ComputerSol

Christophoros Spyretos

2022-10-17

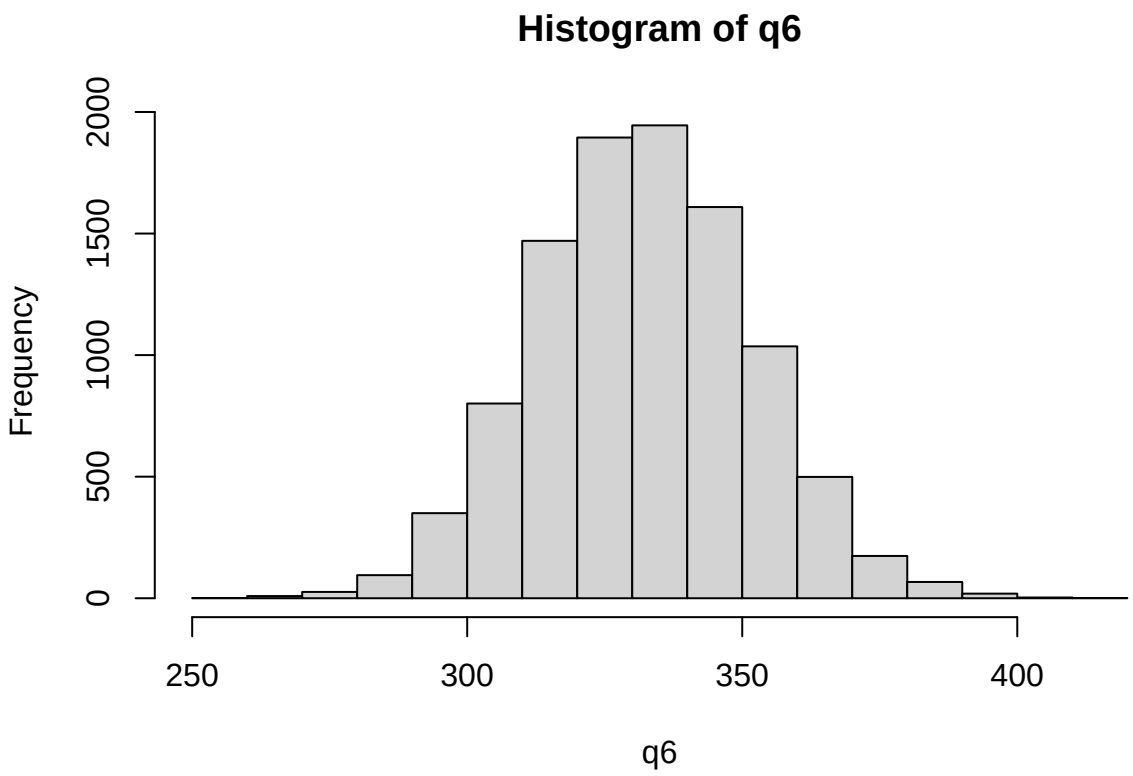
Problem 1

Task a

Hand written exam.

Task b

```
set.seed(12345)

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

theta <- rgamma(nSim, shape = an, rate = bn)
q6 <- rpois(nSim,theta)

hist(q6)
```

