Figure 7 Replication Code

Jane Pan

3/20/2022

The following demo goes over different cases of the Bayesian assurance function using Adcock's precision error condition.

The function below returns the left-hand-side expression of the precision-based solution for determining sample size discussed in Adcock (1997), with 1 - alpha isolated on the right hand side.

```
library(bayesassurance)
```

```
adcock <- function(n, d, sig_sq){
  sigma <- sqrt(sig_sq)
  lhs <- 2 * pnorm(d / (sigma / sqrt(n))) - 1

  # returns left hand side expression, right hand side is 1 - alpha
  return(lhs)
}
```

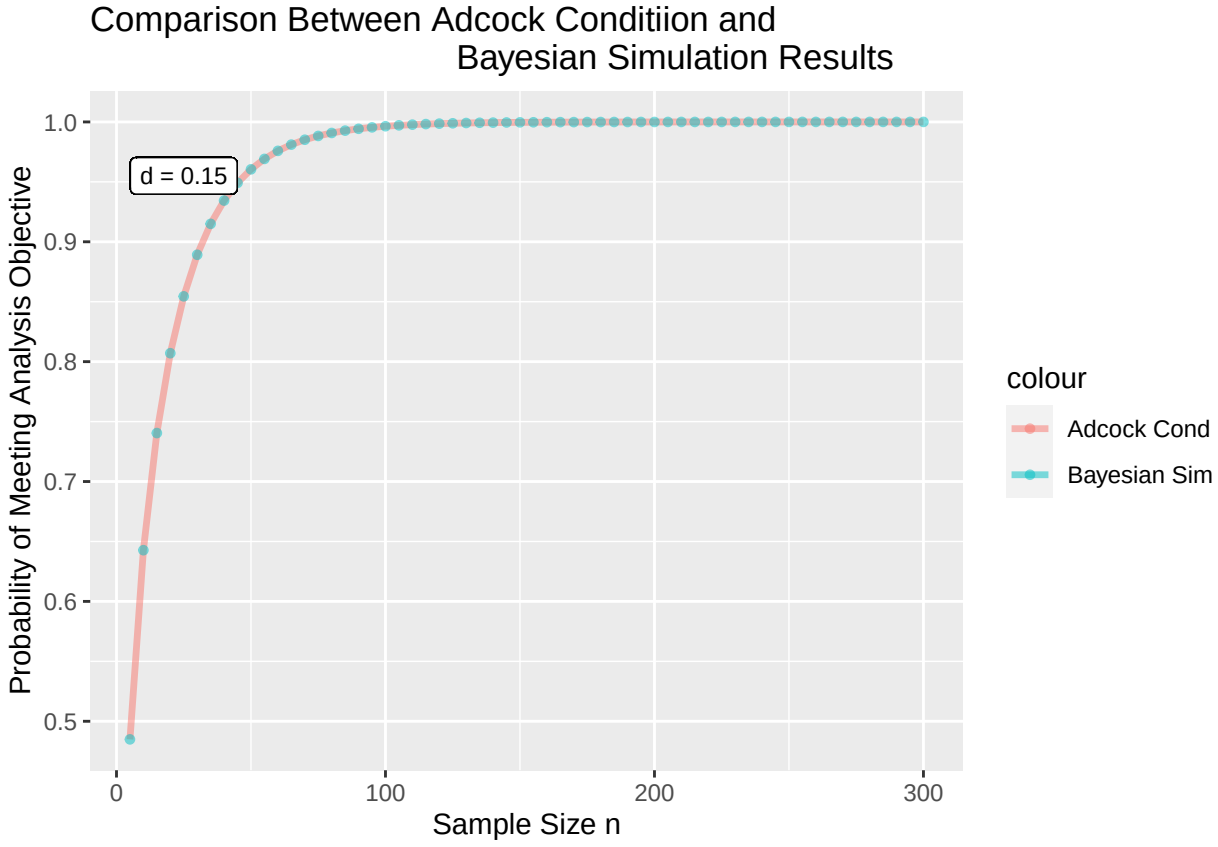
The function below performs the same steps as bayes_adcock() but returns only the left-hand-side expression of the assurance formula rather than computing the assurance. Here, 1 - alpha is also isolated on the right hand side.

```
bayes_adcock_lhs <- function(n, d, mu_beta_a, mu_beta_d, n_a, n_d, sig_sq){

  count <- 0
  maxiter <- 1000
  lhs <- c()

  # Design Stage
  for(i in 1:maxiter){
    var_d <- sig_sq * ((n_d + n) / (n * n_d))
    xbar <- rnorm(n=1, mean = mu_beta_d, sd = sqrt(var_d))

    lambda <- ((n_a * mu_beta_a) + (n * xbar)) / (n_a + n)

    phi_1 <- (sqrt(n_a + n) / sqrt(sig_sq)) * (xbar + d - lambda)
    phi_2 <- (sqrt(n_a + n) / sqrt(sig_sq)) * (xbar - d - lambda)

    lhs <- c(lhs, pnorm(phi_1) - pnorm(phi_2))

  }

  lhs_avg <- mean(lhs)
  return(lhs_avg)
}
```

