Problem1: BDA 2.11 a & 4.1

#Rcode

y = c(-2, -1, 0, 1.5, 2.5)

# θ uniform on [−4, 4].

theta = seq(-4,4,by=0.001) #theta grid

# Unormalized Posterior function

unnorm\_posterior <- function (theta) {

unnorm\_post = prod(1/(1+(y-theta)^2))

return(unnorm\_post)

}

m = length(theta)

# Unnormalized Posterior grid

unnorm\_poster <- matrix(0,m,1)

for (i in 1:m)

unnorm\_poster[i] <- unnorm\_posterior(theta[i])

# Normalized Posterior grid

prob\_y = integrate(Vectorize(unnorm\_posterior),-Inf,Inf)

norm\_post <- unnorm\_poster/prob\_y$value

**# 2.11 a**

plot(theta, norm\_post, title('Normalized Posterior Density Function'), xlab = 'Theta', ylab = 'p(Theta|Y)', col = 'red')



**# 4.1 b**

# First Derivative of log Posterior function

FirstDeriv\_lPost <- function(theta) {

first\_Deriv <- 2 \* sum((y-theta)/(1+(y-theta)^2))

return(first\_Deriv)

}

# Iteratively solving for Posterior Mode

NewtonRhapson <- uniroot(FirstDeriv\_lPost, c(-4,4))

Posterior\_Mode <- NewtonRhapson$root #

Posterior Mode: 0.08759136

**## 4.1 c**

# Fisher Information (Posterior Mode) : I\_PostMode

I\_PostMode\_inv = 1/(2 \* sum ((1-(y-Posterior\_Mode)^2)/(1+(y-Posterior\_Mode)^2)^2))

Norm\_Approx <- dnorm(theta, mean = Posterior\_Mode, sd = sqrt(I\_PostMode\_inv))

plot(theta, Norm\_Approx, title('Normal Approximation of Unnormalized Posterior Density'), xlab = 'Theta', ylab = 'Norm Approx: p(Theta|Y)', col = 'blue')



Normal Approximation is quite close to the Normalized Posterior Density, with mass of Normal approximation concentrated slightly more at the Mode rather than on the right side (positive theta values) as seen in the Normalized Posterior Density plot

Problem 2: BDA 5.13

#Rcode

library(rootSolve)

y <- c(16, 9, 10, 13, 19, 20, 18, 17, 35, 55)

n <- c(74, 99, 58, 70, 122, 77, 104, 129, 308, 119)

**# 5.13b**

## Estimate theta from data (Bin distribution)

theta\_hat <- y/n

estimate\_mean <- mean(theta\_hat) #0.1961412

estimate\_var <- var(theta\_hat) #0.01112067

Estimates: Mean & Variance of Theta: [Mean: 0.1961412, Var: 0.01112067]

## Estimate alpha and beta from estimated mean and var for theta (Beta Distribution)

#par[1] is alpha and par[2] is beta

alpha\_beta\_Estimate <- function(par) {

diff <- numeric(2)

diff[1] <- par[1]/(par[1]+par[2]) - mean(y/n)

diff[2] <- par[1]\*par[2]/(((par[1]+par[2])^2)\*(par[1]+par[2]+1)) - sd(y/n)^2

diff

}

Estimate <- multiroot(alpha\_beta\_Estimate,c(1,1))

alpha\_estimate <- round(Estimate$root[1],1)

beta\_estimate <- round(Estimate$root[2],1)

Estimates: Alpha & Beta: (2.6,10.6)

## Creating alpha-beta grid around the estimates from above

marginal\_poster <- function(alpha, beta) {

post <- 1

for (i in 1:length(y)) {

if (n[i] > 100) n[i] = 100

post = post \* ( ( ( gamma(alpha + beta) ) / ( gamma(alpha) \* gamma(beta) ) ) \*

( ( gamma(alpha + y[i] ) \* gamma(beta + n[i] - y[i]) ) / ( gamma(alpha + beta + n[i]) ) ) )

}

# The hyper prior is defined below

Hyper\_prior(alpha,beta) \* post

}

Hyper\_prior <- function(alpha,beta) {

alpha \* beta \* (alpha + beta)^(-5/2)