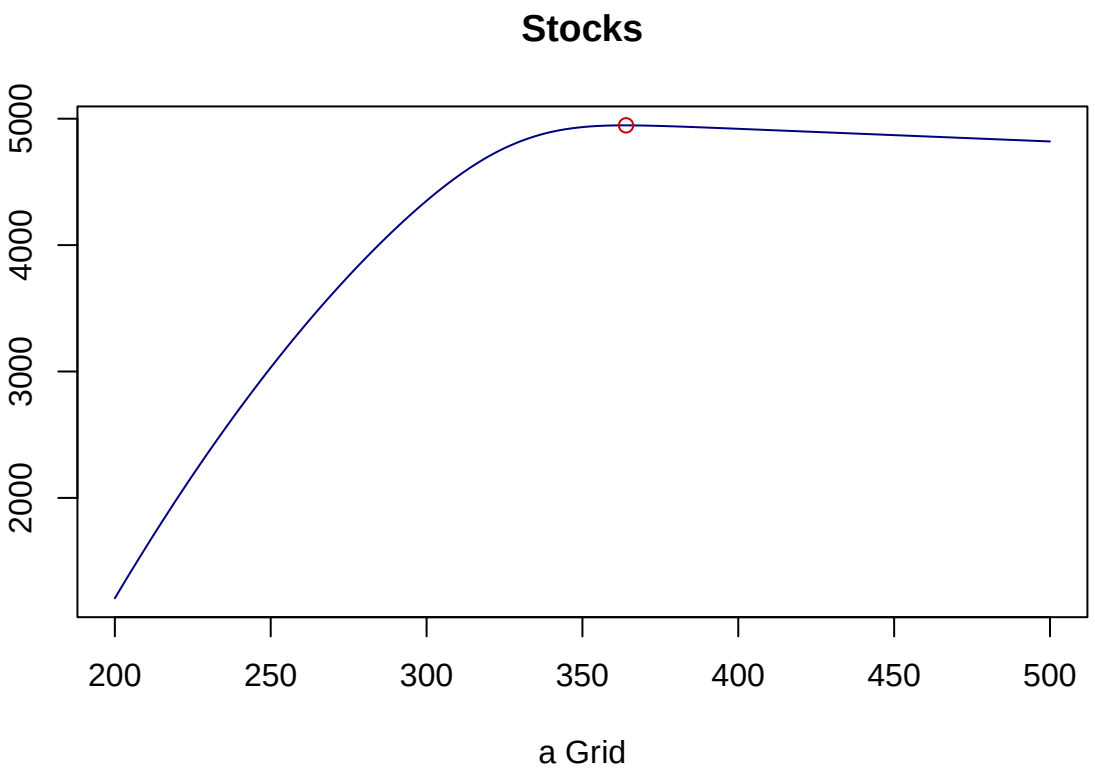



```
prob <- mean(q6>350)
```

The $Pr(Q_6 > 350|q_1, ..., q_5)$ is 0.1799.

Task c

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

aGrid <- seq(200,500,1)
stocks <- matrix(0,length(aGrid),1)

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

aOpt <- aGrid[which.max(stocks)]

plot(aGrid,stocks,type = "l", col = "navy",
     main = "Stocks", xlab = "a Grid", ylab = "")
points(aOpt,mean(utility_function(aOpt,q6)), col = "red3")
```



The optimal number of products in stock is 364.

Problem 2

Task a

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

```
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)

BetasIntervals <- matrix(0,6,2)

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

```
BetasMean <- as.data.frame(BetasMean)
colnames(BetasMean) <- c("Mean Values")
```

```
rownames(BetasMean) <- c("Beta 0","Beta 1","Beta 2",
                         "Beta 3","Beta 4", "Beta 5")
knitr::kable(BetasMean)
```

	Mean Values
Beta 0	-488.8766671
Beta 1	10.7443341
Beta 2	-0.0261714
Beta 3	33.2477399
Beta 4	-0.6083833
Beta 5	-0.1344413

```
BetasIntervals <- data.frame(lower_bound = BetasIntervals[,1], upper_bound = BetasIntervals[,2])
colnames(BetasIntervals) <- c("Lower bound", "Upper bound")
rownames(BetasIntervals) <- c("Beta 0","Beta 1","Beta 2",
                              "Beta 3","Beta 4", "Beta 5")
knitr::kable(BetasIntervals)
```

	Lower bound	Upper bound
Beta 0	-1481.4811157	539.0536182
Beta 1	5.7870148	15.6251718
Beta 2	-0.0364576	-0.0162506
Beta 3	-40.9307993	105.2944392
Beta 4	-1.9988580	0.7934610
Beta 5	-0.3064411	0.0314359

It is the 99% posterior porobability that β_1 belongs between (5.78,15.62).

