June-2021-Exam

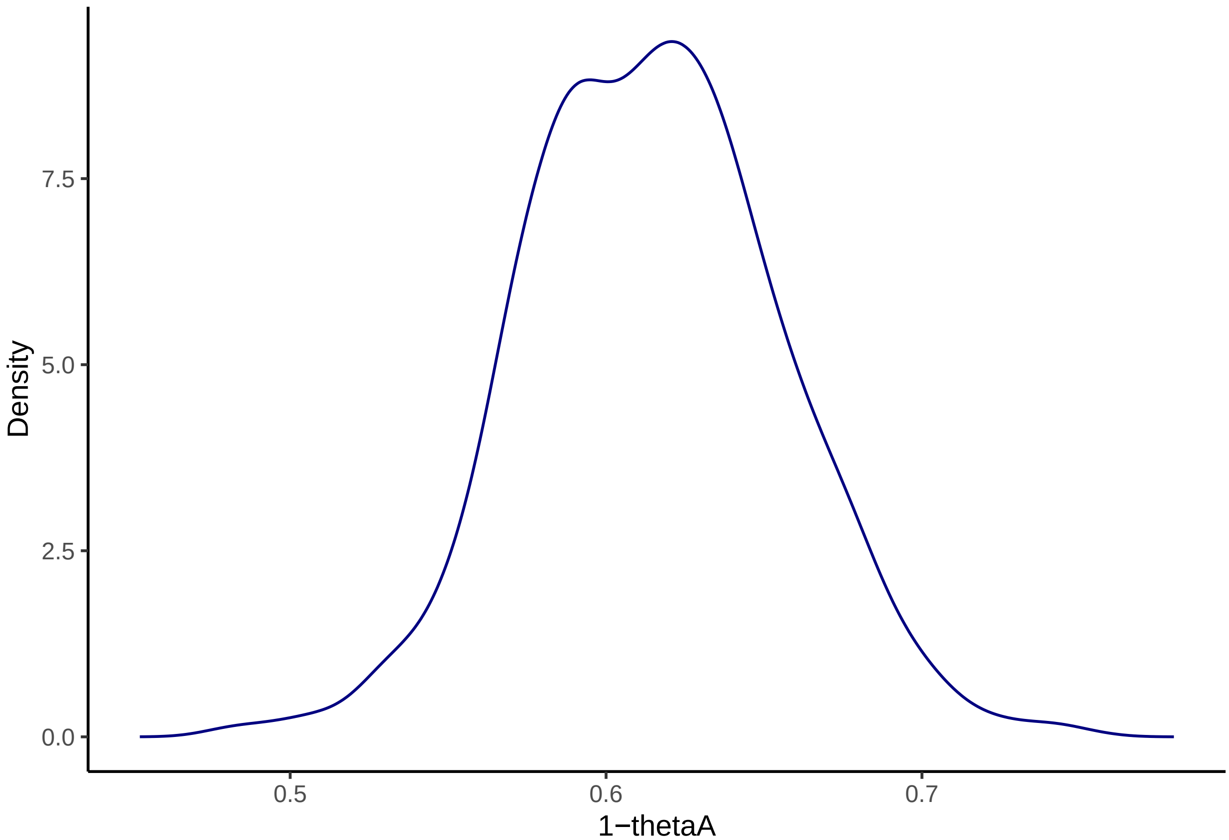
Christophoros Spyretos

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

```r
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

The ratio is the odds of not choosing brand A, i.e. it describes how many more times likely it is to not choose brand A compared to choosing brand A. The credible interval shows the values of the ratio with 95 % probability.

Task 3

