Estimating a Confidence Interval for a Proportion

Using the "Wilson Score Interval" in R

Art Eschenlauer

January 25, 2019

Contents

Introductory remarks

Many ways have been developed to estimate the confidence interval for an observed proportion between success and trials where the probability of success is fixed. This document will explain why I think that the following line of R code is all that you need to estimate the confidence interval:

```
epitools::binom.wilson(x = successes, n = trials, conf.level = 0.95)
```

For a list of several methods for computing the confidence interval, see https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval.

Agresti and Coull (1998) recommended that elementary statistics instruction prefer the "Score" method of Wilson (1927) for estimating the confidence interval over the simpler, but insufficiently conservative, normal approximation. Wilson's formula is presented below.

I personally prefer the Wilson Score method because it performs well for proportions that approach 0 or 1, where the normal approximation breaks down. To facilitate my understanding of the Wilson Score formula, I derived it; see section 'Derivation of Wilson's "Score Interval" formula'.

My test case

For my example, I will assume that 73 successes were observed for 76 trials, and that I want a 95% confidence interval.

```r
successes <- 73
trials <- 76
proportion <- as.double(successes) / as.double(trials)
```

The Wilson Score Interval - from Wilson 1927

The 1927 paper by Edwin Bidwell Wilson presented the following formula as a practical way to estimate the confidence interval for a proportion:

$$p = \frac{p_0 + \frac{t}{2} \pm \sqrt{p_0 q_0 t + \frac{t^2}{4}}}{1+t} \quad \Bigg| \quad \begin{array}{l} \lambda = 1.96 \text{ (for 95 percent confidence)}, \\ t = \frac{\lambda^2}{n}, \quad p_0 = \frac{x}{n}, \quad q_0 = \frac{n-x}{n} \end{array} \tag{1}$$

The CRAN R package `epitools` provides the `binom.wilson` function which can compute this:

```
epi_wilson_ci <-
  epitools::binom.wilson(x = successes, n = trials, conf.level = 0.95)
print(epi_wilson_ci)
```

```
##    x  n proportion     lower     upper conf.level
## 1 73 76  0.9605263 0.8902521 0.9864854       0.95
```

Hoping to demystify the `epitools` implementation somewhat, I coded it directly from Wilson's equation:

```
# computational equivalent of binom.wilson from CRAN 'epitools' package
binom.wilson <-
  function(x, n, conf.level = 0.95) {
    lambda  <- qnorm(0.5*(1 + conf.level))
    # Adapted from Wilson 1927, mostly by multipling numerator
    #   and denominator by n to reduce roundoff error
    nt     <- ( lambda^2 )
    np_0   <- x
    nq_0   <- n - x
    denom   <- n + nt
    center  <- ( np_0 + nt / 2 )
    radical <- ( np_0 * nq_0 + n * (nt / 4) ) * nt / n
    delta   <- sqrt(radical)
    R.lower <- ( center - delta ) / denom
    R.upper <- ( center + delta ) / denom
    data.frame(
      x = x,
      n = n,
      proportion = as.double(np_0) / as.double(n),
      lower = R.lower,
      upper = R.upper,
      conf.level = conf.level
    )
  }
wilson_ci <- binom.wilson(x = successes, n = trials, conf.level = 0.95)
print(wilson_ci)
```

```
##    x  n proportion     lower     upper conf.level
## 1 73 76  0.9605263 0.8902521 0.9864854       0.95
```

There is negligible difference between the `epitools` implementation and my own:

```
print(wilson_ci$lower - epi_wilson_ci$lower)
```

```
## [1] 1.110223e-16
```

```
print(wilson_ci$upper - epi_wilson_ci$upper)
```

```
## [1] 1.110223e-16
```

I have not contrasted the two implementations to see which suffers more from round-off error, but it seems clear that they are consistent with one another. So, having validated that the epitools implementation gives the result that I expect, I can use either one - basically, the choice is between loading another library *vs.* adding a custom function.

A frame of reference for reasonability of the confidence interval

Does the confidence interval seem reasonable?

