ComputerSol

Christophoros Spyretos

2022-10-16

Problem 1

Task a,b and c hand written

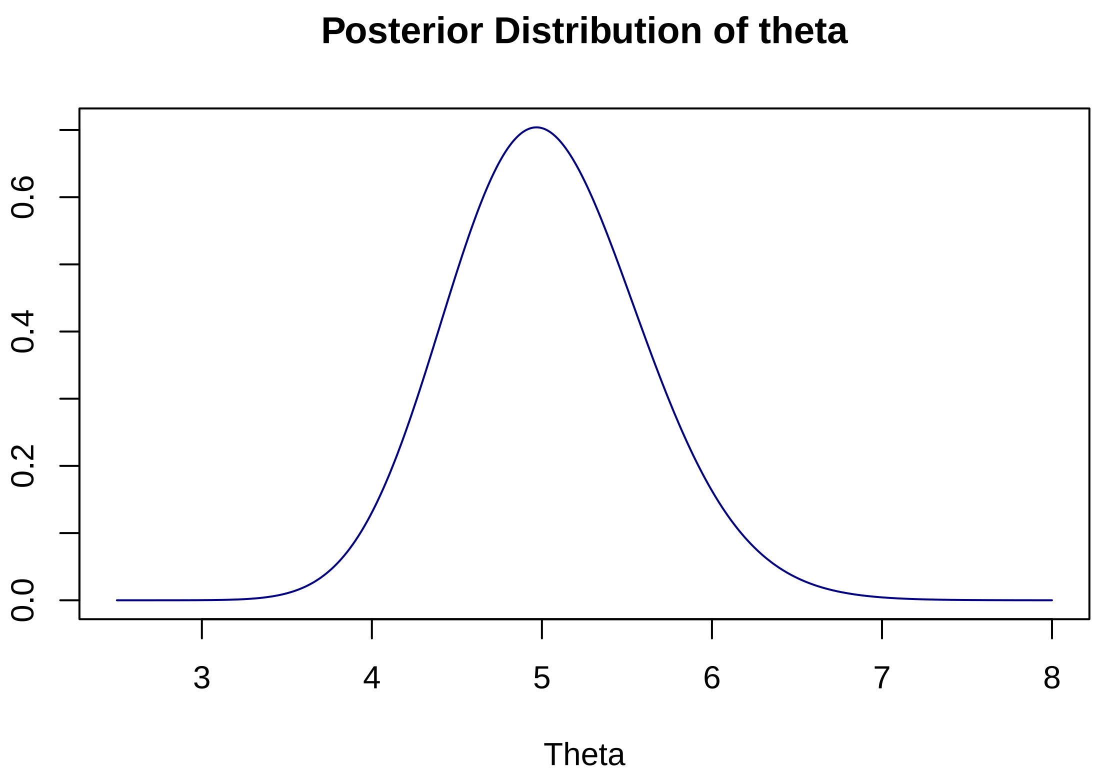
Task d

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

thetaGrid <- seq(2.5,8,0.01)
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 = "")
```



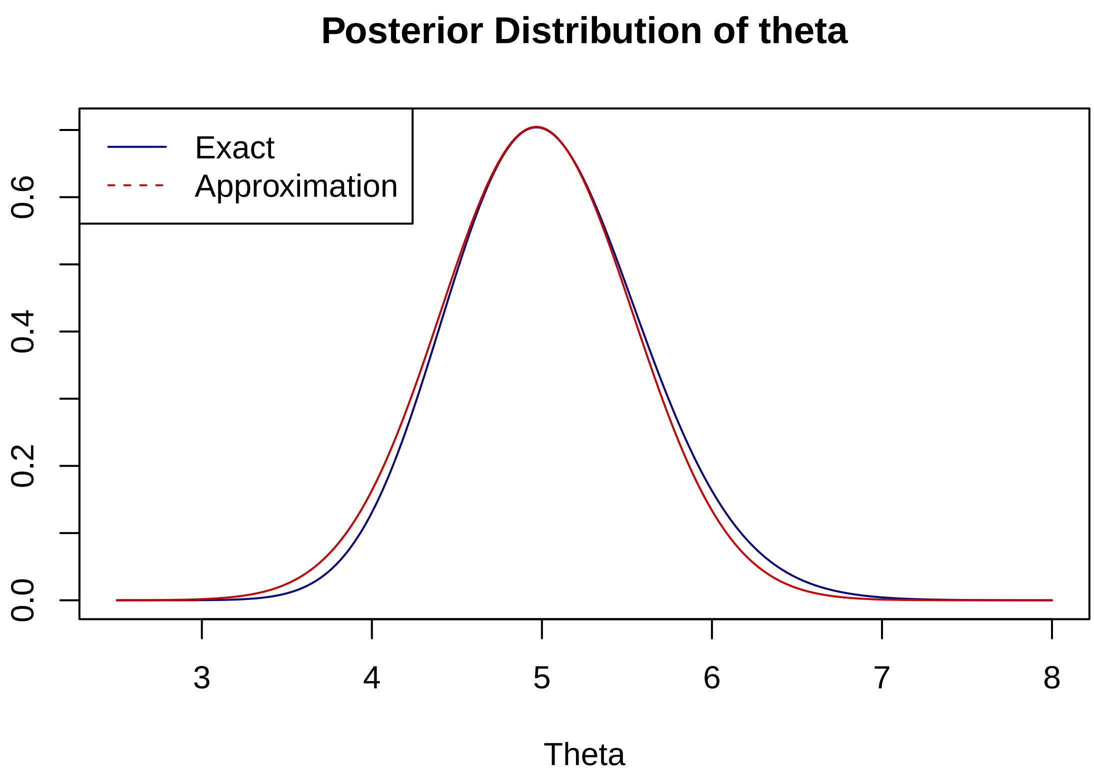
Task e

```
set.seed(12345)

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("topleft", legend = c("Exact","Approximation"),
       col = c("navy", "red3"), lty = 1:2)
```



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

Task f

```
set.seed(12345)

nDraws <- 10000
T_x_rep <- matrix(0,nDraws,1)

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

prob <- mean(T_x_rep >= 14)
```

The posterior predictive p-value is approximately 0.014, hence the probability that the maximum value of 14 from Gunnar originates from the Poisson distribution is very low.

Problem 2

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

Task a

```
library(mvtnorm)
```

```r
nIter <- 10000
mu_0 <- as.vector(rep(0,3))
Sigma_0 <- 16*diag(3)

PostDraws <- BayesLogitReg(y, X, mu_0, Sigma_0, nIter)

Betas <- PostDraws$betaSample

intervalB1 <- quantile(Betas[,2], probs = c(0.05,0.95))

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	0.2108366	1.874749

It is the 90% posterior probability that β_1 is between the above interva.

Task b

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

The probability that $\beta_2 > 0$ is approximately 0.88. It is the probability of x2 having a positive effect on pi when x2 changes from 0 to 1.

