

Assignment2

Abhishek kumar Gupta

March 23, 2019

Contents

| | |
|--|-----------|
| 1. The following will require a derivation and R functions: | 2 |
| a. Write down Bayes' theorem | 2 |
| Baye's theorem | 2 |
| b. Derive the general posterior result for a Beta prior and Binomial likelihood. (Show all working) | 3 |
| c. Plot the three graphs on one interface using R and use the experimental results $n=10$, $x=4$ with a uniform prior. | 3 |
| d. Make a function that will create a similar plot but for different alpha, beta, n and x. Call the function mybeta() | 4 |
| e. Give the output of the function when the following is submitted: | 5 |
| f. Now make a new function (mybeta2()) that will produce the same plots as mybeta() but will release command line Bayesian point estimates and BCI's of whatever equal tail size we wish - all estimating "p". The command line output will be a list containing these estimates (all appropriately labelled). Give the outputs of: | 9 |
| 2. A prior used to summarize belief about "p" can often be well approximated by a Beta distribution with appropriate choice of hyper-parameters alpha and beta. This will not always be the case. Sometimes a mixture of betas will do a better job. A mixture of beta densities can be expressed in the following way $mixbeta(x) = w * dbeta(x, a1, b1) + (1 - w) * dbeta(x, a2, b2)$ where w is a mixing weight and a number between 0 and 1. | 14 |
| a. Show that mixbeta satisfies the first 2/3 properties of a density. The three properties are: . . . | 14 |
| b. Make an R function that will create the mixture density using mixbeta as described above: . . . | 15 |
| c. Make a function that will plot the mixture - call it mymixplot(). Show the output when the following are submitted: | 15 |
| d. Using the mixbeta prior and a single Binomial (x successes in n trials) | 18 |
| i. Find the analytical posterior - that is do the algebra and show that the posterior is a mixture also. | 18 |
| ii. What is the posterior mixing weight? | 19 |
| iii. Plot the prior, likelihood and posterior on the same set of axes when $w = 0.5$, $n=10$, $x = 4$, $a1 = a2=2, b1=b2=4$ | 19 |
| 3. The following problem relates to the above notions of mixtures. Suppose a coin is either "unbiased" or "biased". In which case the chance of a "head" is unknown and is given a uniform prior distribution. We assess a prior probability of 0.9 that it is "unbiased", and then observe 15 heads out of 20 tosses. | 21 |
| a. Explain how the JAGS distribution function dcat() works. | 21 |
| b. Take the above model and use it within JK's code "Jags-ExampleScript.R" adjusting it so that it will create an MCMC sample from the posterior. The two parameters you must monitor are "theta[2]" and "biased" (note: you do not need the inits function - just use something like: initsList = list(pick = 1)) - do NOT attempt to plot the "biased" parameter yet - ONLY "theta[2]". | 21 |
| c. Once you have it working place the script within the body of a function - say mypriormix(), you can then call the script by simply calling the function (no options are needed). | 21 |
| Ans: 3b,c | 21 |
| d. Show the MCMC diagnostics and the posterior sample histograms for theta[2]. | 22 |

| | |
|--|-----------|
| e. Show the summary information of the posterior sample - what is the sample called in the script? Hint: Use <code>summary()</code> ; <code>su = summary(.)</code> ; <code>su\$statistics</code> | 25 |
| f. What is the posterior probability that the coin is biased? | 25 |
| 4. We will work on the same model as in question 3. This time we will examine the model in light of the theory covered in section 10.3.2.1 page 279 and following. The first thing we will need to do is manipulate the list of data created by the JAGS sampler. Locate the mcmc data which will be in the form of a list. | 26 |
| a. Give the structure of the MCMC data in the file produced by the jags script you made in qu. 3 hint: <code>str(.)</code> | 26 |
| b. Use the following code to make <code>mcmcT</code> by filling in the correct object name | 26 |
| c. Using <code>mcmcT</code> and the <code>ggplot2</code> R package make the following plot after first understanding precisely the pseudo prior method. Make sure you have YOUR name on it!! Colors don't have to precisely correspond. | 26 |
| d. Looking at the picture above and considering the model, answer the following: | 27 |
| i. When the parameter <code>biased = 0</code> : <code>pick = ?</code> , | 27 |
| ii. when <code>biased = 1</code> : <code>pick = ?</code> | 27 |
| e. The three variables updated in the Gibbs sampler will go in alphabetical order <code>pick</code>, <code>theta[1]</code> and then <code>theta[2]</code> and then cycling around again and again . . | 27 |
| i. When <code>pick = 1</code> , <code>theta[1]</code> will be sampled from the posterior, then <code>theta[2]</code> will be sampled from what? | 27 |
| ii. When <code>pick = 2</code> what will <code>theta[1]</code> be sampled from? | 27 |
| iii. When <code>pick = 2</code> what will <code>theta[2]</code> be sampled from? | 28 |
| iv. Now explain the plot in d) from the above. | 28 |
| v. Which plot (dark or light) blue would likely represent the true underlying posterior of <code>theta[2]</code> | 28 |
| vi. Now create the following plot which represents the incorrect posterior of <code>theta[2]</code> using <code>ggplot</code> - make sure your name is on it. | 28 |
| vii. Explain the above plot - why does it have the shape it has and then say why it cannot be a true representation of the posterior for <code>theta[2]</code> ? | 29 |
| 5. Now using the method of pseudo priors recode the model and create a function that will produce mcmc output that will be a better representation of the posterior of <code>theta[2]</code>. Hint: Use a beta pseudo prior for <code>theta[2]</code>, the shapes can be calculated using eq. 6.7 pg. 131. | 29 |
| a. Derive the formulae (eq. 6.7) given on pg. 131. | 29 |
| b. You will need to "post-process" the MCMC in order to obtain parameter estimates for the pseudo-prior. | 30 |
| c. Copy and paste your new pseudo prior JAGS model (not all the script JUST the model as in qu 3) | 31 |
| d. Make a function <code>pseudobin()</code> that will run the script - the function should produce | 32 |

1. The following will require a derivation and R functions:

a. Write down Bayes' theorem

Baye's theorem

$$p(A/B) = \frac{p(B/A) * p(A)}{p(B)}$$

b. Derive the general posterior result for a Beta prior and Binomial likelihood. (Show all working)

Since X is a binomial distribution with $X \sim \text{Bin}(n, \theta)$, it can be written as:

$$p(x|\theta) = \binom{n}{x} \theta^x (1 - \theta)^{n-x}$$

Also, given θ follows beta distribution as $\theta \sim \text{Beta}(\alpha, \beta)$

since,

$$p(\theta|\alpha, \beta) = \text{beta}(\theta|\alpha, \beta)$$

So we can expand the beta distribution as follows:

$$p(\theta|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}$$

Since, by Bay's rule we know that

$$\text{posterior} \propto \text{prior} * \text{likelihood}$$

So, we can write as:

$$p(\theta|x) \propto p(\theta)p(x|\theta)$$

and substituting the prior and likelihood function from the above equations, we get

$$p(\theta|x) \propto \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \binom{n}{x} \theta^x (1 - \theta)^{n-x}$$

To remove the proportional we, have a constant (called evidence in Bay's rule), So we have now

$$p(\theta|x) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \frac{1}{p(x, n)} \binom{n}{x} \theta^x (1 - \theta)^{n-x}$$

After re-arranging the terms, we get

$$p(\theta|x) = \theta^{\alpha-1} (1 - \theta)^{\beta-1} \binom{n}{x} \theta^x (1 - \theta)^{n-x} \frac{1}{[B(\alpha, \beta) * p(x, n)]}$$

By collecting powers of same terms, we get

$$p(\theta|x) = \theta^{\alpha+x-1} (1 - \theta)^{\beta+n-x-1} \binom{n}{x} \frac{1}{[B(\alpha, \beta) * p(x, n)]}$$

Since $\binom{n}{x}$ is constant like the bottom terms, we can write together as a function of constant terms as:

$$p(\theta|x) = \theta^{(\alpha+x)-1} (1 - \theta)^{(\beta+n-x)-1} \frac{1}{[B(\alpha + x, \beta + n - x)]}$$

rearranging the terms, we get the beta distribution as seen from below:

$$p(\theta|x) = \theta^{(x+\alpha)-1} (1 - \theta)^{(n-x+\beta)-1} \frac{1}{[B(x + \alpha, n - x + \beta)]}$$

Hence, we get

$$\theta|x \propto \text{Beta}(x + \alpha, n - x + \beta)$$

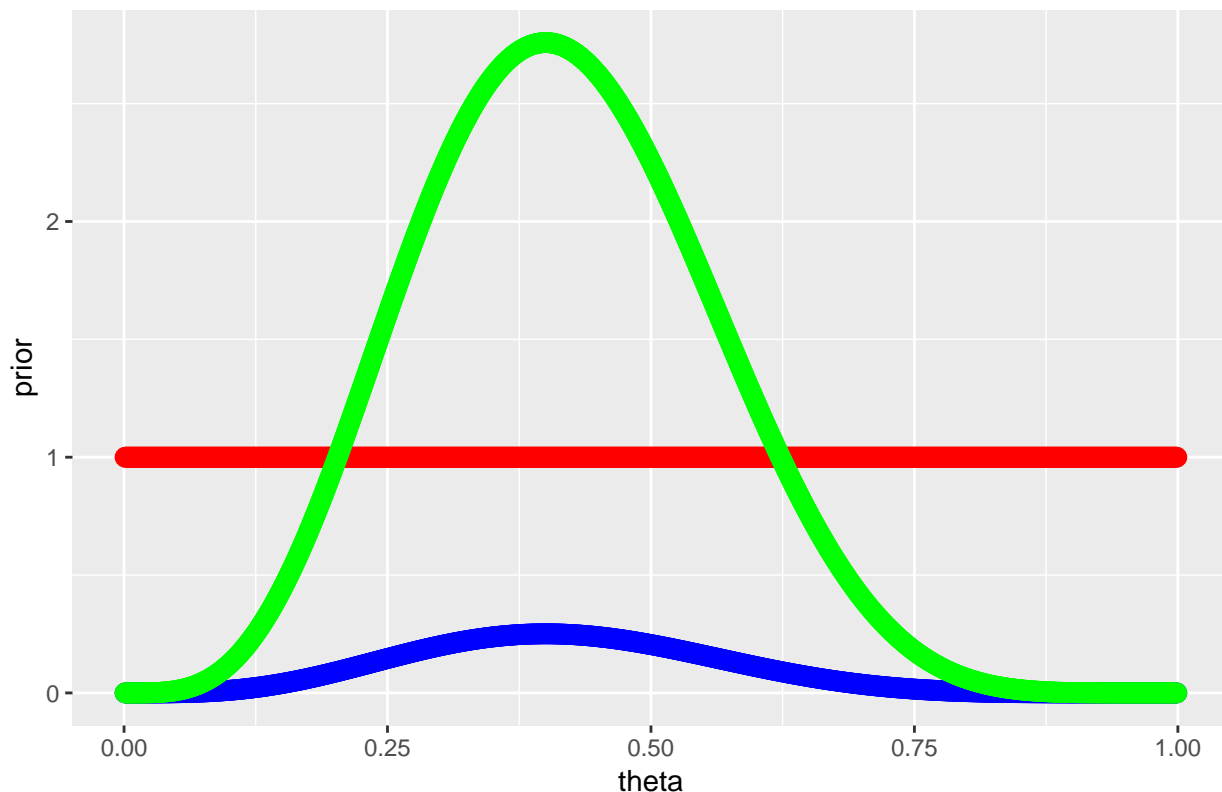
c. Plot the three graphs on one interface using R and use the experimental results $n=10$, $x=4$ with a uniform prior.

```

library(ggplot2)
a=1
b=1
n=10
x=4
theta = seq(0.001,0.999,by=0.001) # points for plotting
prior= dbeta(theta,a,b)
post = dbeta( theta , a+x , b+n-x ) # posterior for plotting
lik= dbinom(x=x,size = n, theta) # likelihood for plotting
df = data.frame(theta, prior, lik, post)
myplot = ggplot(df, aes(theta)) + geom_point(aes(y=prior), col = "red", size = 3) +
  geom_point(aes(y=lik), col = "blue", size =3) +geom_point(aes(y=post), col ="green", size =3) +
  labs(title = "Abhishek Kumar Gupta")
myplot

