ABSTRACT

Given a sample of $n$ Bernoulli random variables, each with probability $p$ of success, we wish to construct a confidence interval for the proportion $x/n$, where $x$ denotes the number of 'successes' out of the $n$ samples. The most widely known method is the normal approximation, but it may have serious shortcomings if $n$ is small or if the proportion is close to 0 or 1. Alternative methods have been developed and this paper presents 12 different methods for constructing such confidence intervals, 11 of which are available in SAS® 9.4. A comparison of the methods is shown and some thoughts on which method to use are given.

BACKGROUND (7 METHODS)

Suppose we have $n$ subjects with $x$ 'successes'. We can define the proportion, $p$, of successes as $p = x/n$. Let $z$ correspond to the $a/2$-percentile of the normal distribution. Thus, for a 95%CI (ie, $a=0.05$), the corresponding $z$ value would be 1.96.

Newcombe (1998) provides 7 methods for constructing confidence intervals for a single binomial proportion. They are as follows:

1. Wald method without continuity correction

\[
p \pm z\sqrt{p(1-p)/n}
\]

2. Wald method with continuity correction

\[
p \pm (z\sqrt{p(1-p)/n + 1/(2n)})
\]

3. Wilson's 'score' method without continuity correction

\[
\left(2np + z^2 \pm z\sqrt{z^2 + 4np(1-p)n}/2(n+z^2)\right)
\]

4. Score method with continuity correction

\[
\frac{2np + z^2}{2(n+z^2)} \pm \frac{1 + z\sqrt{z^2 - 2 - 1/n + 4p(1-p) + 1}}{2(n+z^2)}
\]

5. 'Exact' binomial (Clopper-Pearson)

Find the interval $[L, U]$ with $L \leq p \leq U$ such that for all $\theta$ in the interval:

(i) If $L \leq \theta \leq p$: $\sum_{j=x}^{\min(x, n)} p_j \leq 1 - a/2$

(ii) If $p \leq \theta \leq U$: $\sum_{j=x}^{\min(x, n)} p_j \geq a/2$

where $p_j = \Pr(N = j) = \binom{n}{j} \theta^j (1 - \theta)^{n-j}$, $j = 0, 1, \ldots, n$. $N$ denotes the random variable for which $j$ is the realization, and $k = 1$. Empty summations are understood to be zero.

PROC FREQ uses the $F$ distribution to compute the exact confidence limits according to the following:

Lower bound: \(1 + \frac{n-x+1}{x^2(1-a/2, x, 2(n-x+1))}^{-1}\) \quad Upper bound: \(1 + \frac{n-x}{(x+1)F(a/2, 2(x+1), 2(n-x))}^{-1}\)

where $F(a, b, c)$ is the $a$th percentile of the $F$ distribution with $b$ and $c$ degrees of freedom.
6. 'Mid-p' binomial method

This method is the same as Method 5, but with $k = 1/2$.

7. Likelihood-based method

All $\theta$ satisfying: $x \ln(\theta) + (n - x) \ln(1 - \theta) \geq x \ln(p) + (n - x) \ln(1 - p) - z^2/2$

FOUR ADDITIONAL METHODS

In SAS® 9.4, all seven of these methods are available using the FREQ procedure. Moreover, 4 additional methods are also available.

8. Agresti-Coull

This method has the same form as the Wald method (Method 1), but the parameters have been updated as follows:

$$\hat{p} \pm z\sqrt{\hat{p}(1 - \hat{p})/\hat{n}}$$

where $\hat{n} = n + z^2$ and $\hat{p} = (x + z^2/2)/\hat{n}$

9. Jeffreys

Lower bound: $\beta(\alpha/2, x + 1/2, n - x + 1/2)$
Upper bound: $\beta(1 - \alpha/2, x + 1/2, n - x + 1/2)$

where $\beta(a, b, c)$ is the $a^{th}$ percentile of the beta distribution with parameters $b$ and $c$. The lower confidence limit is set to 0 when $x = 0$ and the upper confidence limit is set to 1 when $x = n$.

