# Assignment 1

*2017-04-10*

## Question 1: Inference for the binomial parameter:

(a) Develop an R function to calculate HPD intervals for data $(x, n)$, given a *beta(a,b)* prior. I actually developed two functions. My initial attempts were based around the idea that you could optimise for some height 'h'. However, when it came to actually implementing this method in future questions I ran into trouble. So, I instead opted to try and optimise the length of the interval between the upper and lower limits, which produced results that are useable in parts (b)-(e). Here is my first attempt.

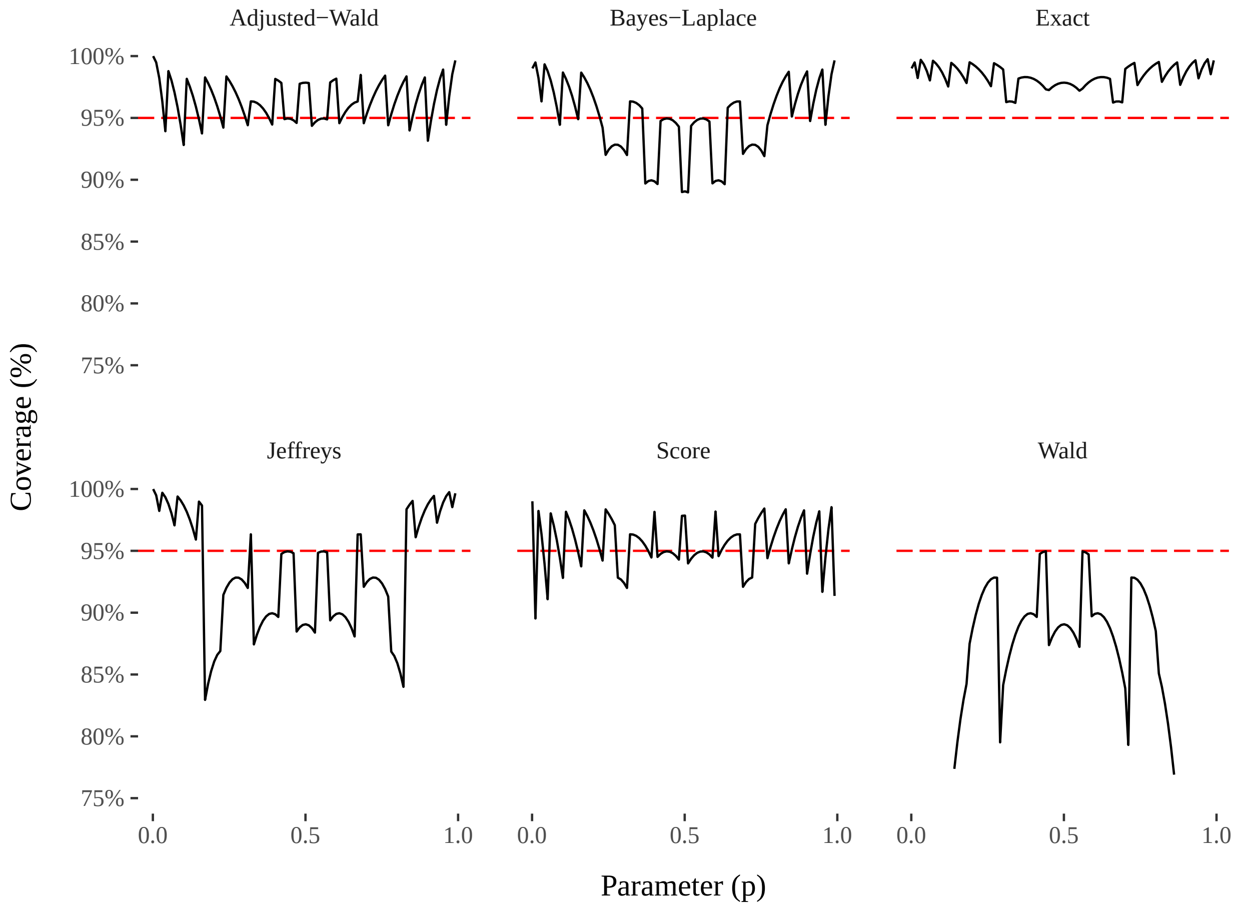
```r
solve.HPD.beta = function(shape1, shape2, credint = 0.95, one.sided = FALSE,...){
  if(shape1 <= 1 | one.sided == TRUE){
    lt = 0
    ut = qbeta(credint, shape1, shape2)
    coverage = credint
    results = data_frame("Lower"    = lt,
                         "Upper"    = ut,
                         "Coverage" = coverage,
                         "Height"   = ut)
    return(results)
  }
  if(shape1 > n){
    lt = qbeta(1-credint, shape1, shape2)
    ut = 1
    coverage = credint
    results = data_frame("Lower"    = lt,
                         "Upper"    = ut,
                         "Coverage" = coverage,
                         "Height"   = lt)
    return(results)
  } else {
    hpdfunc <- function(h, shape1, shape2){
      mode = (shape1 - 1)/(shape1 + shape2 - 2)