```

Abhishek Kumar Gupta



d. Make a function that will create a similar plot but for different alpha, beta, n and x. Call the function mybeta()

```

mybeta = function(a,b,n,x)
{
  theta = seq(0.001,0.999,by=0.001) # points for plotting
  prior= dbeta(theta,a,b)
  post = dbeta( theta , x+a , n-x+b ) # posterior for plotting
  lik= dbinom(x=x,size = n, theta) # likelihood for plotting

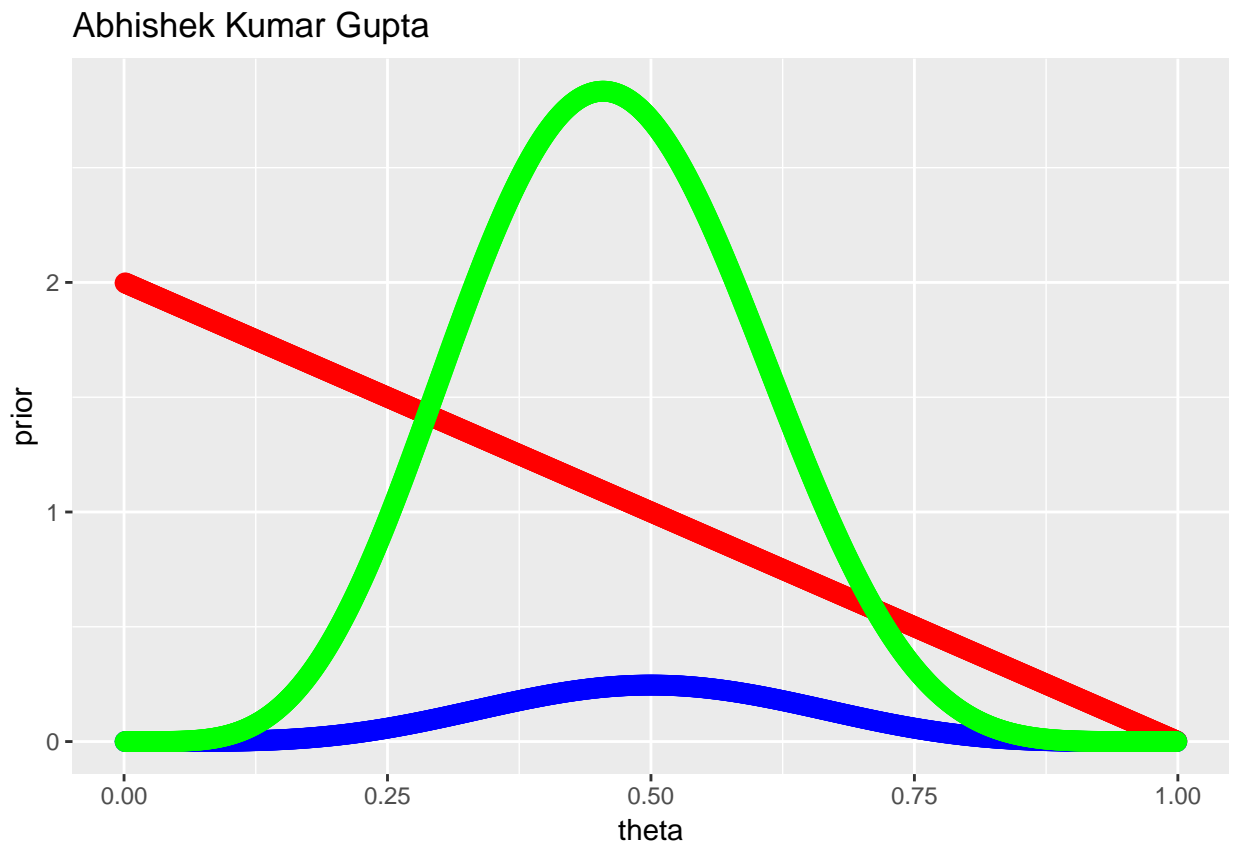
```

```

df = data.frame(theta, prior, lik, post)
myplot = ggplot(df, aes(theta)) + geom_point(aes(y=prior), col = "red", size = 3) +
  geom_point(aes(y=lik), col = "blue", size = 3) + geom_point(aes(y=post), col = "green", size = 3) +
  labs(title = "Abhishek Kumar Gupta")
return (myplot)
}

mybeta(1,2,10,5)

```

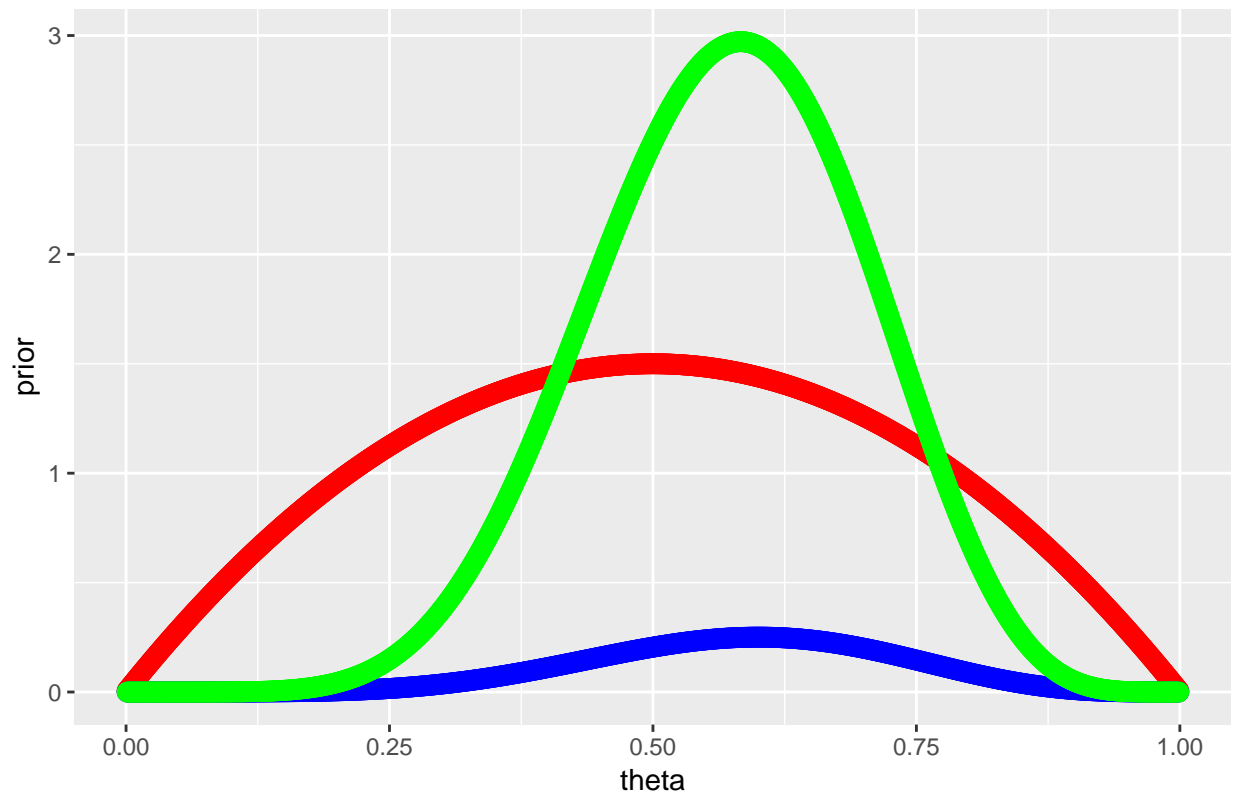


e. Give the output of the function when the following is submitted:

