Figure 9 Replication Code

Jane Pan

3/20/2022

This demo intends to demonstrate overlapping scenarios between the Bayesian and frequentist settings assessing the assurance using credible interval based conditions. This demo focuses on the case when p_1 and p_2 are unknown.

```
library(bayesassurance)
```

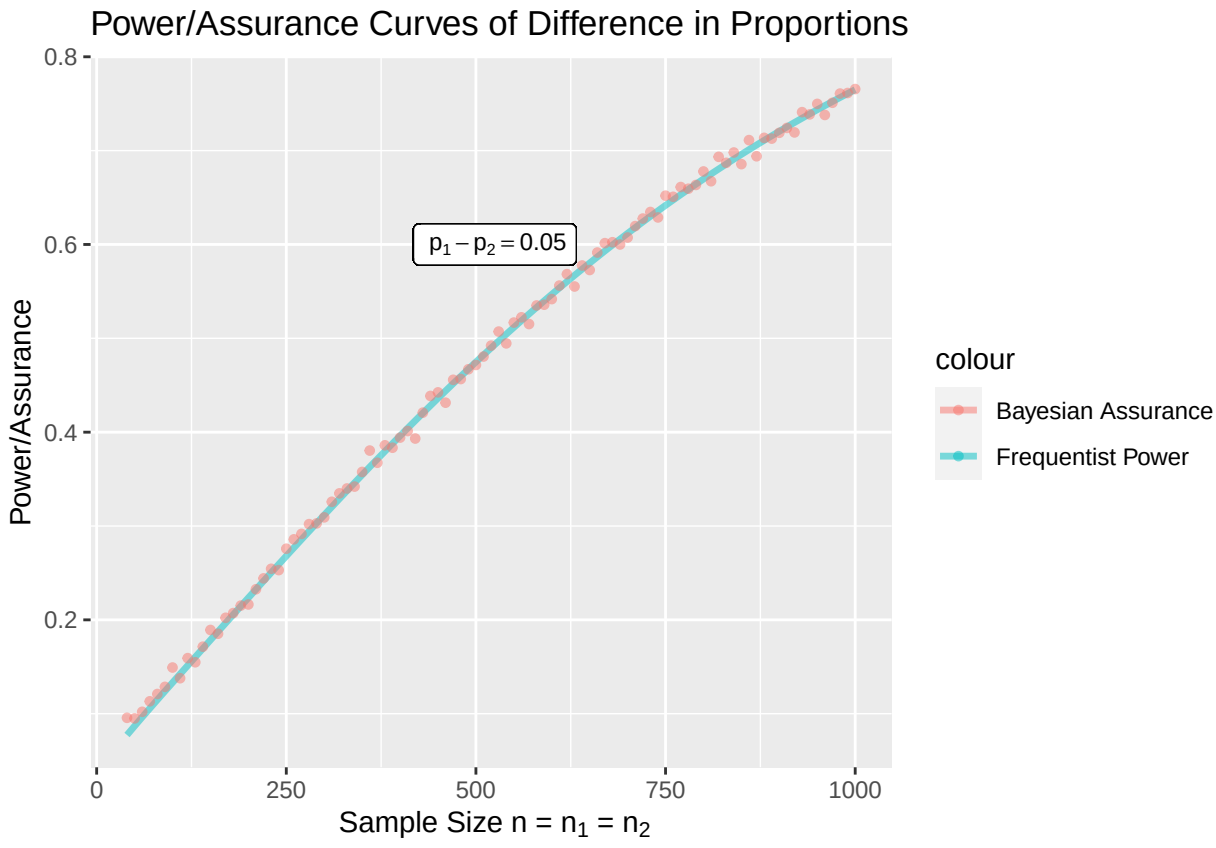
```
propdiffCI_classic <- function(n, p1, p2, alpha_1, beta_1, alpha_2, beta_2, sig_level){
  set.seed(1)
  if(is.null(p1) == TRUE & is.null(p2) == TRUE){
    p1 <- rbeta(n=1, alpha_1, beta_1)
    p2 <- rbeta(n=1, alpha_2, beta_2)
  }else if(is.null(p1) == TRUE & is.null(p2) == FALSE){
    p1 <- rbeta(n=1, alpha_1, beta_1)
  }else if(is.null(p1) == FALSE & is.null(p2) == TRUE){
    p2 <- rbeta(n=1, alpha_2, beta_2)
  }
  p <- p1 - p2

  power <- pnorm(sqrt(n / ((p1*(1-p1)+p2*(1-p2)) / (p)^2)) - qnorm(1-sig_level/2))
  return(power)
}
```

The following set of examples assign different values for p1 and p2 that adhere to different sets of critical differences while maintaining the same 0.5 throughout.

