to demonstrate these two methods is as follows:

```
data m1;
      r = 81;
      n = 263;
      alpha = 0.05;
      p = r / n;
      z = probit (1-alpha/2);

      *standard error;
      se = (sqrt(n*p*(1-p)))/n;

      L =  p - z * se;
      U  = p + z * se;

      put L= U= ;
run;
```

```
data m2;
      r = 81;
      n = 263;
      alpha = 0.05;
      p = r / n;
      z = probit (1-alpha/2);

      *standard error;
      se = (sqrt(n*p*(1-p)))/n;

      *continuity correction;
      cc = 1/(2*n);

      L = p - (z * se + cc);
      U = p + (z * se + cc);

      put  L= U=;
run;
```

A2 Method #13

The Logit method (#13) is actually very similar to Wald method in structure. Instead using p = r / n as the empirical proportion, a logit transformation is applied in Logit method, namely, exp[log (p / (1-p))]:

```
data m13;
      r = 81;
      n = 263;
      alpha = 0.05;
      p = r / n;
      z = probit (1-alpha/2);

      L = exp(log(p/(1-p)) - z*sqrt(n/(r*(n-r)))) /
                  (1+exp(log(p/(1-p)) - z*sqrt(n/(r*(n-r)))));
      U = exp(log(p/(1-p)) + z*sqrt(n/(r*(n-r)))) /
                  (1+exp(log(p/(1-p)) + z*sqrt(n/(r*(n-r)))));
      put L= U=;
run;
```

Logit method is often used for odds ratios. Like the Wald method, it is not guaranteed satisfactory when n is small or p is close to 0 or 1.

A3 Method #3, #4

Wilson score method (#3) is considered the simplest acceptable alternative to the Wald approach. It gets better performance when n is small and when p is close to 0 or 1.

Wilson method is also not boundary-respecting. A continuity correction can be applied to get method #4:

```
data m3;
      r = 81;
      n = 263;
      alpha = 0.05;
      p = r / n;
      q = 1-p;
      z = probit (1-alpha/2);

      L = ( 2*r+z**2 -
(z*sqrt(z**2+4*r*q)) ) /
(2*(n+z**2));
      U = ( 2*r+z**2 +
(z*sqrt(z**2+4*r*q)) ) /
(2*(n+z**2));
      put L= U=;
run;
```

```
data m4;
      r = 81;
      n = 263;
      alpha = 0.05;
      p = r / n;
      q = 1-p;
      z = probit (1-alpha/2);

      L = ( 2*r+z**2 -1 -
z*sqrt(z**2 - 2- 1/n +
4*p*(n*q+1))) / (2*(n+z**2));
      U = ( 2*r+z**2 +1 +
z*sqrt(z**2 + 2- 1/n + 4*p*(n*q-
1))) / (2*(n+z**2));
      put L= U=;
run;
```

A4 Method #5

The Wald-like intervals described as above are all asymptotic intervals. The so called Clopper-Pearson 'exact' method (#5) is quite different since it's very conservative. It's very computationally convenient and only one inverse Beta function is used:

```
data m5;
      r = 81;
      n = 263;
      alpha = 0.05;

      L = 1 - betainv(1 - alpha/2,n-r+1,r);
      U =     betainv(1 - alpha/2,r+1   ,n-r);
      put L= U=;
run;
```

A5 Method #6

The method #5 is so conservative that sometimes it's even unnecessary. A similar 'mid-p' (#6) is used to reduce the conservatism. Method #6 accumulates the tail areas and it's relatively complicated compared to any methods mentioned above. The full code is available in the master code (of session '`method 6`'):

https://raw.github.com/Jiangtang/Programming-SAS/master/CI_Single_Proportion.sas

A6 Method #7

Likelihood-based approach (#7) is said to be theoretically appealing. The full program is available here (of session '`method 7`'):

https://raw.github.com/Jiangtang/Programming-SAS/master/CI_Single_Proportion.sas

A7 Method #9-#12

Methods #9-#12 come from the family of 'pseduo-frequency methods'. Instead of using p = r / n, 4 methods are developed when ψ (pronounced as psi) gets values $z_2/2$, 2, 1, 3:

$$p_\psi = (r + \psi)/(n + 2\psi)$$

```
data m9_12;
```

```
        r = 81;
        n = 263;
        alpha = 0.05;
        z = probit (1-alpha/2);

        do psi = z**2/2, 2, 1, 3;
              p2=(r+psi)/(n+2*psi);

              L = p2 - z*(sqrt(p2*(1-p2)/(n+2*psi)));
              U = p2 + z*(sqrt(p2*(1-p2)/(n+2*psi)));
              put L= U=;
              output;
        end;
run;
```

A8 Method #8

The methods mentioned above are all frequentist methods. The alternative Bayesian methods are sometimes preferable because they incorporate prior information. I'm not well equipped with Bayesian methods, but one called 'Jeffreys' method is pretty straightforward from programing point of view:

```
data m8;
        r = 81;
        n = 263;
        alpha = 0.05;

        L = betainv(  alpha/2, r+0.5,n-r+0.5);
        U = betainv(1-alpha/2, r+0.5,n-r+0.5);
        put L= U=;
run;
```

