Figure 10 Replication Code

Jane Pan

3/20/2022

The following demo is meant to demonstrate overlapping scenarios between the Bayesian and frequentist settings for assessing the assurance using credible interval based conditions. This demo looks at the case when p_1 and p_2 are unknown. In this setting, we recall
the normal distribution can be used to approximate binomial distributions for large sample sizes given that the Beta distribution is approximately normal when its parameters α and β are set to be equal and large. Hence, we manually assign such values to illustrate these parallels. Running the follow segments of code will reproduce Figure 10 in the paper.

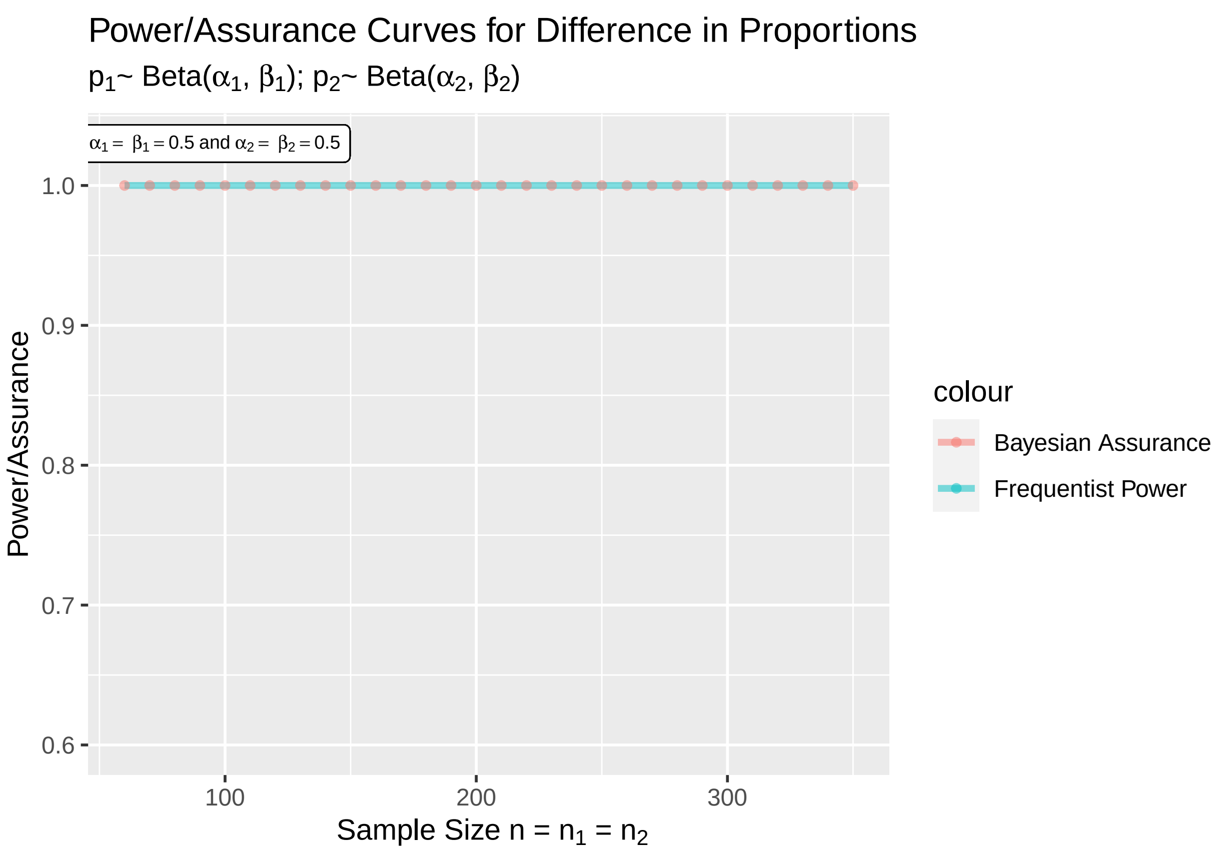
```
library(bayesassurance)
```

```
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)
}
```

