June-2021-Exam

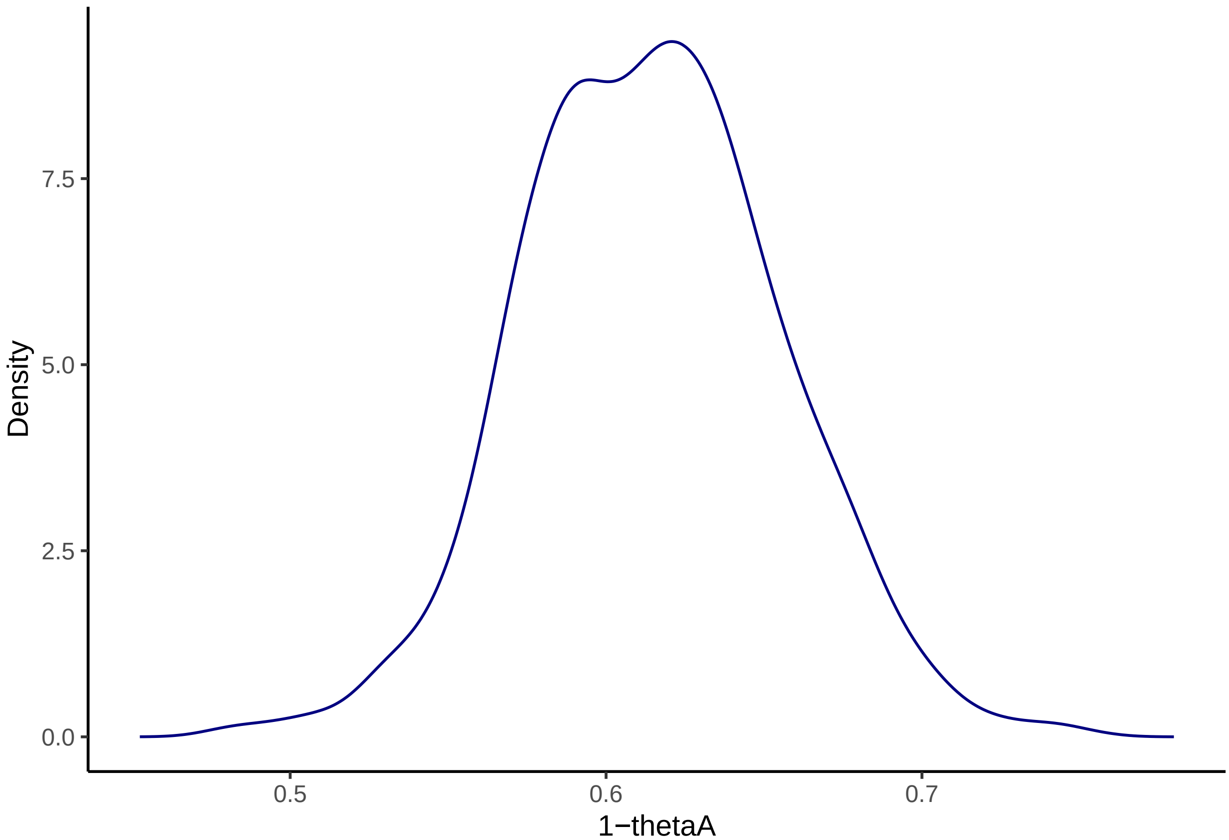
Christophoros Spyretos

Exercise 1 *Customers' choice of brands*

```
n <- 100
s <- 38
f <- 62
a <- 16
b <- 24
```

Task 1

```
set.seed(12345)

#generates 1,000 random deviates.
theta_A <- rbeta(n = 1000, shape1 = a+s, shape2 = b+f)

#exact posterior prob
#pbeta the distribution function
thetaA_prob <- pbeta(q = 0.4, shape1 = a+s, shape2 = b+f, lower.tail = FALSE)

dens_one_minus_thetaA <- density(1 - theta_A)

df_plot_density <- data.frame("x" = dens_one_minus_thetaA$x, "y"= dens_one_minus_thetaA$y)

ggplot(df_plot_density) +
  geom_line(aes(x=x, y=y),color = "navy") +
  xlab("1-thetaA") +
  ylab("Density") +
  theme_classic()
```



The posterior probability of $\theta_A > 0.4$ equals approximately 0.36.

