

# BayesExam

Christophoros Spyretos

2022-10-18

## Problem 1

### Task b

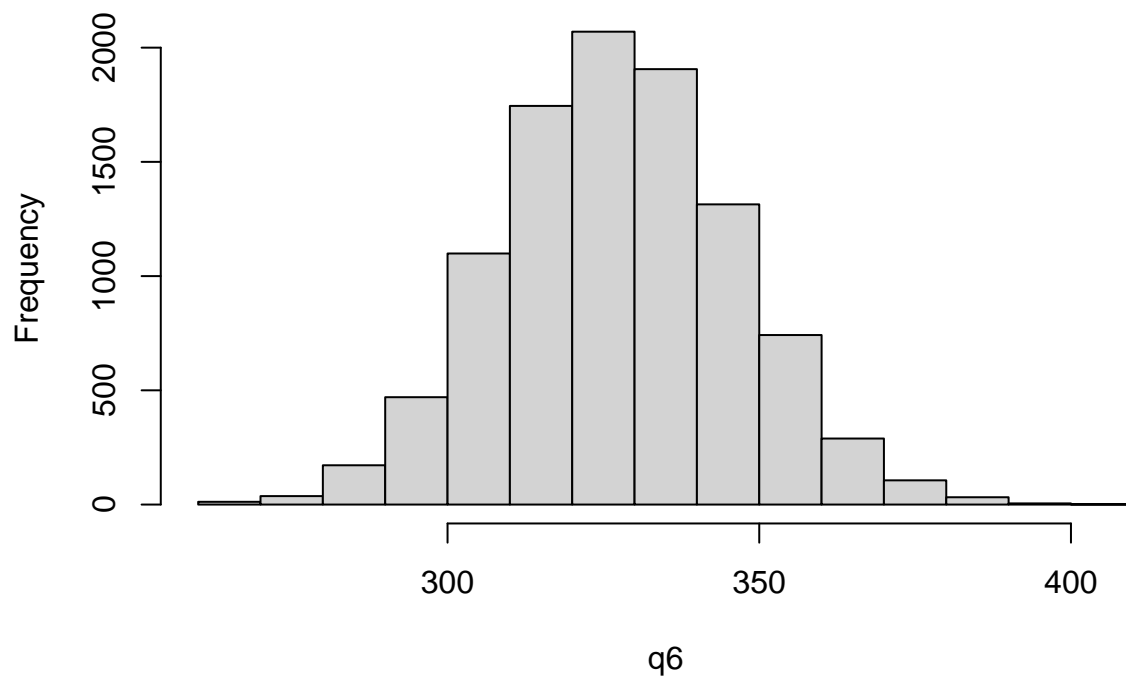
```
set.seed(12345)

q <- c(322,248,385,341,310)
an <- 2326
bn <- 7
n <- 5
nDraws <- 10000

theta <- rgamma(nDraws,shape = an + sum(q), rate = bn + n)
q6 <- rpois(nDraws,theta)

hist(q6)
```