```r
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], rate = 1)
}

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

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

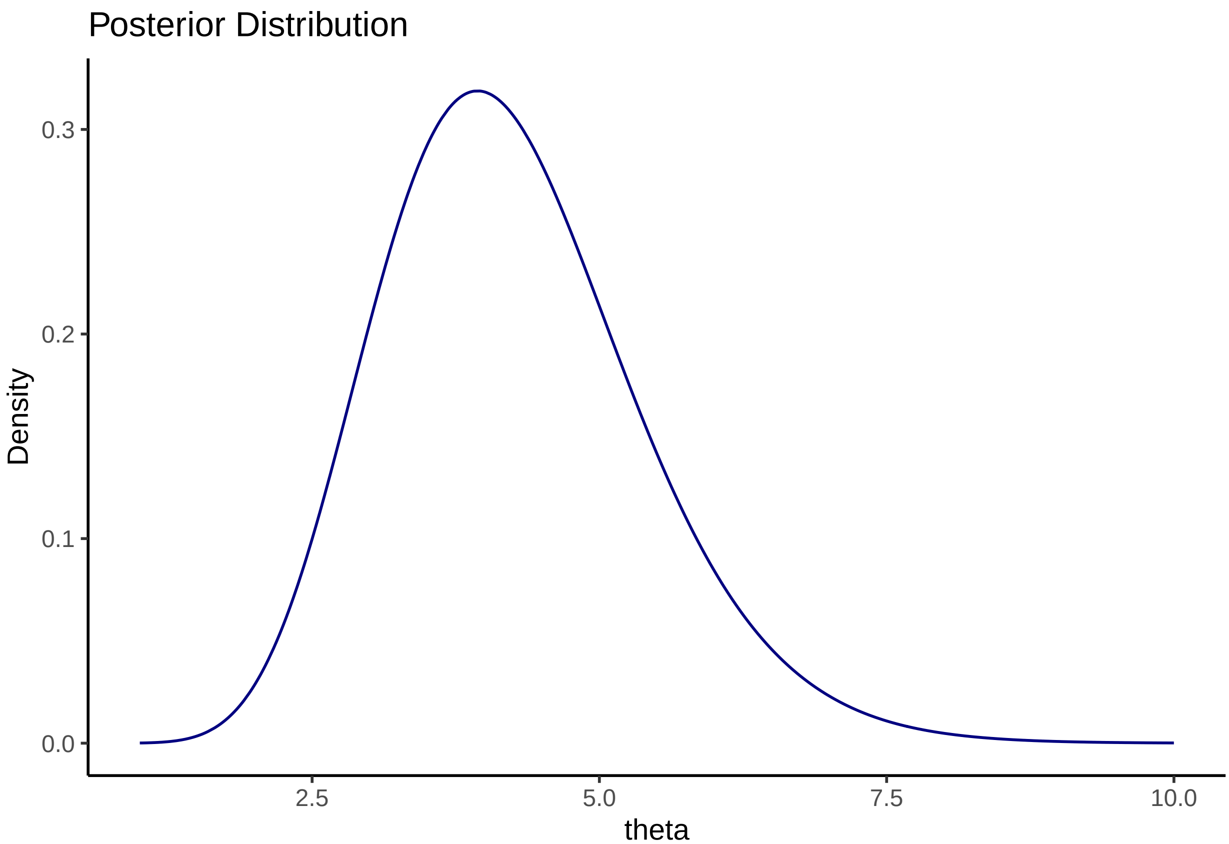
The posterior probability is 0.656.

Exercise 2

Task a,b and c are hand written solutions.