i. `Mybeta(alpha = 2, beta=2, n=10, x=6)`

```
mybeta(2,2,10,6)
```

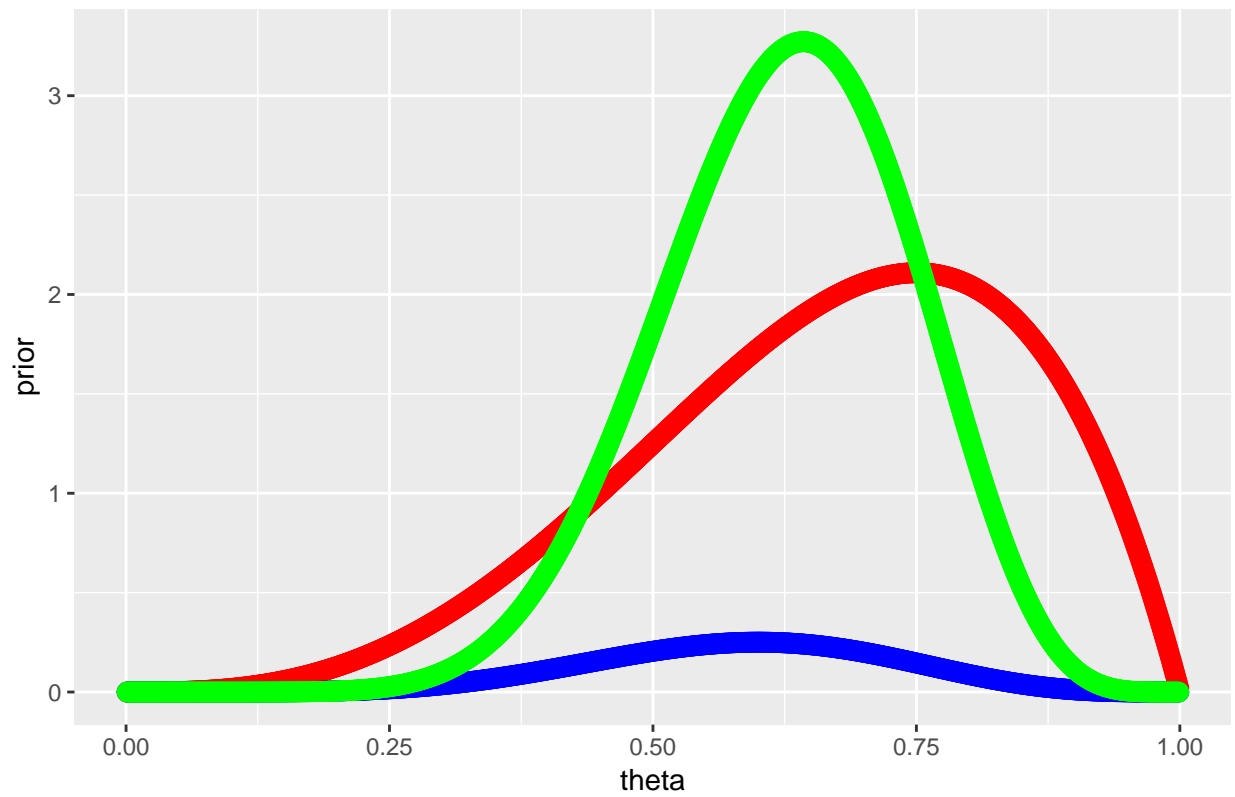
Abhishek Kumar Gupta



ii. `Mybeta(alpha = 4, beta=2, n=10, x=6)`

```
mybeta(4,2,10,6)
```

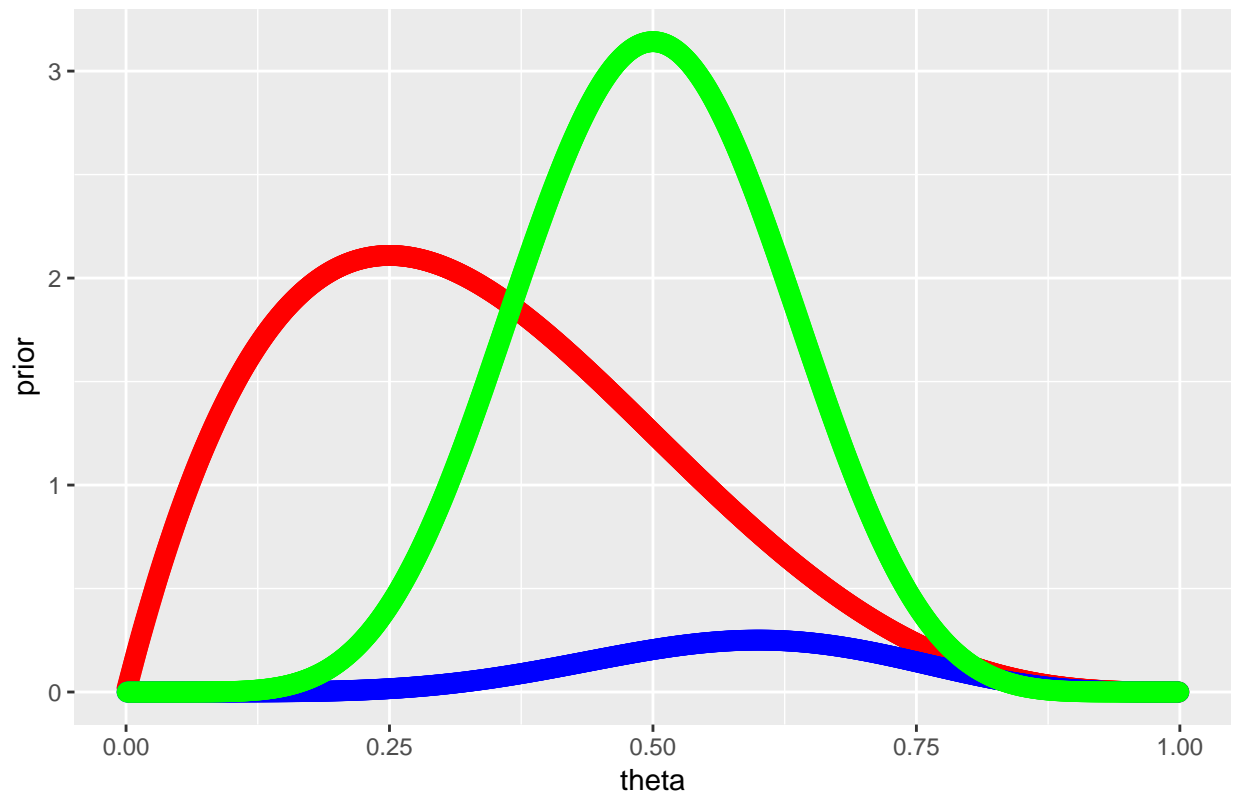
Abhishek Kumar Gupta



iii. `Mybeta(alpha = 2, beta=4, n=10, x=6)`

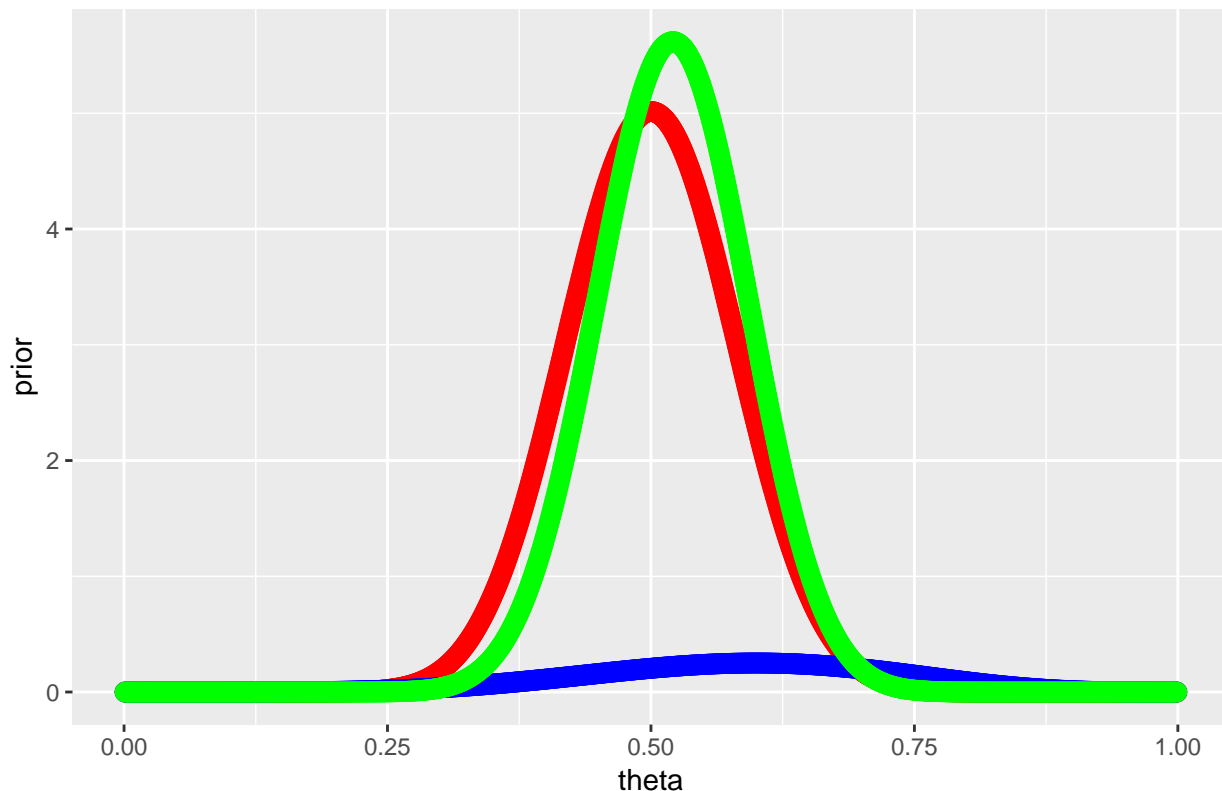
```
mybeta(2,4,10,6)
```

Abhishek Kumar Gupta



iv. `Mybeta(alpha = 20, beta=20, n=10, x=6)`

```
mybeta(20,20,10,6)
```

f. Now make a new function (`mybeta2()`) that will produce the same plots as `mybeta()` but will release command line Bayesian point estimates and BCI's of whatever equal tail size we wish - all estimating "p". The command line output will be a list containing these estimates (all appropriately labelled). Give the outputs of:

```
HDIofICDF = function( ICDFname , credMass=0.95 , tol=1e-8 , ... ) {
  # Arguments:
  #   ICDFname is R's name for the inverse cumulative density function
  #     of the distribution.
  #   credMass is the desired mass of the HDI region.
  #   tol is passed to R's optimize function.
  # Return value:
  #   Highest density interval (HDI) limits in a vector.
  # Example of use: For determining HDI of a beta(30,12) distribution, type
  #   HDIofICDF( qbeta , shape1 = 30 , shape2 = 12 )
  # Notice that the parameters of the ICDFname must be explicitly named;
  # e.g., HDIofICDF( qbeta , 30 , 12 ) does not work.
  # Adapted and corrected from Greg Snow's TeachingDemos package.
  incredMass = 1.0 - credMass
  intervalWidth = function( lowTailPr , ICDFname , credMass , ... ) {
    ICDFname( credMass + lowTailPr , ... ) - ICDFname( lowTailPr , ... )
  }
  optInfo = optimize( intervalWidth , c( 0 , incredMass ) , ICDFname=ICDFname ,
    credMass=credMass , tol=tol , ... )
```

```

HDIlowTailPr = optInfo$minimum
return( c( ICDFname( HDIlowTailPr , ... ) ,
            ICDFname( credMass + HDIlowTailPr , ... ) ) )
}

mybeta2 = function(a,b,n,x,alpha)
{
  library(ggplot2)
  theta = seq(0.001,0.999,by=0.001) # points for plotting
  prior= dbeta(theta,a,b)
  post = dbeta( theta , x+a , n-x+b ) # posterior for plotting
  lik= dbinom(x=x,size = n, theta) # likelihood for plotting
  df = data.frame(theta, prior, lik, post)
  myplot = ggplot(df, aes(theta)) + geom_point(aes(y=prior), col = "red", size = 5) +
    geom_point(aes(y=lik), col = "blue", size =3) +geom_point(aes(y=post), col ="green", size =3) +
    labs(title = "Abhishek Kumar Gupta")
  mean = a/(a+b)
  mode = (a-1)/(a+b-2)
  HDImass = 1-alpha
  HDIinfo = HDIofICDF( qbeta , shape1=a , shape2=b , credMass=HDImass )

  return (list(mean = mean, mode = mode,BCI = HDIinfo, myplot))
}

```

i. Mybeta2(alpha = 2, beta=2, n=10, x=4, alphalevel=0.05

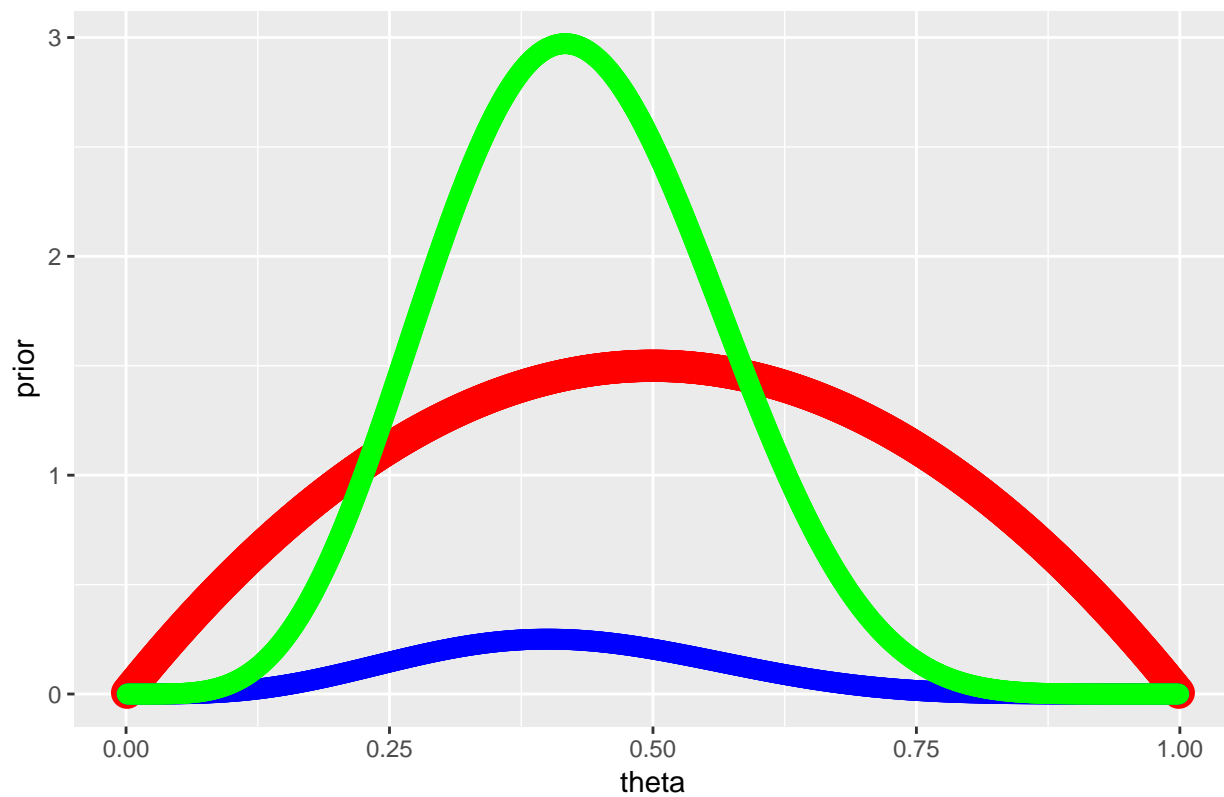
```

mybeta2(2,2,10,4,0.05)

## $mean
## [1] 0.5
##
## $mode
## [1] 0.5
##
## $BCI
## [1] 0.09429932 0.90570068
##
## [[4]]

```

Abhishek Kumar Gupta

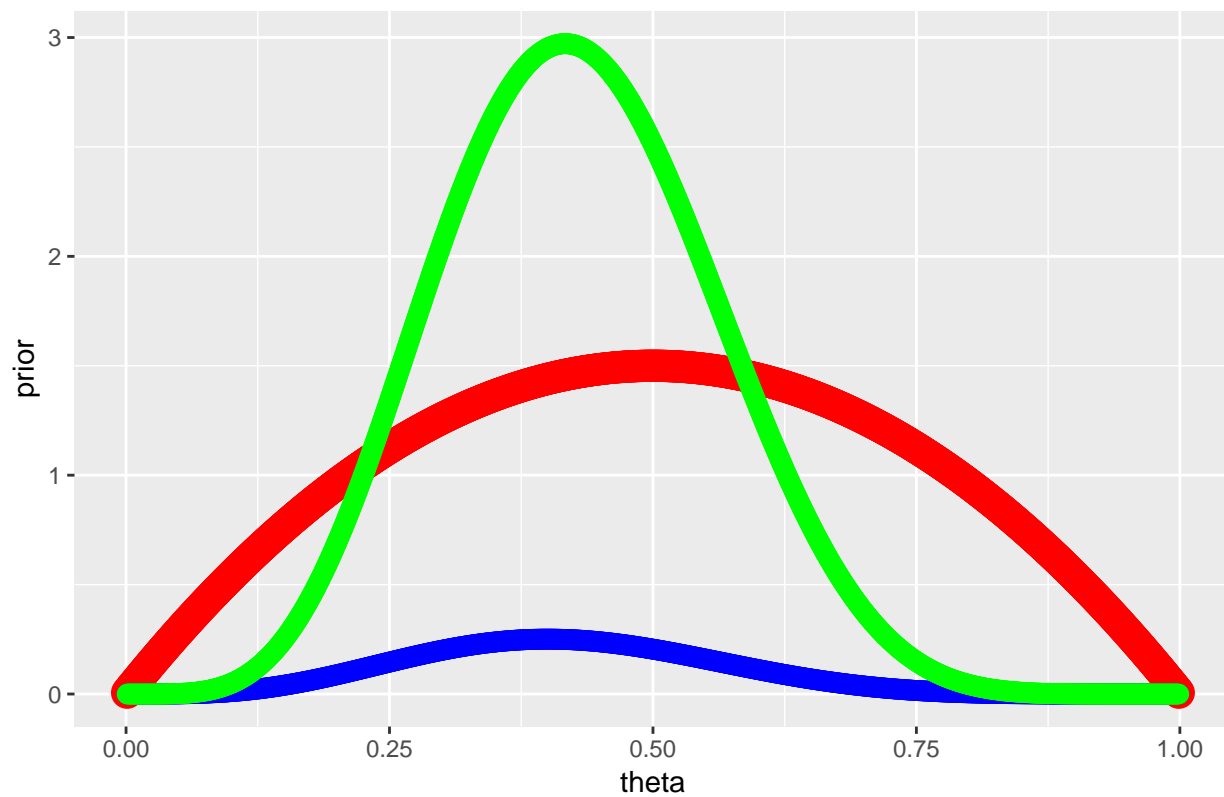


ii. `Mybeta2(alpha = 2, beta=2, n=10, x=4, alphalevel=0.10)`

```
mybeta2(2,2,10,4,0.10)
```

```
## $mean
## [1] 0.5
##
## $mode
## [1] 0.5
##
## $BCI
## [1] 0.1353504 0.8646496
##
## [[4]]
```

Abhishek Kumar Gupta

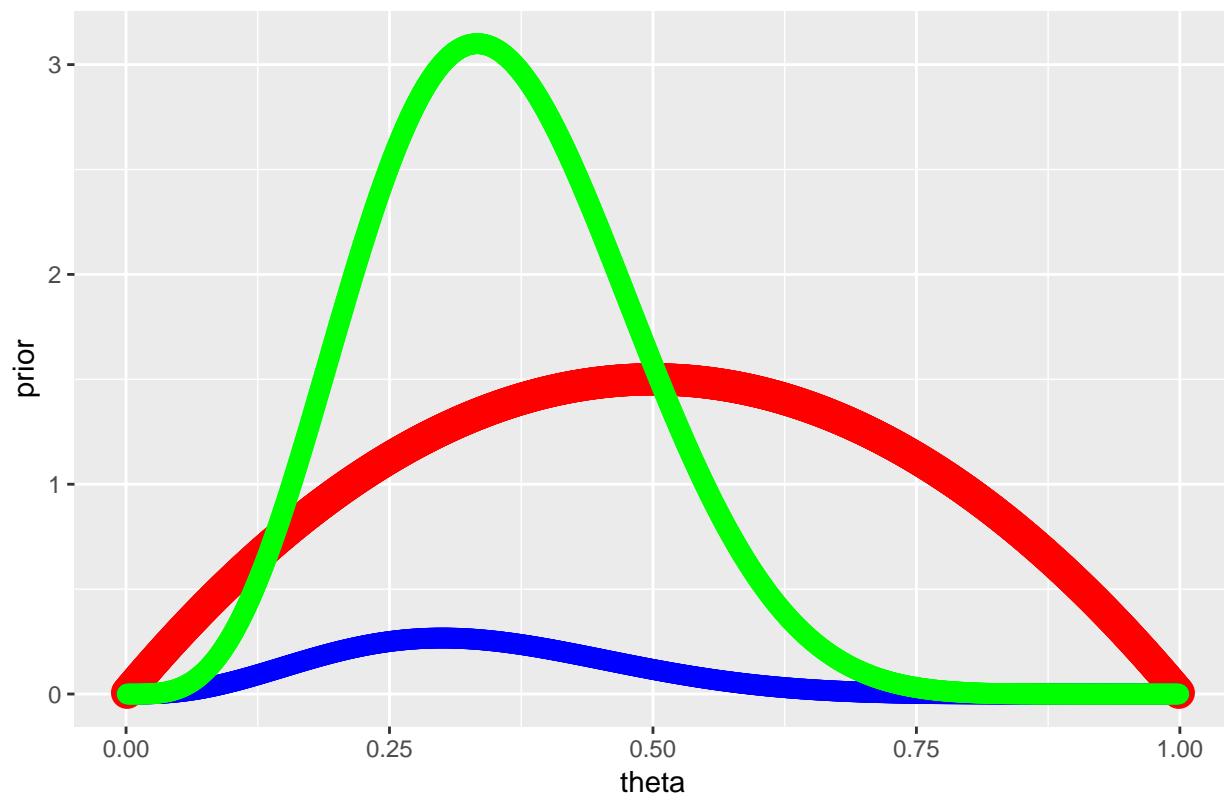


iii. `Mybeta2(alpha = 2, beta=2, n=10, x=3, alphalevel=0.05)`

```
mybeta2(2,2,10,3,0.05)
```

```
## $mean
## [1] 0.5
##
## $mode
## [1] 0.5
##
## $BCI
## [1] 0.09429932 0.90570068
##
## [[4]]
```