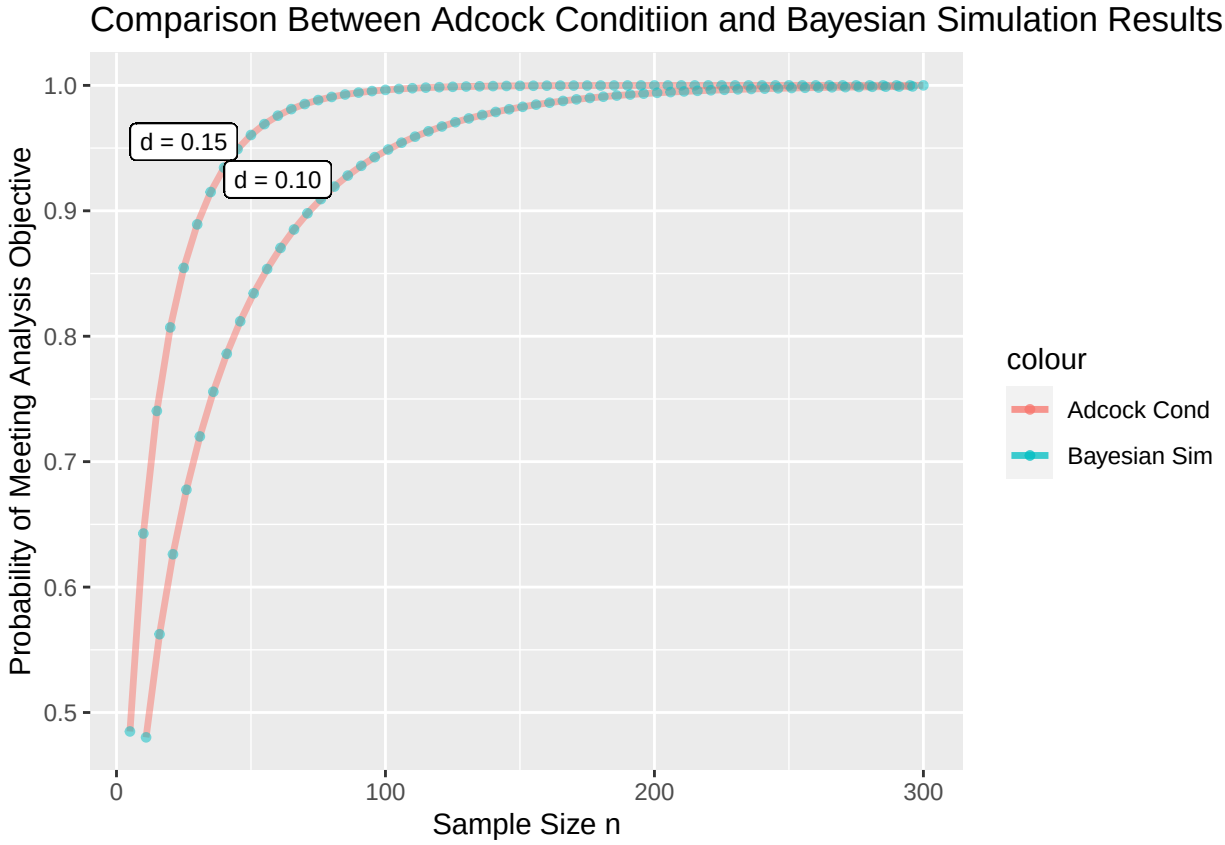
The next several blocks of code implement the above two functions for various assignments of precision,

d. The results should overlap one another perfectly when n_a = 0, as shown in the figures. Figures are produced using ggplot2.