Task c

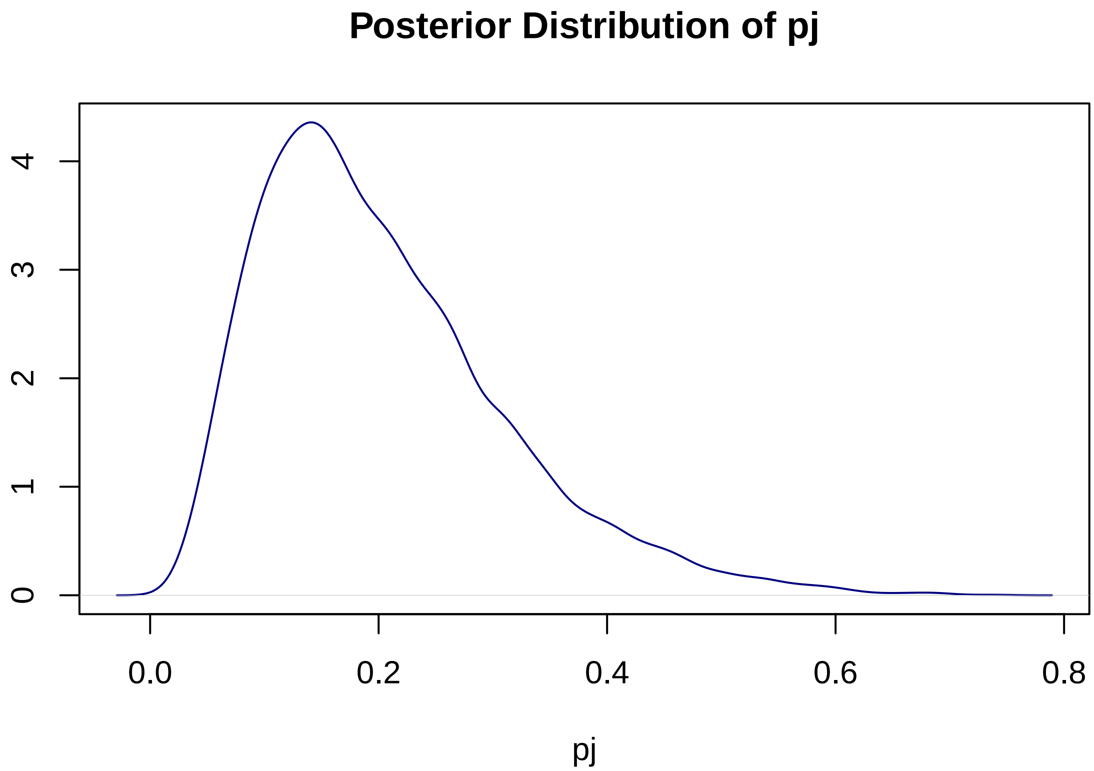
```r
prob <- mean(Betas[,2] > 0 & Betas[,3] > 0)
```

The probability that $\beta_1 > 0$ and $\beta_2 > 0$ is approximately 0.87.

Task d

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

plot(density(patient), type = "l", col = "navy",
     main = "Posterior Distribution of pj",
     xlab = "pj", ylab = "")
```




```r
prob <- mean(patient>0.5)
```

The posterior probability that $p_j > 0.5$ for this patient is approximately 0.015.

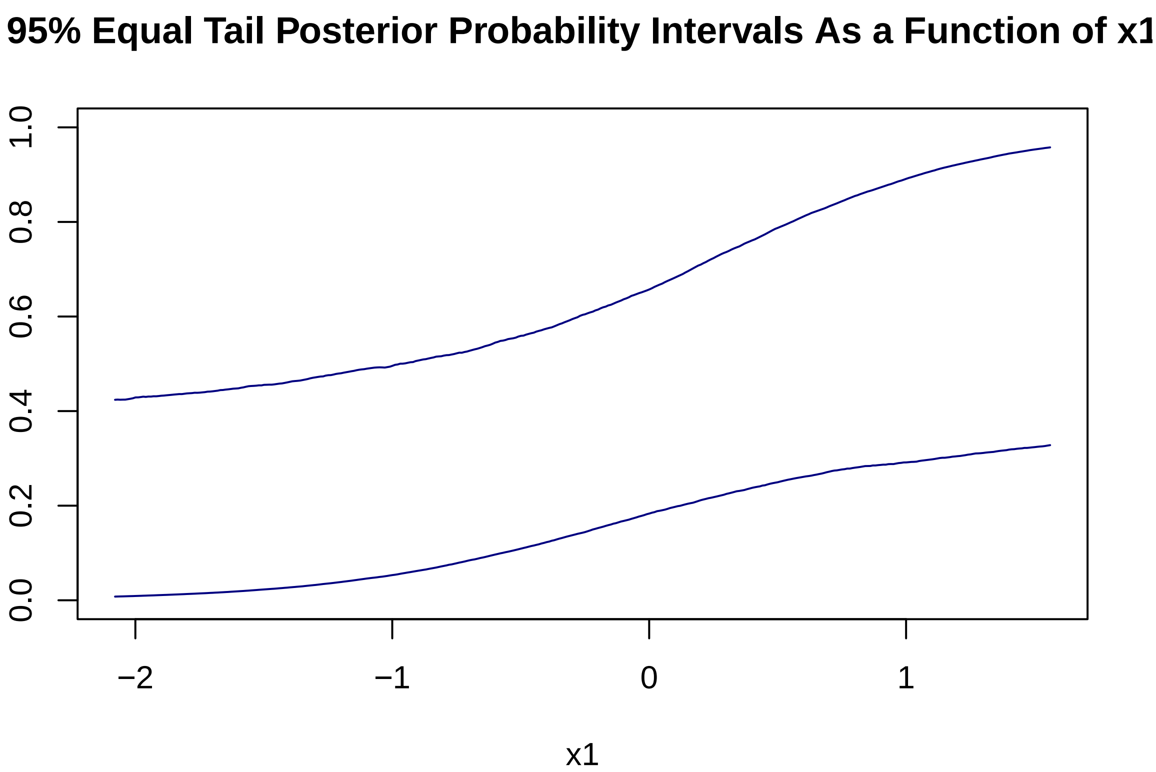
Task e