10. Logit method

The data is log-transformed via $Y = \log[p/(1-p)]$ and CIs for $Y$ are computed as follows:

Lower bound: $Y_L = \log[p/(1-p)] - z\sqrt{n/(x(n-x))}$
Upper bound: $Y_U = \log[p/(1-p)] + z\sqrt{n/(x(n-x))}$

Then the confidence limits are remapped (by taking exponents) to obtain approximate confidence limits

Lower bound: $\exp(Y_L/(1 + \exp(Y_L)))$
Upper bound: $\exp(Y_U/(1 + \exp(Y_U)))$

11. Blaker method

The confidence interval consists of all values of $\theta$ for which $\{\theta: B(\theta, x) > a\}$, where

$$B(\theta, x) = \Pr[y(\theta, X) \leq y(\theta, x)|\theta]$$
$$y(\theta, x) = \min(\Pr[X \geq x|\theta], \Pr[X \leq x|\theta])$$

Looking at Method 8 (Agresti-Coull), when $\alpha = 0.05$, $z \approx 2$ and thus $z^2 = 4$ and $z^2/2 = 2$. Thus, for $\alpha = 0.05$, Method 8 could be approximated by replacing $x$ with $x + 2$ and $n$ with $n + 4$. This known as the “add 2 successes and 2 failures” version of the Agresti-Coull method.

IMPLEMENTATION IN SAS®

As noted above, Methods 1-11 are available in SAS® 9.4. The code to generate these CIs is listed below:
Constructing Confidence Intervals for a Single Binomial Proportion in SAS®, Continued

```sas
data testdata;
    input trial x n alpha;
datalines;
    1 81 263 0.05
    2 0 20 0.05
;
data testdat1;
    set testdata;
    y = n - x;
    p = x/n;
    z = probit(1-alpha/2);
run;
proc transpose data = testdat1 out = x_data(rename=(_NAME_=outcome COL1=count));
    by trial;
    var x y;
run;
ods output BinomialCLs = CLs1;
proc freq data = x_data;
    tables outcome /binomial(cl=(ac blaker exact jeffreys lr logit midp wald score));
    weight count;
run;
ods output BinomialCLs = CLs2;
proc freq data = x_data;
    tables outcome /binomial(cl=(wald(correct) score(correct)));
    weight count;
run;
In earlier versions of SAS® (eg, SAS® 9.2), only Methods 1-3, 5 and 8-9 are available in SAS® 9.2 via PROC FREQ.
ods output BinomialCLs = CLs1;
proc freq data = x_data;
    by trial;
    tables outcome /binomial(cl=(ac exact jeffreys wald score));
    weight count / zero;
run;
ods output BinomialCLs = CLs2;
proc freq data = x_data;
    by trial;
    tables outcome /binomialc(cl=(wald));
    weight count / zero;
run;
We need to stack the two datasets together, and update labels if using SAS® 9.2 or 9.3.

data CLs;
    set CLs1(in=a) CLs2(in=b);
    *** Fix labels for SAS 9.2 or 9.3 ***;
    if b then do;
        if type = "Wald" then type = "Wald (Corrected)";
    end;
run;
The remaining methods (Methods 4, 6, 7, 10, and 11) need to be written using the formulas above. Code to do so appears in Appendix A.

As a side note, Hu (2011) wrote a macro which generates CIs using methods 1-11 above, all in open code (without invoking SAS® procedures).
```

3
ONE MORE METHOD

In passing there is one more method worth mentioning proposed by Blyth and Still (1983). It is an exact method that has the shortest possible intervals and is referred to as the Blyth-Still-Casella method (Method 12). -

However, the resulting CIs are not necessarily nested (Blaker [2000], Theorem 2). That is, the 90% CI is not necessarily contained within the 95% CI. Blaker’s method (Method 11) modifies the Blyth-Still-Casella method to ensure this nesting property (although the CIs are wider).

SAS® does not currently have an implementation for the Blyth-Still-Casella method, but StatXact does. The code using Proc StatXact appears in Appendix B.

COMPARING THE METHODS

Newcombe (1998) provides a table of examples for the initial 7 methods. The table is recast below, with entries for the 5 additional methods as well.

