

# Solution to computer exam in Bayesian learning

Bertil Wegmann

2022-06-03

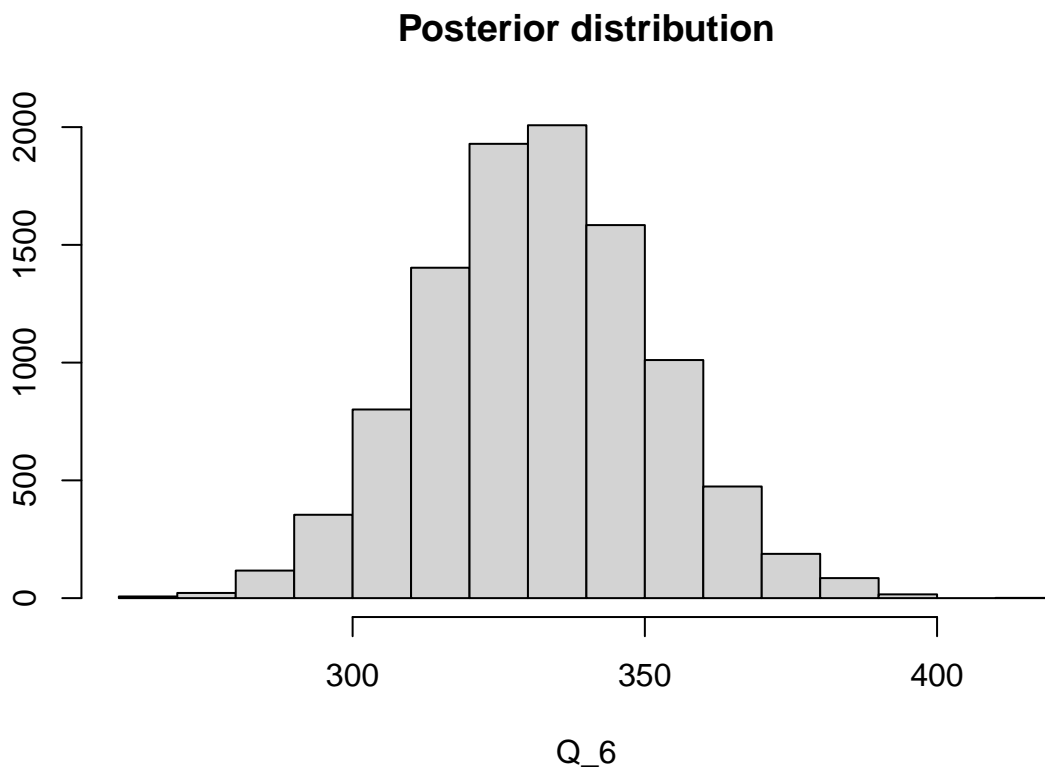
First load all the data into memory by running the R-file given at the exam

```
rm(list=ls())  
source("ExamData.R")  
set.seed(1)
```

## Problem 1

1b

```
alpha_n <- 2326  
beta_n <- 7  
theta <- rgamma(1e4, shape = alpha_n, rate = beta_n)  
Q_6 <- rpois(1e4, theta)  
hist(Q_6, main="Posterior distribution", xlab="Q_6", ylab="")
```