A9 Method #14

Although CIs using the Blaker method are available with options from a SAS FREQ procedure, the program below manually computes the CI lower and upper bounds using the Blaker method:

```
data m14;
      r = 81;
      n = 263;
      alpha = 0.05;

      tolerance=1e-05;
      lower = 0;
      upper = 1;

      if r ^= 0 then do;
         lower = quantile('BETA',alpha/2, r, n-r+1);
       do while (acceptbin(r, n, lower + tolerance) < (alpha));
               lower = lower + tolerance;
       end;
      end;


      if r ^= n then do;
      upper = quantile('BETA',1 - alpha/2, r+1, n-r);
      do while (acceptbin(r, n, upper - tolerance) < (alpha));
                  upper = upper - tolerance;
        end;
      end;
```

```
    L=lower;
    U =upper;
      put L= U=;
run;
```

B. THE WORKHORSE: SAS FREQ

The individual SAS programs above can be used for educational purpose to better understand the various methods one can use to calculate CIs. I compiled them together into a macro and can be called as

```
filename CI url 'https://raw.github.com/Jiangtang/Programming-
SAS/master/CI_Single_Proportion.sas';
%include CI;
%CI_Single_Proportion(r=81,n=263);
```

which can print out all outputs from the14 methods together for comparison.

Alternatively, SAS PROC FREQ (I use SAS 9.4 TS1M2 in Windows 7 when writing this paper) can be used to compute most of the confidence intervals (11 to be precisely):

```
data test;
input grp outcome $ count;
datalines;
1 f 81
1 u 182
;

/*2.3 CL=enumeration:  #1,#3, #5,#6, #7,#8, #9,#13, #14*/
ods select BinomialCLs;
proc freq data=test;
      tables outcome / binomial (CL=
                                             WALD
                                             WILSON
                                             CLOPPERPEARSON
                                             MIDP
                                             LIKELIHOODRATIO
                                             JEFFREYS
                                             AGRESTICOULL
                                             LOGIT
                                             BLAKER
                                            );
      weight Count;
run;

/*2.4 CL=CC : #2,#4*/
ods select BinomialCLs;
proc freq data=test;
    tables outcome / binomial (CL =
                                            WILSON(CORRECT)
                                            WALD(CORRECT)
                                            );
    weight Count;
run;
```

The output:

The FREQ Procedure

Confidence Limits for the Binomial Proportion		
outcome = f		
Proportion = 0.3080		
Type	95% Confidence Limits	
Agresti-Coull 9	0.2552	0.3663
Blaker 14	0.2539	0.3665
Clopper-Pearson (Exact) 5	0.2527	0.3676
Jeffreys 8	0.2545	0.3656
Likelihood Ratio 7	0.2542	0.3655
Logit 13	0.2551	0.3664
Mid-p 6	0.2544	0.3658
Wald 1	0.2522	0.3638
Wilson 3	0.2553	0.3662

The FREQ Procedure

Confidence Limits for the Binomial Proportion		
outcome = f		
Proportion = 0.3080		
Type	95% Confidence Limits	
Wald (Corrected) 2	0.2503	0.3657
Wilson (Corrected) 4	0.2535	0.3682

Only methods #10 - #12 from 'pseduo-frequency methods' family are not available in SAS PROC FREQ.

A programming note:

1) If no confidence interval option assigned, the Wald and the 'exact' CIs will be presented.
2) If option `CL = All` is applied, the following 5 CIs will be computed: Agresti-Coull, Clopper-Pearson (Exact), Jeffreys, Wald, Wilson.

C. OTHER SAS PRECEDURES

SAS PROC MCMC can be used to compute Bayesian confidence interval but I'm not studying it yet.

D. CONCLUSION

SAS PROC FREQ is pretty impressive to compute varieties of confidence intervals. The customized SAS programs in this paper are best used for educational purposes, or to serve as a start to customize your own pieces of CI programs.

REFERENCES

[1] *Confidence Intervals for Proportions and Related Measures of Effect Size by Robert Newcombe, 2012, CRC Press*

[2] *Interval estimation for the difference between independent proportions: comparison of eleven methods by Robert Newcombe, 1998, STATISTICS IN MEDICINE*

[3] *Two-sided confidence intervals for the single proportion: comparison of seven methods by Robert Newcombe, 1998, STATISTICS IN MEDICINE*

ACKNOWLEDGMENTS

I highly recommend Robert newcombe’s book and papers to jump into the world of confidence intervals. I personally gain a lot from them.

CONTACT INFORMATION

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

Jiangtang(‘JT’) Hu
Life Sciences Consultant
d-Wise Technologies, Inc.
1500 Perimeter Park Dr., Suite 150, Morrisville, NC, 27560
www.d-Wise.com

(O) 919-334-6096
(C) 919-801-9659
(F): 888-563-0931
(O) Jiangtang.Hu@d-Wise.com
(P) jiangtanghu@gmail.com
http://jiangtanghu.com/