Task b

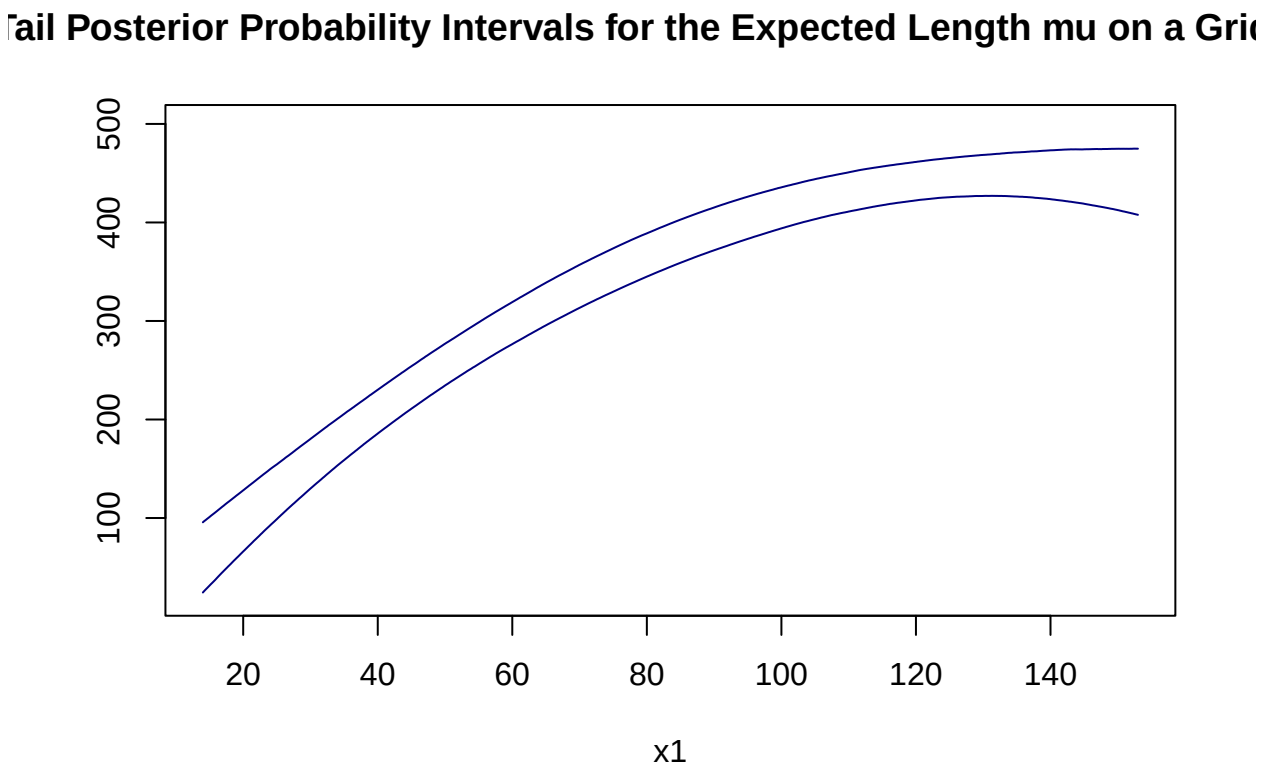
```
Sigma2 <- PostDraws$sigma2Sample

Sigma <- sqrt(Sigma2)

SigmaMean <- mean(Sigma)
SigmaMedian <- median(Sigma)
```

The posterior mean and posterior median of the standard deviation are approximately 40.019 and 39.56.