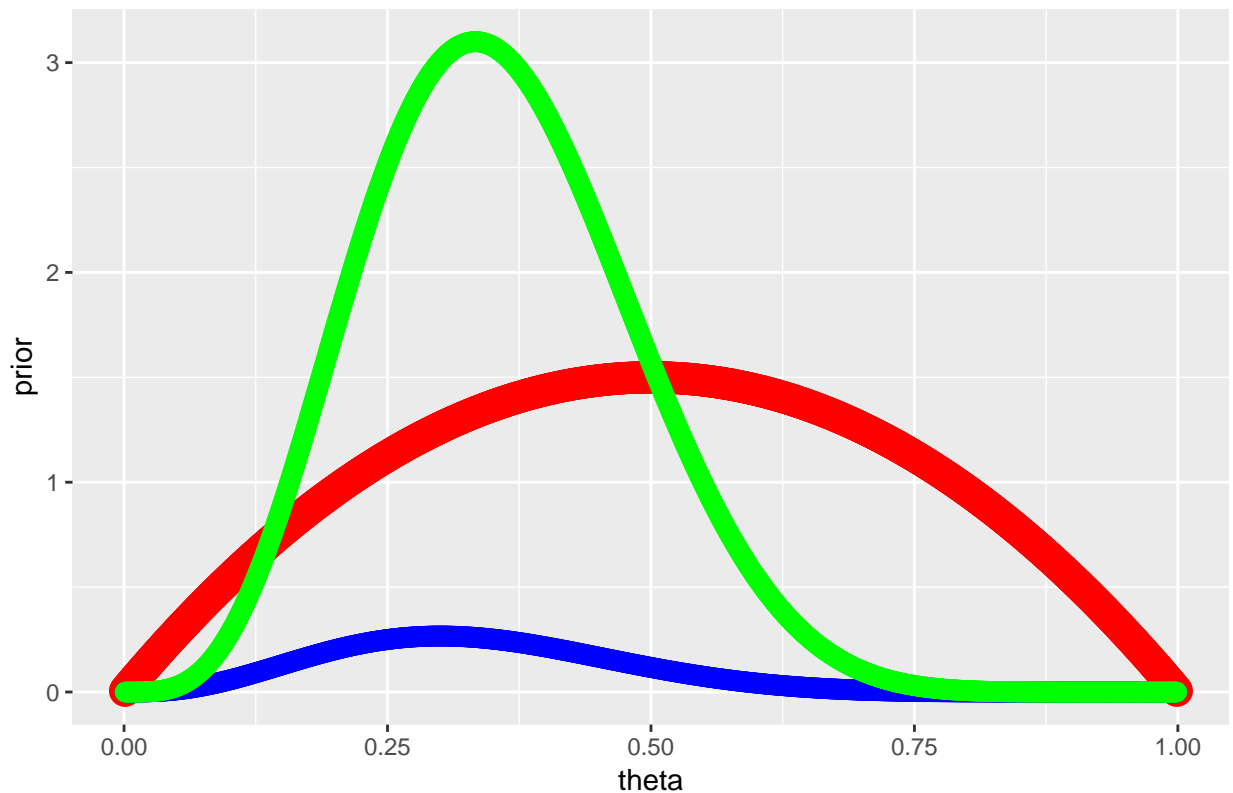
Abhishek Kumar Gupta



iv. `Mybeta2(alpha = 2, beta=2, n=10, x=3, alphalevel=0.01)`

```
mybeta2(2,2,10,3,0.01)
```

```
## $mean
## [1] 0.5
##
## $mode
## [1] 0.5
##
## $BCI
## [1] 0.04140015 0.95859985
##
## [[4]]
```



2. A prior used to summarize belief about “p” can often be well approximated by a Beta distribution with appropriate choice of hyper-parameters alpha and beta. This will not always be the case. Sometimes a mixture of betas will do a better job. A mixture of beta densities can be expressed in the following way $mixture(x) = w * dbeta(x, a1, b1) + (1 - w) * dbeta(x, a2, b2)$ where w is a mixing weight and a number between 0 and 1.

a. Show that mixture satisfies the first 2/3 properties of a density. The three properties are:

i $f(x) \geq 0$

Since its a probability density function, it must be always positive. Also, beta function will give the positive values

ii $\int_{-\infty}^{\infty} f(x)dx = 1$

Area under the curve of any probability density function over all the domain values will give 1.

iii. $p(a < X < b) = \int_a^b f(x)dx$

b. Make an R function that will create the mixture density using mixbeta as described above:

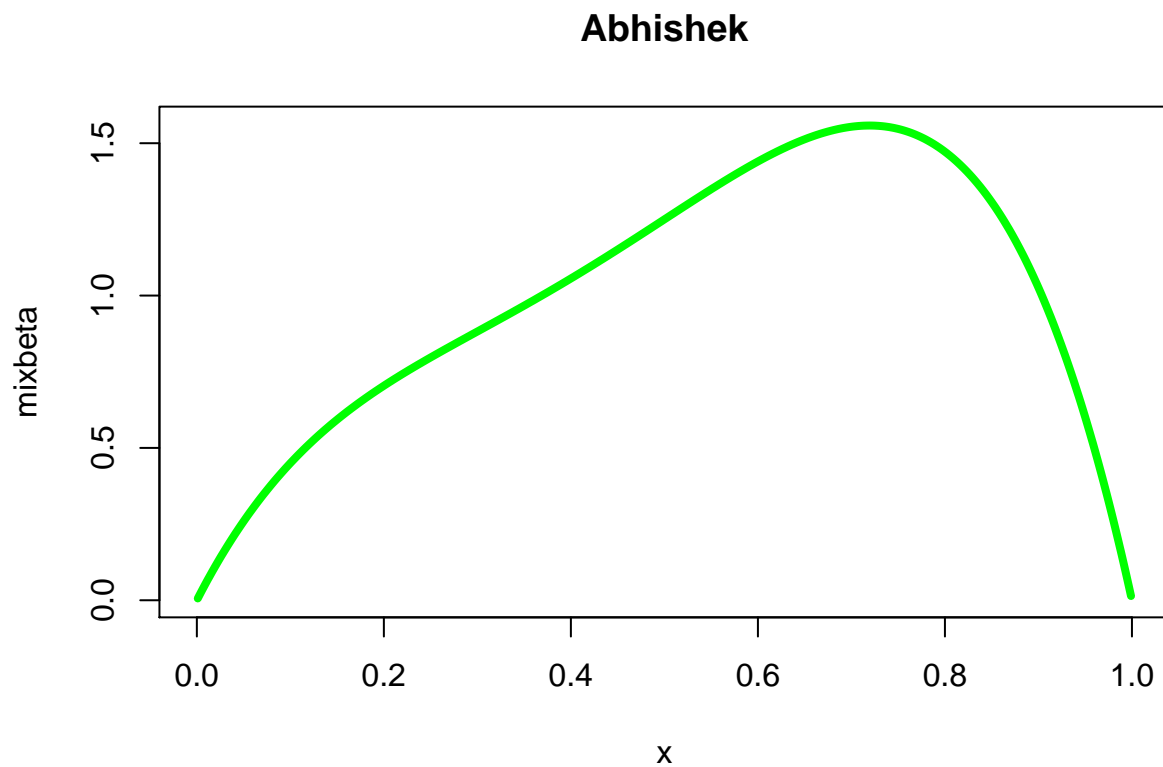
```
mymix = function(w,a1,b1,a2,b2)
{
  w*dbeta(x,a1,b1) + (1-w)*dbeta(x,a2,b2)
}
```

c. Make a function that will plot the mixture - call it mymixplot(). Show the output when the following are submitted:

```
mymixplot = function(w,a1,b1,a2,b2)
{
  x= seq(0.001,0.999,by=0.001)
  y<- w*dbeta(x,a1,b1) + (1-w)*dbeta(x,a2,b2)
  plot(y=y,x=x,xlim = c(0,1), ylab = "mixbeta", type = "l", col = "green", lwd = 4, main = "Abhishek")
}
```

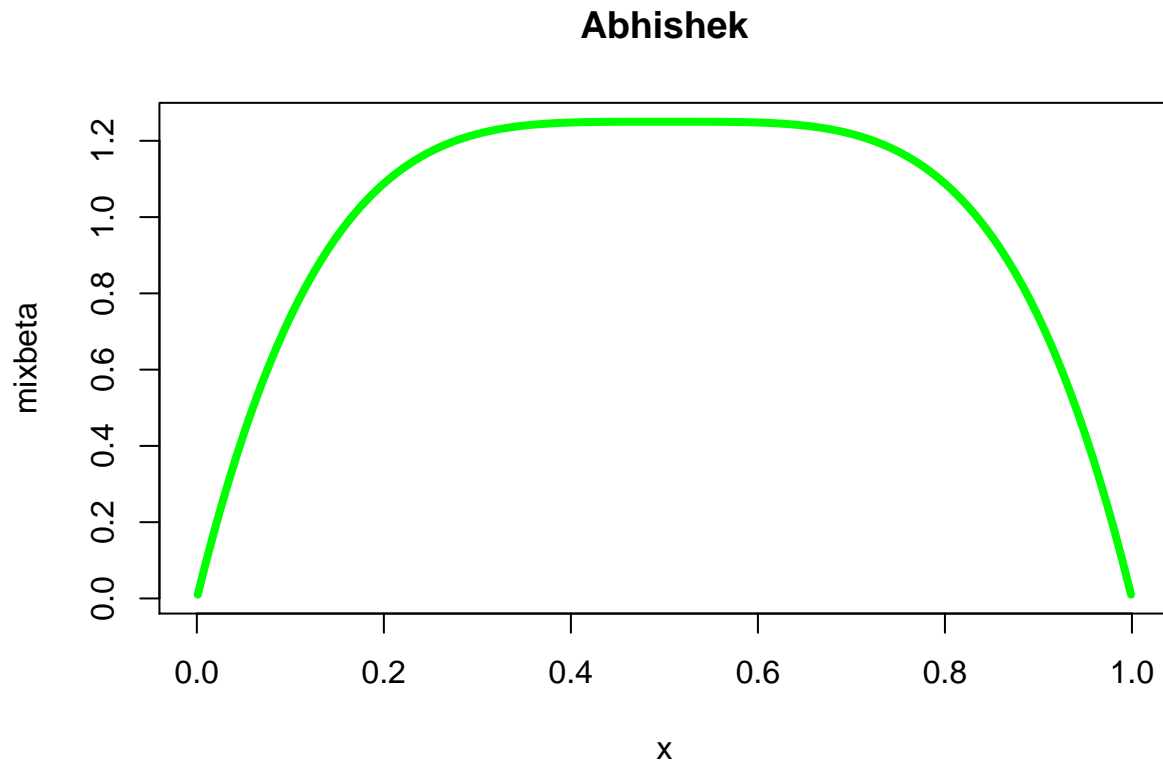
i. Mymixplot(w=0.3, a1=2,b1=4,a2=4,b2=2)

```
mymixplot(0.3,2,4,4,2)
```



