# Conflicting Outputs in R

Joshua Ingram

1/7/2020

## Motivations

We want to see if there are any conflicting outputs with the confidence intervals and p-values in the `prop.test` function in R. This is because the function uses Wald Confidence Interval, whereas a Score test is used to find the p-value.

## Methods and Criteria

To see if there are any conflicts, we will compare the confidence interval and p-value with $alpha = 0.05$. If the confidence interval includes a 0 and the p-value is less than 0.05, then there is a conflict. We will also check to see the cases in which the confidece interval does not include 0 but the p-value is greater than 0.05.

## Conflict Checking Function

Below is the code used to find conflicts amongst 26,502,500 different combinations of inputs for prop.test(). This checks for conflicts where the the corrections are enabled and disabled in the prop.test() function.

```
num.cores <- detectCores()

cluster <- makeCluster(num.cores - 2)

n1 <- seq(1,100)
n2 <- seq(1,100)
y1 <- seq(0,100)
y2 <- seq(0,100)

df <- expand.grid(y1, y2, n1, n2)
colnames(df) <- c("y1", "y2", "n1", "n2")

df <- subset(df, y1 <= n1 & y2 <= n2)
df <- df[which(df$y1 > 0 | df$y2 > 0),]
df <- df[which(df$y1 != df$n1 | df$y2 != df$n2),]

conflict.check <- function(vector.data){
  y1 <- vector.data[1]
  y2 <- vector.data[2]
  n1 <- vector.data[3]
```

```
  n2 <- vector.data[4]

  prop.info.correction <- prop.test(c(y1, y2), c(n1, n2), correct = TRUE)

  p.value.correction <- prop.info.correction$p.value
  conf.int.low.correction <- prop.info.correction$conf.int[1]
  conf.int.up.correction <- prop.info.correction$conf.int[2]

  if (conf.int.low.correction < 0 & 0 < conf.int.up.correction & p.value.correction < 0.05){
    conflict.correction <- TRUE
  } else if (conf.int.low.correction < 0 & conf.int.up.correction < 0 & p.value.correction > 0.05){
    conflict.correction <- TRUE
  } else if (conf.int.low.correction > 0 & conf.int.up.correction > 0 & p.value.correction > 0.05){
    conflict.correction <- TRUE
  } else {
    conflict.correction <- FALSE
  }

  prop.info.no <- prop.test(c(y1, y2), c(n1, n2), correct = FALSE)

  p.value.no <- prop.info.no$p.value
  conf.int.low.no <- prop.info.no$conf.int[1]
  conf.int.up.no <- prop.info.no$conf.int[2]

  if (conf.int.low.no < 0 & 0 < conf.int.up.no & p.value.no < 0.05){
    conflict.no <- TRUE
  } else if (conf.int.low.no < 0 & conf.int.up.no < 0 & p.value.no > 0.05){
    conflict.no <- TRUE
  } else if (conf.int.low.no > 0 & conf.int.up.no > 0 & p.value.no > 0.05){
    conflict.no <- TRUE
  } else {
    conflict.no <- FALSE
  }

  info.vector <- c(n1, n2, y1, y2, conflict.correction, conflict.no)

  return(info.vector)

}

object.export.names <- c("df", "conflict.check")
clusterExport(cluster, object.export.names)

output <- parApply(cl = cluster, X = df, 1, FUN = conflict.check)
output <- t(output)
conflict.output <- as.data.frame(output)
colnames(conflict.output) <- c("n1", "n2", "y1", "y2", "conflict.correction", "conflict.no.correction")

View(conflict.output)

stopCluster(cluster)
```

## Results

Out of the 26,502,500 combinations, there were 571,288 conflicts (2.1556%) when there was a correction and 575,266 conflicts (2.17061%) when there was no correction. There were also 133,240 (0.0503%) combinations that had conflicts with both a correction and not a correction. In total, there were 1,013,314 conflicts (3.823466%).