**Histogram of q6**



```
prob1b <- mean(q6>350)
```

The  $Pr(Q_6 > 350|q_1, \dots, q_5)$  is 0.1175.

```
set.seed(12345)
```

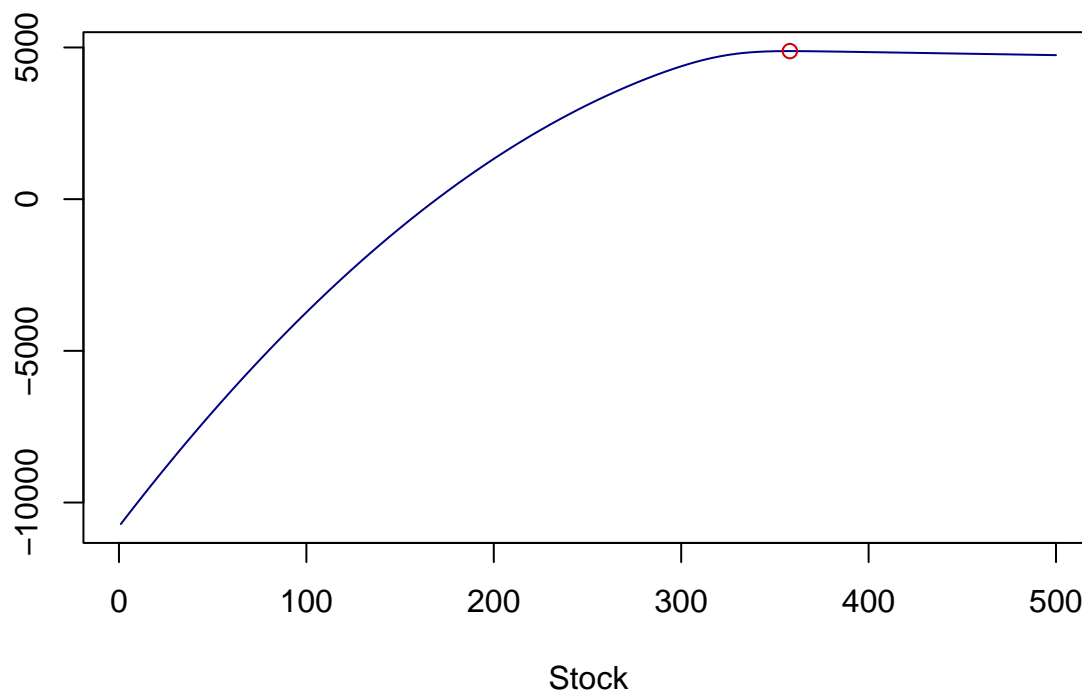
```
utility_function <- function(a,q6){  
  res <- ifelse(q6 <=a,15*q6 - (a-q6),15*a-0.1*(q6-a)^2)  
  return(res)  
}
```

```
aGrid <- seq(1,500,1)  
products <- matrix(0,length(aGrid),1)
```

```
for (i in 1:length(aGrid)) {  
  products[i] <- mean(utility_function(aGrid[i],q6))  
}
```

```
aOpt <- aGrid[which.max(products)]
```

```
plot(aGrid,products, type = "l", col = "navy",  
      xlab = "Stock", ylab = "")  
points(aOpt,mean(utility_function(aOpt,q6)), col = "red3")
```



The stock that should be held for the next month is 358.

## Problem 2

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

## Task a

```
set.seed(12345)

mu_0 <- as.vector(rep(0,6))
Omega_0 <- (1/100)*diag(6)
v_0 <- 1
sigma2_0 <- 100^2
nIter <- 10000

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

Betas <- PostDraws$betaSample

BetasMean <- colMeans(Betas)

BetasMean <- data.frame(BetasMean)
colnames(BetasMean) <- c("Mean Values")
rownames(BetasMean) <- c("Beta0", "Beta1", "Beta2",
                        "Beta3", "Beta4", "Beta5")

knitr::kable(BetasMean)
```

	Mean Values
Beta0	-488.8766671
Beta1	10.7443341
Beta2	-0.0261714
Beta3	33.2477399
Beta4	-0.6083833
Beta5	-0.1344413

```
BetasInterval <- matrix(0,6,2)

for (i in 1:6) {
  BetasInterval[i,] <- quantile(Betas[,i],probs = c(0.005,0.995))
}

BetasInterval <- data.frame(BetasInterval)
colnames(BetasInterval) <- c("Lower Bound", "Upper Bound")
rownames(BetasInterval) <- c("Beta0", "Beta1", "Beta2",
                            "Beta3", "Beta4", "Beta5")

knitr::kable(BetasInterval)
```

	Lower Bound	Upper Bound
Beta0	-1481.4811157	539.0536182
Beta1	5.7870148	15.6251718
Beta2	-0.0364576	-0.0162506
Beta3	-40.9307993	105.2944392
Beta4	-1.9988580	0.7934610
Beta5	-0.3064411	0.0314359

The 99% posterior probability that  $\beta_1$  is in the interval.

## Task b

```
sd <- sqrt(PostDraws$sigma2Sample)
```

```
MeanSD <- mean(sd)
```

```
MedianSD <- median(sd)
```

The posterior mean and posterior median of the standard deviation is approximately 40.01 and 39.56 respectively.