```r
x1Grid <- seq(min(X[,2]), max(X[,2]),0.01)

intervals <- matrix(0, nrow = length(x1Grid), ncol = 2)

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

plot(x1Grid,intervals[,1], type = "l", col = "navy",
     main = "95% Equal Tail Posterior Probability Intervals As a Function of x1",
     xlab = "x1", ylab = "", ylim = c(0,1))
lines(x1Grid,intervals[,2], type = "l", col = "navy")
```



Problem 3

Task a

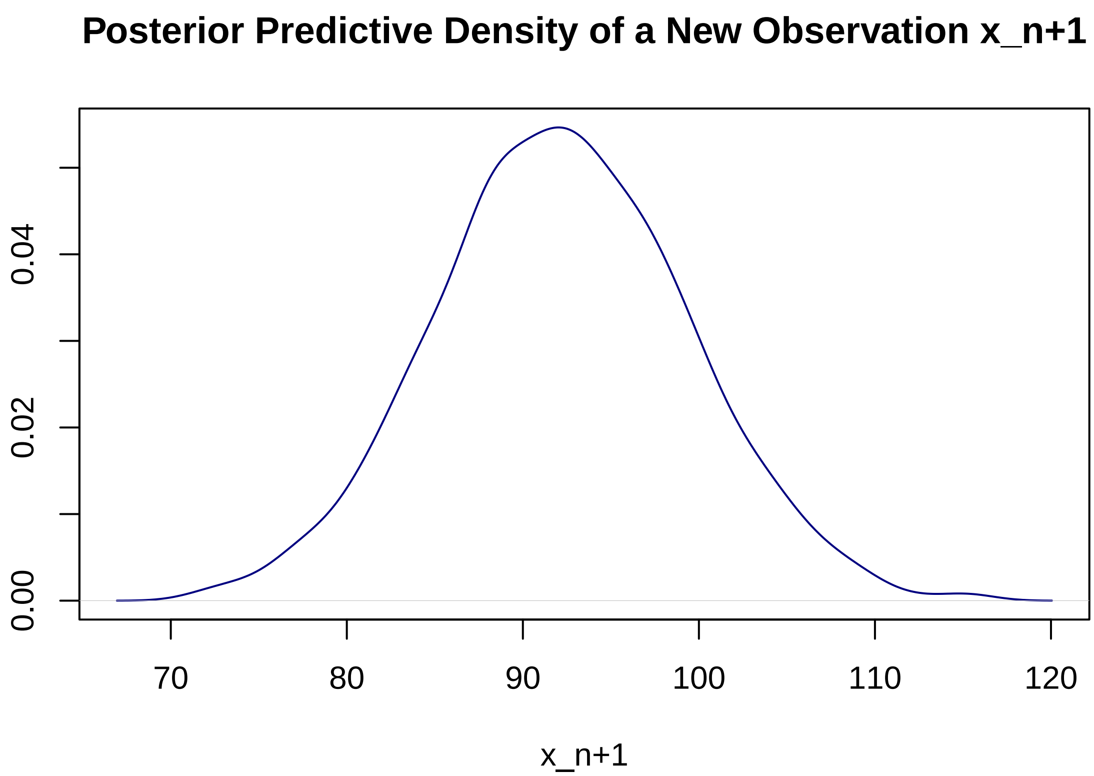
Hand written solution.

Task b

