Solution to computer exam in Bayesian learning

Bertil Wegmann

2021-08-18

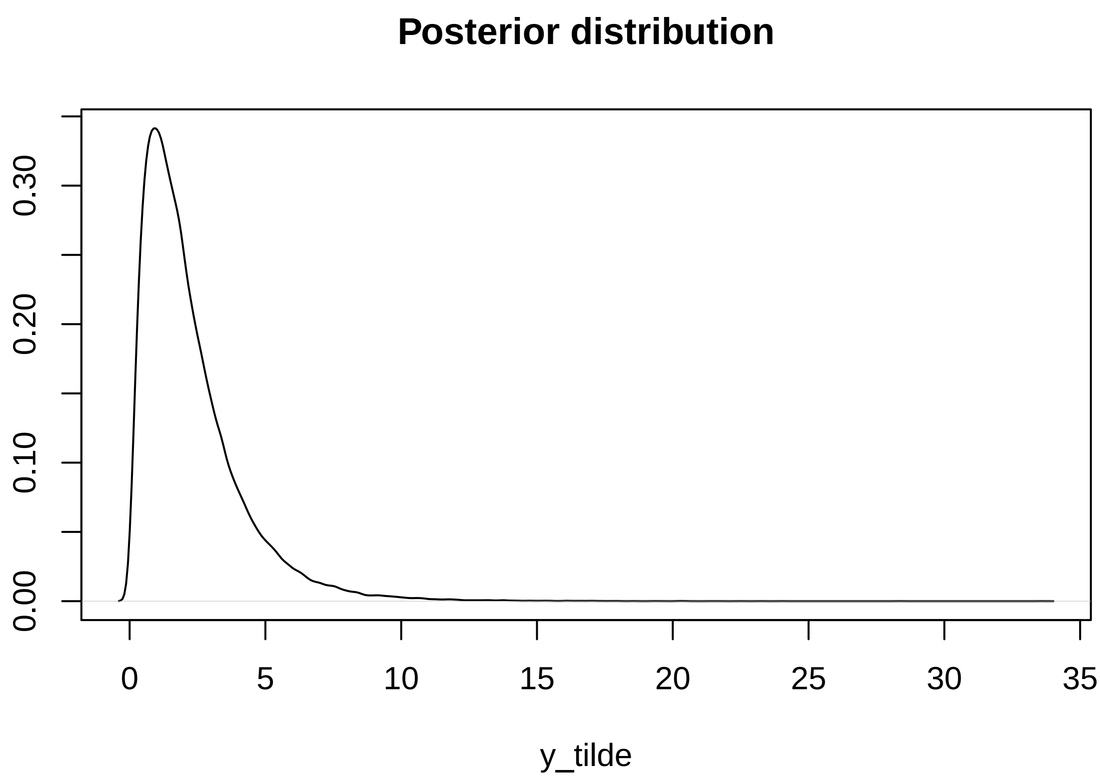
First load all the data into memory by running the R-file given at the exam

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

Problem 1

1b

```
y1 <- c(2.32,1.82,2.40,2.08,2.13)
n <- length(y1)
theta <- rgamma(1e5,shape = 2*n+1,rate = 0.5+sum(y1))
y_tilde <- rgamma(1e5, shape = 2, rate = theta)
plot(density(y_tilde),type="l",main="Posterior distribution",xlab="y_tilde",ylab="")
```



```
mean(y_tilde < 1.9)
```

```
## [1] 0.5359
```

The posterior predictive probability is roughly 0.53. The posterior distribution is plotted above.

1c

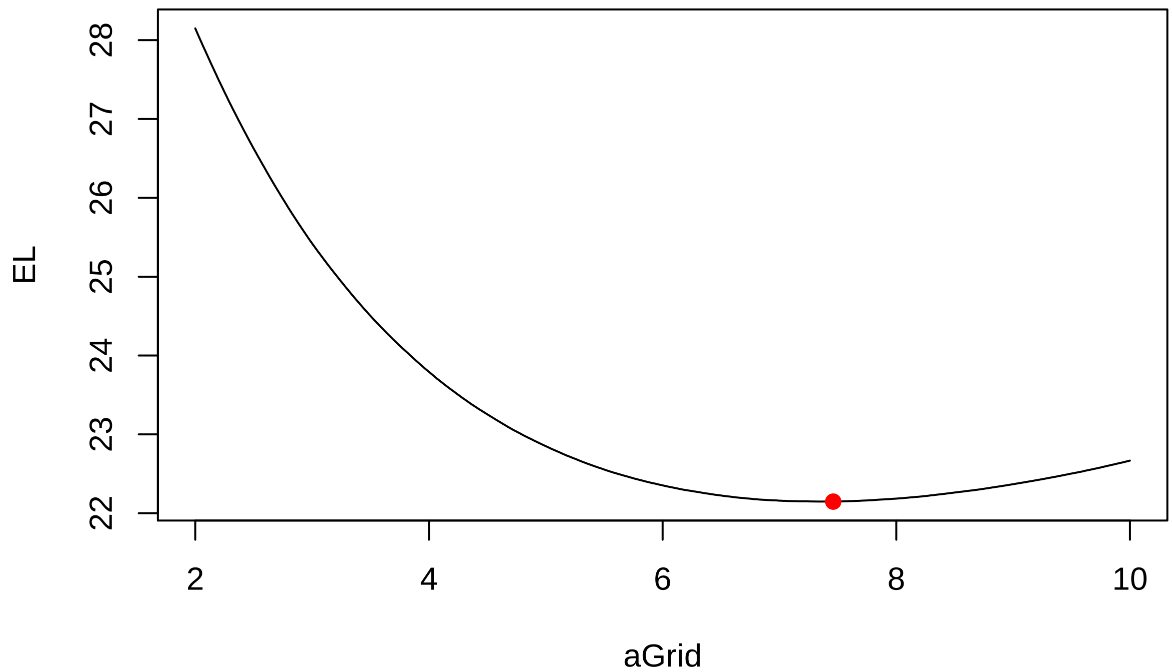
```
nSim <- 1e5
nWeeks <- 30
WeeklyWeights <- matrix(NA,nSim,nWeeks)
for (i in 1:nSim){
  thetas <- rgamma(nWeeks,shape = 2*n+1,rate = 0.5+sum(y1))
  WeeklyWeights[i,] <- t(rgamma(nWeeks, shape = 2, rate = thetas))
}
ExceedingWeeks <- rowSums(WeeklyWeights > 2.4)
mean(ExceedingWeeks)
```

```
## [1] 10.50021
```