      lt = uniroot(f=function(x){ dbeta(x,shape1, shape2) - h},
                   lower=0, upper=mode)$root
      ut = uniroot(f=function(x){ dbeta(x,shape1, shape2) - h},
                   lower=mode, upper=1)$root
      coverage = pbeta(ut, shape1, shape2) - pbeta(lt, shape1, shape2)

      hpdval = abs(credint-coverage)

      return(hpdval)
    }
    upper = max(dbeta(seq(0,1, by = 0.001), shape1, shape2))

    h = optimize(hpdfunc,
                 interval = seq(0,upper,by = 0.001),
                 lower = 0,
                 tol = .Machine$double.eps,
                 shape1,
                 shape2)
```

```r
    h <- h$minimum
    mode = (shape1 - 1)/(shape1 + shape2 - 2)
    lt = uniroot(f=function(x){ dbeta(x,shape1, shape2) - h},
                 lower=0, upper=mode)$root
    ut = uniroot(f=function(x){ dbeta(x,shape1, shape2) - h},
                 lower=mode, upper=1)$root
    coverage = pbeta(ut, shape1, shape2) - pbeta(lt, shape1, shape2)
    results = data_frame("Lower"    = lt,
                         "Upper"    = ut,
                         "Coverage" = coverage,
                         "Height"   = h)
    return(results)
  }}

y=1; n=100; a=1; b=1; p=0.95
solve.HPD.beta(shape1 = y + a, shape2 = n - y + b)
```

```
## # A tibble: 1 × 4
##          Lower      Upper Coverage   Height
##          <dbl>      <dbl>    <dbl>    <dbl>
## 1 0.0004144375 0.04630291     0.95 4.286167
```