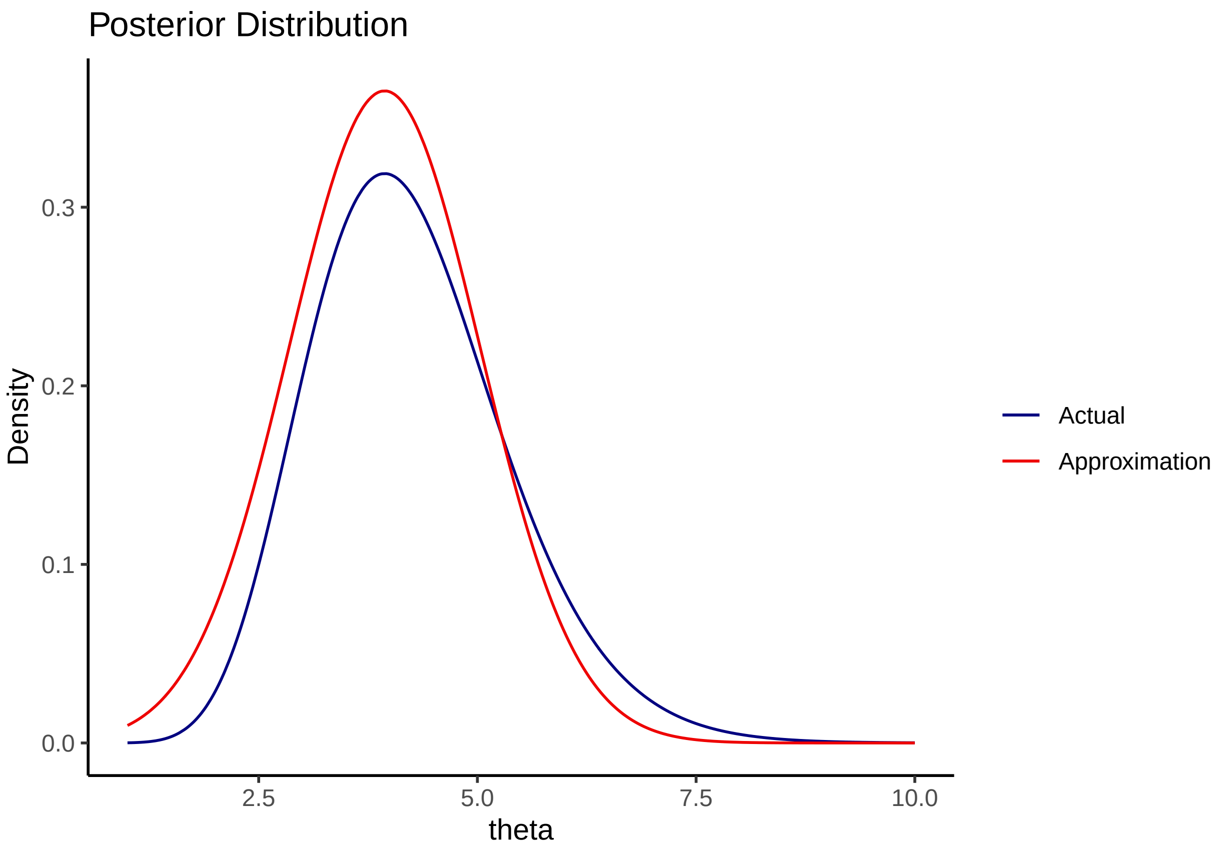
Task d

```
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)
library(mvtnorm)

mu_0 <- rep(0,7)
Omega_0 <- 1/25 * diag(7)
v_0 <- 1
sigma2_0 <- 4
nIter <- 1000

PostDraws <- BayesLinReg(y, X, mu_0, Omega_0, v_0, sigma2_0, nIter)

Betas <- PostDraws$betaSample

BetasMean <- colMeans(Betas)

intervals <- matrix(NA, nrow = 7, ncol = 2)

for (i in 1:ncol(Betas)){
```

```
  intervals[i,] <- quantile(Betas[,i], probs = c(0.025, 0.975))
}

colnames(intervals) <- c("2.5%","97.5")
rownames(intervals) <- c("b0","b1","b2","b3","b4","b5","b6")
knitr::kable(intervals)
```

	2.5%	97.5
b0	1.1529187	1.4674344
b1	0.5377850	0.8711292
b2	0.0499087	0.2702009
b3	0.0225842	0.8239922
b4	-0.3683783	0.0361629
b5	-0.2882946	0.4179264
b6	-0.4500895	-0.0432380

Task b

```
Sigma2 <- PostDraws$sigma2Sample
Sigma2Median <- median(sqrt(Sigma2))
```