I manually adjusted some "guesses"" to obtain a 2.5% area under probability density function (PDF) curve for each tail of the binomial distribution. Because, in actuality, this would require integration, the better choice is to use the cumulative density function (CDF), which is the integration of the PDF. For R, the `stats::pbinom` function implements CDF for the binomial distribution.

So, what proportion do I expect to be near the lower limit of the CI? I expect `low.guess` to be near, *i.e.*:

```
low.guess <- 0.90816
1 - pbinom(q = successes, size = trials, prob = low.guess)
```

```
## [1] 0.02501495
```

And, I expect `high.guess` to be near the upper limit of the CI:

```
high.guess <- 0.99178
1 - pbinom(q = successes, size = trials, prob = high.guess, lower.tail = FALSE)
```

```
## [1] 0.02503437
```

How does the actual proportion behave? I would expect the CDF for to be 50%, but it is only somewhat close:

```
1 - pbinom(q = successes, size = trials, prob = proportion)
```

```
## [1] 0.4186734
```

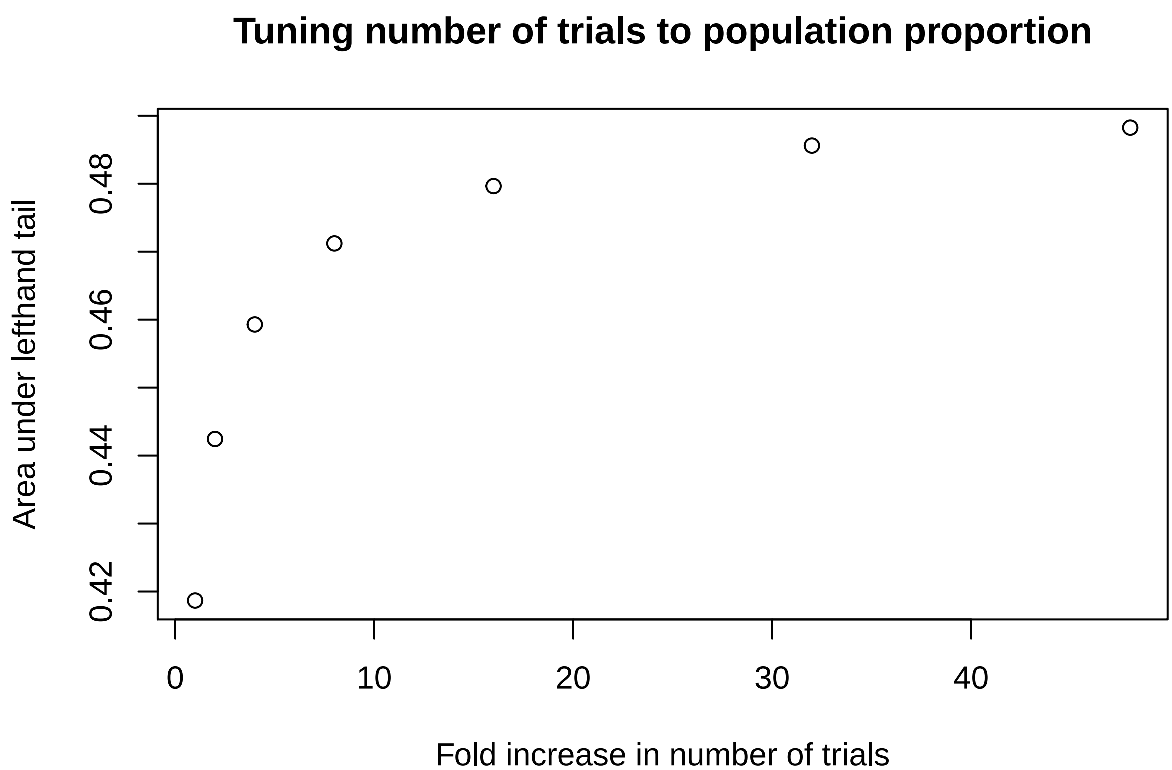
Increasing the number of successes and trials without changing the proportion shows that the small number of trials explains the deviation from my expectation.

```
1 - pbinom(q = 100 * successes, size = 100 * trials, prob = proportion)
```

```
## [1] 0.4918569
```