ii. `Mymixplot(w=0.5, a1=2,b1=4,a2=4,b2=2)`

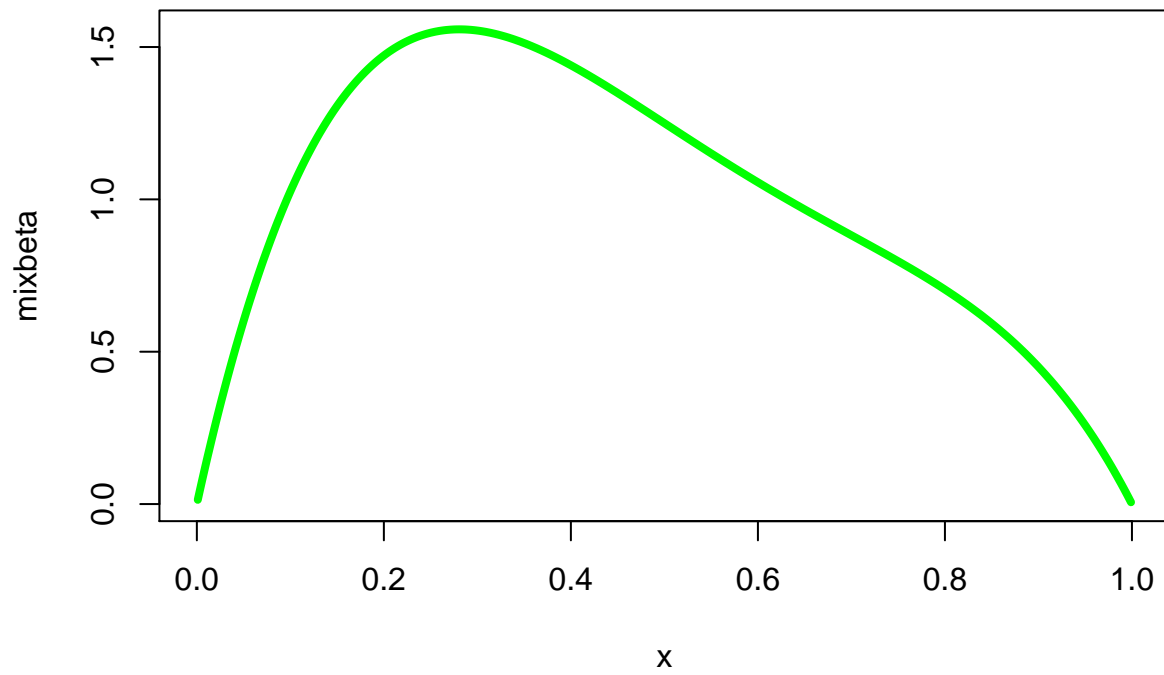
```
mymixplot(0.5,2,4,4,2)
```



iii. `Mymixplot(w=0.7, a1=2,b1=4,a2=4,b2=2)`

```
mymixplot(0.7,2,4,4,2)
```

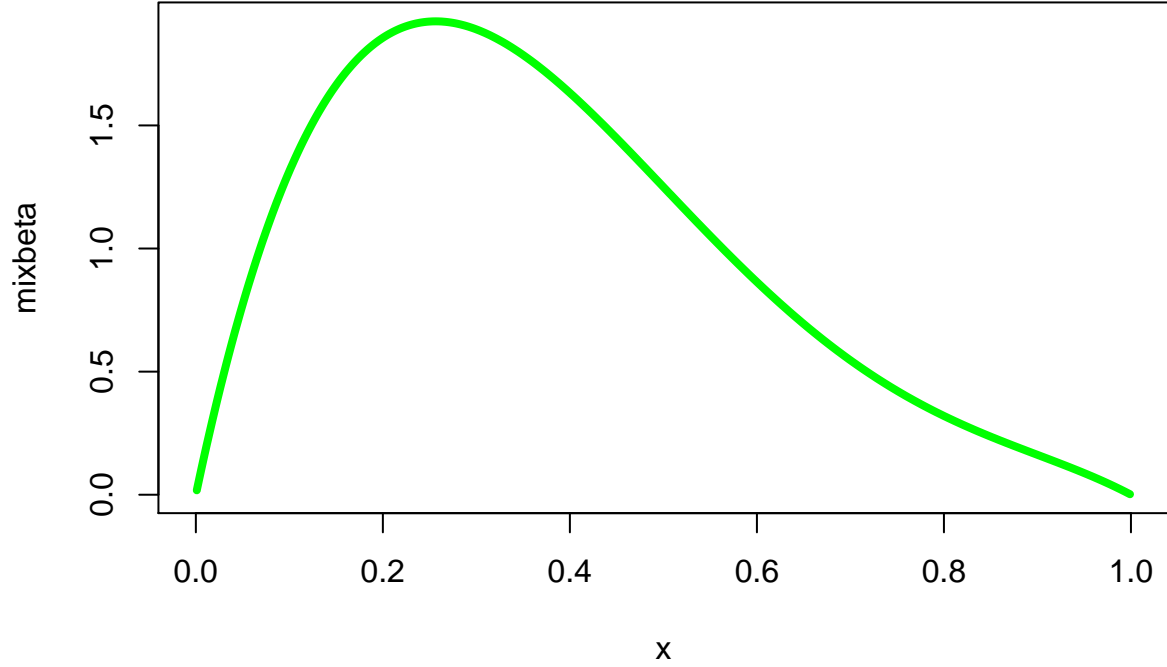

Abhishek



iv. `Mymixplot(w=0.9, a1=2,b1=4,a2=4,b2=2)`

```
mymixplot(0.9,2,4,4,2)
```

Abhishek



d. Using the mixbeta prior and a single Binomial (x successes in n trials)

i. Find the analytical posterior - that is do the algebra and show that the posterior is a mixture also.

So, we have prior as mixbeta

$$\text{mixbeta}(x) = w * \text{dbeta}(x, a1, b1) + (1 - w) * \text{dbeta}(x, a2, b2)$$

using the beta expansion, we get

$$\text{prior} = (w * \frac{\theta^{\alpha_1-1} * (1-\theta)^{\beta_1-1}}{B(\alpha_1, \beta_1)} + (1-w) * \frac{\theta^{\alpha_2-1} * (1-\theta)^{\beta_2-1}}{B(\alpha_2, \beta_2)})$$

and the likelihood is given as:

$$\text{likelihood} = \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

Since,

$$\text{Posterior} \propto \text{prior} * \text{likelihood}$$

$$\text{MixPosterior} \propto (w * \frac{\theta^{\alpha_1-1} * (1-\theta)^{\beta_1-1}}{B(\alpha_1, \beta_1)} + (1-w) * \frac{\theta^{\alpha_2-1} * (1-\theta)^{\beta_2-1}}{B(\alpha_2, \beta_2)}) * \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

Remoiving the proportionality and putting equality, we get

$$MixPosterior = (w * \frac{\theta^{\alpha_1-1} * (1-\theta)^{\beta_1-1}}{B(\alpha_1, \beta_1)} + (1-w) * \frac{\theta^{\alpha_2-1} * (1-\theta)^{\beta_2-1}}{B(\alpha_2, \beta_2)}) * \frac{\binom{n}{x} \theta^x (1-\theta)^{n-x}}{p(x, n)}$$

Multiplying the terms,

$$MixPosterior = (w * \frac{\theta^{\alpha_1-1} * (1-\theta)^{\beta_1-1}}{B(\alpha_1, \beta_1)} * \frac{\binom{n}{x} \theta^x (1-\theta)^{n-x}}{p(x, n)} + ((1-w) * \frac{\theta^{\alpha_2-1} * (1-\theta)^{\beta_2-1}}{B(\alpha_2, \beta_2)}) * \frac{\binom{n}{x} \theta^x (1-\theta)^{n-x}}{p(x, n)})$$

After rearranging the terms and collecting powers, we get

$$MixPosterior = \frac{w * \binom{n}{x} * \theta^{\alpha_1+x-1} * (1-\theta)^{n-x+\beta_1-1}}{p(x, n) * B(\alpha_1, \beta_1)} + \frac{(1-w) * \binom{n}{x} * \theta^{\alpha_2+x-1} * (1-\theta)^{n-x+\beta_2-1}}{p(x, n) * B(\alpha_2, \beta_2)}$$

Since $\binom{n}{x}$ is constant like the bottom terms, we can write together as a function of constant terms as:

$$MixPosterior = \frac{w * \theta^{\alpha_1+x-1} * (1-\theta)^{n-x+\beta_1-1}}{B(\alpha_1 + x, \beta_1 + n - x)} + \frac{(1-w) * \theta^{\alpha_2+x-1} * (1-\theta)^{n-x+\beta_2-1}}{B(\alpha_2 + x, \beta_2 + n - x)}$$

Here, we see that posterior is also a beta distribution whihc can be written as:

$$MixPosterior = w * dbeta(\alpha_1 + x, \beta_1 + n - x) + (1-w) * dbeta(\alpha_2 + x, \beta_2 + n - x)$$

Therefore, we see that posterior is also a beta mixture.

ii. What is the posterior mixing weight?

Let the two prior distribution given above as \$P_1\$ and \$P_2\$ with mixture weight \$w\$ So, mixture distribution \$P(\theta)\$ can be given as:

$$P(\theta) = w * P_1(\theta) + (1-w) * P_2(\theta)$$

After observing the data \$y\$, posterior will be given as:

$$P(\theta|y) = w' * P_1(\theta|y) + (1-w') * P_2(\theta|y)$$

where \$w'\$ is the posterior mix weight. since, by Bay's rule we have

$$p_i(\theta|y) \propto p(y_i|\theta)p_i(\theta)$$

So, posterior mix weight can be calculated as :

$$w' = \frac{w p_1(y)}{w p_1(y) + w p_2(y)}$$

where

$$p_i(y) = \int p(y|\theta)p_i(\theta)d\theta$$

iii. Plot the prior, likelihood and posterior on the same set of axes when \$w = 0.5\$, \$n=10\$, \$x =4\$, \$a1 =a2=2\$, \$b1=b2=4\$

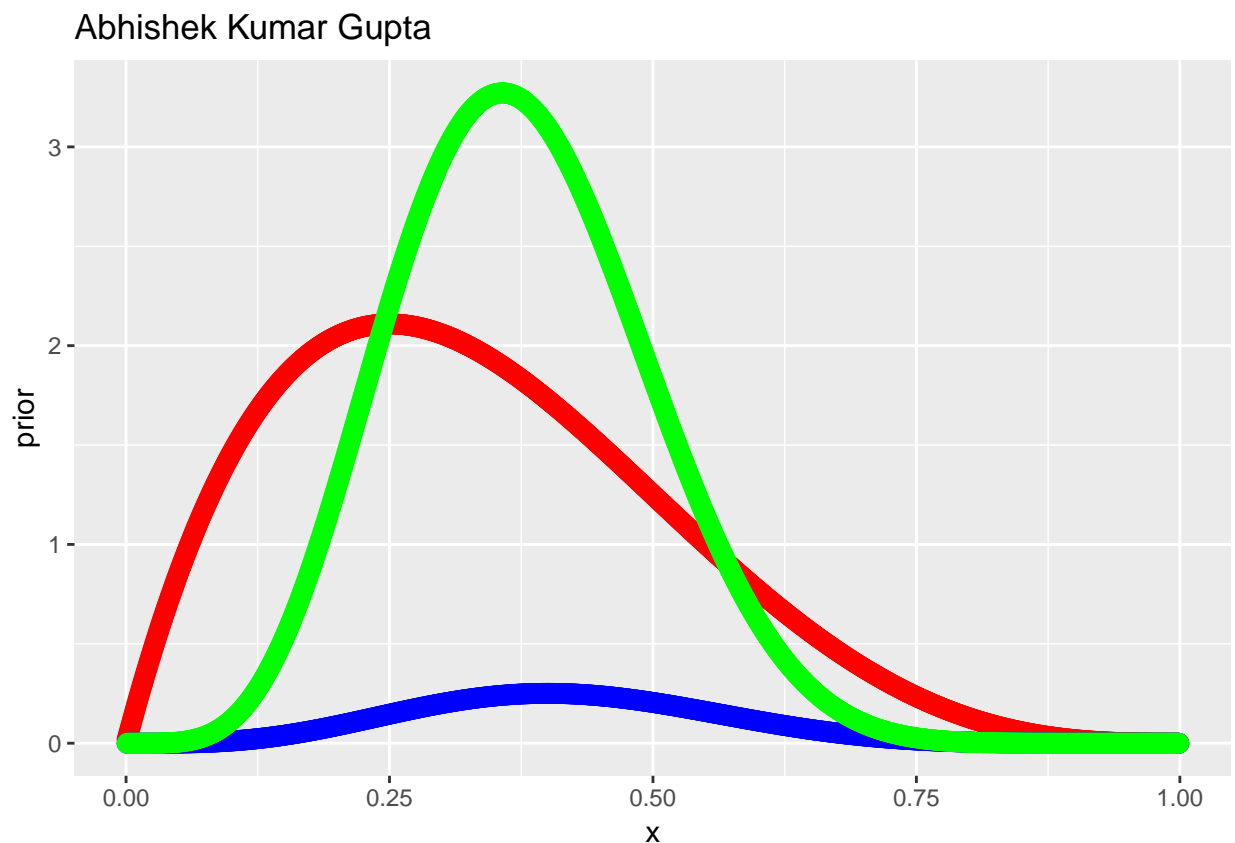
```

mymix = function(w,a1,b1,a2,b2)
{
  w*dbeta(x,a1,b1) + (1-w)*dbeta(x,a2,b2)
}

mymix_post = function(w,a1,b1,a2,b2,n,p)
{
  w*dbeta(x,a1+p,n-p+b1) + (1-w)*dbeta(x,a2+p,n-p+b2)
}

x = seq(0.001,0.999,by=0.001) # points for plotting
prior= mymix(0.5,2,4,2,4)
lik= dbinom(x=4,size = 10, x) # likelihood for plotting
post = mymix_post(0.5,2,4,2,4,10,4) # posterior for plotting
df = data.frame(x, prior, lik, post)
library(ggplot2)
myplot = ggplot(df, aes(x)) + geom_point(aes(y=prior), col = "red", size = 3) +
  geom_point(aes(y=lik), col = "blue", size =3) +geom_point(aes(y=post), col ="green", size =3) +
  labs(title = "Abhishek Kumar Gupta")
myplot

```



3. The following problem relates to the above notions of mixtures. Suppose a coin is either “unbiased” or “biased”. In which case the chance of a “head” is unknown and is given a uniform prior distribution. We assess a prior probability of 0.9 that it is “unbiased”, and then observe 15 heads out of 20 tosses.

a. Explain how the JAGS distribution function `dcat()` works.

The `dcat()` function will produce the index (1 or 2) based on what's inside it. In our case, it will produce either 1 or 2 based on the probability `q[1]` and `q[2]`.

b. Take the above model and use it within JK's code “Jags-ExampleScript.R” adjusting it so that it will create an MCMC sample from the posterior. The two parameters you must monitor are “`theta[2]`” and “biased” (note: you do not need the `inits` function - just use something like: `initsList = list(pick = 1)`) - do NOT attempt to plot the “biased” parameter yet - ONLY “`theta[2]`”.

c. Once you have it working place the script within the body of a function - say `mypriormix()`, you can then call the script by simply calling the function (no options are needed).