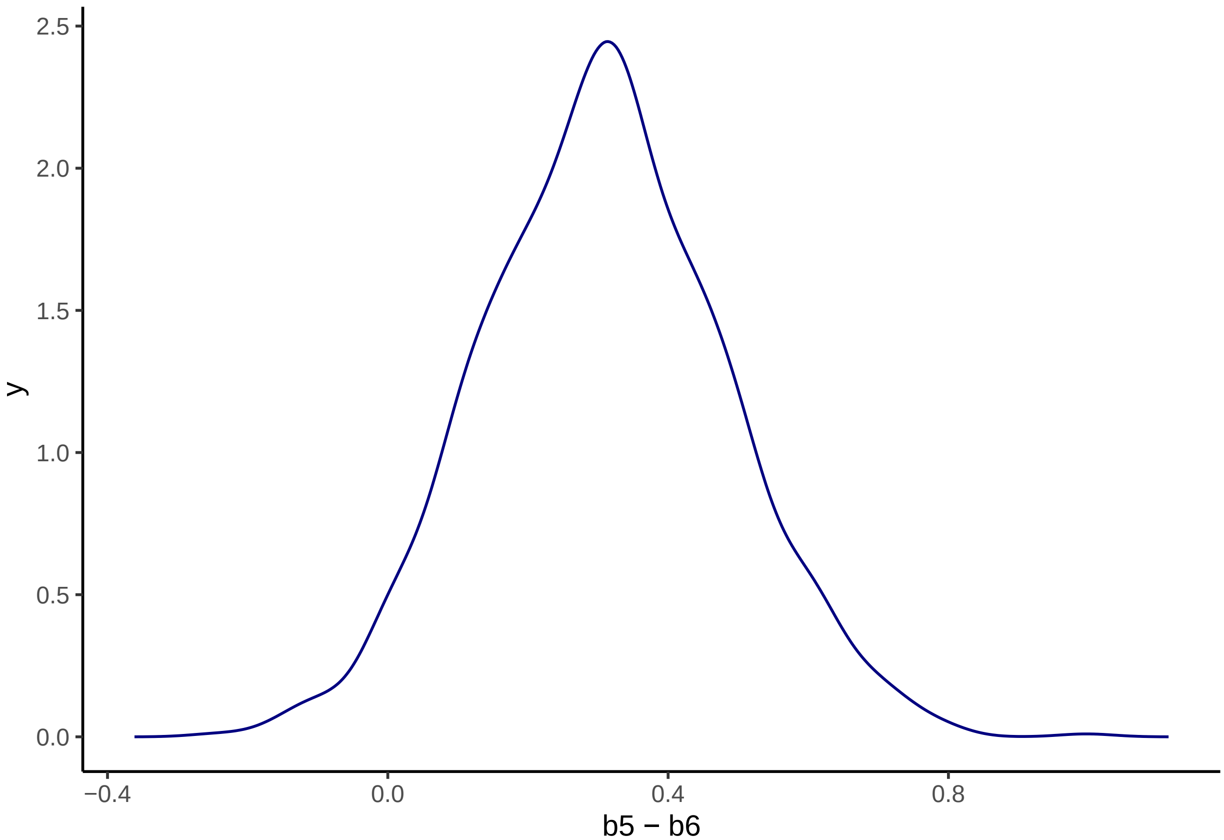
The posterior median of the standard deviation σ is approximately 0.64.