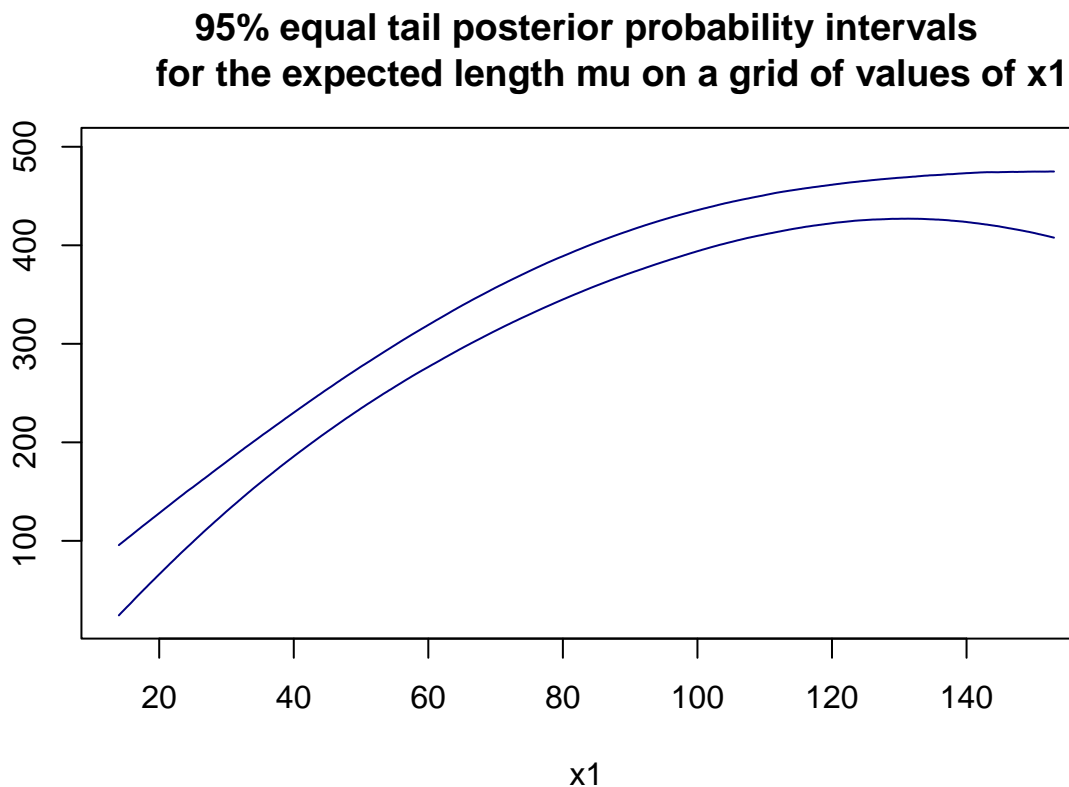
## Task c

```
x1Grid <- seq(min(X[,2]),max(X[,2]),0.1)
```