Task c

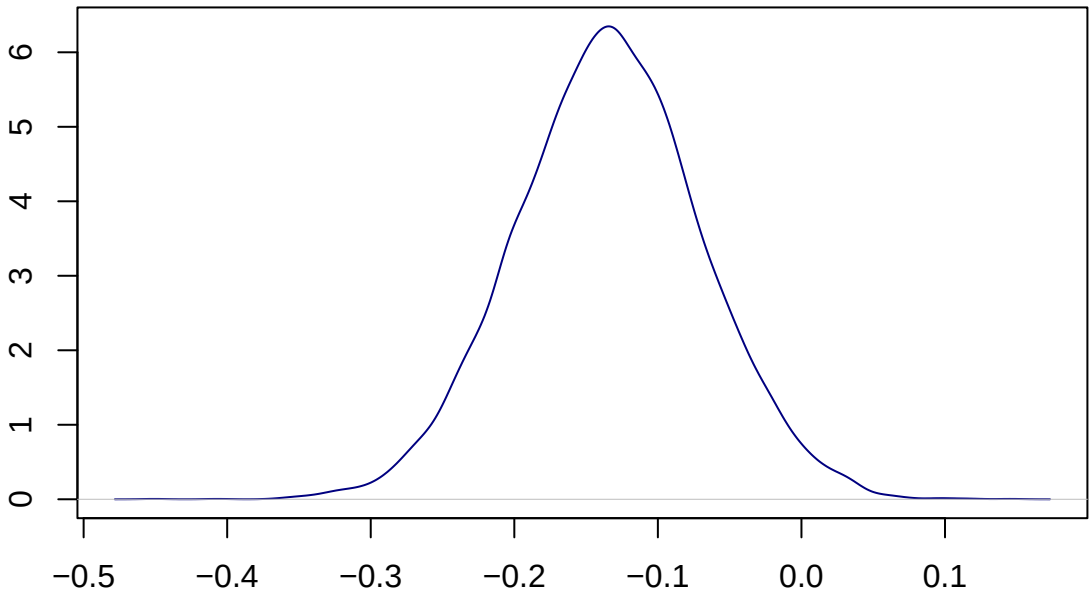
```
x1Grid <- seq(min(X[,2]),max(X[,2]),0.1)

intervals <- matrix(0,length(x1Grid),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 Valu
     xlab = "x1", ylab = "", ylim = c(20,500))
lines(x1Grid,intervals[,2], type = "l", col = "navy")
```



Task d

```
Effect <- Betas[,6]

plot(density(Effect), type = "l", col = "navy",
     main = "Effect on y from x1 Depends on x2",
     xlab = "Beta 5", ylab = "")
```



