# BayesExam

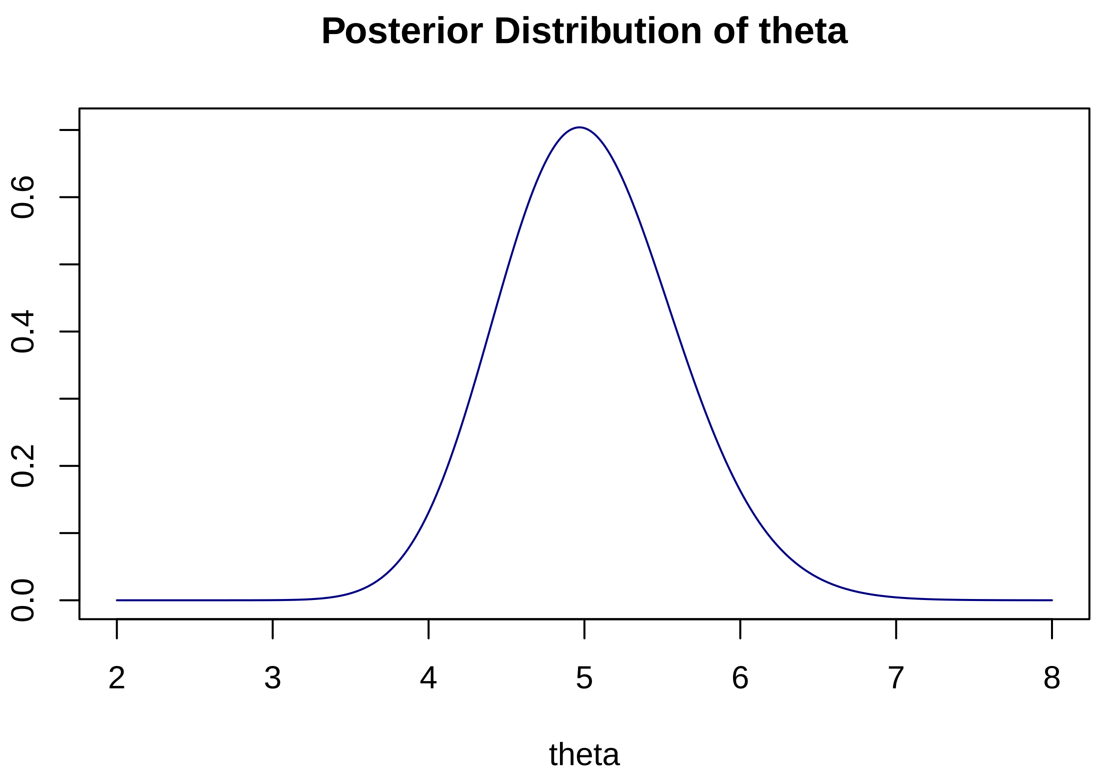
Christophoros Spyretos

2022-10-18

## Problem 1

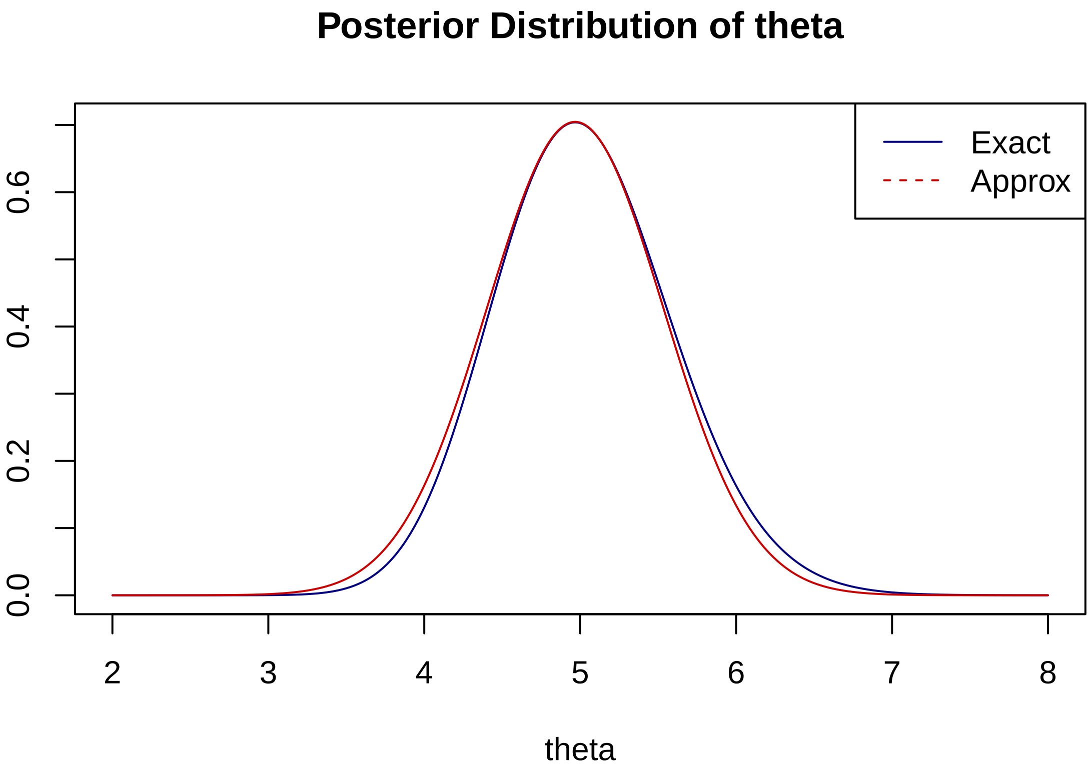
### Task d

```
set.seed(12345)

LogPost <- function(theta,n,sumx){
  res <- (2+sumx)*log(theta) - theta*(0.5+n)
  return(res)
}

thetaGrid <- seq(2,8,0.01)
n <- 15
sumx <- 75

LogPost_propto <- exp(LogPost(thetaGrid,n,sumx))
LogPost_dens <- LogPost_propto/(0.01*sum(LogPost_propto))

plot(thetaGrid, LogPost_dens, type = "l", col = "navy",
     main = "Posterior Distribution of theta",
     xlab = "theta", ylab = "")
```




### Task e

```
OptimRes <- optim(4, LogPost, gr = NULL, n, sumx,
                  method = c("L-BFGS-B"),lower = 3,
                  control = list(fnscale = -1),
                  hessian = TRUE)

approx <- dnorm(thetaGrid, mean = OptimRes$par, sd = sqrt(diag(-solve(OptimRes$hessian))))

plot(thetaGrid, LogPost_dens, type = "l", col = "navy",
     main = "Posterior Distribution of theta",
     xlab = "theta", ylab = "")
lines(thetaGrid,approx, type = "l", col = "red3")
legend("topright", legend = c("Exact", "Approx"), col = c("navy","red3"), lty = 1:2)
```




The posterior approximation very accurate. The exact posterior is very slightly skewed to the right.

### Task f

```
set.seed(12345)

nSim <- 1000
T_x_rep <- matrix(0,nSim,1)