Ans: 3b,c

```
# Jags-ExampleScript.R
# Optional generic preliminaries:
#graphics.off() # This closes all of R's graphics windows.
#rm(list=ls()) # Careful! This clears all of R's memory!

# Load the functions used below:
source("DBDA2E-utilities.R") # Must be in R's current working directory.

##
## *****
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *****

## Warning: package 'rjags' was built under R version 3.5.3
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs

require(rjags) # Must have previously installed package rjags.

fileNameRoot="Jags-Assignment2" # For output file names.

modelString = "
model {
  x ~ dbin( p, n )
  p <- theta[pick]
  pick ~ dcat(q[]) # categorical 1 produced prob q[1], etc
```

```

# pick is 2 if biased 1 unbiased
q[1]<-0.9
q[2]<-0.1
theta[1] <-0.5 # unbiased
theta[2] ~ dunif(0,1) # biased
biased <- pick - 1
}
" # close quote for modelString

writeLines( modelString , con="TEMPmodel.txt" )

data.List = list( x = 15, n=20)
initsList = list(pick=1)

# Run the chains:
jagsModel = jags.model( file="TEMPmodel.txt" , data=data.List , inits=initsList ,
                        n.chains=3 , n.adapt=500 )

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 1
##   Unobserved stochastic nodes: 2
##   Total graph size: 12
##
## Initializing model

update( jagsModel , n.iter=500 )
codaSamples = coda.samples( jagsModel , variable.names=c("theta[1]","biased","theta[2]") ,
                           n.iter=3334 )
save( codaSamples , file=paste0(fileNameRoot,"Mcmc.Rdata") )

```

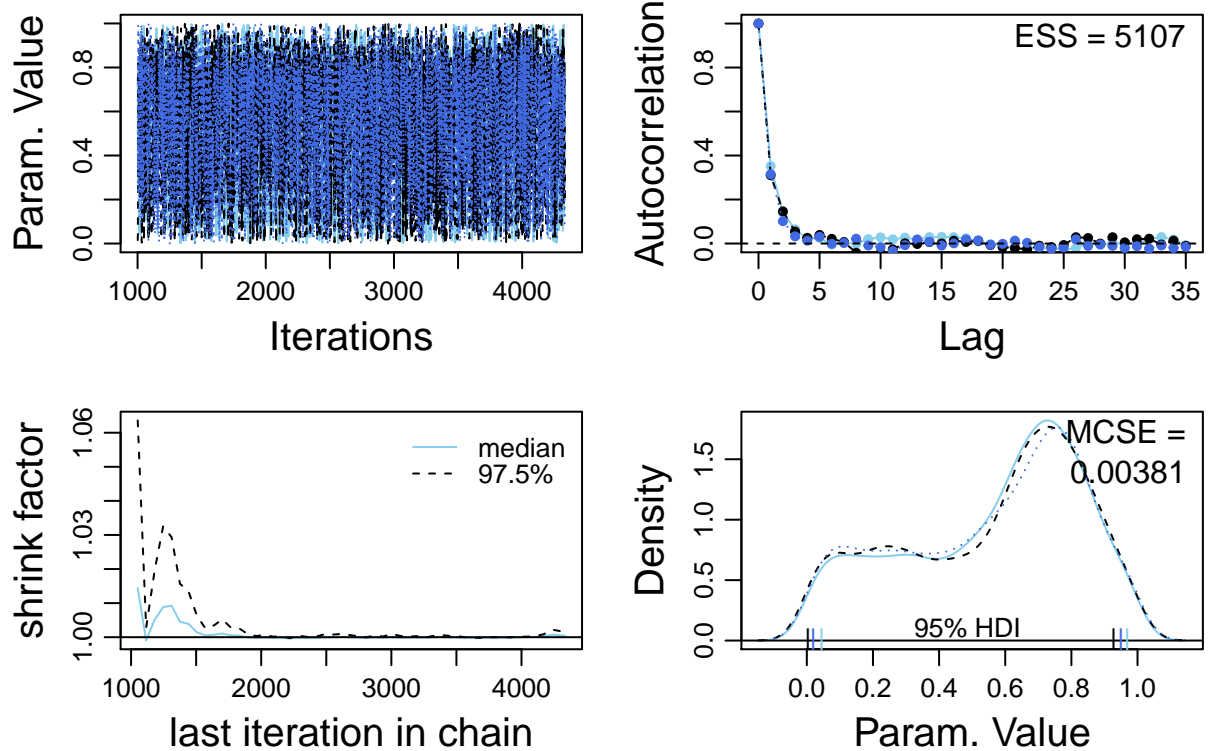
d. Show the MCMC diagnostics and the posterior sample histograms for $\theta[2]$.

```

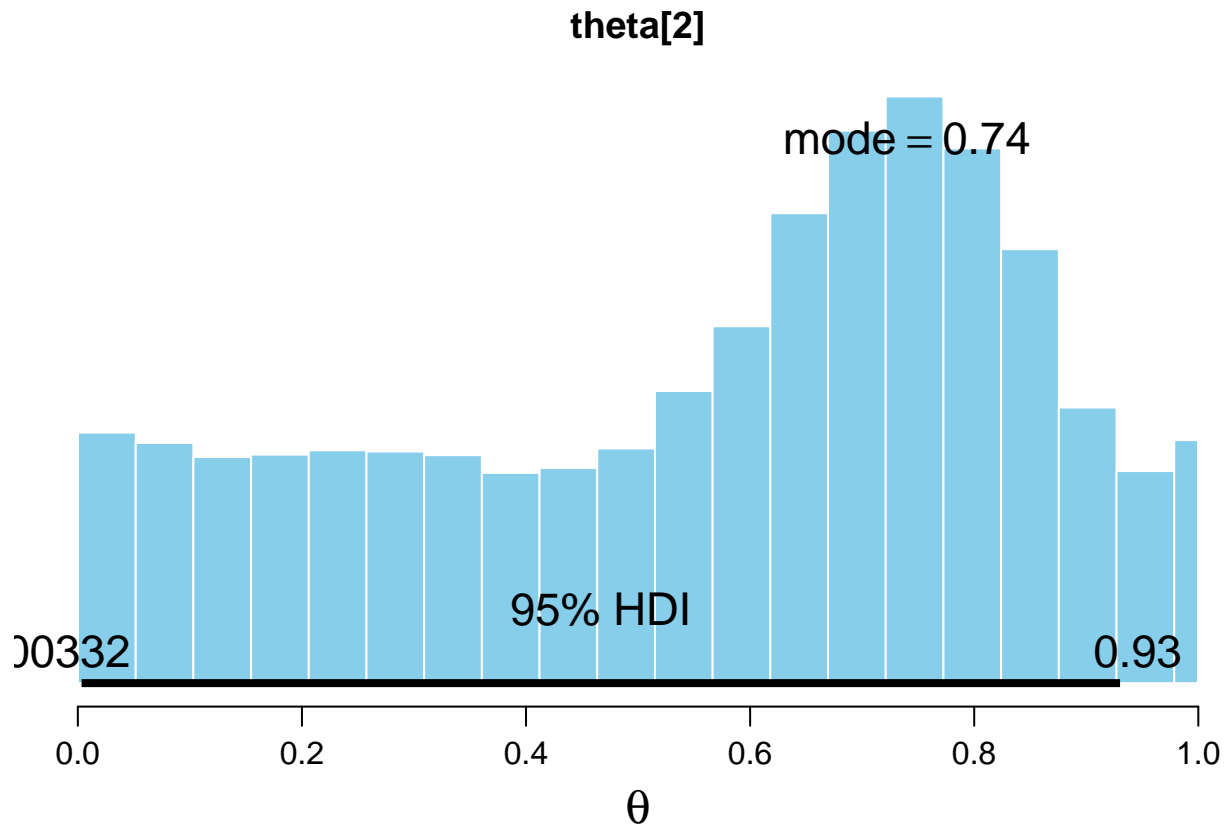
# Examine the chains:
# Convergence diagnostics:
diagMCMC( codaObject=codaSamples , parName="theta[2]" )
saveGraph( file=paste0(fileNameRoot,"ThetaDiag") , type="eps" )

```

theta[2]

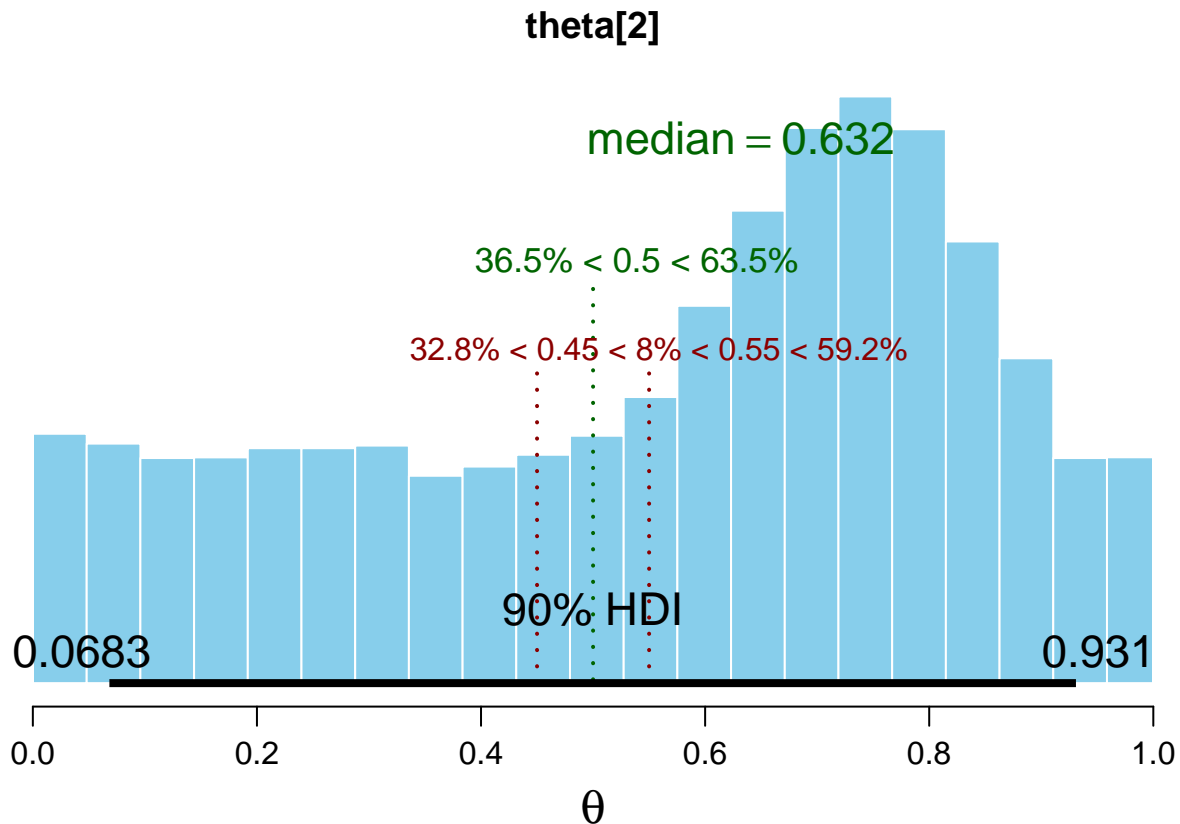


```
# Posterior descriptives:
openGraph(height=3,width=4)
par( mar=c(3.5,0.5,2.5,0.5) , mgp=c(2.25,0.7,0) )
plotPost( codaSamples[, "theta[2]" ] , main="theta[2]" , xlab=bquote(theta) )
```



```
##          ESS      mean    median    mode hdiMass      hdiLow  hdiHigh
## theta 4707.206 0.5598235 0.6319723 0.7399005    0.95 0.003315618 0.9301776
##      compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE pGtROPE
## theta      NA         NA      NA      NA      NA      NA      NA
```

```
saveGraph( file=paste0(fileNameRoot,"ThetaPost") , type="eps" )
# Re-plot with different annotations:
plotPost( codaSamples[, "theta[2]"] , main="theta[2]" , xlab=bquote(theta) ,
           cenTend="median" , compVal=0.5 , ROPE=c(0.45,0.55) , credMass=0.90 )
```

```
##          ESS      mean    median      mode hdiMass    hdiLow    hdiHigh
## theta 4707.206 0.5598235 0.6319723 0.7399005      0.9 0.06830903 0.9309631
##      compVal pGtCompVal ROPElow ROPEhigh  pLtROPE    pInROPE    pGtROPE
## theta      0.5   0.634973   0.45     0.55 0.3279344 0.08028394 0.5917816
```

```
saveGraph( file=paste0(fileNameRoot,"ThetaPost2") , type="eps" )
```

e. Show the summary information of the posterior sample - what is the sample called in the script? Hint: Use `summary()`; `su = summary(.)`; `su$statistics`

The sample is called `codaSamples` in the script.

```
su=summary(codaSamples)
su$statistics
```

```
##          Mean      SD    Naive SE Time-series SE
## biased   0.2727455 0.4453933 0.004453488    0.006038207
## theta[1] 0.5000000 0.0000000 0.000000000    0.000000000
## theta[2] 0.5598235 0.2720647 0.002720375    0.003808949
```

f. What is the posterior probability that the coin is biased?

From the above statistics, we see that posterior probability that the coin is biased is now $\sim 26\%$

4. We will work on the same model as in question 3. This time we will examine the model in light of the theory covered in section 10.3.2.1 page 279 and following. The first thing we will need to do is manipulate the list of data created by the JAGS sampler. Locate the mcmc data which will be in the form of a list.

a. Give the structure of the MCMC data in the file produced by the jags script you made in qu. 3 hint: `str(.)`

```
str(codaSamples)

## List of 3
## $ : 'mcmc' num [1:3334, 1:3] 1 1 0 0 1 1 1 1 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##     .. ..$ : NULL
##     .. ..$ : chr [1:3] "biased" "theta[1]" "theta[2]"
##   ..- attr(*, "mcpair")= num [1:3] 1001 4334 1
## $ : 'mcmc' num [1:3334, 1:3] 0 0 0 0 0 0 0 0 1 1 0 ...
##   ..- attr(*, "dimnames")=List of 2
##     .. ..$ : NULL
##     .. ..$ : chr [1:3] "biased" "theta[1]" "theta[2]"
##   ..- attr(*, "mcpair")= num [1:3] 1001 4334 1
## $ : 'mcmc' num [1:3334, 1:3] 0 0 0 0 0 0 0 0 0 0 1 ...
##   ..- attr(*, "dimnames")=List of 2
##     .. ..$ : NULL
##     .. ..$ : chr [1:3] "biased" "theta[1]" "theta[2]"
##   ..- attr(*, "mcpair")= num [1:3] 1001 4334 1
## - attr(*, "class")= chr "mcmc.list"
```