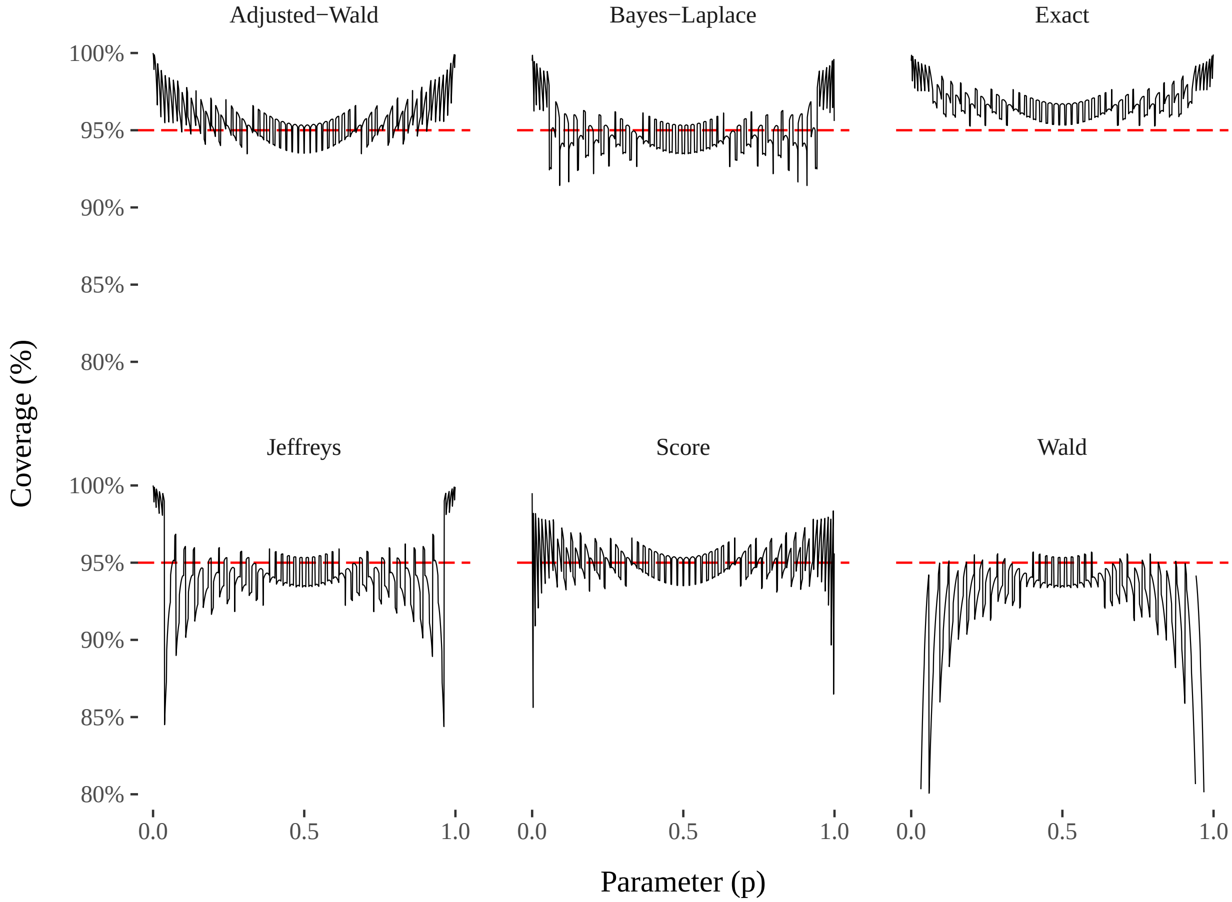
(b) Reproduce Agresti & Coull's (1998) Figure 4 ($n = 10$), and replicate for the Score and Bayes-Laplace & Jeffreys HPD intervals The code for generating the intervals in questions (b) - (e) is given below. The values for the sawtooth graphs are obtained by using vapply and a vector of values for the parameter between 0 and 1: e.g for the Wald we use "vapply(p,waldcover,0)".

```r
n <- 10 # successes
a <- 0.05 # desired alpha
p <- seq(0.001,0.999,1/100) # vector of possible param. values.
z <- abs(qnorm(.5*a,0,1)) # z-score to use for some of the intervals.
source("Question1/1bData.R") # contains the code for generating the req.values: see appendix.
source("Question1/1bChart.R") # contains code for plotting the values.
q1bchart # calls the chart.
```