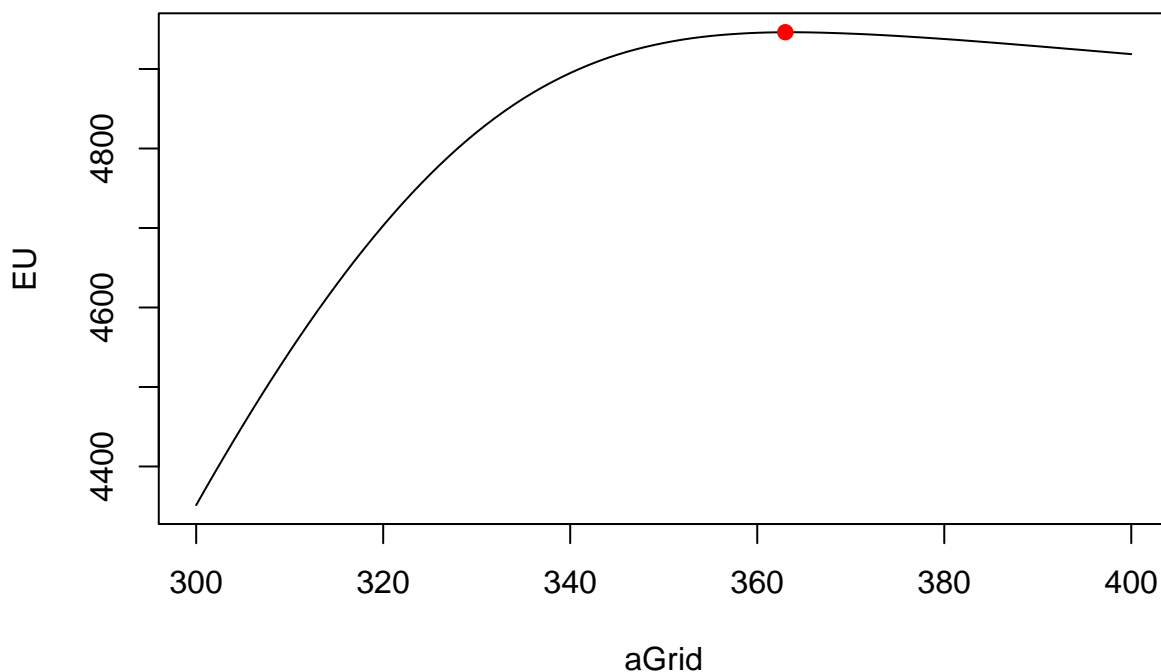
```
mean(Q_6 > 350)
```

```
## [1] 0.1775
```

The posterior probability is 0.178. The posterior distribution is plotted above.

1c

```
aGrid <- seq(300,400,by=1)
EU <- rep(NA,length(aGrid),1)
count <- 0
for (a in aGrid){
  count <- count + 1
  EU[count] <- mean(utility_func(a,Q_6))
}
plot(aGrid, EU, type = "l")
a0pt = aGrid[which.max(EU)] # This is the optimal a
points(a0pt,mean(utility_func(a=a0pt, Q_6)), col = "red",pch=19)
```



```
a0pt
```

```
## [1] 363
```

The optimal number of products in stock (a) is 363.

## Problem 2

2a

```
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

Means <- colMeans(Betas)
CredInt <- matrix(0,6,2)
for (j in 1:6){
  CredInt[j,] <- quantile(Betas[,j],probs=c(0.005,0.995))
}
PostRes <- matrix(0,6,3)
PostRes[,1] <- t(Means)
PostRes[,2:3] <- CredInt
PostRes

```

```

##           [,1]           [,2]           [,3]
## [1,] -490.4249992 -1.486995e+03 498.94407149
## [2,]  10.7268860  5.704980e+00 15.81988500
## [3,]  -0.0260867 -3.647196e-02 -0.01578911
## [4,]  33.3430410 -3.809424e+01 108.06775472
## [5,]  -0.6092026 -2.024708e+00  0.75924254
## [6,]  -0.1344050 -3.121198e-01  0.03562525

```

It is 99 % posterior probability that beta\_1 is on the interval (5.7,15.8).