Task c

```
Effect_B <- Betas[,2] + Betas[,6]
Effect_C <- Betas[,2] + Betas[,7]

Diff <- Effect_B - Effect_C

DensDiff <- density(Diff)

df_DensDiff <- data.frame("x" = DensDiff$x, "y" = DensDiff$y)

DiffInterval <- quantile(Diff,probs=c(0.025,0.975))

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

	lower bound	upper bound
95% Equal Tail Credible Interval	-0.0108499	0.6558339

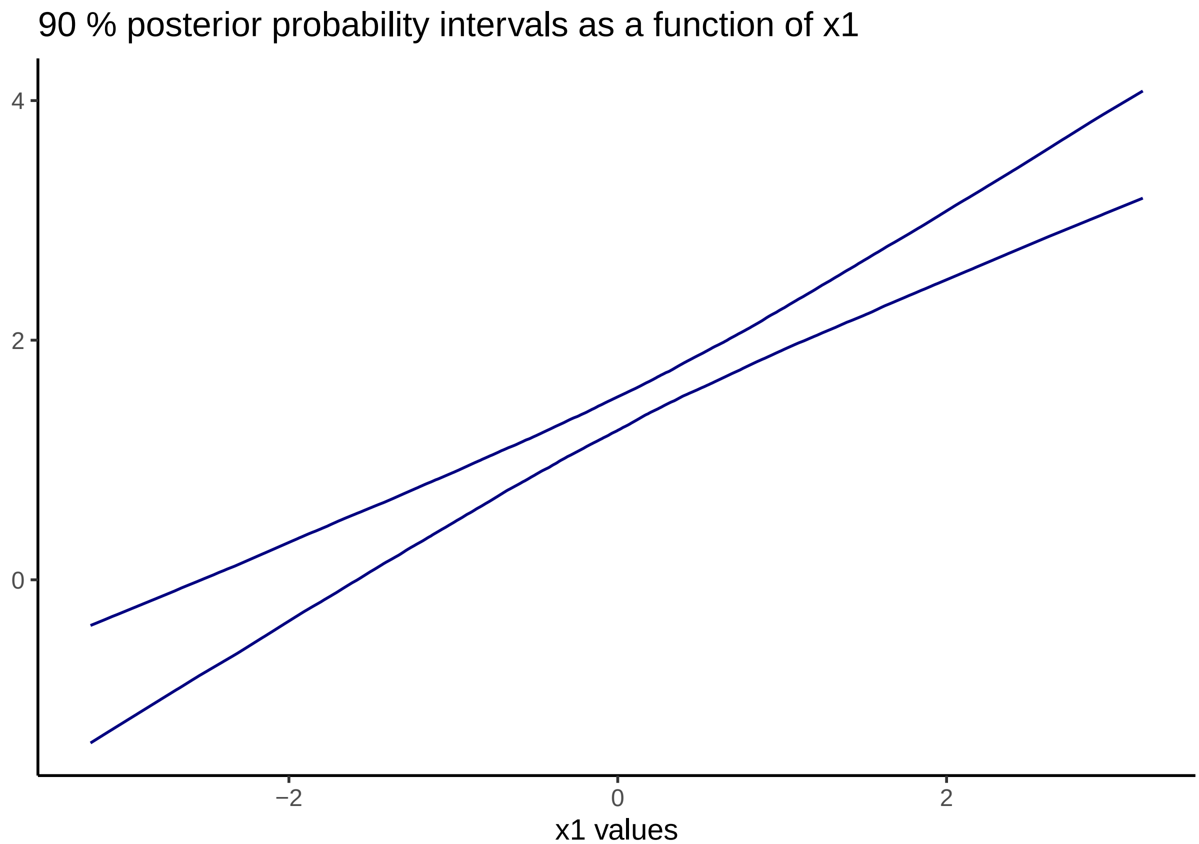
```
ggplot(df_DensDiff) +
  geom_line(aes(x=x, y=y), color = "navy") +
  xlab("b5 - b6") +
  theme_classic()
```



There is substantial probability mass that the effect on y from x1 is larger in high school B compared to high school C. However, the 95 % equal tail credible interval for the difference of the slopes of x1 between the high schools reveals that the difference can be either negative or positive. Hence, the probability is not that high that this effect in high school B is larger than in high school C.

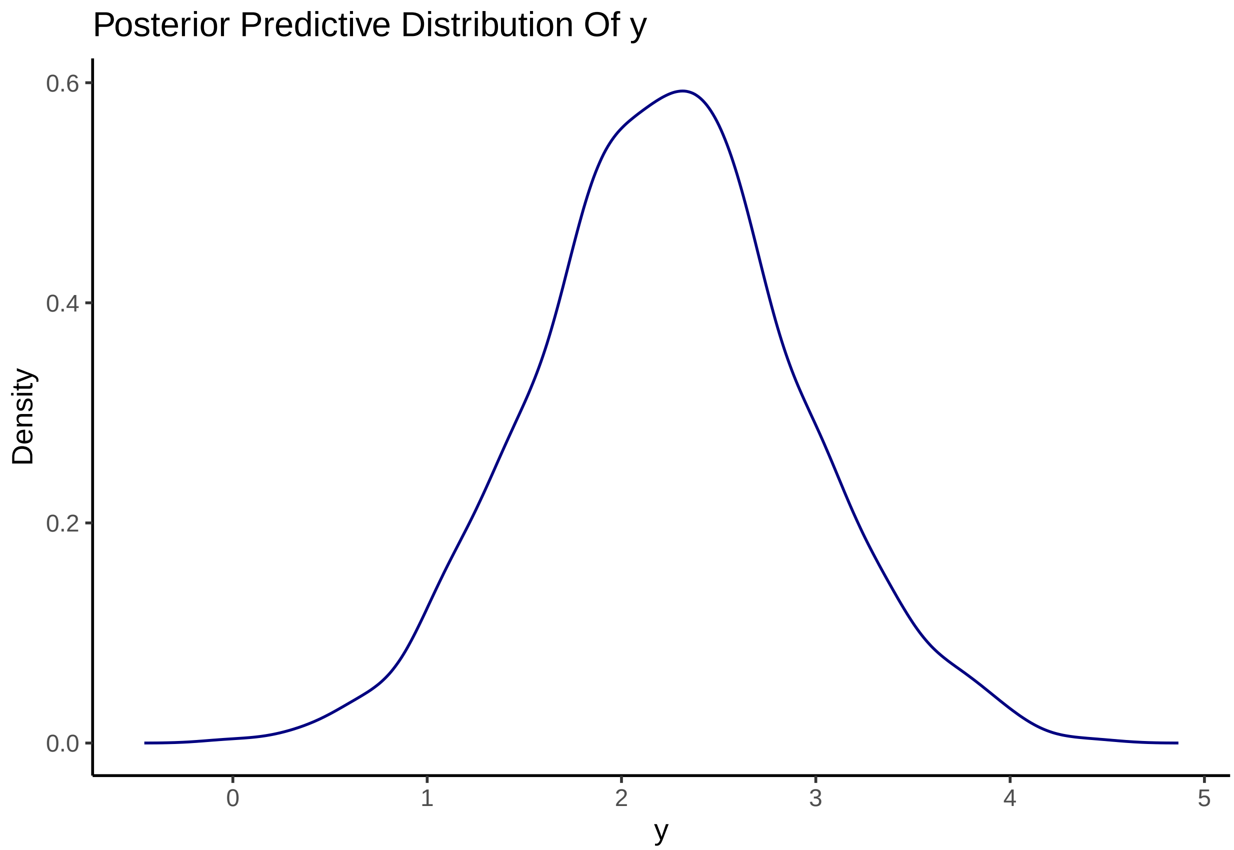
Task d