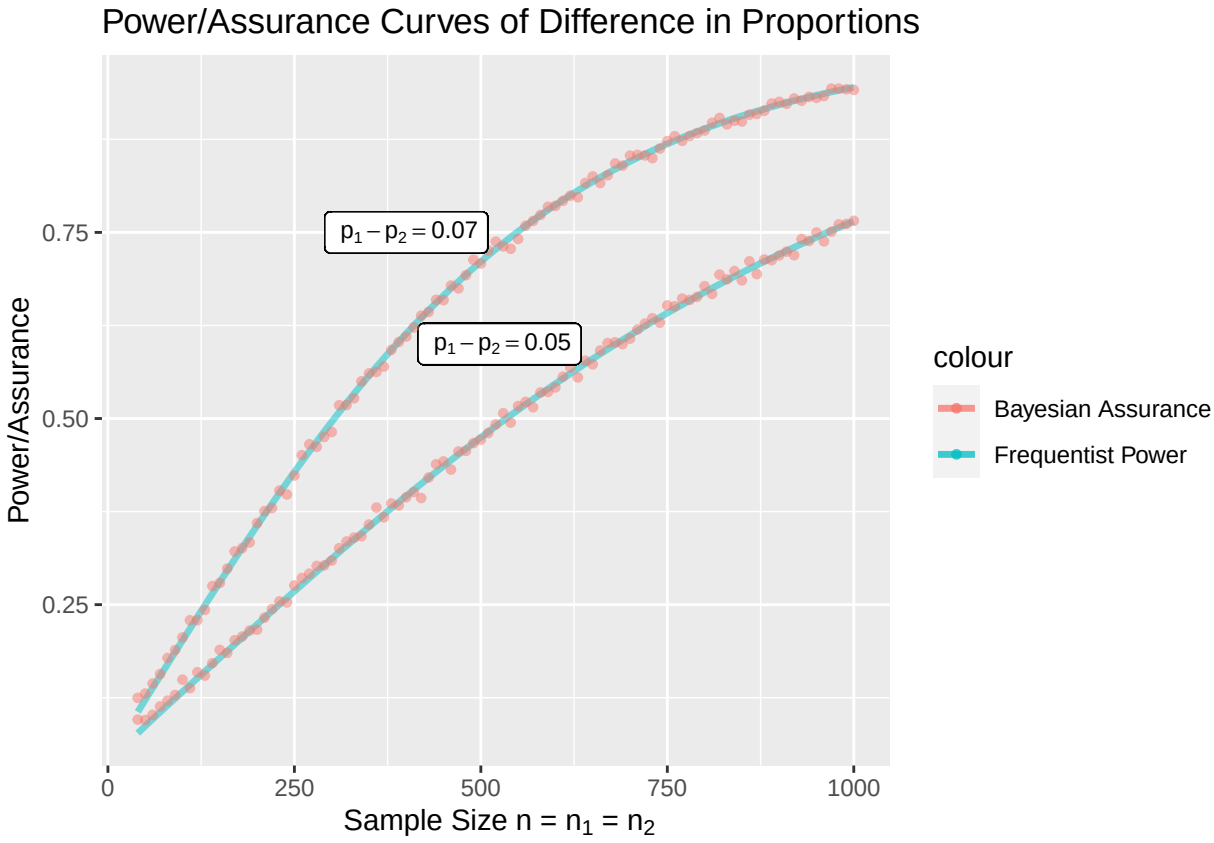
Case 1: p1 - p2 = 0.1

```
# Simulation
n <- seq(40, 1000, 10)
############################################################
# alpha1 = 0.5, beta1 = 0.5, alpha2 = 0.5, beta2 = 0.5 ##
############################################################

power_vals <- c()
for(j in n){
  temp_power <- propdiffCI_classic(j, 0.25, 0.2, NULL, NULL, NULL, NULL, 0.05)
  power_vals <- c(power_vals, temp_power)
}
df1 <- as.data.frame(cbind(n, power_vals))

set.seed(3)
assur_out <- bayes_sim_betabin(n1 = n, n2 = n, p1 = 0.25, p2 = 0.2, alpha_1 = 0.5,
                               beta_1 = 0.5, alpha_2 = 0.5, beta_2 = 0.5,
```

```
                                    sig_level = 0.05, alt = "two.sided")
df2 <- as.data.frame(cbind(n, assur_out$assurance_table$Assurance))
colnames(df2) <- c("n", "assur_vals")

library(ggplot2)
p1 <- ggplot(df1, alpha = 0.5, aes(x = n, y = power_vals,
                      color="Frequentist Power"))
p1 <- p1 + geom_line(alpha = 0.5, aes(x = n, y = power_vals, color="Frequentist Power"),
                      lwd = 1.2)

p2 <- p1 + geom_point(data = df2, alpha = 0.5, aes(x = n, y = assur_vals,
      color = "Bayesian Assurance"), lwd = 1.2) +
      ylab("Power/Assurance") + xlab(~ paste("Sample Size n = ", "n"[1], " = ", "n"[2])) +
      ggtitle("Power/Assurance Curves of Difference in Proportions")

p2 <- p2 + geom_label(aes(525, 0.6, label = "~p[1]-p[2] == 0.05"), parse = TRUE,
                       color = "black", size = 3)

p2
```



Case 2: p1 - p2 = 0.07



Case 3: p1 - p2 = 0.05