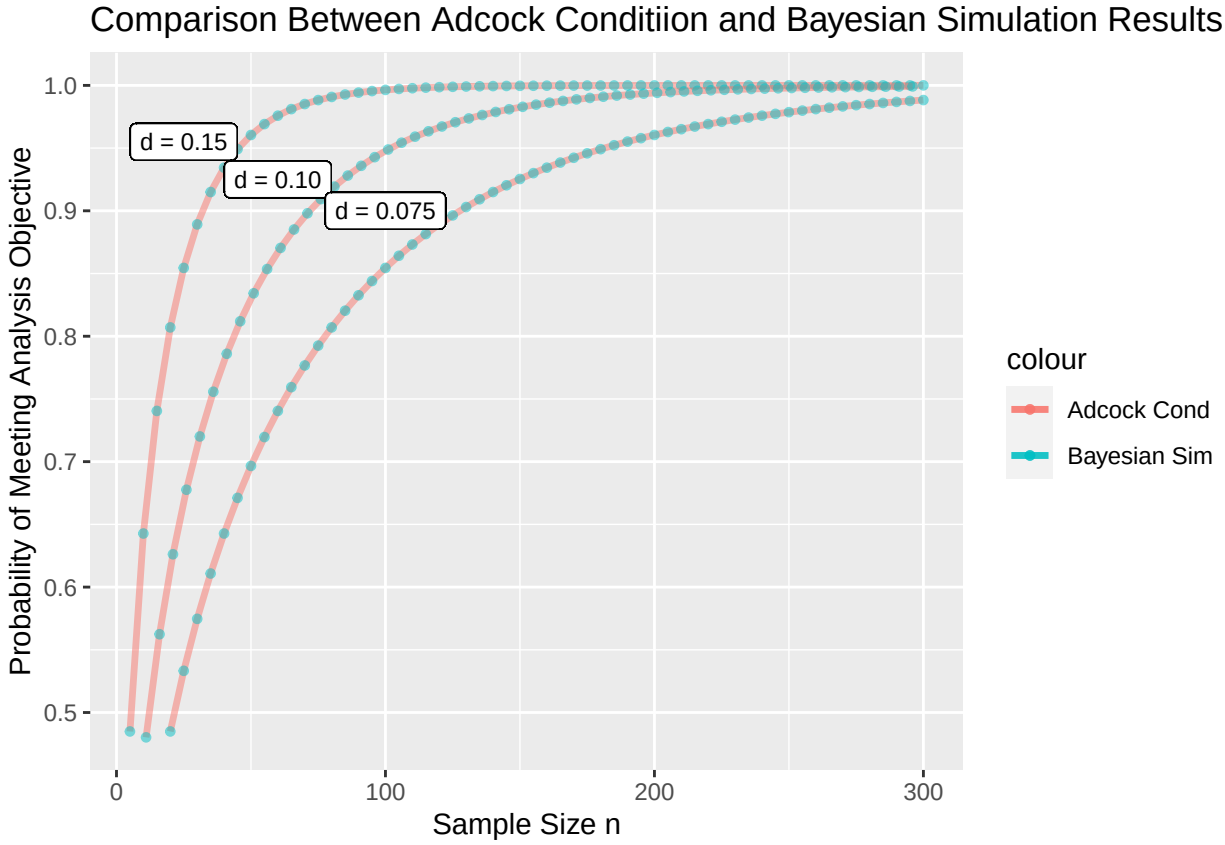
Case 1: d = 0.15



Case 2: d = 0.10


Comparison Between Adcock Conditiion and Bayesian Simulation Results
Probability of Meeting Analysis Objective
Sample Size n
d = 0.15
d = 0.10
1.0
0.9
0.8
0.7
0.6
0.5
0
100
200
300
colour
Adcock Cond
Bayesian Sim

Case 3: $\mathbf{d = 0.075}$



Case 4: $\mathbf{d = 0.30}$