```
x1_grid <- seq(min(X[,2]), max(X[,2]), 0.01)
ExpectedMu <- matrix(NA, nrow = length(x1_grid), ncol = 2)

for (i in 1:length(x1_grid)){
  Mu <- Betas[,1] + Betas[,2]*x1_grid[i] + Betas[,3]*0.5
  ExpectedMu[i,] <- quantile(Mu, probs = c(0.05,0.95))
}

df_ExpectedMu <- data.frame("x1"= x1_grid,
                            "lower" = ExpectedMu[,1],
                            "upper" = ExpectedMu[,2])

ggplot(df_ExpectedMu) +
  geom_line(aes(x=x1, y=lower),color = "navy") +
  geom_line(aes(x=x1, y=upper),color = "navy") +
  ggtitle("90 % posterior probability intervals as a function of x1") +
  xlab("x1 values") +
  ylab("") +
  theme_classic()
```



Task e

```
Mu <- Betas[,1] + Betas[,2]*0.4 + Betas[,3] + Betas[,4] + Betas[,6]
N <- 1000

y <- rnorm(N, mean = Mu, sd = sqrt(Sigma2))

yDens <- density(y)

df_y <- data.frame("x"=yDens$x, "y"=yDens$y)

ggplot(df_y) +
  geom_line(aes(x=x,y=y), color = "navy") +
  ggtitle("Posterior Predictive Distribution Of y") +
  xlab("y") +
  ylab("Density") +
  theme_classic()
```


Posterior Predictive Distribution Of y
Density
0.6
0.4
0.2
0.0
0
1
2
3
4
5
y