Case 1: alpha1 = 0.5, beta1 = 0.5, alpha2 = 0.5, beta2 = 0.5

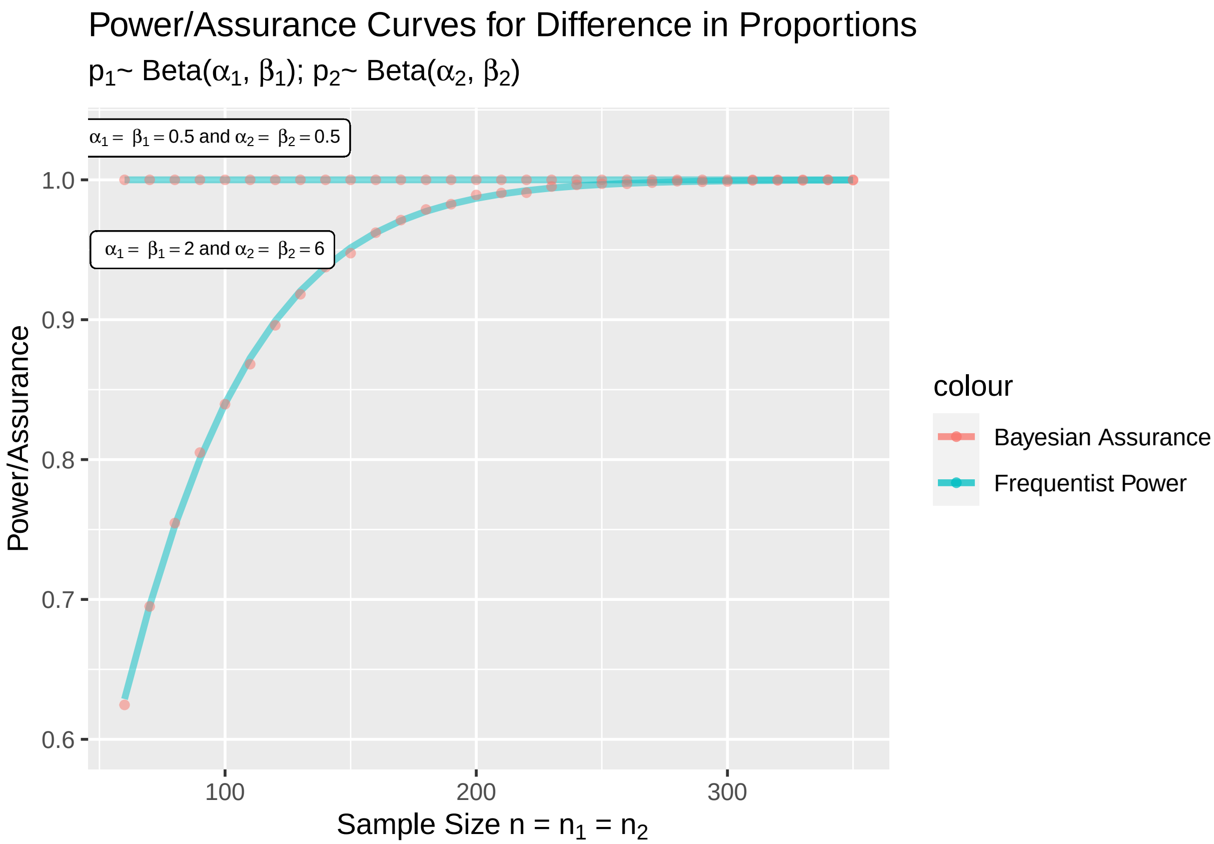
```
# Simulation
n <- seq(60, 350, 10)
########################################################
# Noninformative Beta Priors
# alpha1 = 0.5, beta1 = 0.5, alpha2 = 0.5, beta2 = 0.5 #
########################################################
power_vals <- c()
for(j in n){
  temp_power <- propdiffCI_classic(j, p1 = NULL, p2 = NULL, alpha_1 = 0.5, beta_1 = 0.5,
                                   alpha_2 = 0.5, beta_2 = 0.5, 0.05)
  power_vals <- c(power_vals, temp_power)
}
df1 <- as.data.frame(cbind(n, power_vals))
```

```r
set.seed(3)
assur_out <- bayes_sim_betabin(n1 = n, n2 = n, p1 = NULL, p2 = NULL, 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 for Difference in Proportions",
      subtitle = expression(paste(p[1], "~ Beta(", alpha[1], ", ", beta[1], "); ",
      p[2], "~ Beta(", alpha[2], ", ", beta[2], ")")))

p2 <- p2 + geom_label(aes(95, 1.03, label="~alpha[1] == ~beta[1] ==
      0.5~and~alpha[2] == ~beta[2] == 0.5"), parse=TRUE, color = "black",
      size = 2.5) + ylim(0.6, 1.03) + xlim(60, 350)

p2
```