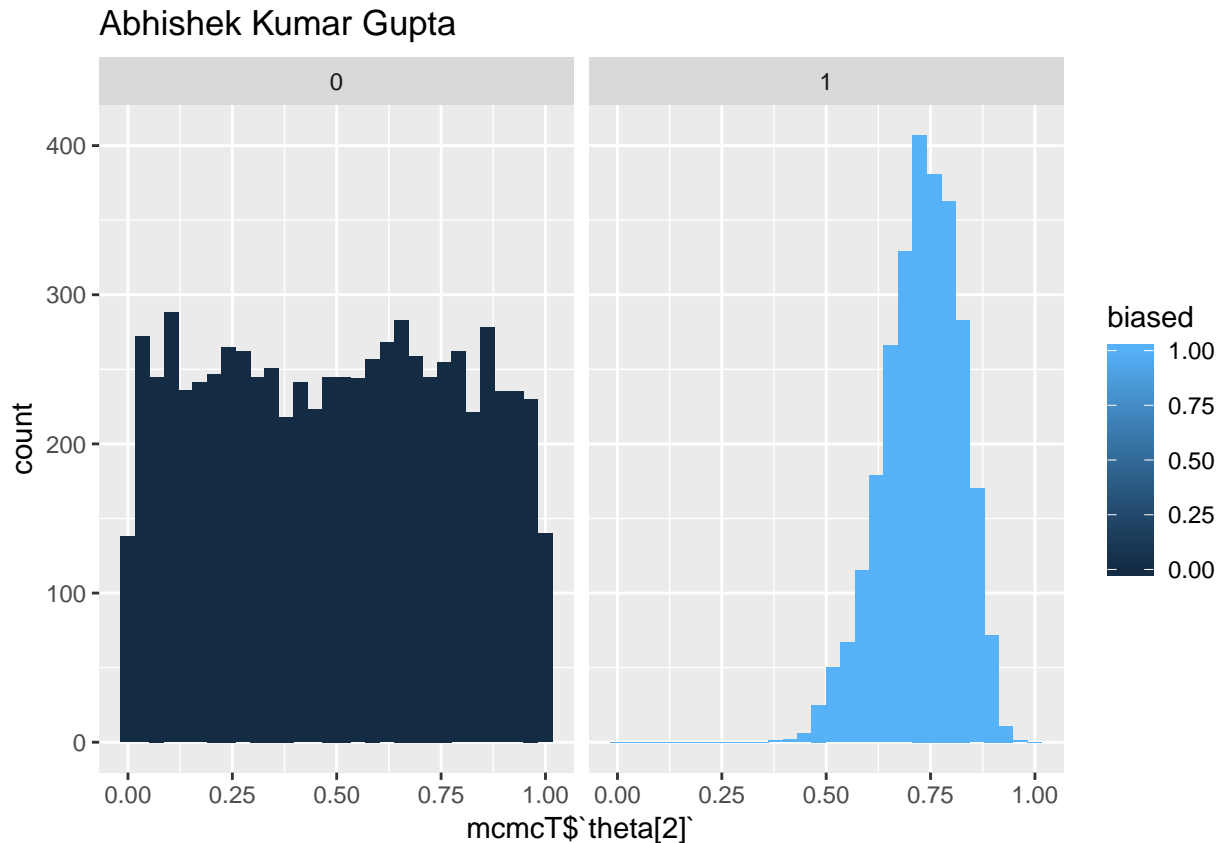
b. Use the following code to make mcmcT by filling in the correct object name

```
mcmc1 = as.data.frame(codaSamples[[1]])
mcmc2 = as.data.frame(codaSamples[[2]])
mcmc3 = as.data.frame(codaSamples[[3]])
mcmcT = rbind.data.frame(mcmc1,mcmc2,mcmc3)
```

c. Using mcmcT and the ggplot2 R package make the following plot after first understanding precisely the pseudo prior method. Make sure you have YOUR name on it!! Colors don't have to precisely correspond.

```
t<- ggplot(data = mcmcT, aes(mcmcT$`theta[2]`,fill = biased)) +geom_histogram(stat = "bin")
t+facet_wrap(~biased)+ labs(title = "Abhishek Kumar Gupta")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



d. Looking at the picture above and considering the model, answer the following:

i. When the parameter $\text{biased} = 0$: $\text{pick} = ?$,

$\text{pick} = 1$

ii. when $\text{biased} = 1$: $\text{pick} = ?$

$\text{pick} = 2$

e. The three variables updated in the Gibbs sampler will go in alphabetical order pick , $\text{theta}[1]$ and then $\text{theta}[2]$ and then cycling around again and again . .

i. When $\text{pick} = 1$, $\text{theta}[1]$ will be sampled from the posterior, then $\text{theta}[2]$ will be sampled from what?

Prior

ii. When $\text{pick} = 2$ what will $\text{theta}[1]$ be sampled from?

Prior

iii. When `pick = 2` what will `theta[2]` be sampled from?

Posterior

iv. Now explain the plot in d) from the above.

This plot explains the observed answers in parts i,ii,iii i.e, when `pick = 1` we have `biased equals 0` which corresponds to the left plot. This indicates that, when `biased=0` `theta[1]` is sampled from posterior and `theta[2]` is sampled from prior as can be seen from the uniform prior in the plot

Similarly, when `pick` is 2, we have `biased 1` which means the right plot. From this, we see that when `pick=2`, `theta[2]` is sampled from posterior as can be seen from the non-uniform distribution and `theta[1]` will be sampled from the prior.

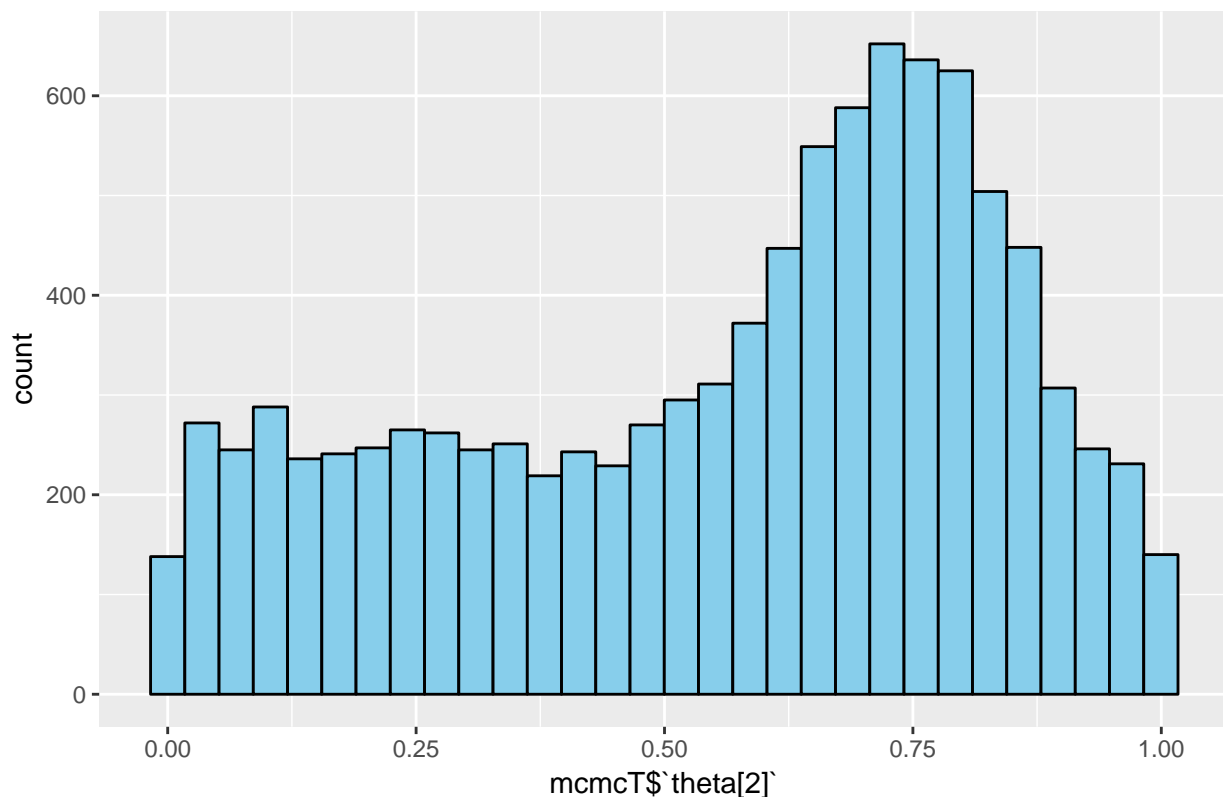
v. Which plot (dark or light) blue would likely represent the true underlying posterior of `theta[2]`

Light blue

vi. Now create the following plot which represents the incorrect posterior of `theta[2]` using `ggplot` - make sure your name is on it.

```
ggplot(data = mcmcT, aes(mcmcT$`theta[2]`))+geom_histogram(stat = "bin", col="black", fill = "skyblue")  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Abhishek Kumar Gupta



vii. Explain the above plot - why does it have the shape it has and then say why it cannot be a true representation of the posterior for theta[2]?

The above plot represents the posterior of theta[2]. It shows the posterior is non uniform and slightly sifted towards right. It can't be true representation of true theta[2] because the model diagnostics shows its having high autocorrelation. This happens because when working for model pick = 1 or biased =0, theta[1] is sampled from posterior but theta[2] is sampled from prior. Similarly, when pick is 2, we have biased 1, theta[2] is sampled from posterior and theta[1] will be sampled from the prior. This issue leads to high autocorrelation. It can be resolved using pseudo priors.

5. Now using the method of pseudo priors recode the model and create a function that will produce mcmc output that will be a better representation of the posterior of theta[2]. Hint: Use a beta pseudo prior for theta[2], the shapes can be calculated using eq. 6.7 pg. 131.

a. Derive the formulae (eq. 6.7) given on pg. 131.

Since, we know mean and standard deviation of beta distribution is given as:

$$mean = \mu = \frac{a}{a+b}$$

$$StandardDeviation = \sigma^2 = \frac{\mu(1-\mu)}{a+b+1}$$

Solving mean, we can write using above equations

$$a+b = a/\mu$$

or,

$$\frac{a+b}{a} = \frac{1}{\mu}$$

after rearranging, we get,

$$b = a * \frac{1-\mu}{\mu}$$

again solving standard deviation, we can write

$$a+b+1 = \frac{\mu(1-\mu)}{\sigma^2}$$

Substituting the value of a+b from above and substituting in the above equation, we get

$$a = \mu * \left(\frac{\mu(1-\mu)}{\sigma^2} - 1 \right)$$

Substituting a in the above relation, we get b

$$b = (1-\mu) * \left(\frac{\mu(1-\mu)}{\sigma^2} - 1 \right)$$

b. You will need to “post-process” the MCMC in order to obtain parameter estimates for the pseudo-prior.

i. Write a function that will create a list of the hyper-parameter estimates using the summary stats from a previous run of the MCMC sampler. Put into the rmd document.

```
pick1 = subset(mcmcT, biased ==0, )
pick2 = subset(mcmcT, biased ==1, )

mu = mean(pick2$`theta[2]`)
sd = sqrt(var(pick2$`theta[2]`))

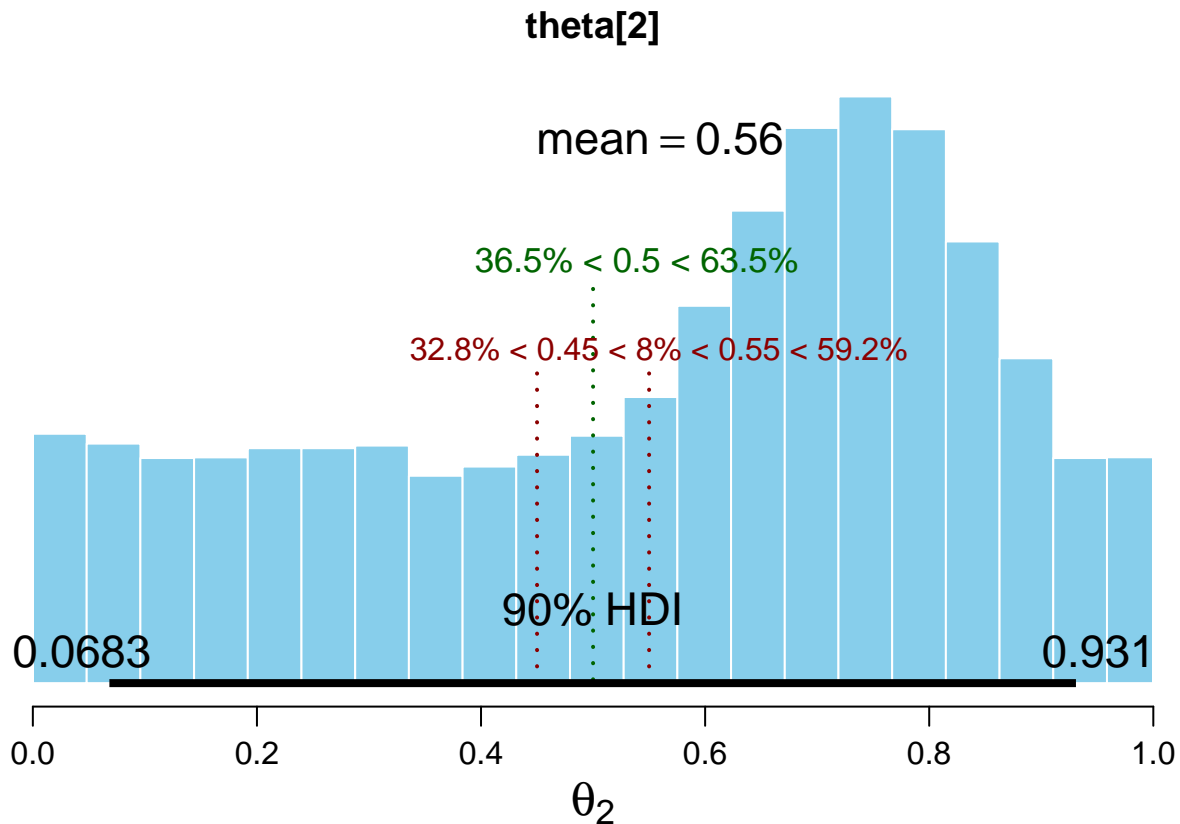
calcab = function(mu,sd)
{
  a = mu*(mu*(1-mu)/(sd^2)-1)
  b = (1-mu)*(mu*(1-mu)/(sd^2)-1)
  list(a,b)
}
```

ii. Give the output of the function for a previous MCMC run of the model (without pseudo priors)

```
calcab(mu,sd)

## [[1]]
## [1] 16.45894
##
## [[2]]
## [1] 6.155807

diagMCMC( codaObject=codaSamples , parName="theta[2]" )
openGraph(height=3,width=4)
  par( mar=c(3.5,0.5,2.5,0.5) , mgp=c(2.25,0.7,0) )
  plotPost( codaSamples[, "theta[2]" ] , main="theta[2]" , xlab=bquote(theta[2]) ,
            cenTend="mean" , compVal=0.5 , ROPE=c(0.45,0.55) , credMass=0.90 )
```



```
##          ESS      mean    median      mode hdiMass    hdiLow
## theta[2] 4707.206 0.5598235 0.6319723 0.7399005    0.9 0.06830903
##          hdiHigh compVal pGtCompVal ROPElow ROPEhigh  pLtROPE
## theta[2] 0.9309631    0.5  0.634973  0.45    0.55 0.3279344
##          pInROPE  pGtROPE
## theta[2] 0.08028394 0.5917816
```

iii. What part of the MCMC chain did you use (look at the picture below)?