## 2b

```

Sigma2 <- PostDraws$sigma2Sample
mean(sqrt(Sigma2))

```

```
## [1] 40.05494
```

```
median(sqrt(Sigma2))
```

```
## [1] 39.63477
```

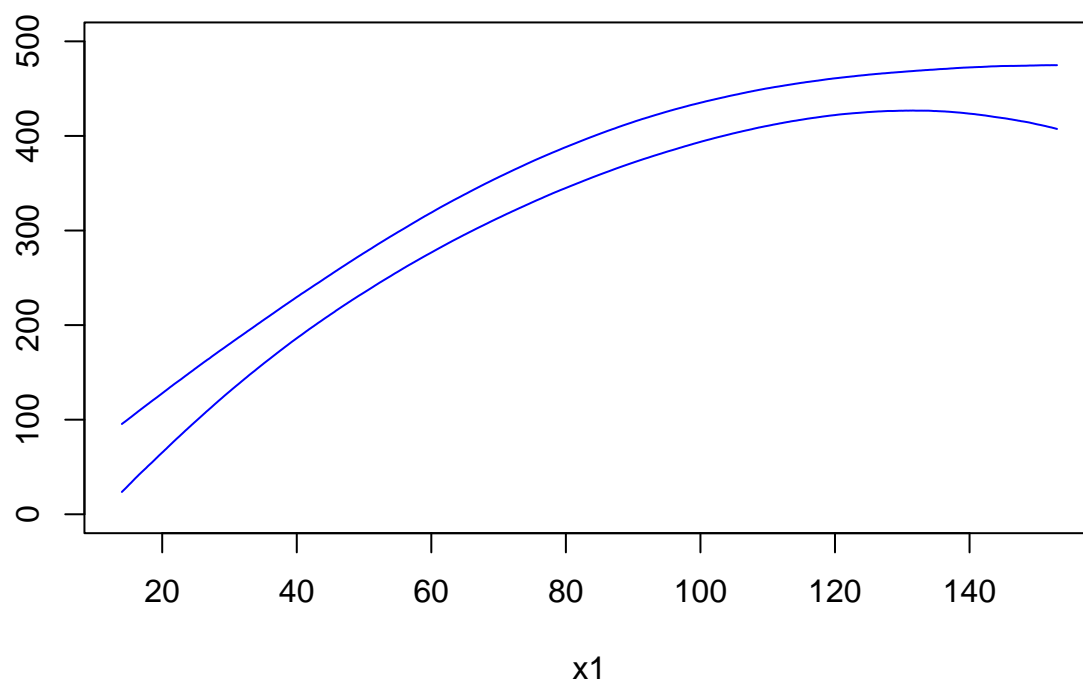
## 2c

```

x1_grid <- seq(min(X[,2]),max(X[,2]),0.1)
Mu_draws <- matrix(0,length(x1_grid),2)
for (ii in 1:length(x1_grid)){
  Curr_x <- c(1,x1_grid[ii],x1_grid[ii]**2,27,27**2,x1_grid[ii]*27)
  CurrMu <- Betas %*% Curr_x
  Mu_draws[ii,] <- quantile(CurrMu,probs=c(0.025,0.975))
}
plot(x1_grid,Mu_draws[,1],"n",main="95 % posterior probability intervals as a function of x1",
     xlab="x1", ylab="",ylim=c(0,500))
lines(x1_grid,Mu_draws[,1],col="blue")
lines(x1_grid,Mu_draws[,2],col="blue")