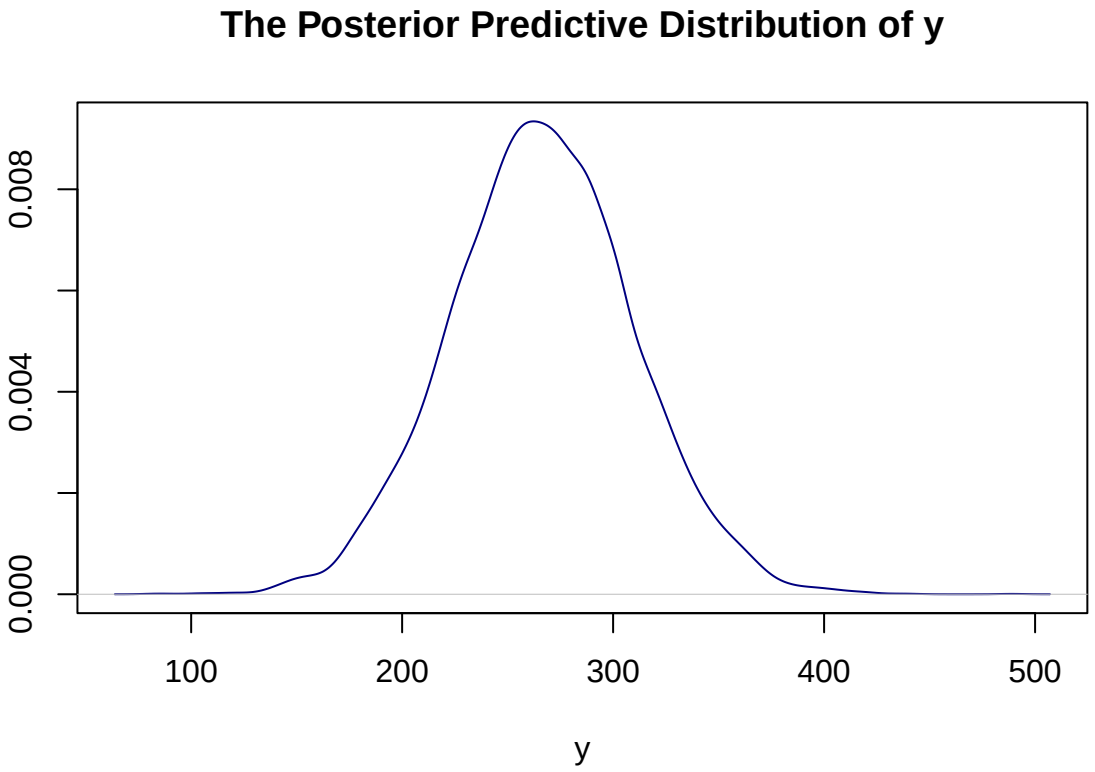

```
Beta6Interval <- quantile(Effect,probs=c(0.025,0.975))
Beta6Interval <- data.frame(lower_bound = Beta6Interval[1], upper_bound = Beta6Interval[2])
colnames(Beta6Interval) <- c("Lower bound", "Upper bound")
rownames(Beta6Interval) <- c("95% Equal Tail Credible Interval")
knitr::kable(Beta6Interval)
```

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

There is a substantial mass probability that the the effect on y from x1 depends on x2. Also, the 95% equal tail posterior probability interval strengthens this assumption.

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, mean = mu, sd = Sigma)

plot(density(y_values), type = "l", col = "navy",
     main = "The Posterior Predictive Distribution of y",
     xlab = "y", ylab = "")
```



Task f

```
set.seed(12345)