}

axis\_x <- seq(log(alpha\_estimate/beta\_estimate)\*1.5,

log(alpha\_estimate/beta\_estimate)/1.5,length.out =151) # log(alpha/beta)

axis\_y <- seq(log(alpha\_estimate+beta\_estimate)/2.5,

log(alpha\_estimate+beta\_estimate)\*1.5,length.out =151) # log(alpha + beta)

beta <- exp(axis\_y)/(exp(axis\_x)+1)

alpha <- exp(axis\_y+axis\_x)/(exp(axis\_x)+1)

log\_marginal <- function(x1,x2){

log(marginal\_poster(x1, x2))

}

posterior\_dens <- outer(alpha,beta,log\_marginal)

posterior\_dens <- exp(posterior\_dens - max(posterior\_dens))

posterior\_dens <- posterior\_dens/sum(posterior\_dens)

contours <- seq(min(posterior\_dens), max(posterior\_dens), length=10)

contour(axis\_x, axis\_y, posterior\_dens,levels=contours, xlab=expression( log(alpha/beta) ),

ylab=expression( log(alpha+beta) ), xlim=c( min( axis\_x ), max( axis\_x ) ) ,

ylim=c( min( axis\_y ), max( axis\_y ) ),

drawlabels=FALSE, main="Joint posterior density: p(alpha,beta|y)")



## Draw Samples from p(alpha,beta|y)

samples <- 1000

# Sum over all beta to get the marginal of alpha

marginal\_alpha\_dens <- apply(posterior\_dens ,1, sum)

# sample\_log\_x: log(alpha/beta)

sample\_log\_x <- sample(axis\_x, samples, replace=TRUE, prob = marginal\_alpha\_dens)

# Compute conditional probability (p(beta|alpha))

conditional\_prob\_beta <- function(x)

{

posterior\_dens[which(axis\_x == sample\_log\_x[x]),]

}

# Sample beta according the the conditional probatility above

#sample\_log\_y: log(alpha + beta)

sample\_log\_y <- sapply(1:samples,function(x) sample(axis\_y,1,replace=TRUE,prob=conditional\_prob\_beta (x)))

# Add a uniform random jitter centered at zero with width equal to the grid spacing to make

# simulation draws more continuous.

grid.alpha <- axis\_x[2] - axis\_x[1]

grid.beta <- axis\_y[2] - axis\_y[1]

sample\_log\_y <- sample\_log\_y + runif(length(sample\_log\_y),-grid.beta/2,grid.beta/2)

sample\_log\_x <- sample\_log\_x + runif(length(sample\_log\_x),-grid.alpha/2,grid.alpha/2)

# Plot the sampled values

points(sample\_log\_x, sample\_log\_y,col = 'red', xlab=expression( log(alpha/beta)^s ),

ylab=expression( log(alpha+beta)^s ), xlim=c( min(axis\_x) , max(axis\_x) ) ,

ylim=c( min(axis\_y) , max(axis\_y) ),

main="Sample Draws of log(alpha/beta) and log(alpha+beta)")



sample\_beta <- exp( sample\_log\_y ) / ( exp(sample\_log\_x)+1 )

sample\_alpha <- exp( sample\_log\_y + sample\_log\_x ) / ( exp(sample\_log\_x)+1 )

**## 5.13c**

# For each draw of hyper-parameters, draw a sample of θ from p(θ|alpha,beta,y)

theta\_dist <- sapply(1:10, function(x) rbeta(1000, sample\_alpha +y[x], sample\_beta + n[x] - y[x]))

theta\_dist <- apply(theta\_dist,2,sort)

plot(0:600/1000, 0:600/1000, type="l", xlab="Observed rate",

ylab="95% CI and median of posterior")

jitter.x <- y/n + runif(length(y),-0.01,0.01)

points(jitter.x, theta\_dist[500,])

segments(jitter.x,theta\_dist[25,], jitter.x,theta\_dist[975,] )

title(main="Posterior Distribution of Bike rates for all 10 streets")



**## 5.13d**

# 1000 draws from Beta(sample\_alpha,sample\_beta):

sample\_theta <- rbeta(1000, shape1 =sample\_alpha , shape2 = sample\_beta)

CI <- round(sample\_theta[order(sample\_theta)][c(25,975)],2)

The posterior interval for θ\_estimate = (0.02, 0.46)

**5.13e**

The posterior interval predicts with 95% confidence that Theta lies between 1 and 48. This interval is too large and hence not very informative.

**5.13f**

Posterior estimates did not show much shrinkage, hence the beta assumption might not be very reasonable.