```

## 95 % posterior probability intervals as a function of x1

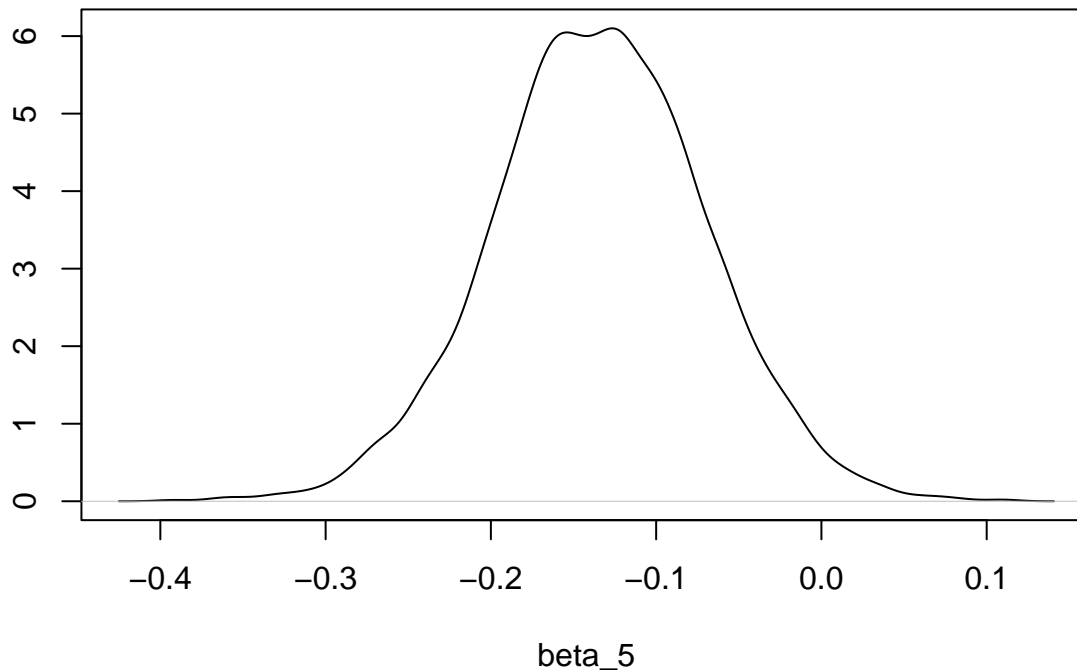


The limits of the posterior probability intervals as a function of  $x_1$  are plotted above.

2d

```
Effect_x1x2 <- Betas[,6]  
plot(density(Effect_x1x2),main="Posterior distribution",xlab="beta_5", ylab="")
```

## Posterior distribution



```
quantile(Effect_x1x2,probs=c(0.025,0.975))
```

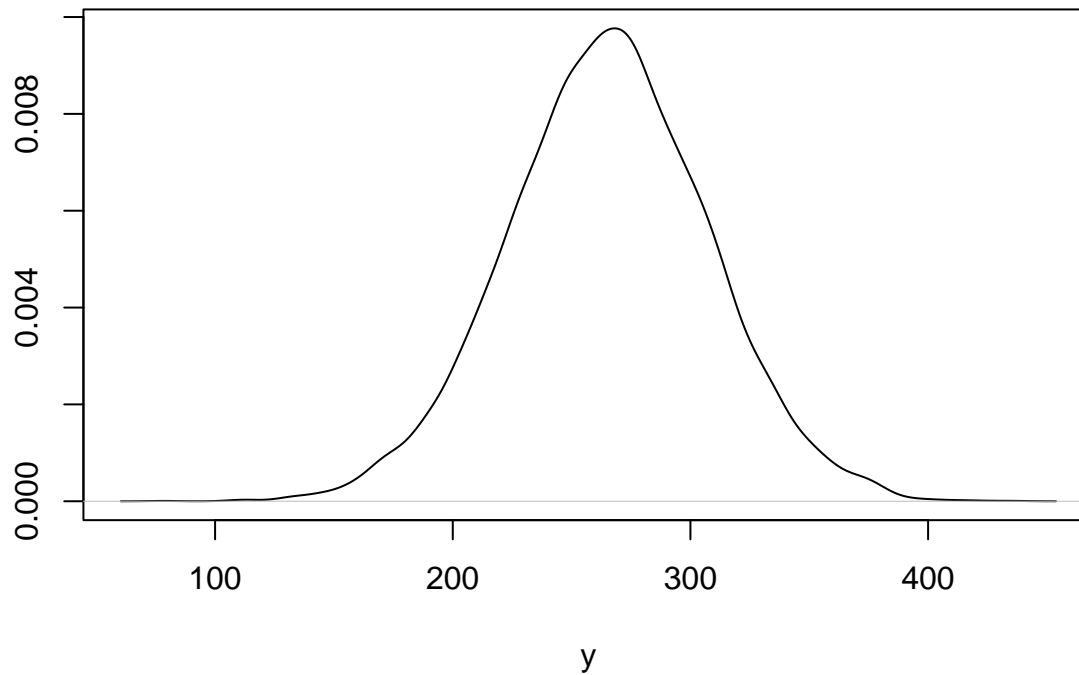
```
##          2.5%          97.5%  
## -0.265031107 -0.007967831
```