This begs the question: "If you are truly obsessed with having a 'nicely behaved' proportion, how much should you increase the number of trials?" Here is a quick plot to address that question:

```r
plot(
  x = x <- c(1,2,4,8,16,32,48),
  y = y <- sapply(
    X = x,
    FUN = function(x){
      1 - pbinom(q = x * successes, size = x * trials, prob = proportion)
    }
  ),
  xlab = "Fold increase in number of trials",
  ylab = "Area under lefthand tail",
  main = "Tuning number of trials to population proportion"
)
```



So, it looks like the point of diminishing returns would be around an eight-fold increase in the number of trials.

Conclusions

It turns out that my reasonability guesses are pretty close to the limits of the confidence interval:

```r
cat(
  sprintf("For %d successes over %d trials:"
  , wilson_ci$x
  , wilson_ci$n
  ),
  sprintf("Proportion: %f"
  , wilson_ci$proportion
  ),
```

```r
  sprintf("Wilson estimate of confidence interval for proportion: [%f,%f]"
  , wilson_ci$lower
  , wilson_ci$upper
  ),
  sprintf("My low and high guesses: %f,%f"
  , low.guess
  , high.guess
  ),
  sprintf("Differences from my guesses: %f, %f"
  , wilson_ci$lower - low.guess
  , wilson_ci$upper - high.guess
  ),
  sep = "\n"
)
```

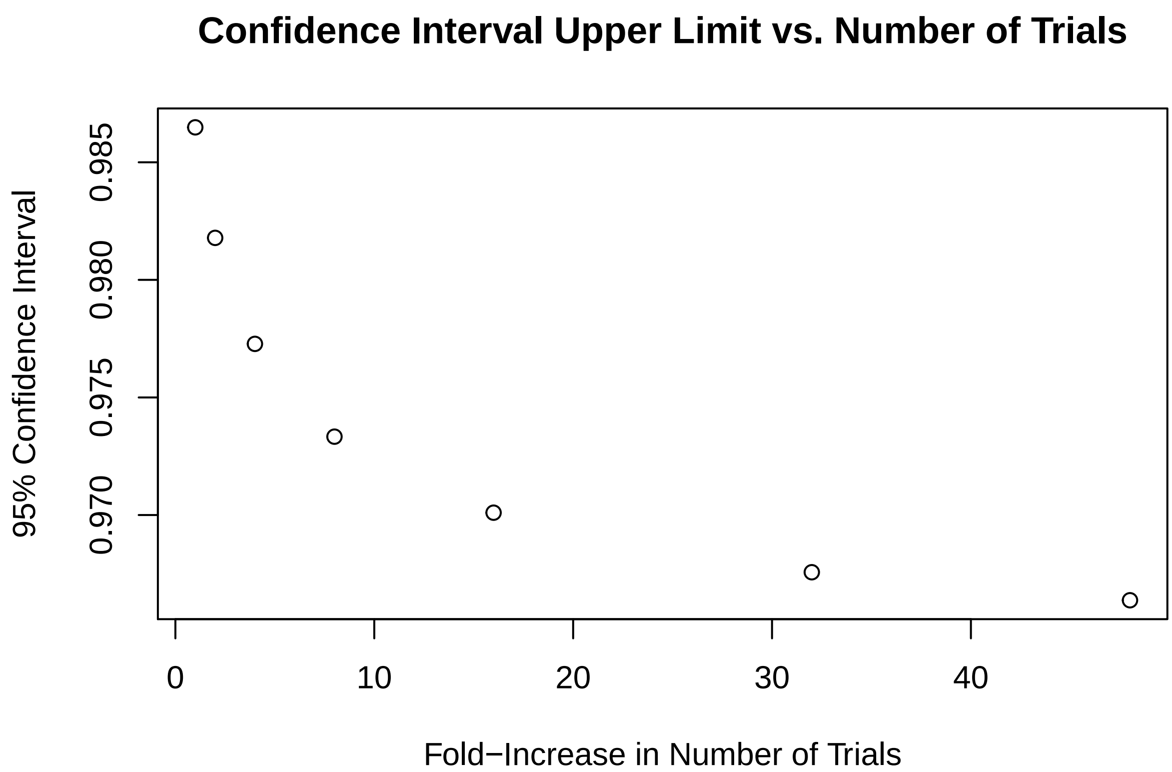
```
## For 73 successes over 76 trials:
## Proportion: 0.960526
## Wilson estimate of confidence interval for proportion: [0.890252,0.986485]
## My low and high guesses: 0.908160,0.991780
## Differences from my guesses: -0.017908, -0.005295
```