Case 2: alpha1 = 2, beta1 = 2, alpha2 = 6, beta2 = 6

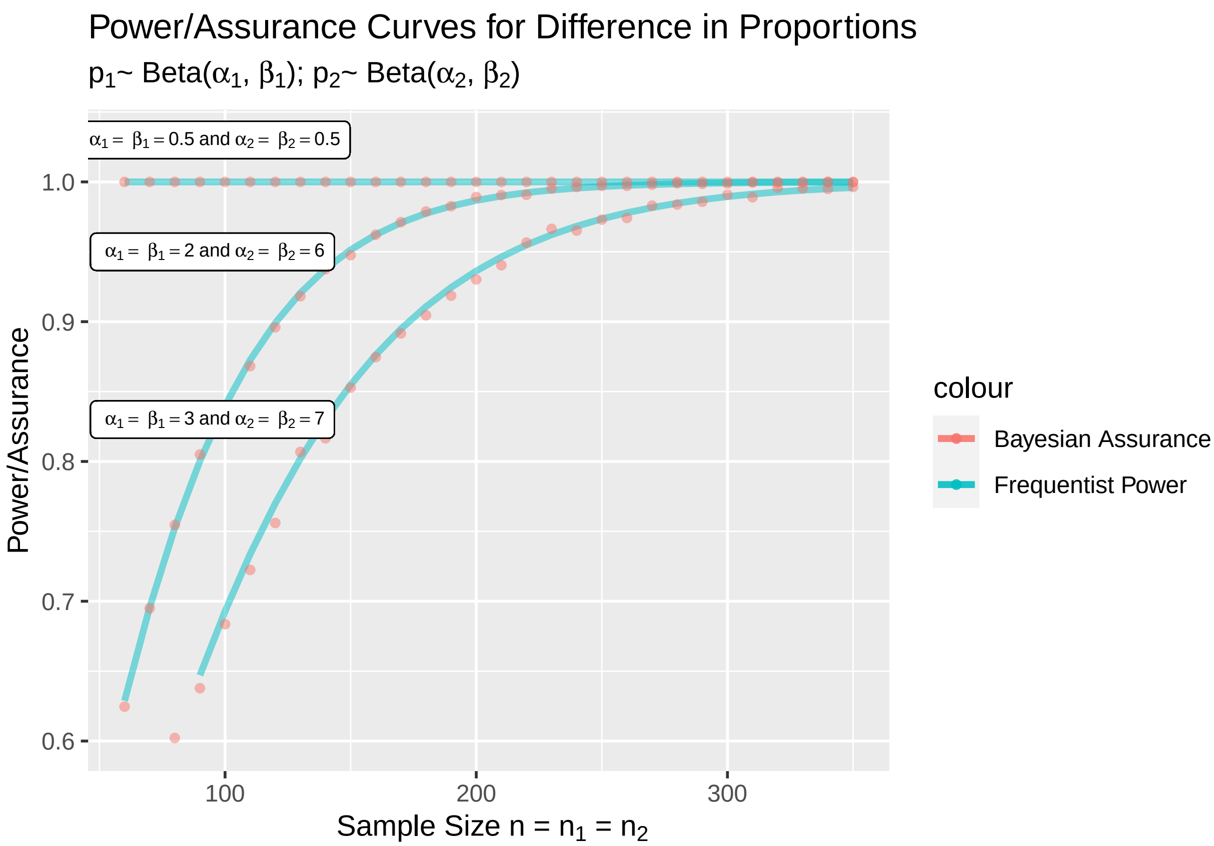
```
################################################
# alpha1 = 2, beta1 = 2, alpha2 = 6, beta2 = 6 #
################################################
power_vals <- c()
for(j in n){
  temp_power <- propdiffCI_classic(j, p1 = NULL, p2 = NULL, 2, 2, 6, 6, 0.05)
  power_vals <- c(power_vals, temp_power)
}
df1 <- as.data.frame(cbind(n, power_vals))


set.seed(3)
assur_out <- bayes_sim_betabin(n1 = n, n2 = n, p1 = NULL, p2 = NULL, alpha_1 = 2,
                                beta_1 = 2, alpha_2 = 6, beta_2 = 6, sig_level = 0.05,
                                alt = "two.sided")

df2 <- as.data.frame(cbind(n, assur_out$assurance_table$Assurance))
colnames(df2) <- c("n", "assur_vals")

p3 <- p2 + geom_line(data = df1, alpha = 0.5, aes(x = n, y = power_vals,
      color="Frequentist Power"), lwd = 1.2)

p4 <- p3 + geom_point(data = df2, alpha = 0.5, aes(x = n, y = assur_vals,
      color="Bayesian Assurance"),lwd=1.2)
p4 <- p4 + geom_label(aes(95, 0.95, label="~alpha[1] == ~beta[1] ==
      2~and~alpha[2] == ~beta[2] == 6"), parse=TRUE,color = "black", size = 2.5)
p4
```



