

More simulations: Approximate binomial intervals and robustness

Patrick Breheny

October 15, 2014

In the previous lab, we saw the power of simulations to investigate the accuracy of theoretical results in mathematically intractable situations. We will continue using simulations today, investigating the accuracy of approximate binomial intervals and (time permitting), the robustness of the t -test and variance interval to departures from normality.

1 Simulation: Binomial CI coverage, Part 2

First, let's revisit our binomial CI coverage simulation, but add some additional "competitors" to see how their coverage compares to the Clopper-Pearson interval. Specifically, let's add the Wald interval, the score interval, and the Bayesian HPD interval. I have posted a function online, `bayes.binom`, that we can use to calculate HPD intervals for binomial data (as we discussed previously, there is no standard R function for this). The score interval is available via the R function `prop.test` (for inference concerning proportions). Lastly, there is no R function that returns the Wald interval, but it is trivial to write one. So, after we source the following:

```
> source("http://myweb.uiowa.edu/pbreheny/571/f14/labs/binom.bayes.R")
> prop.wald <- function(x, n, level=0.95) {
+   pi.hat <- x/n
+   SE <- sqrt(pi.hat*(1-pi.hat)/n)
+   z <- qnorm(c((1-level)/2, 1-(1-level)/2))
+   pi.hat + z*SE
+ }
> prop.test(31,39) ## Just to show how prop.test works
```

1-sample proportions test with continuity correction

```
data: 31 out of 39, null probability 0.5
X-squared = 12.41, df = 1, p-value = 0.000427
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.6306 0.9013
sample estimates:
      p
0.7949
```

We can now run our simulation loop as in the last lab, using the `binom.test`, `binom.bayes`, `prop.test`, and `prop.wald` functions to calculate our various intervals. Here, for the sake of organization, I'll store all the coverage results in a matrix (previously, a vector was sufficient).

```

> source("http://myweb.uiowa.edu/pbreheny/571/f14/labs/displayProgressBar.R")
> N <- 10000
> n <- 20
> pi <- c(0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.5, 0.68, 0.84, 0.92, 0.96, 0.98, 0.99)
> coverage <- matrix(NA, length(pi), 4, dimnames=list(pi, c("CP", "Bayes", "Score", "Wald")))
> for (j in 1:length(pi)) {
+   x <- rbinom(N, prob=pi[j], size=n)
+   covered <- matrix(NA, N, 4)
+   for (i in 1:N) {
+     ci <- binom.test(x[i], n)$conf
+     covered[i,1] <- (ci[1] < pi[j]) & (pi[j] < ci[2])
+     ci <- binom.bayes(x[i], n)$ci.hpd
+     covered[i,2] <- (ci[1] < pi[j]) & (pi[j] < ci[2])
+     ci <- prop.test(x[i], n, correct=FALSE)$conf
+     covered[i,3] <- (ci[1] < pi[j]) & (pi[j] < ci[2])
+     ci <- prop.wald(x[i], n)
+     covered[i,4] <- (ci[1] < pi[j]) & (pi[j] < ci[2])
+   }
+   coverage[j,] <- apply(covered, 2, mean)
+   displayProgressBar(j, length(pi))
+ }

```

Progress:

```

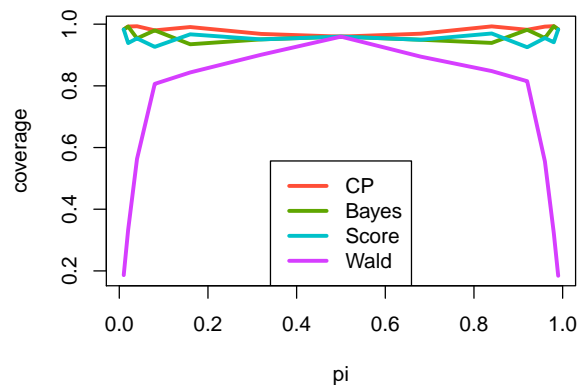
|          |
*****

```

```

> col <- hcl(seq(15, 375, len = 5), 150, 60)[1:4] ## Setting up some colors
> matplot(pi, coverage, type="l", lwd=3, lty=1, col=col)
> legend("bottom", legend=colnames(coverage), col=col, lwd=3)

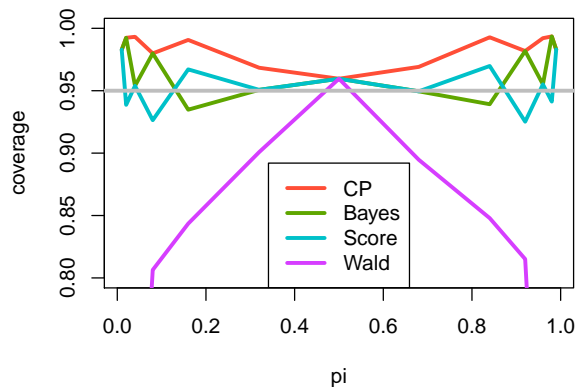
```



```

> matplot(pi, coverage, type="l", lwd=3, lty=1, col=col, ylim=c(0.8,1))
> legend("bottom", legend=colnames(coverage), col=col, lwd=3)
> abline(h=0.95, col="gray", lwd=3)

```



Note the use of `apply` here. `apply` is a very useful function that applies a function across all rows or columns of a matrix (or a larger array). We could write a for loop to calculate the average coverage for each method, but with `apply`, we can obtain the result we're interested in with just a single line of code. Also, the function `matplot` works a lot like `plot`, but allows you to plot several lines at once.