(c) Repeat (b) for $n = 50$.

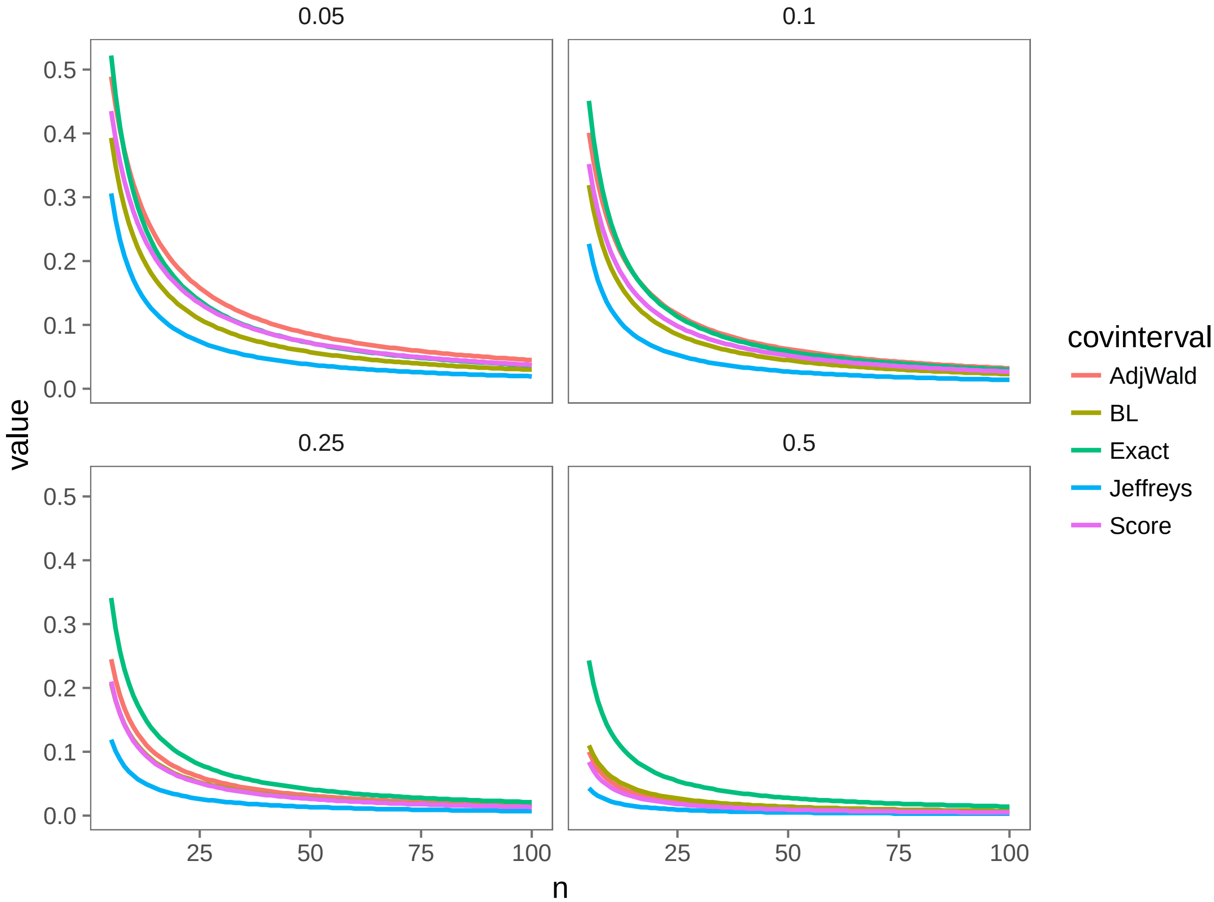
```
n <- 50
a <- 0.05
p <- seq(0.0001,0.9999,1/1000)
z <- abs(qnorm(.5*a,0,1))
source("Question1/1cData.R")
```




(d) Compare the minimum coverage of the six graphs at (c)

| covinterval | mean(dens) | median(dens) | min(dens) |
|---|---|---|---|
| Adjusted-Wald | 0.9579977 | 0.9556593 | 0.9346807 |
| Bayes-Laplace | 0.9500028 | 0.9468787 | 0.9141711 |
| Exact | 0.9693205 | 0.9677614 | 0.9526640 |
| Jeffreys | 0.9418993 | 0.9408915 | 0.8438360 |
| Score | 0.9518805 | 0.9530135 | 0.8562090 |
| Wald | 0.9005129 | 0.9348509 | 0.0049878 |

(e) The adjusted Wald interval appears to perform well with respect to frequentist coverage, if close to nominal combined with reasonable minimum coverage is aimed for. From a Bayesian point of view, performance of individual intervals is just as, if not more, important. Given $x = 0$, compare the adjusted Wald interval with the exact & Score intervals (all two-sided), and with the Bayes-Laplace & Jeffreys HPD intervals, for a range of values of $n$ and $\alpha$, comment on its limitations, and give an appropriate graphical illustration.