There is substantial probability mass that the effect on  $y$  from  $x_1$  depends on  $x_2$ , where the slope for  $x_1$  decreases (becomes more negative) as the value of  $x_2$  increases. This is also supported by the 95 % equal tail credible interval for  $\beta_5$  with only negative values.

2e

```
Mu <- Betas[,1] + Betas[,2]*50 + Betas[,3]*50**2 + Betas[,4]*25 + Betas[,5]*25**2 + Betas[,6]*50*25  
Sigma <- sqrt(Sigma2)  
y_Vals <- rnorm(10000,Mu,Sigma)  
plot(density(y_Vals),main="Posterior predictive distribution of y",xlab="y", ylab="")
```

## Posterior predictive distribution of y



The posterior predictive distribution of y is plotted above.

2f

```
T_y <- max(y)
T_y_rep <- matrix(0,nIter,1)
Mu <- Betas %*% t(X)
for (ii in 1:nIter){
  y_Vals <- rnorm(length(y),Mu[ii,],Sigma[ii])
  T_y_rep[ii,1] <- max(y_Vals)
}
mean(T_y_rep >= T_y)
```

```
## [1] 0.9885
```

The posterior predictive p-value is 0.989, which is far away from 0.5. Hence, the model can not replicate the length of the largest mollusc in the data in a good way.

## Problem 3

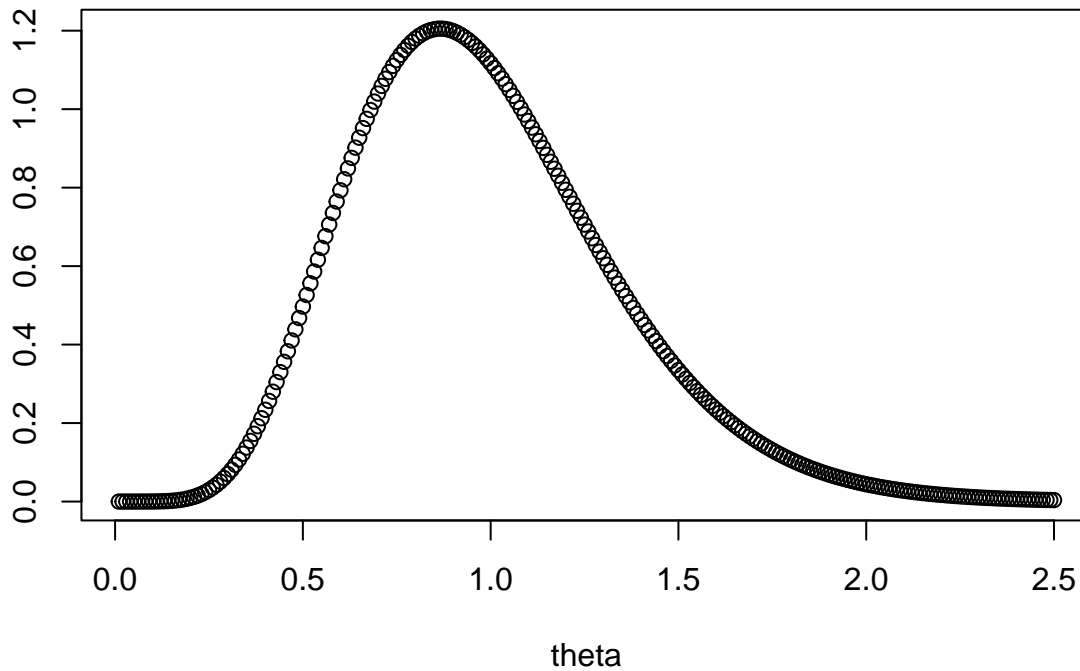
3d

```
LogPost <- function(theta,n,Sumx3){
  logLik <- n*log(theta) - Sumx3*theta;
  logPrior <- 2*log(theta) - 4*theta;

  return(logLik + logPrior)
}
theta_grid <- seq(0.01,2.5,0.01)
```

```
PostDens_propto <- exp(LogPost(theta_grid,5,4.084))
PostDens <- PostDens_propto/(0.01*sum(PostDens_propto))
plot(theta_grid,PostDens,main="Posterior distribution",xlab="theta", ylab="")
```