<table>
<thead>
<tr>
<th>Method</th>
<th>n = 263; x = 81</th>
<th>n = 20; x = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple asymptotic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Without CC</td>
<td>0.2522, 0.3638</td>
<td>0.0000, 0.0000*</td>
</tr>
<tr>
<td>2 With CC</td>
<td>0.2503, 0.3657</td>
<td>&lt;0.0000*, 0.0250</td>
</tr>
<tr>
<td>Score method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Without CC</td>
<td>0.2553, 0.3662</td>
<td>0.0000, 0.1611</td>
</tr>
<tr>
<td>4 With CC</td>
<td>0.2535, 0.3682</td>
<td>0.0000, 0.2005</td>
</tr>
<tr>
<td>Binomial-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 ‘Exact’</td>
<td>0.2527, 0.3676</td>
<td>0.0000, 0.1684</td>
</tr>
<tr>
<td>6 Mid-p</td>
<td>0.2544, 0.3658</td>
<td>0.0000, 0.1391</td>
</tr>
<tr>
<td>Likelihood-based</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.2542, 0.3655</td>
<td>0.0000, 0.0916</td>
</tr>
<tr>
<td>Additional methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 Agresti-Coull</td>
<td>0.2552, 0.3663</td>
<td>0.0000, 0.1898</td>
</tr>
<tr>
<td>9 Jeffreys</td>
<td>0.2545, 0.3656</td>
<td>0.0000, 0.1166</td>
</tr>
<tr>
<td>10 Logit</td>
<td>0.2551, 0.3664</td>
<td>0.0000, 0.1598</td>
</tr>
<tr>
<td>11 Blaker</td>
<td>0.2539, 0.3665</td>
<td>0.0000, 0.1601</td>
</tr>
<tr>
<td>12 Blyth-Still-Casella</td>
<td>0.2527, 0.3653</td>
<td>0.0000, 0.1539</td>
</tr>
</tbody>
</table>

CC: continuity correction  *aberrations (directly calculated limits outside [0, 1] or zero-width intervals)

When comparing the initial 7 methods, Newcombe noted problems with the simple asymptotic methods, in that they are very anti-conservative and results in asymmetrical coverage, especially near the boundaries of 0 and 1.

Moreover, it is noted that the binomial-based methods perform well, though the ‘Exact’ method (Method 5) is too conservative. Method 6 (Mid-p) seems to be the most preferred, though Method 3 (Score) also performs well and is easy to calculate by hand. However, given that all methods are readily available in SAS®, this is not a convincing argument. However, Reiczigel (2003) argued that the asymptotic behavior of the Mid-p method (Method 6) is not that great.

Brown, Cai and DasGupta (2001) argue for Method 3 (Score) or Method 9 (Jeffreys) for small n and Method 8 (Agresti-Coull) for larger n.

Thus, there is no single consensus on what constitutes the most appropriate confidence interval for a binomial proportion. In this author’s opinion, Method 3 (Score) seems to be a good choice as an approximation and Method 11 (Blaker) seems to be the best option in terms of an “exact” confidence interval that guarantees nominal coverage.

SUMMARY

Multiple methods exist for computing confidence intervals for binomial proportions and there is no single ‘correct’ method. The general consensus is that the most widely taught method (Method 1) is inappropriate in most situations and should not be used. Binomial-based methods do well at providing exact coverage, although they may be overly conservative. In terms of approximation methods, the Score method (Method 3) seems to perform well.
REFERENCES


CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Name: Will Garner
Company: Gilead Sciences, Inc.
Address: 333 Lakeside Dr
City, State ZIP: Foster City, CA 94404
Work Phone: (650) 522-5691
E-mail: will.garner@gilead.com

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.
APPENDIX A – SAS® CODE TO GENERATE METHODS 4, 6, 7, 10 AND 11

Below is SAS® code to implement the methods not available in SAS® 9.2 or 9.3.

/* Method 4: Continuity-Corrected Wilson Score */
data method4(keep=trial LowerCL UpperCL method);
  set testdat1;
  length method $25.;
  method = " 4. Score With CC";
  LowerCL = (2*n*p+z**2-1-z*sqrt(z**2-2-1/n+4*p*(n*(1-p)+1)))/(2*(n+z**2));
  UpperCL = (2*n*p+z**2+1+z*sqrt(z**2+2-1/n+4*p*(n*(1-p)-1)))/(2*(n+z**2));
  if x = 0 then LowerCL = 0;  if x = n then UpperCL = 1;
run;

/* Method 6: Mid-p */
data method6;
  set testdat1;
  by trial;
  do j = 0.000001 to 0.999999 by 0.00001;
    if (x > 0 and x < n) then a2 = 0.5*probnml(j,n,x-1)+0.5*probnml(j,n,x);
    output;
  end;
run;
data max(keep=j rename=(j=UpperCL));
  set method6;
  where a2 <= alpha/2 and x > 0 and x < n;
  by trial;
  if first.trial;
run;
data min(keep=j rename=(j=LowerCL));
  set method6;
  where a2 > 1-alpha/2 and x > 0 and x < n;
  by trial;
  if last.trial;
run;
data method6(keep=trial LowerCL UpperCL method);
  merge testdat1 min max;
  length method $25.;
  method = " 6. Mid-p";
  if x = 0 then do;
    LowerCL = 0;  UpperCL = 1-alpha**(1/n);
  end;
  if x = n then do;
    LowerCL = alpha**(1/n);  UpperCL = 1;
  end;
run;