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

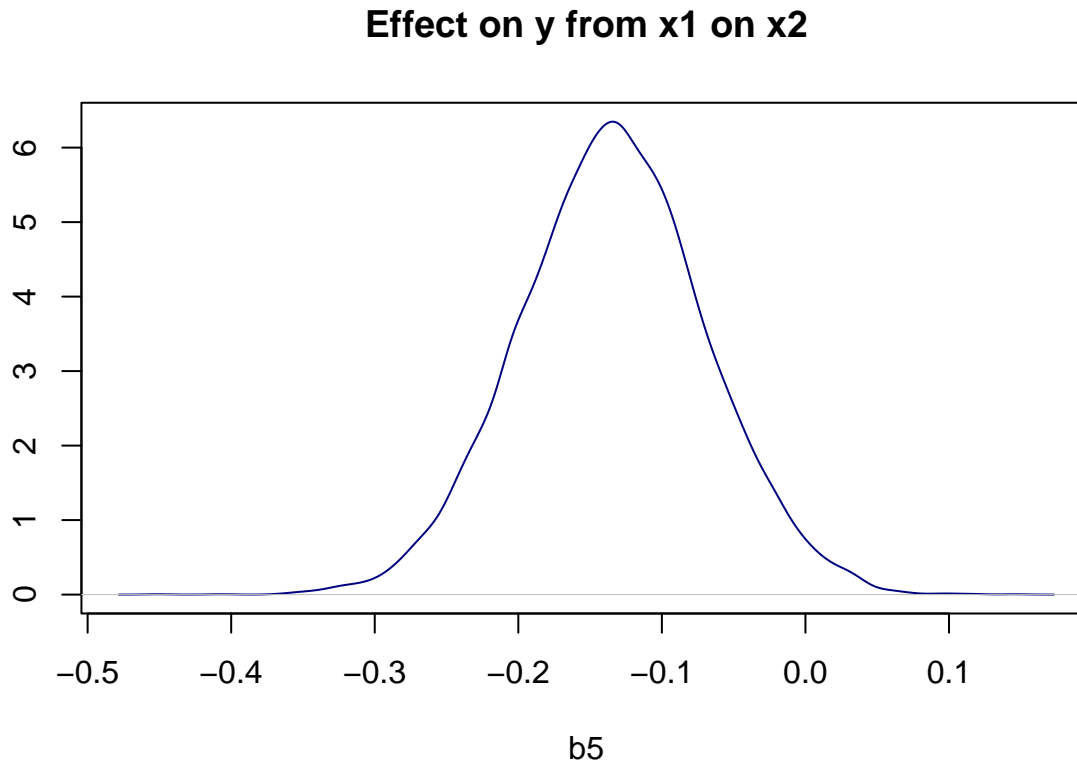
```
for (i in 1:length(x1Grid)) {  
  mu <- Betas[,1] + Betas[,2]*x1Grid[i] + Betas[,3]*(x1Grid[i]^2) +  
    Betas[,4]*27 + Betas[,5]*(27^2) + Betas[,6]*x1Grid[i]*27  
  intervals[i,] <- quantile(mu, probs = c(0.025,0.975))  
}
```

```
plot(x1Grid,intervals[,1], type = "l", col = "navy",  
     main = "95% equal tail posterior probability intervals  
for the expected length mu on a grid of values of x1",  
     xlab = "x1", ylab = "", ylim = c(20,500))  
lines(x1Grid,intervals[,2], type = "l", col = "navy")
```



## Task d

```
plot(density(Betas[,6]), type = "l", col = "navy",  
     main = "Effect on y from x1 on x2",  
     xlab = "b5", ylab = "")
```



```
intervalB5 <- quantile(Betas[,6], probs = c(0.025,0.975))  
  
intervalB5 <- data.frame(lower_bound = intervalB5[1], upper_bound = intervalB5[2])  
colnames(intervalB5) <- c("Lower Bound", "Upper Bound")  
rownames(intervalB5) <- c("95% Equal Tail Credible Interval")  
knitr::kable(intervalB5)
```