Task 2

```
ratio <- (1-theta_A)/theta_A
interval <- quantile(ratio,probs = c(0.025,0.975))

#table for the interval
df_intervals <- data.frame("lower_bound" = interval[1], "upper_bound" = interval[2])
colnames(df_intervals) <- c("lower bound", "upper bound")
rownames(df_intervals) <- c("95% Equal Tail Credible Interval")
knitr::kable(df_intervals)
```

	lower bound	upper bound
95% Equal Tail Credible Interval	1.14581	2.272752

Task 3

```
beta(a+s,b+f)/beta(a,b)
```

```
## [1] 7.556771e-30
```

Task 4

```
set.seed(12345)
counts <- c(38,27,35)
c <- 2
a <- c*c(1,1,1)

N <- 1000
xDraws <- matrix(0,N,length(a))
thetaDraws <- matrix(0,N,length(a))

for (i in 1:length(a)){
  xDraws[,i] <- rgamma(N,a[i]+counts[i])
}

for (i in 1:N){
  thetaDraws[i,] <- xDraws[i,]/sum(xDraws[i,])
}

mean_val <- mean(thetaDraws[,1] > thetaDraws[,3])
```

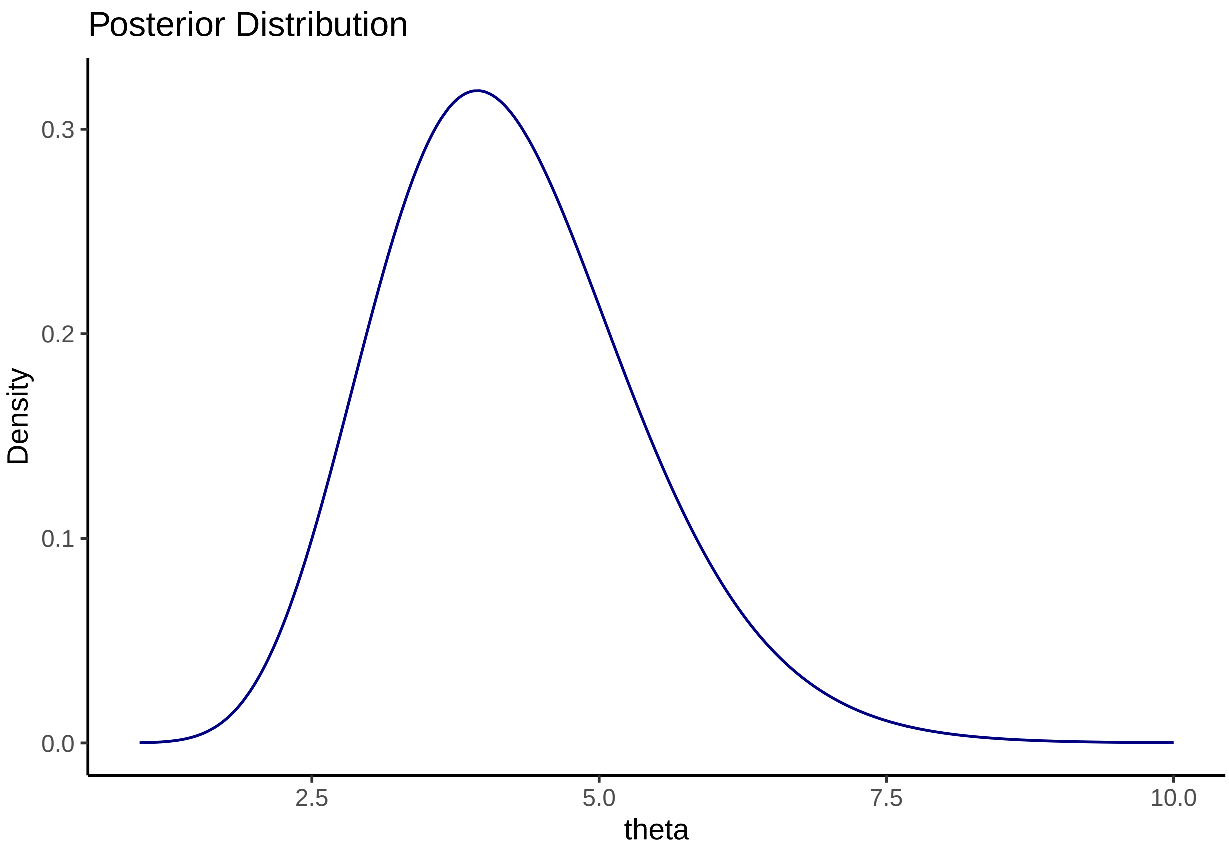
The posterior probability is 0.656.

Exercise 2

Task a,b and c are hand written solutions.