I think that the results differ in part because the guesses checks are based on different actual proportions and in part because the number of trials is low. I don't think that this casts doubt upon the confidence interval; rather, increasing the number of trials narrows the confidence interval:

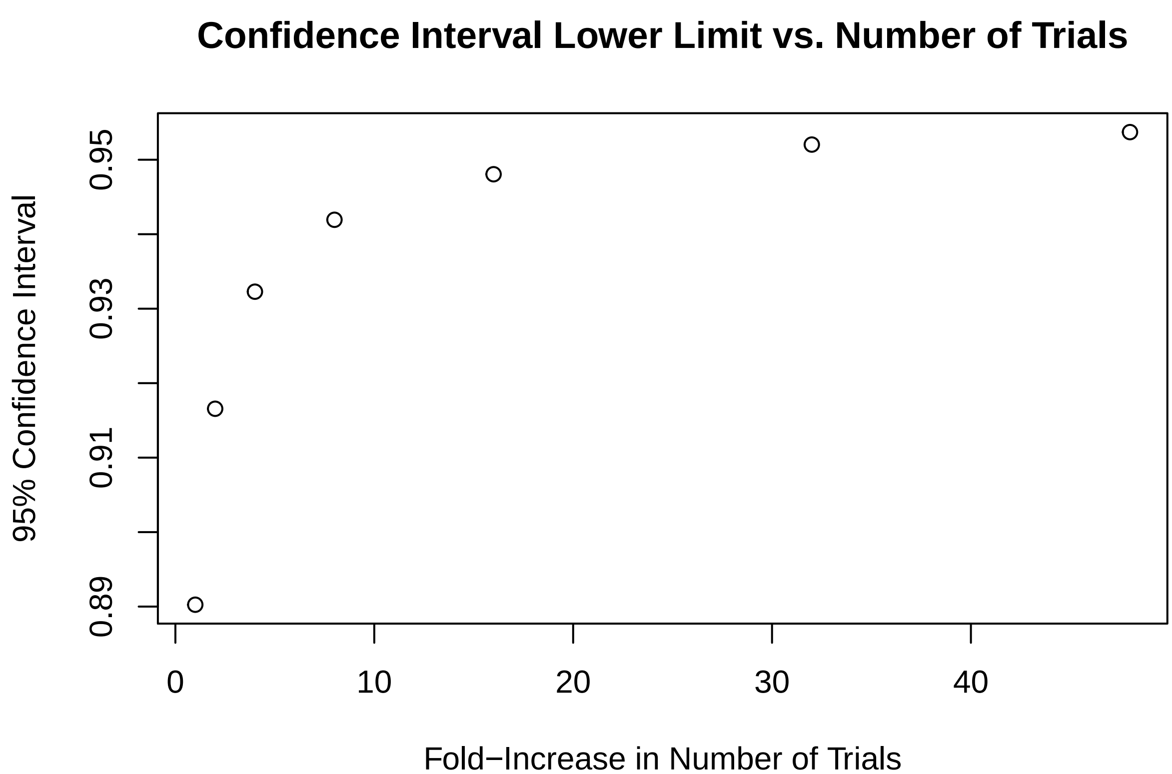
```r
print(binom.wilson(x = 8 * successes, n = 8 * trials, conf.level = 0.95))
```

```
##     x   n proportion     lower     upper conf.level
## 1 584 608  0.9605263 0.9419373 0.9733325       0.95
```

```r
x <- c(1,2,4,8,16,32,48)
y.lower <- sapply(
  X = x,
  FUN = function(x){
    binom.wilson(x = x * successes, n = x * trials, conf.level = 0.95)$lower
  }
)
y.upper <- sapply(
  X = x,
  FUN = function(x){
    binom.wilson(x = x * successes, n = x * trials, conf.level = 0.95)$upper
  }
)
lbl_upper <- "Confidence Interval Upper Limit vs. Number of Trials"
lbl_lower <- "Confidence Interval Lower Limit vs. Number of Trials"
lbl_x <- "Fold-Increase in Number of Trials"
lbl_y <- "95% Confidence Interval"
plot(x = x, y = y.upper, main = lbl_upper, xlab = lbl_x, ylab = lbl_y)
```




```
plot(x = x, y = y.lower, main = lbl_lower, xlab = lbl_x, ylab = lbl_y)
```