	Lower Bound	Upper Bound
95% Equal Tail Credible Interval	-0.2627378	-0.0057662

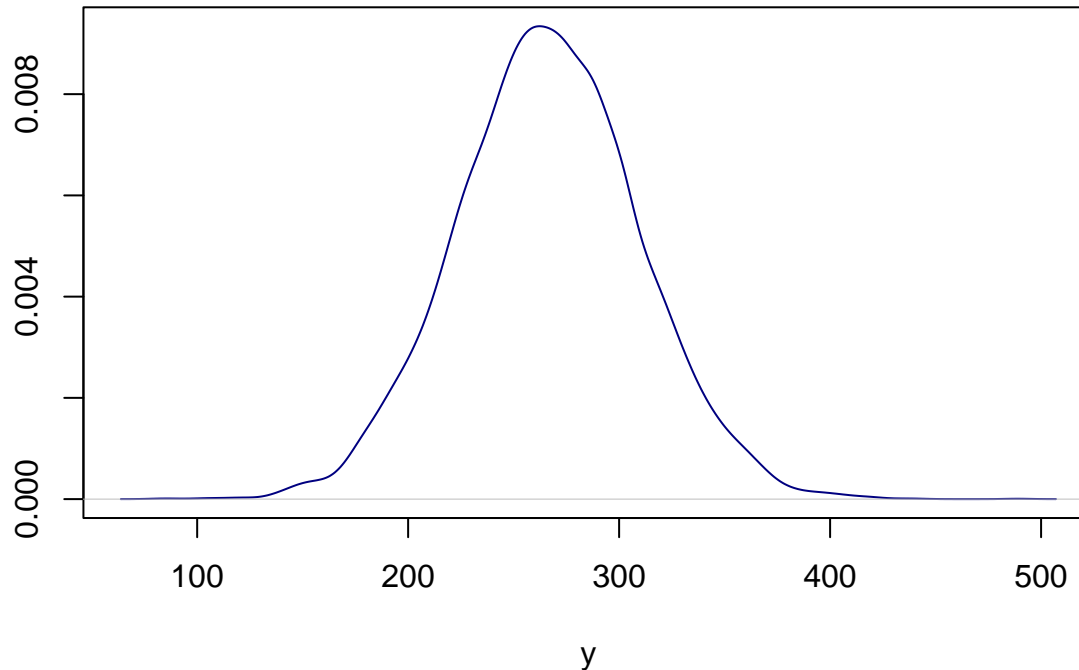
There is a substantial mass probability that the effect on y from x1 depends on x2. The 95% equal tail credible intervals strengthens the assumption as  $\beta_5$  is on the interval.

## Task e

```
mu <- Betas[,1] + Betas[,2]*50 + Betas[,3]*(50^2) +  
      Betas[,4]*25 + Betas[,5]*(25^2) + Betas[,6]*50*25  
  
y_values <- rnorm(nIter,mu,sd)  
  
plot(density(y_values), col = "navy",
```

```
main = "Posterior predictive distribution of y for a new mollusc",
xlab = "y", ylab = "")
```

## Posterior predictive distribution of y for a new mollusc



### Task f

```
set.seed(12345)

T_y_rep <- matrix(0,nIter,1)
mu <- Betas%*%t(X)

for (i in 1:nIter) {
  y_values <- rnorm(length(y), mean = mu[i], sd = sd[i])
  T_y_rep[i,] <- max(y_values)
}

prob2f <- mean(T_y_rep >= max(y))
```

The posterior predictive p-value is 0.0026. Hence, the model can not replicate the length of the largest mollusc in the data in a good way.

## Problem 3

### Task d

```
LogPost <- function(theta,n,sumx3){
  res <- (2+n)*log(theta) - theta*(4+sumx3)
  return(res)
}
```

```

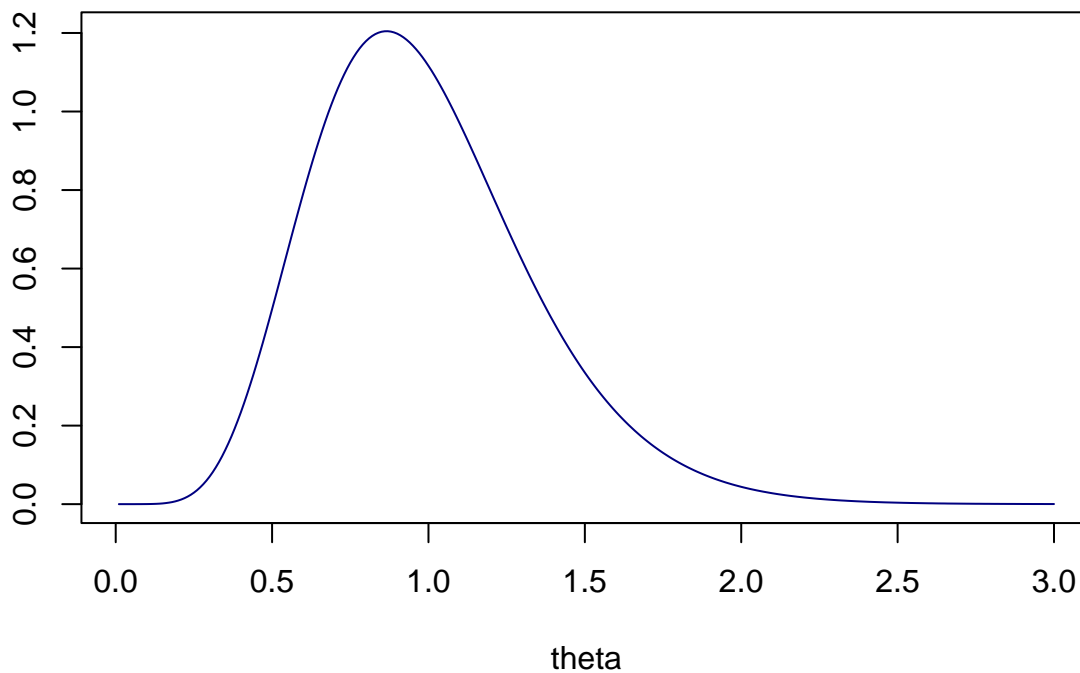
thetaGrid <- seq(0.01,3,0.01)
n <- 5
sumx3 <- 2*(0.8^3) + 1.1^3 + 0.9^3 + 1

LogPost_propto <- exp(LogPost(thetaGrid,n,sumx3))
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 = "")

