STA 032 R Homework 5

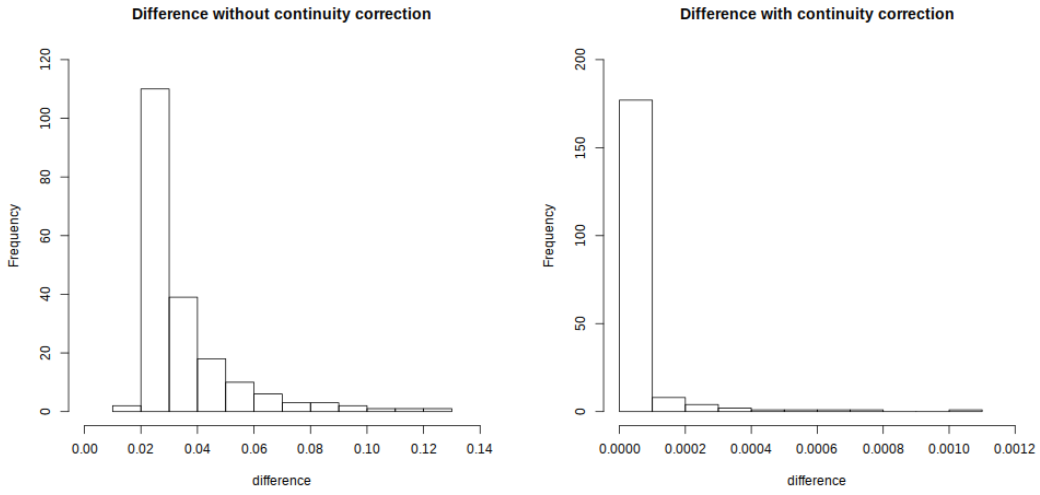
Hardy Jones
999397426
Professor Melcon
Winter 2015

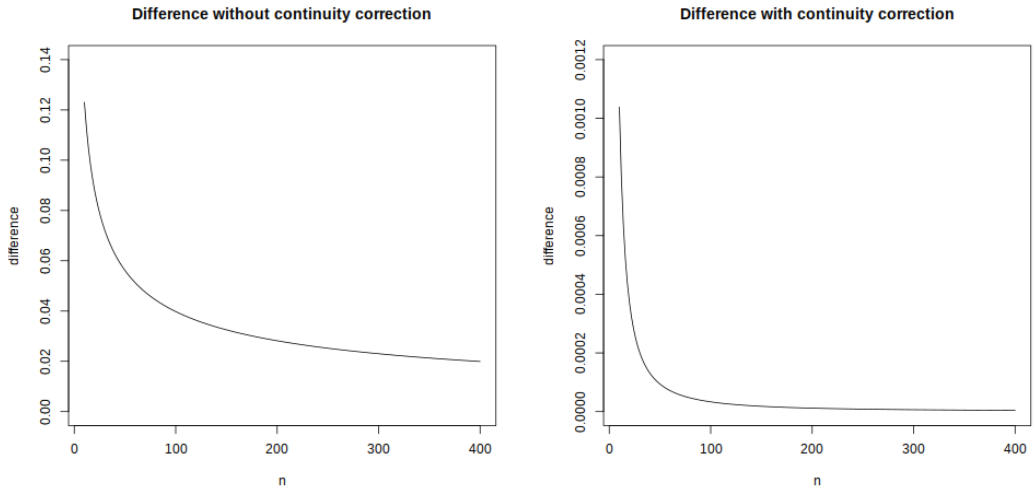
1. (a) 0.110
   (b) 0.0746
   (c) 0.0551
   (d) 2.96×10^{-10}

2. (a) 0.005 89
   (b) 0.008 45
   (c) 0.005 39
   (d) 2.96×10^{-10}

3. (a) The mean of the uncorrected column is 3.45×10^{-2}.
   The mean of the corrected column is 4.78×10^{-5}.

   (b)





(c)

(d) Yes, I think implementing the continuity correction is worthwhile if you are approximating a binomial probability with a normal distribution.

The additional complexity to compute the correction is a constant factor. So, when using the correction, the difference in computation time is not noticeable. However, the results are non-trivial.

In the above example, the correction provided a better approximation at all values of n than any uncorrected approximation at any n.

Appendix A R code

First create some common functionality.

```
adjust_discrete <- function(op, x) {
  # Adjust for the discrete values used in `pbinom`.
  switch( op
        , '<'  = list(x = x - 1, lower = TRUE)
        , '<=' = list(x = x,     lower = TRUE)
        , '>'  = list(x = x,     lower = FALSE)
        , '>=' = list(x = x - 1, lower = FALSE)
        )
}
```

Problem 1

```
source('./preamble.R')

prob_diff <- function(n, p, whichWay, x) {
  # Since `pbinom` and `pnorm` only work for P(X <= x) or P(X > x),
  # we have to use the properties of integers and inequalities.
  adj <- adjust_discrete(whichWay, x)
  true_prob <- pbinom(adj$x, n, p, adj$lower)
  approx_prob <- pnorm(adj$x, n * p, sqrt(n * p * (1 - p)), adj$lower)

  abs(true_prob - approx_prob)
}
```

Problem 2

```
source('./preamble.R')

prob_diff_corrected <- function(n, p, whichWay, x) {
  # Since `pbinom` and `pnorm` only work for P(X <= x) or P(X > x),
  # we have to use the properties of integers and inequalities.
  adj <- adjust_discrete(whichWay, x)
  true_prob <- pbinom(adj$x, n, p, adj$lower)
  # We can add 0.5 to the adjusted `x` and get the proper `x`.
  approx_prob <- pnorm(adj$x + 0.5, n * p, sqrt(n * p * (1 - p)), adj$lower)

  abs(true_prob - approx_prob)
}
```

Problem 3

```
source('./prob1.R')
source('./prob2.R')

prob_both <- function(ns, p, op) {
  no_cor   <- sapply(ns, function(n) prob_diff(n, p, op, n / 2))
  with_cor <- sapply(ns, function(n) prob_diff_corrected(n, p, op, n / 2))
  matrix(c(no_cor, with_cor), nrow = length(ns), ncol = 2)
}
```

(a)

```
all.n <- seq(10, 400, 2)
p <- 0.50
whichWay <- '<='
both <- prob_both(all.n, p, whichWay)
colMeans(both)
```

(b)

```
png('prob3b1.png')
hist(both[, 1], main = 'Difference without continuity correction',
  xlab = 'difference', xlim = c(0, 0.14), ylim = c(0, 120))
dev.off()

png('prob3b2.png')
hist(both[, 2], main = 'Difference with continuity correction',
  xlab = 'difference', xlim = c(0, 0.0012), ylim = c(0, 200))
dev.off()
```

(c)

```
png('prob3c1.png')
plot(all.n, both[, 1], type = 'n', main = 'Difference without continuity correction',
  xlab = 'n', ylab = 'difference', ylim = c(0, 0.14))
lines(all.n, both[, 1])
dev.off()

png('prob3c2.png')
plot(all.n, both[, 2], type = 'n', main = 'Difference with continuity correction',
  xlab = 'n', ylab = 'difference', ylim = c(0, 0.0012))
lines(all.n, both[, 2])
dev.off()
```