0.05
0.1
0.25
0.5
value
n
0.0
0.1
0.2
0.3
0.4
0.5
25
50
75
100
covinterval
AdjWald
BL
Exact
Jeffreys
Score

## Question 2: Inference for the Cauchy parameter:

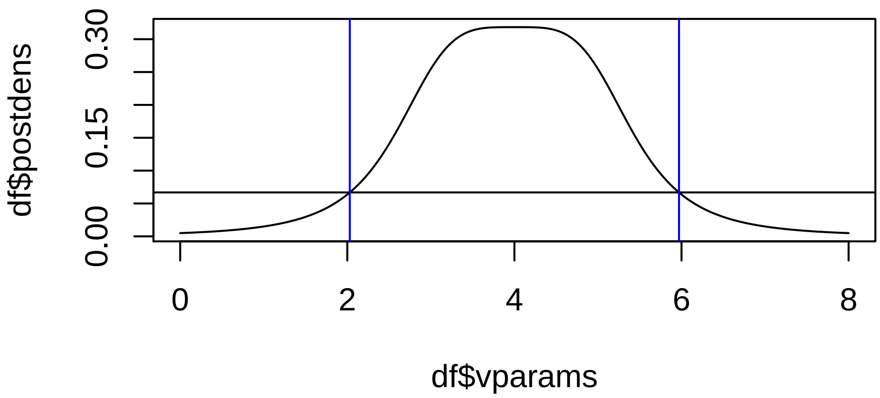
(a) Develop an R function to find percentiles of a (general) Cauchy posterior as discussed by Jaynes (1976, Example 6) and Box & Tiao (1973, p.64), to be used for the examples below.

```
source("Question2/2a.R")
CauchyPercentage <- function(x, # a vector of samples
                             p, # a vector of possible parameters
                             y = NULL # a value to test Pr[p < y]
) {
  cauchydens <- function(x,p){

    cauchydist <- function(x,p) {
      H = vector(mode = "numeric",length = length(x))
      for(i in 1:length(x)){
        H[i] = (1+(x[i] - p)^2)^(-1)
      }
      return(prod(H))
    }
    dens = vapply(p,cauchydist,min(p), x = x)
    c = (integrate(dens, -Inf, Inf)$value)^(-1)
    data.frame(vparams = p, postdens = c*dens(p))
  }

  df = cauchydens(x,p)

  yind = ifelse(length(which(df$vparams == y))==1,
                which(df$vparams == y),
                max(which(df$vparams < y)))

  plot(df$vparams, df$postdens, type = 'l')
  abline(v = df$vparams[yind])
  abline(h = df[yind,"postdens"])

  cumdist = c*integrate(dens,-Inf,y)$value

  return((c(yind,df[yind,"postdens"],cumdist)))
}

CauchyHPD <- function(x, # vector of samples
                      p, # vector of possible parameter values
                      alpha = 0.95, # HPD interval value
                      tol = 0.0001) { # level of tolerance for exact HPD interval

  cauchydens <- function(x,p){

    cauchydist <- function(x,p) {
      H = vector(mode = "numeric",length = length(x))
      for(i in 1:length(x)){
        H[i] = (1+(x[i] - p)^2)^(-1)
      }
      return(prod(H))
    }
    dens = function(p) vapply(p,cauchydist,min(p), x = x)
```

```
    c = (integrate(dens, -Inf, Inf)$value)^(-1)
    data.frame(vparams = p, postdens = c*dens(p))
  }

  df = cauchydens(x,p)

  cumdist = cumsum(df$postdens)*diff(df$vparams)[1]
  post_median = which.min(abs(cumdist-0.5))

  HPDlimits <- function(post_dens) { ## find lower and upper values for which
    ## prob dens is closest to target value
    lower = which.min(abs(df$postdens[1:post_median]-post_dens))
    upper = which.min(abs(df$postdens[(post_median+1):length(df$postdens)]-post_dens))+post_median
    limits = c(lower,upper)
  }

  HPDlimitarea <- function(post_dens) {
    limitints = HPDlimits(post_dens)
    limitarea = sum(df$postdens[limitints[1]:limitints[2]])*diff(df$vparams)[1]
  }
  ## find credible interval
  v2 = seq(0,max(df$postdens),by=tol)
  vals = sapply(v2,HPDlimitarea)
  w = which.min(abs(vals-alpha))
  r = c(df$vparams[HPDlimits(v2[w])])
  names(r) = c("lower","upper")
  par(mfrow = c(1,2))
  plot(df$vparams, cumdist, type = 'l')
  abline(h = 0.5)
  abline(v = df$vparams[post_median], col = 'red')
  abline(v = r["upper"], col = 'blue')
  abline(v = r["lower"], col = 'blue')
  plot(df$vparams, df$postdens, type = 'l')
  abline(v = df$vparams[post_median], col = 'red')
  abline(h = df[HPDlimits(v2[w])[1],"postdens"])
  abline(v = r["upper"], col = 'blue')
  abline(v = r["lower"], col = 'blue')
  return(r)
}
```