for (i in 1:nSim){
  theta <- rgamma(1, shape = 3 + sumx, rate = 0.5 + n)
  x_rep <- rpois(n,theta)
  T_x_rep[i,] <- max(x_rep)
}

prob1f <- mean(T_x_rep>14)
```

The posterior predictive p-value is 0.002, which is a very low value; thus, it is not reasonable to think that the maximum value of 14 from Gunnar originates from the Poisson distribution in this problem.

## Problem 2

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

### Task a

```
mu_0 <- as.vector(rep(0,3))
Sigma2_0 <- 16*diag(3)
```

```r
nIter <- 10000
X <- as.matrix(X)

PostDraws <- BayesLogitReg(y, X, mu_0, Sigma2_0, nIter)
```

```r
Betas <- PostDraws$betaSample

intervalB1 <- quantile(Betas[,2], probs = c(0.05, 0.95))
# table for the interval
intervalB1 <- data.frame(lower_bound = intervalB1[1], upper_bound = intervalB1[2])
colnames(intervalB1) <- c("Lower bound", "Upper bound")
rownames(intervalB1) <- c("90% Equal Tail Credible Interval")
knitr::kable(intervalB1)
```

| | Lower bound | Upper bound |
|---|---|---|
| 90% Equal Tail Credible Interval | 0.2099814 | 1.887007 |

The 90% posterior probability that $\beta_1$ is on the interval.

### Task b

```r
prob2b <- mean(Betas[,3]>0)
```

The posterior probability that $\beta_2 > 0$ is 0.8828. The effect from variable x2 on p is positive when it changes from 0 to 1.