## Posterior distribution



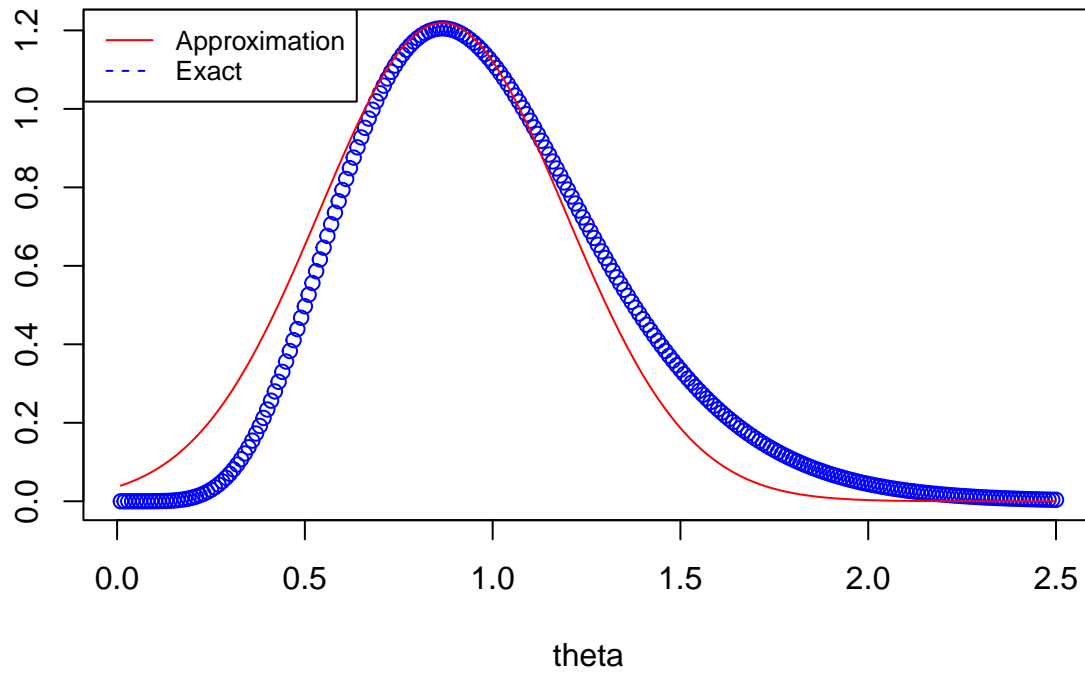
The posterior distribution is given above.

3e

```
n <- 5
Sumx3 <- 4.084
OptRes <- optim(0.5,LogPost,gr=NULL,n,Sumx3,method=c("L-BFGS-B"),lower=0.1,
               control=list(fnscale=-1),hessian=TRUE)

plot(theta_grid,PostDens,col="blue",main="Posterior distribution",xlab="theta", ylab="")
lines(theta_grid,dnorm(theta_grid,mean = OptRes$par,sd = sqrt(-1/OptRes$hessian)),col="red")
legend("topleft", legend=c("Approximation", "Exact"), col=c("red", "blue"), lty=1:2, cex=0.8)
```

## Posterior distribution



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