In this case left part i.e. dark blue one part of the curve has been used.

c. Copy and paste your new pseudo prior JAGS model (not all the script JUST the model as in qu 3)

```
modelString = "
model {
  x ~ dbin(p,n)
  p <- theta[pick]
  #theta[1] represents fixed unbiased value
  theta[1] <- equals(pick,1)*fixtheta1 + equals(pick,2)*fixtheta1
  theta[2] <- equals(pick,1)*pseudotheta2 + equals(pick,2)*theta2
  pick ~ dcat(q[]) # categorical 1 produced prob q[1], etc
  # pick is 2 if biased 1 unbiased
  q[1]<-0.9
  q[2]<-0.1
  fixtheta1 <-0.5 # unbiased; fixed probability
```

```

theta2 ~ dunif(0,1) #true prior of theta[2]
#shape parameters for pseudo prior will be part of dataList
pseudotheta2 ~ dbeta(15.92,5.94)
biased <- pick - 1
}
" # close quote for modelString

```

d. Make a function `pseudobin()` that will run the script - the function should produce

i. a ggplot of the posterior sample of `theta[2]` with `fill = biased`, make sure your name is on the title (`ggtitle()`)

```

pseudobin = function()
{
  source("DBDA2E-utilities.R") # Must be in R's current working directory.
  require(rjags)               # Must have previously installed package rjags.

  fileNameRoot="Jags-Ass2e" # For output file names.

  modelString = "
  model {
    x ~ dbin(p,n)
    p <- theta[pick]
    #theta[1] represents fixed unbiased value
    theta[1] <- equals(pick,1)*fixtheta1 + equals(pick,2)*fixtheta1
    theta[2] <- equals(pick,1)*pseudotheta2 + equals(pick,2)*theta2
    pick ~ dcat(q[]) # categorical 1 produced prob q[1], etc
    # pick is 2 if biased 1 unbiased
    q[1]<-0.9
    q[2]<-0.1
    fixtheta1 <-0.5 # unbiased; fixed probability
    theta2 ~ dunif(0,1) #true prior of theta[2]
    #shape parameters for pseudo prior will be part of dataList
    pseudotheta2 ~ dbeta(15.92,5.94)
    biased <- pick - 1
  }
  " # close quote for modelString

  writeLines( modelString , con="TEMPmodel.txt" )
  data.List = list(x= 15, n=20)
  initsList = list(pick=1)

  # Run the chains:
  jagsModel = jags.model( file="TEMPmodel.txt" , data=data.List ,
                          n.chains=3 , n.adapt=500 )
  update( jagsModel , n.iter=500 )
  codaSamples2 = coda.samples( jagsModel , variable.names=c("biased","theta[1]","theta[2]") ,
                              n.iter=3334 )
  save( codaSamples , file=paste0(fileNameRoot,"Mcmc.Rdata") )
  p1<- plotPost( codaSamples[, "theta[2]" ] , main="theta[2]" , xlab=bquote(theta[2]) )

```



```

mcmc1 = as.data.frame(codaSamples2[[1]])
mcmc2 = as.data.frame(codaSamples2[[2]])
mcmc3 = as.data.frame(codaSamples2[[3]])
mcmcT = rbind.data.frame(mcmc1,mcmc2,mcmc3)

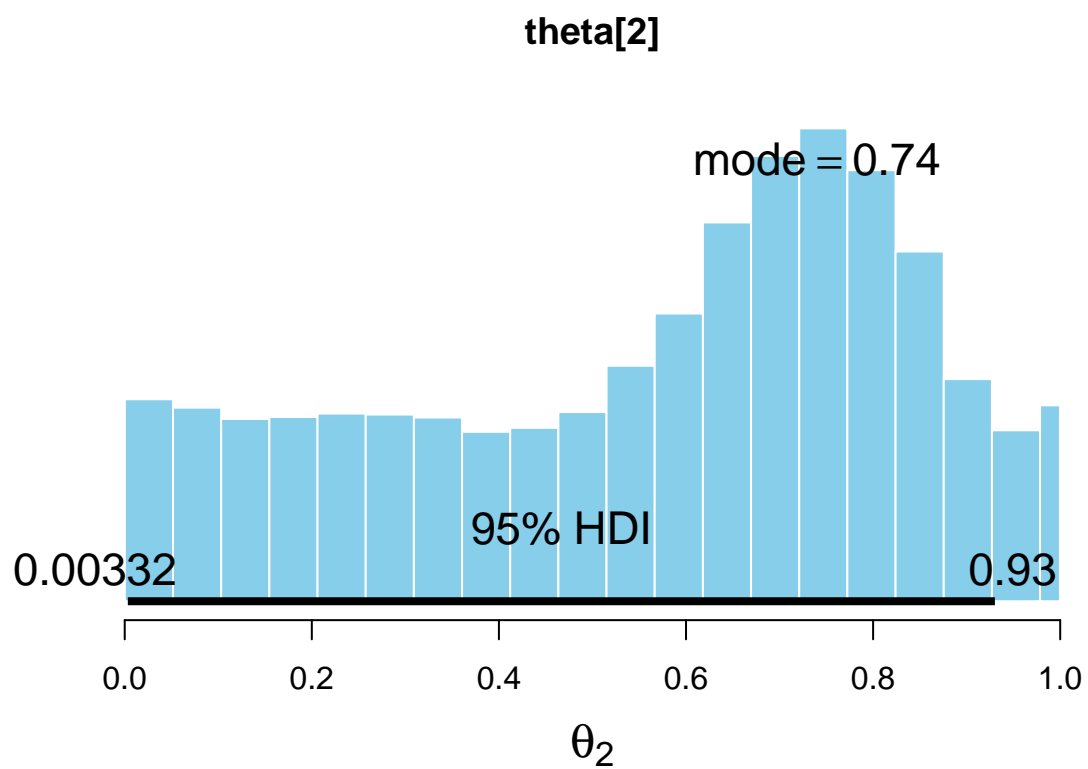
library(ggplot2)
p2<- ggplot(data = mcmcT, aes(mcmcT$`theta[2]`,fill = biased)) +geom_histogram(stat = "bin") + fac

p3<- ggplot(data = mcmcT, aes(mcmcT$biased)) +geom_histogram(stat = "bin")+ labs(title = "Abhishek Ku
su<- summary(codaSamples2)
stats<- su$statistics
return(list(d.i = p2, d.ii = p1, d.iii = p3,stats = stats))
}

pseudobin()

##
## *****
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *****
##
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 1
##   Unobserved stochastic nodes: 3
##   Total graph size: 24
##
## Initializing model

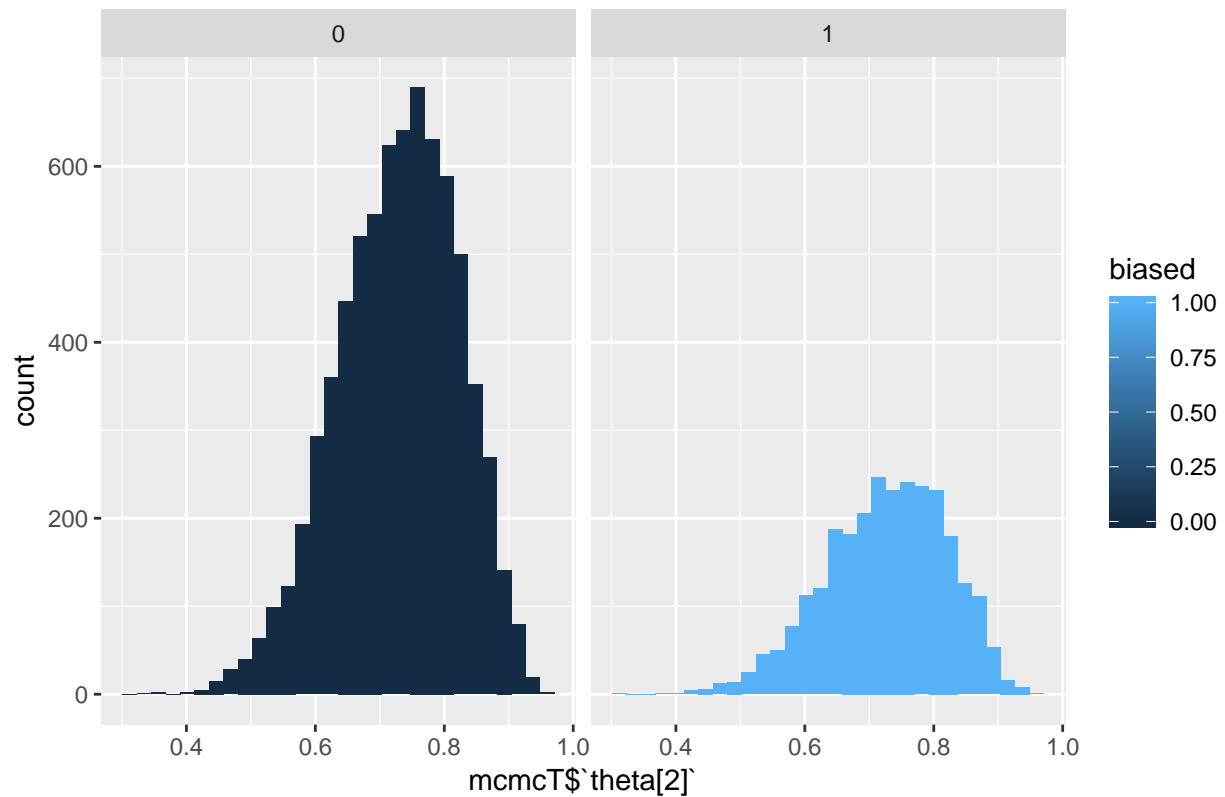
```



```
## $d.i
```

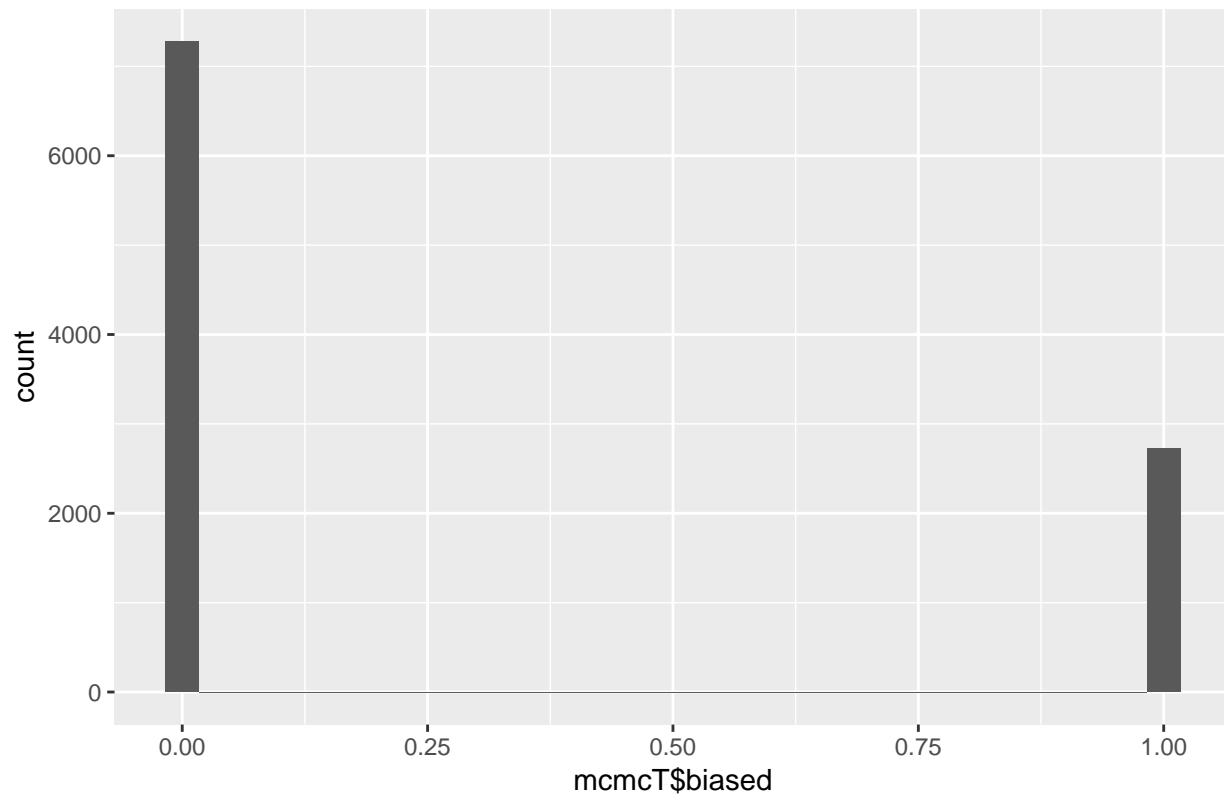
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Abhishek Kumar Gupta



```
##
## $d.ii
##           ESS      mean    median      mode hdiMass      hdiLow
## theta[2] 4707.206 0.5598235 0.6319723 0.7399005    0.95 0.003315618
##           hdiHigh compVal pGtCompVal ROPElow ROPEhigh pLtROPE pInROPE
## theta[2] 0.9301776      NA          NA      NA      NA      NA      NA
##           pGtROPE
## theta[2]      NA
##
## $d.iii
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Abhishek Kumar Gupta



```
##
## $stats
##           Mean          SD      Naive SE Time-series SE
## biased    0.2725455 0.4452912 0.0044524670 0.006368744
## theta[1] 0.5000000 0.0000000 0.0000000000 0.000000000
## theta[2] 0.7275105 0.0935322 0.0009352285 0.001075217
```

e. Looking at the above plot and the model used to make it - why are there more counts in the first facet than the second?

This is because the first facet also contains the samples of pseudo priors along with the posterior while the right one has no pseudo priors and contain only posterior samples.