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

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

prob <- mean(T_y_rep >= T_y)
```

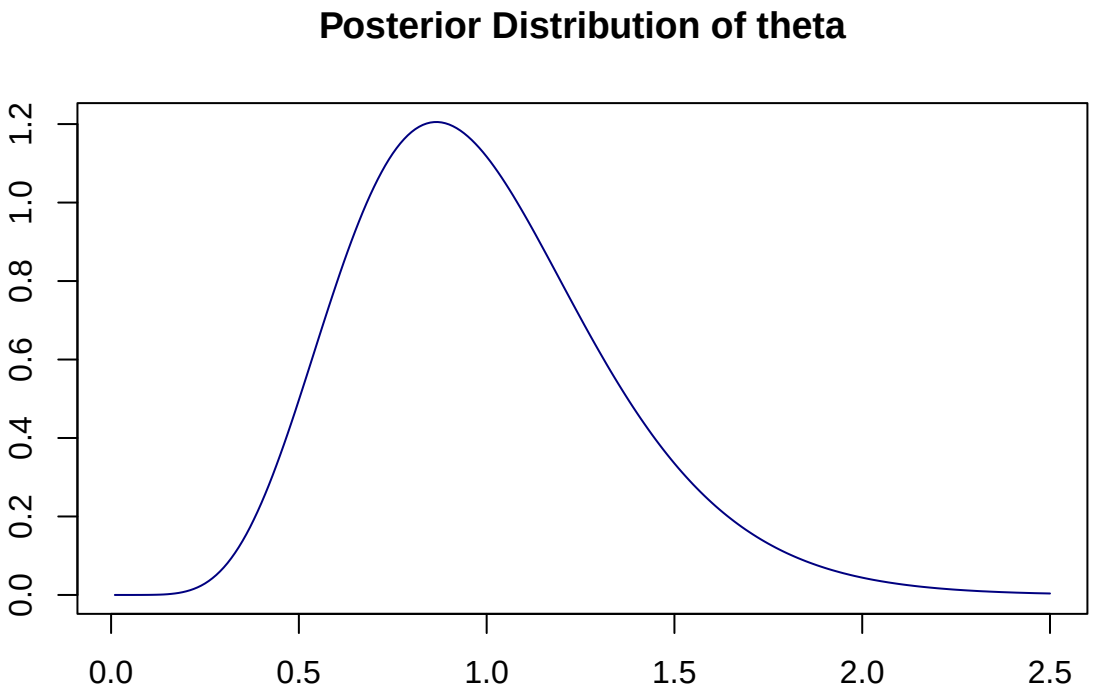
The posterior predictive p-value is approximately 0.99, which means that the model cannot replicate the length of the largest mollusc in this data.

Problem 3

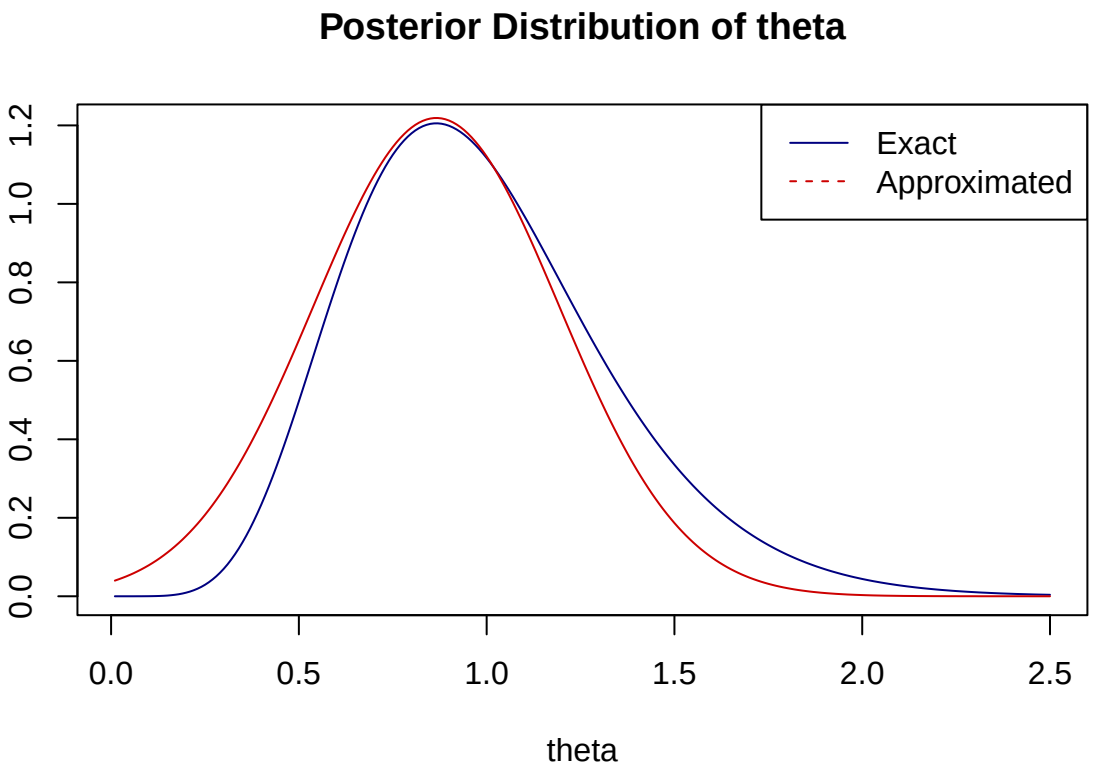
Task d

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

thetaGrid <- seq(0.01,2.5,0.01)
n <- 5
```

```r
sumx3 <- sum(c(0.8^3,1.1^3,0.8^3,0.9^3,1))

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 = "")
```




```r
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)
```



The posterior approximation is not that accurate, the actual posterior is slightly sweked to the right.