```
n <- 1000
x <- matrix(0, nrow = n, ncol = 1)

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

plot(density(x), type = "l" , col = "navy",
     main = "Posterior Predictive Density of a New Observation x_n+1",
     xlab = "x_n+1", ylab = "")
```



Task c

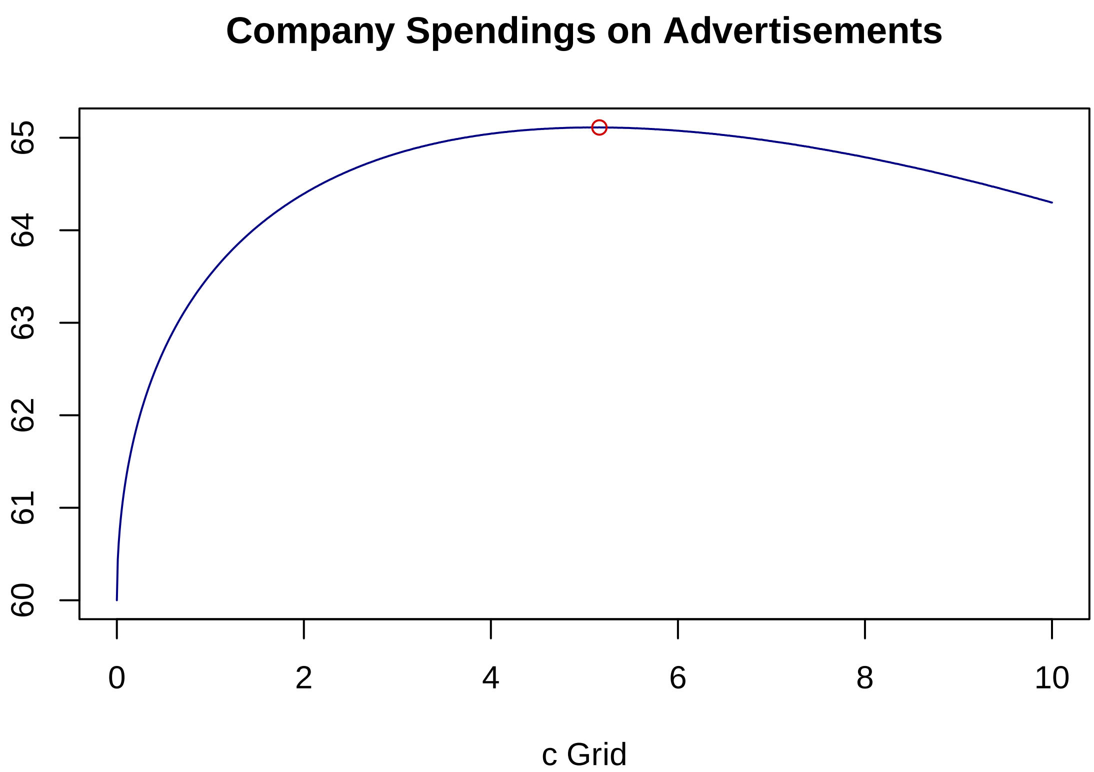
```
utility_function <-  function(c,mu){
  res <- 60 + sqrt(c)*log(mu) - c
  return(res)
}

n <- 10000
cGrid <- seq(0,10,0.01)
spendings <- matrix(0, nrow = length(cGrid), ncol = 1)

for (i in 1:length(cGrid)){
  mu <- mean(rnorm(n,mean = 92, sd = 2))
  spendings[i,] <- utility_function(cGrid[i],mu)
}

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

plot(cGrid,spendings, type = "l", col = "navy",
     main = "Company Spendings on Advertisements",
     xlab = "c Grid", ylab = "")
points(cOpt,utility_function(cOpt,mean(rnorm(n,mean = 92, sd = 2))),
       col = "red3")
```



The optimal spend is around 5.11 MSEK.