Again, an eight-fold increase in the number of trials appears to be the point of diminishing returns.

Derivation of Wilson's "Score Interval" formula

Let X be a binomally distributed random variable and let $q = 1 - p$:

$$X\tilde{}B(n,p), \qquad \overline{x} = np, \qquad {\sigma_x}^2 = np(1-p) = npq \tag{2}$$

Dividing by n:

$$p = \frac{\overline{x}}{n}, \qquad {\sigma_p}^2 = \frac{{\sigma_x}^2}{n^2} = \frac{pq}{n} \tag{3}$$

For a 95 percent confidence interval:

$$\lambda = 1.96, \qquad \text{confidence interval (score interval)} = [p - \lambda\sigma, p + \lambda\sigma] \tag{4}$$

For simplicity, define t:

$$t = \frac{\lambda^2}{n} \tag{5}$$

For p_0 (the observed value p) at either confidence limit:

$$(p_0 - p)^2 = \frac{\lambda^2 pq}{n} = p(1-p)t \tag{6}$$

Substituting for λ:

$$0 = {p_0}^2 - 2pp_0 + p^2 - pt + p^2 t \tag{7}$$

Distributing $1 + t$:

$$0 = (1+t)p^2 - 2(p_0 + t)p + {p_0}^2 \tag{8}$$

Dividing by 2:

$$0 = \frac{(1+t)}{2}p^2 - (p_0 + \frac{t}{2})p + \frac{1}{2}{p_0}^2 \tag{9}$$

Applying the quadratic formula:

$$p = \frac{p_0 + \frac{t}{2} \pm \sqrt{{p_0}^2 + p_0 t + \frac{t^2}{4} - ({p_0}^2 + {p_0}^2 t)}}{1+t} \tag{10}$$

Applying $0 = {p_0}^2 - {p_0}^2$:

$$p = \frac{p_0 + \frac{t}{2} \pm \sqrt{p_0 t + \frac{t^2}{4} - {p_0}^2 t}}{1+t} \tag{11}$$

Commuting $p_0 t - {p_0}^2 t$:

$$p = \frac{p_0 + \frac{t}{2} \pm \sqrt{p_0(1-p_0)t + \frac{t^2}{4}}}{1+t} \tag{12}$$

About this document

This document was produced from an RMarkdown file, the source of which should be at:

https://eschen42.github.io/propci/propci.Rmd

Revisions:

- January 23, First revision
- January 25, Added upper confidence limits plot

References

Agresti, Alan; Coull, Brent A. (1998). "Approximate is better than 'exact' for interval estimation of binomial proportions". The American Statistician. 52: 119-126. doi:10.2307/2685469.

Wilson, E. B. (1927). "Probable inference, the law of succession, and statistical inference". Journal of the American Statistical Association. 22: 209-212. doi:10.1080/01621459.1927.10502953.