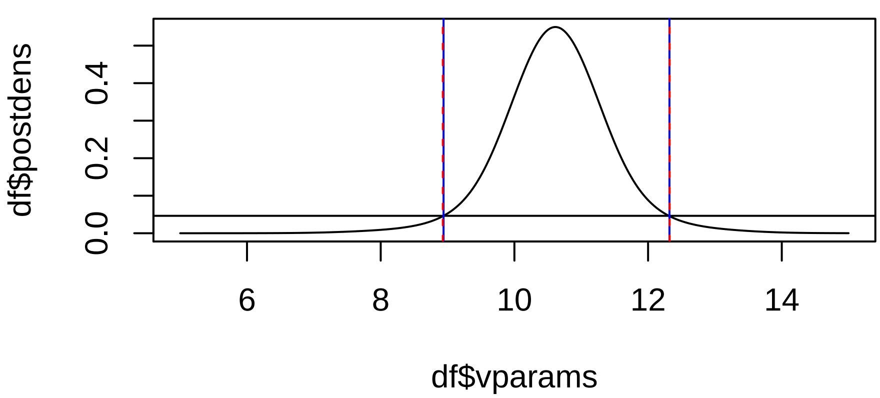
(b) Consider Jaynes' example of $n = 2$ observations $(3, 5)$: plot the posterior and calculate the 90% central credible interval. Explain why it is quite different from the confidence interval derived by Jaynes (p.202).

```
source("Question2/2b.R")
```



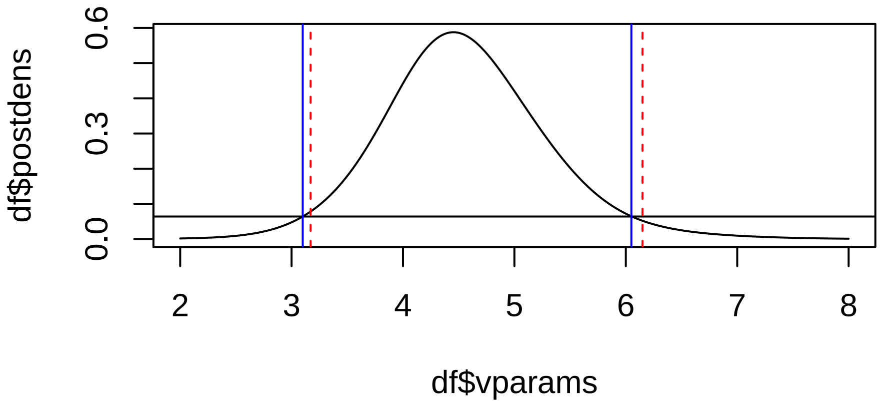

(c) Consider Box & Tiao's example of $n = 5$ observations $(11.4, 7.3, 9.8, 13.7, 10.6)$: plot the posterior and calculate 95% central and HPD credible intervals and check $Pr[\theta < 11.5]$ given by Box & Tiao.

```
source("Question2/2c.R")
```




(d) Consider Berger's (1985, p.141) example of $n = 5$ observations $(4.0, 5.5, 7.5, 4.5, 3.0)$: calculate 95% central and HPD credible intervals, with and without Berger's restriction $(\theta > 0)$.

```
source("Question2/2d.R")
```




(e) Clearly, Berger's restriction ($\theta > 0$) will sometimes lead to a posterior quite different from the unrestricted posterior. Plot this restricted posterior for the hypothetical negative version of Berger's example: i.e. $(-4.0, -5.5, -7.5, -4.5, -3.0)$, and calculate the 95% HPD interval.

```
source("Question2/2e.R")
```