```

## Posterior Distribution of theta



## Task e

```

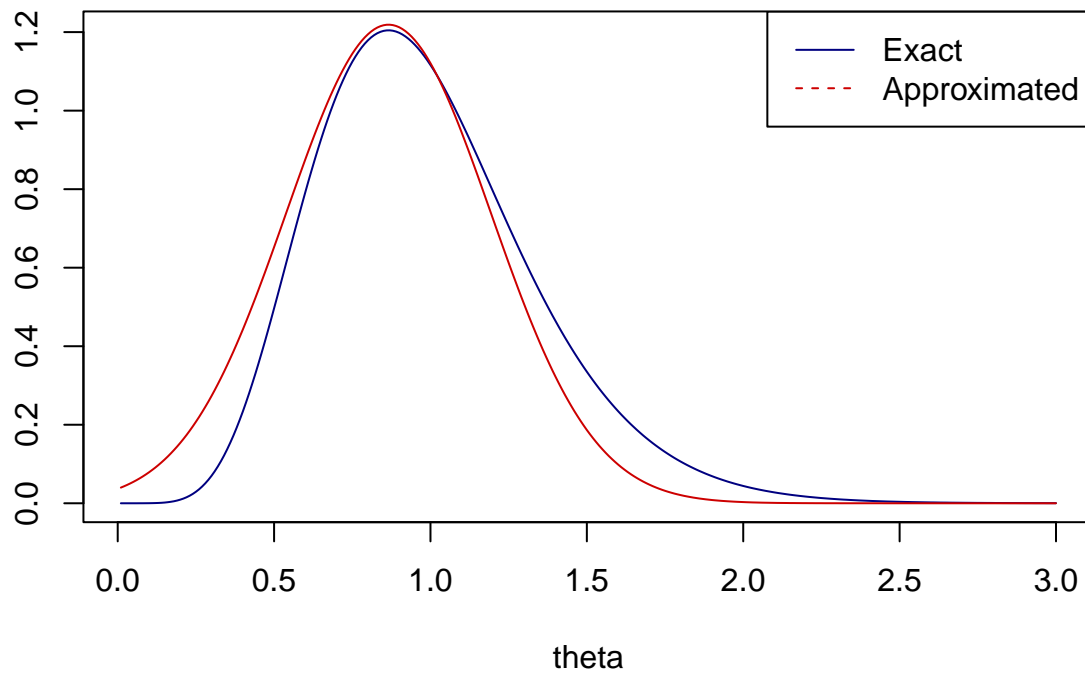
OptimRes <- optim(0.5, LogPost, gr = NULL, n, sumx3,
                 method = c("L-BFGS-B"), lower = 0.1,
                 control = list(fnscale = -1), hessian = TRUE)

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", "Approximated"), col = c("navy", "red3"), lty = 1:2)

```

## Posterior Distribution of theta



The posterior approximation is not that accurate, the exact posterior is skewed to the right.