Case 3: alpha1 = 3, beta1 = 3, alpha2 = 7, beta2 = 7

```
#################################################
# alpha1 = 3, beta1 = 3, alpha2 = 7, beta2 = 7 #
#################################################
power_vals <- c()
for(j in n){
  temp_power <- propdiffCI_classic(j, p1 = NULL, p2 = NULL, 3, 3, 7, 7, 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 = NULL, p2 = NULL, alpha_1 = 3,
                               beta_1 = 3, alpha_2 = 7, beta_2 = 7, sig_level = 0.05,
                               alt = "two.sided")

df2 <- as.data.frame(cbind(n, assur_out$assurance_table$Assurance))
colnames(df2) <- c("n", "assur_vals")


p5 <- p4 + geom_line(data = df1, alpha = 0.5, aes(x = n, y = power_vals,
      color="Frequentist Power"), lwd = 1.2)

p6 <- p5 + geom_point(data = df2, alpha = 0.5, aes(x = n, y = assur_vals,
      color="Bayesian Assurance"),lwd=1.2)
```

```
p6 <- p6 + geom_label(aes(95, 0.83,
      label="~alpha[1] == ~beta[1] == 3~and~alpha[2] == ~beta[2] == 7"),
      parse=TRUE, color = "black", size = 2.5)
p6
```



Case 4: alpha1 = 4, beta1 = 4, alpha2 = 8, beta2 = 8

```
###############################################
# alpha1 = 4, beta1 = 4, alpha2 = 8, beta2 = 8 #
###############################################
power_vals <- c()
for(j in n){
  temp_power <- propdiffCI_classic(j, p1 = NULL, p2 = NULL, 4, 4, 8, 8, 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 = NULL, p2 = NULL, alpha_1 = 4,
                               beta_1 = 4, alpha_2 = 8, beta_2 = 8, sig_level = 0.05,
                               alt = "two.sided")

df2 <- as.data.frame(cbind(n, assur_out$assurance_table$Assurance))
colnames(df2) <- c("n", "assur_vals")

p7 <- p6 + geom_line(data = df1, alpha = 0.5, aes(x = n, y = power_vals,
```

```
      color="Frequentist Power"), lwd = 1.2)

p8 <- p7 + geom_point(data = df2, alpha = 0.5, aes(x = n, y = assur_vals,
      color="Bayesian Assurance"),lwd=1.2)
p8 <- p8 + geom_label(
  aes(95, 0.72, label="~alpha[1] == ~beta[1] == 4~and~alpha[2] == ~beta[2] == 8"),
  parse=TRUE,color = "black", size = 2.5
)
p8
```