/* Method 7: Likelihood ratio */
data method7;
  set testdat1;
  by trial;
  k = -cinv(1-alpha,1)/2;
  do j = 0.000001 to 0.999999 by 0.000001;
    lik=pdf('Binomial',x,j,n);  output;
  end;
run;
proc sort data = method7 out = max;
  by trial descending lik;
run;
data max(keep=trial lik rename=(lik=max));
   set max;
   by trial;
   if first.trial;
run;

data method7;
   merge method7 max;
   by trial;
   if lik ^= 0 then logLR = log(lik/max);
run;

data min(keep=j rename=(j=LowerCL)) max(keep=j rename=(j=UpperCL));
   set method7;
   by trial;
   where logLR > k;
   if first.trial then output min;
   if last.trial then output max;
run;

data method7(keep=trial LowerCL UpperCL method);
   merge testdat1 min max;
length method $25.;
   method = " 7. Likelihood Ratio";
   if x = 0 then LowerCL = 0;  if x = n then UpperCL = 1;
run;

/* Method 10: Logit */
data method10(keep=trial LowerCL UpperCL method);
   set testdat1;
length method $25.;
   method = "10. Logit";
   if x = 0 then LowerCL = 0;  else if x = n then UpperCL = 1;
else do;
   LowerY = log(p/((1-p)) - z*sqrt(n/(x*(n-x))));
   UpperY = log(p/((1-p)) + z*sqrt(n/(x*(n-x))));
   LowerCL = exp(LowerY)/(1+exp(LowerY));
   UpperCL = exp(UpperY)/(1+exp(UpperY));
end;
run;

/* Method 11: Blaker */
data method11(keep=trial n x alpha);
   set testdat1;
run;

data min(keep=trial LowerCL);
   set method11;
   if x = 0 then LowerCL = 0;
else if x > 0 then do;
   value = quantile('Beta',alpha/2,x,n-x+1);
   do while (gamma < alpha);
      p1 = 1-cdf('Binomial',x-1,value,n);
      p2 = cdf('Binomial',x,value,n);
      a1 = p1+cdf('Binomial',quantile('Binomial',p1,value,n)-1,value,n);
      a2 = p2+1-cdf('Binomial',quantile('Binomial',1-p2,value,n),value,n);
      gamma = min(a1, a2);
      value = value + 0.00001;
   end;
   LowerCL = value;
end;
run;
data max(keep=trial UpperCL);
  set method11;
  if x = n then UpperCL = 1;
  else if x < n then do;
    value = quantile('Beta', 1-alpha/2, x+1, n-x);
    do while (gamma < alpha);
      p1 = 1 - cdf('Binomial', x-1, value, n);
      p2 = cdf('Binomial', x, value, n);
      a1 = p1 + cdf('Binomial', quantile('Binomial', p1, value, n) - 1, value, n);
      a2 = p2 + 1 - cdf('Binomial', quantile('Binomial', 1 - p2, value, n), value, n);
      gamma = min(a1, a2);
      value = value - 0.00001;
    end;
    UpperCL = value;
  end;
run;

data method11(keep=trial LowerCL UpperCL method);
  merge method11 min max;
  by trial;
  length method $25.;
  method = "11. Blaker";
run;

APPENDIX B – SAS® CODE TO GENERATE METHOD 12 USING PROC STATXACT

Below is SAS® code to implement Method 12 (Blyth-Still-Casella) using PROC StatXact.

proc binomial data = x_data out=exact;
  by trial;
  bi/bs;
  ou outcome;
  weight count;
run;

data exactci;
  set exact(keep=item value test by_val);
  if item in ('EST_PI' 'B_LCI_PI' 'B_UCI_PI');
  trial = input(by_val, 8.);
run;

proc sort data = exactci;
  by test trial;
run;

proc transpose data = exactci out = exactci;
  by test trial;
  var value;
  id item;
run;

data final(keep=trial Proportion LowerCL UpperCL method);
  set exactci;
  label Proportion = "Proportion"
    LowerCL = "95% Lower Confidence Limit"
    UpperCL = "95% Upper Confidence Limit";
  Proportion = est_pi;
  LowerCL = b_lci_pi;
  UpperCL = b_uci_pi;
  length method $25.;
  method = "12. Blyth-Still-Casella";
run;