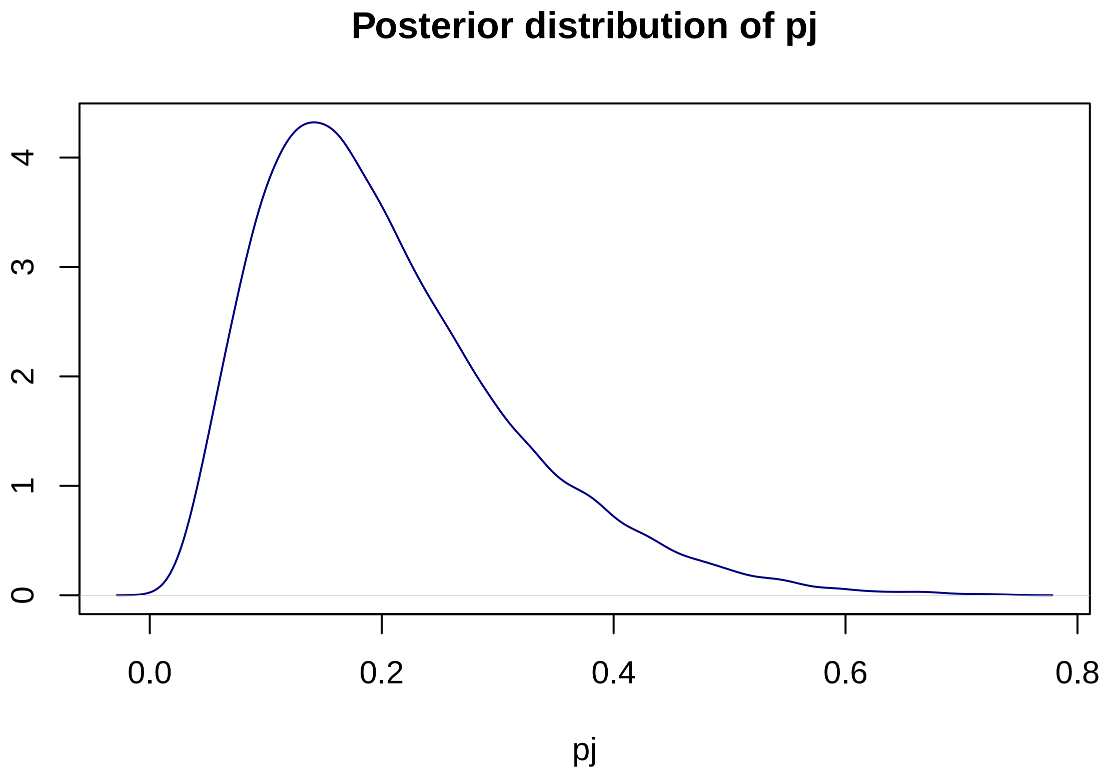
### Task c

```r
prob2c <- mean(Betas[,2]>0 & Betas[,3]>0)
```

The joint posterior probability that both $\beta_1 > 0$ and $\beta_2 > 0$ is 0.8711.

### Task d

```r
numerator <- exp(Betas[,1])
pj <- numerator/(1+numerator)

plot(density(pj), col = "navy",
     main = "Posterior distribution of pj",
     xlab = "pj", ylab = "")
```