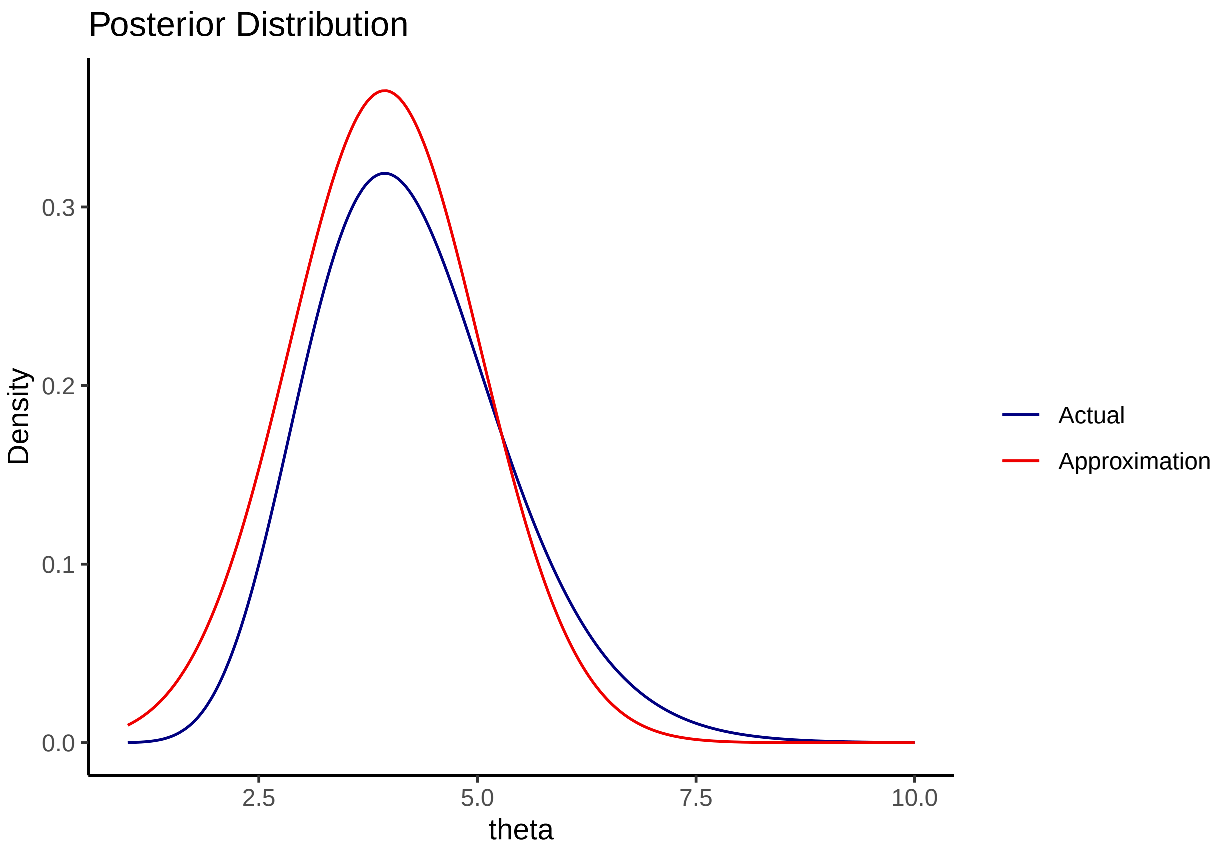
Task d

```
LogPost <- function(theta,n, sum_x2){

  logLik <- n*log(theta) - theta*sum_x2
  logPrior <- -0.5 * theta

  return(logLik + logPrior)
}

theta <- runif(1000,1,10)
n <- 13
sum_x2 <- 2.8

post_dens <- exp(LogPost(theta,n, sum_x2))

#normalise posterior density

post_dens <- post_dens/(0.01 * sum(post_dens))

df_plot <- data.frame("theta" = theta, "posterior" = post_dens)

ggplot(df_plot) +
  geom_line(aes(x=theta, y=posterior), color ="navy") +
  ggtitle("Posterior Distribution") +
  ylab("Density") +
  theme_classic()
```



Task e

```
OptimRes <- optim(3,LogPost,gr=NULL,n,sum_x2,method=c("BFGS"),control=list(fnscale=-1),hessian=TRUE)

normal_approx <- dnorm(theta, OptimRes$par, sqrt(diag(-solve(OptimRes$hessian))))

df_plot$approximation <- normal_approx

ggplot(df_plot) +
  geom_line(aes(x=theta, y=posterior, color ="navy")) +
  geom_line(aes(x=theta, y=approximation, color = "red2")) +
  theme(legend.position="right") +
  scale_color_manual(values=c('navy','red2'),
                     name = "",
                     labels = c("Actual","Approximation")) +
  ggtitle("Posterior Distribution") +
  ylab("Density") +
  theme_classic()
```



The posterior approximation is quite accurate, but the actual posterior distribution is skewed to the right.

Exercise 3

```
#source("ExamData.R")
```

Task a

```
#BayesLinReg <- function(y, X, mu_0, Omega_0, v_0, sigma2_0, nIter)

mu_0 <- as.vector(rep(0,7))
Omega_0 <- (1/25) * diag(7)
sigma2 <- 4
nIter <- 1000

#sample <- BayesLinReg(y, X, mu_0, Omega_0, v_0, sigma2_0, nIter)
```