Some observations I would make:

- The Wald interval is simply unacceptable. Its coverage is nowhere even close to 95%. The other three intervals are at least reasonable here, never falling too far below 95%.
- The Clopper-Pearson interval is the most conservative, but is also the only interval that always provides at least 95% coverage.
- The Bayesian interval is not a confidence interval at all – it provides an interval for the middle 95% of the posterior distribution and is not concerned with coverage in the long-run frequency sense – but still has reasonable frequentist properties. This is often the case for Bayesian methods.

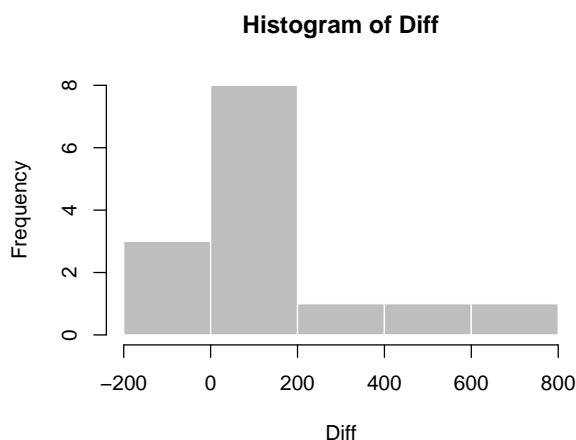
As a brief aside, the default behavior for `prop.test` is to carry out a minor “continuity correction”. This is a fairly simple adjustment, but we didn't really discuss it in class. To get the version of the interval we discussed in class, you can turn off the correction with `correct=FALSE`.

2 Simulation: NHANES lipid data

2.1 *t*-tests and confidence intervals

Carrying out one-sample *t*-tests and obtaining the corresponding confidence intervals is fairly straightforward in R using the function `t.test`. To illustrate with the cystic fibrosis study we discussed in class:

```
> cf <- read.delim("http://myweb.uiowa.edu/pbreheny/data/cystic-fibrosis.txt")
> Diff <- cf$Placebo - cf$Drug
> hist(Diff, col="gray", border="white")
```



```
> t.test(Diff)

One Sample t-test

data:  Diff
t = 2.288, df = 13, p-value = 0.03949
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 7.643 265.357
sample estimates:
mean of x
 136.5
```

As we saw in class, the p -value for testing the null hypothesis that amiloride has no effect on the average reduction in FVC is 0.04, and the confidence interval for the average difference is $[8, 265]$. As we see from the histogram, the distribution of differences doesn't look exactly normal, but with only 14 observations, it's difficult to tell.

2.2 How well do t intervals do when the data is not normally distributed?

As we said in class, the derivation of the t distribution (and, therefore, its resulting confidence intervals) is based on an assumption of normality. How well do these intervals work when the data is not normally distributed? Or to put it a different way, how robust are these results to departures from normality? As we did in the previous lab, let's investigate how well this works by resampling the NHANES triglyceride data:

```
> lipids <- read.delim("http://myweb.uiowa.edu/pbreheny/data/lipids.txt")
> pop <- lipids$TRG
```

Recall that the underlying distribution here is somewhat skewed to the right. Does this cause problems for the t intervals?

```
> N <- 10000          ## Number of simulations to run
> n <- 25             ## Sample size
> covered <- numeric(N)
```

```

> for (i in 1:N) {
+   sam <- sample(pop, n)
+   covered[i] <- t.test(sam, mu=mean(pop))$p.value > 0.05
+ }
> mean(covered)

[1] 0.9317

```

The coverage is slightly under 95%, but the t intervals aren't doing too bad here. What if we lower the sample size to 15? To 5?

Now let's look at the variance. As we discussed in class, there is a nice pivotal quantity for the normal distribution that allows us to obtain confidence intervals for the variance. How about its coverage in this situation? Unfortunately there is no standard R function for constructing this interval, but it's pretty easy to code:

```

> N <- 10000          ## Number of simulations to run
> n <- 25             ## Sample size
> covered <- numeric(N)
> for (i in 1:N) {
+   sam <- sample(pop, n)
+   ci <- var(sam)*(n-1)/qchisq(c(0.975,0.025), n-1)
+   covered[i] <- (ci[1] < var(pop)) & (var(pop) < ci[2])
+ }
> mean(covered)

[1] 0.8412

```

The coverage here is far worse than we had for the mean. As we said in class, the central limit theorem works for *any* distribution. For that reason, the t test is often a reasonable approximation even if the data departs somewhat from the normal distribution. There is no corresponding theorem for the variance: the pivotal CI works for the normal distribution and is not guaranteed to be accurate – even approximately – for other distributions.

This is a specific instance of an important general point in statistics: even though both of the above confidence interval methods are derived based on the assumption of normality, the two methods are not equally sensitive to that assumption. The t test is quite robust to departures from normality, while the variance interval is very sensitive to such departures. This is one reason why simulations are so important and useful in statistics: to examine the performance of various methods when assumptions are not met (this is typically difficult to address analytically).