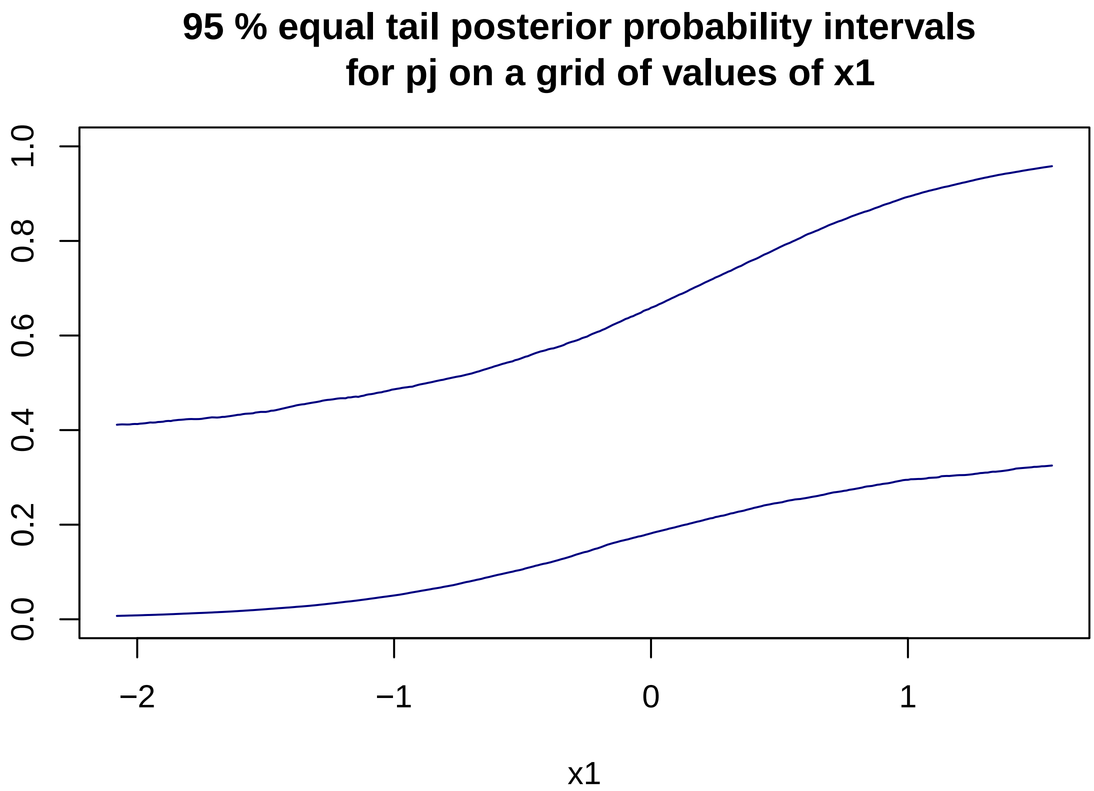



```
prob2d <- mean(pj>0.5)
```

The posterior probability that $p_j > 0.5$ for this patient is 0.0156.

### Task e

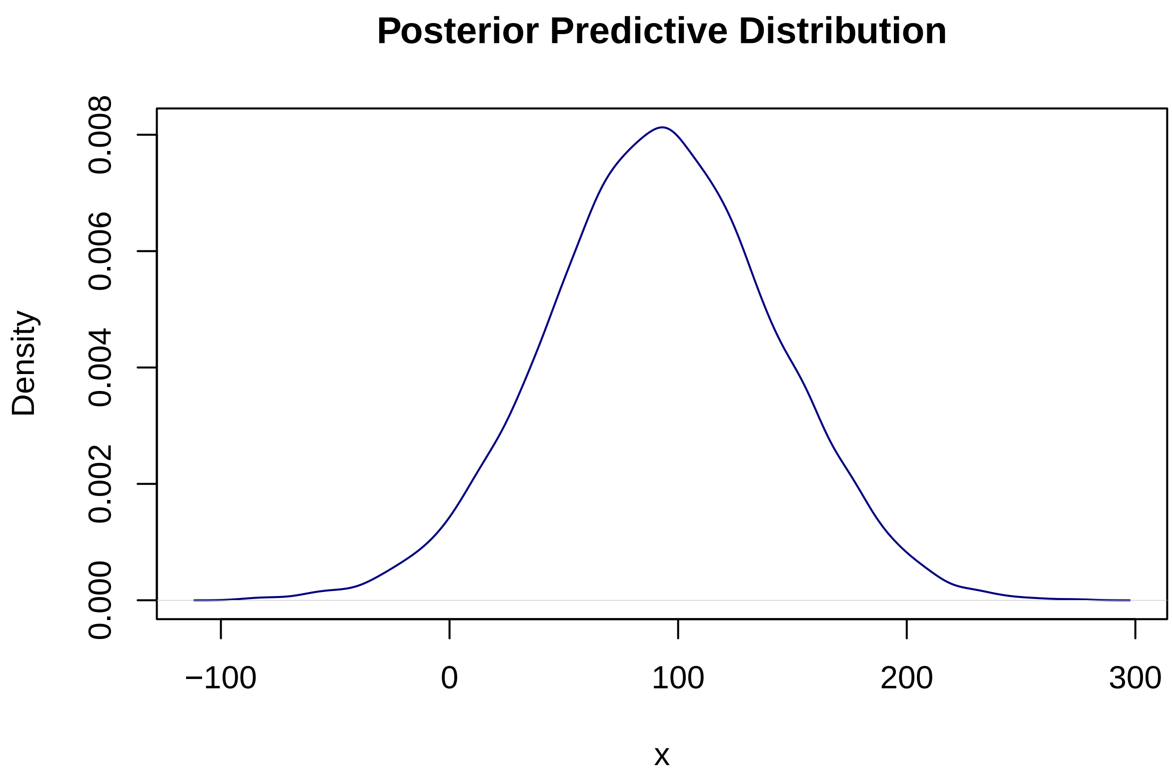
```
x1Grid <- seq(min(X[,2]),max(X[,2]),0.01)
intervals <- matrix(0,length(x1Grid),2)

for (i in 1:length(x1Grid)){
  numerator <- exp(Betas[,1] + Betas[,2]*x1Grid[i] +Betas[,3])
  pj <- numerator/(1+numerator)
  intervals[i,] <- quantile(pj, probs = c(0.025,0.975))
}

plot(x1Grid, intervals[,1], type = "l", col = "navy",
     main = "95 % equal tail posterior probability intervals
     for pj on a grid of values of x1",
     xlab = "x1", ylab = "", ylim = c(0,1))
lines(x1Grid, intervals[,2], type = "l", col = "navy")
```




## Problem 3

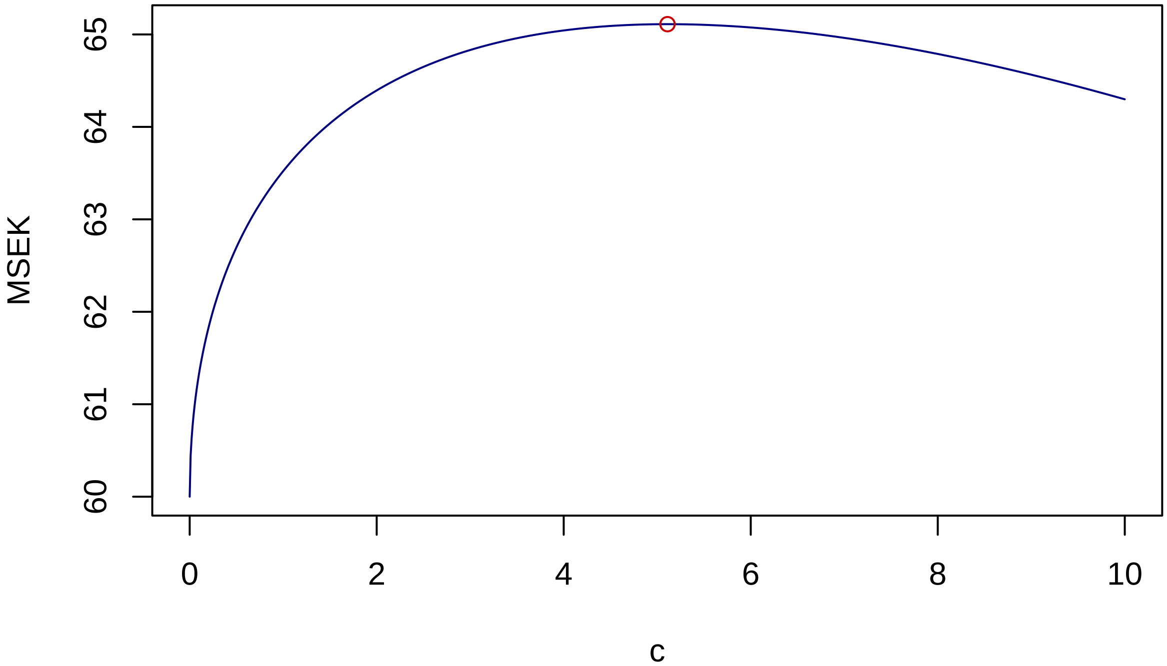
### Task b

```r
set.seed(12345)

nSim <- 10000
x <- matrix(0,nSim,1)

for (i in 1:nSim) {
  mu <- rnorm(1, mean = 92, sd = 2)
  x[i,] <- rnorm(1, mean = mu, sd = 50)
}


plot(density(x), col = "navy",
     main = "Posterior Predictive Distribution",
     xlab = "x")
```




### Task c

```
set.seed(12345)

utility_fun <- function(c,mu){
  res <- 60 + sqrt(c)*mean(log(mu)) - c
  return(res)
}

cGrid <- seq(0,10,0.01)
mu <- rnorm(nSim, mean = 92, sd = 2)
money <- matrix(0,length(cGrid),1)

for (i in 1:length(cGrid)){
  money[i,] <- utility_fun(cGrid[i],mu)
}

cOpt <- cGrid[which.max(money)]

plot(cGrid,money, type = "l", col = "navy",
     xlab = "c", ylab = "MSEK")
points(cOpt,utility_fun(cOpt,mu), col = "red3")
```




The industry spending c MSEK on advertisements is 5.11.