The expected number of weeks is roughly 10.5.

1d

```
ExpectedLoss <- function(a, WeeklyWeights){
  EL <- a + mean(rowSums(WeeklyWeights > 0.9*log(a)))
  return(EL)
}
aGrid <- seq(2,10,by = 0.01)
EL <- rep(NA,length(aGrid),1)
count <- 0
for (a in aGrid){
  count <- count + 1
  EL[count] = ExpectedLoss(a, WeeklyWeights)
}
plot(aGrid, EL, type = "l")
aOpt = aGrid[which.min(EL)] # This is the optimal a
points(aOpt,ExpectedLoss(a=aOpt, WeeklyWeights), col = "red",pch=19)
```




```
aOpt
```

```
## [1] 7.46
```

The optimal build cost (a) is roughly 7.5.

Problem 2

2a

```
mu_0 <- as.vector(rep(0,8))
Omega_0 <- (1/9)*diag(8)
v_0 <- 1
sigma2_0 <- 9
nIter <- 10000
library(mvtnorm)
```

```
## Warning: package 'mvtnorm' was built under R version 4.0.5
```

```
X <- as.matrix(X)
PostDraws <- BayesLinReg(y, X, mu_0, Omega_0, v_0, sigma2_0, nIter)
Betas <- PostDraws$betaSample
quantile(Betas[,2],probs=c(0.005,0.995))
```

```
##       0.5%      99.5%
## -0.3285869  1.8429748
```

It is 99 % posterior probability that beta_1 is on the interval (-0.33,1.84).

2b

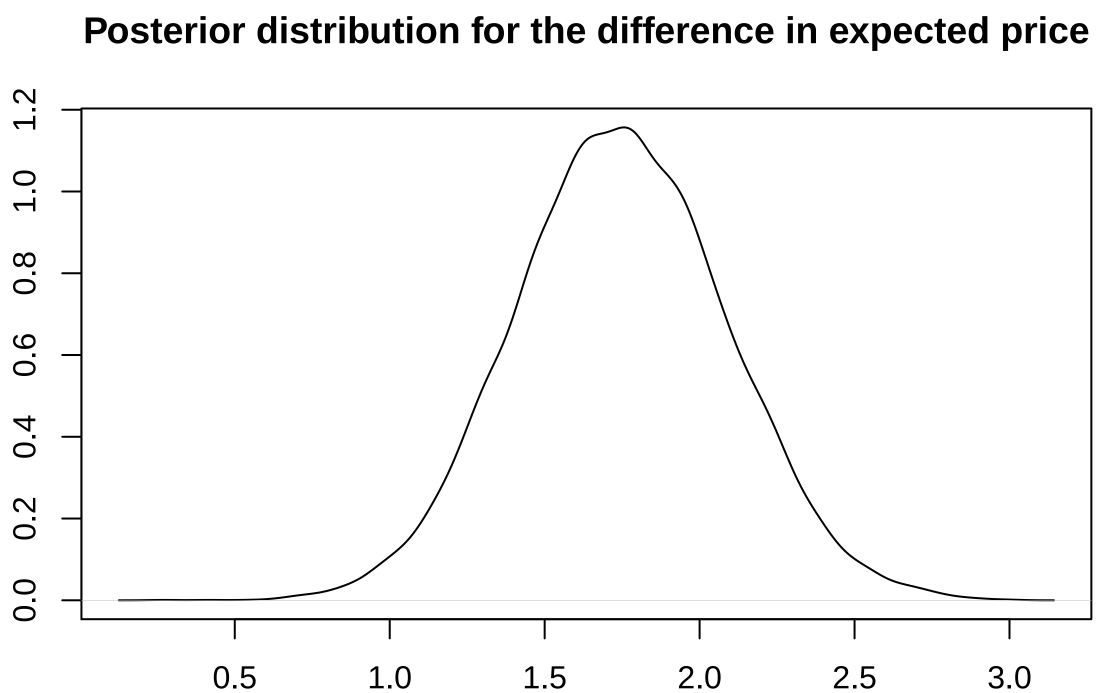
```
Mu_draws <- Betas%*%as.vector(c(1,1,1,0.5,0,1,0,1))
Sigma_draws <- sqrt(PostDraws$sigma2Sample)
median(Sigma_draws/Mu_draws)
```

```
## [1] 1.825709
```

The median of CV is given by roughly 1.83.

2c

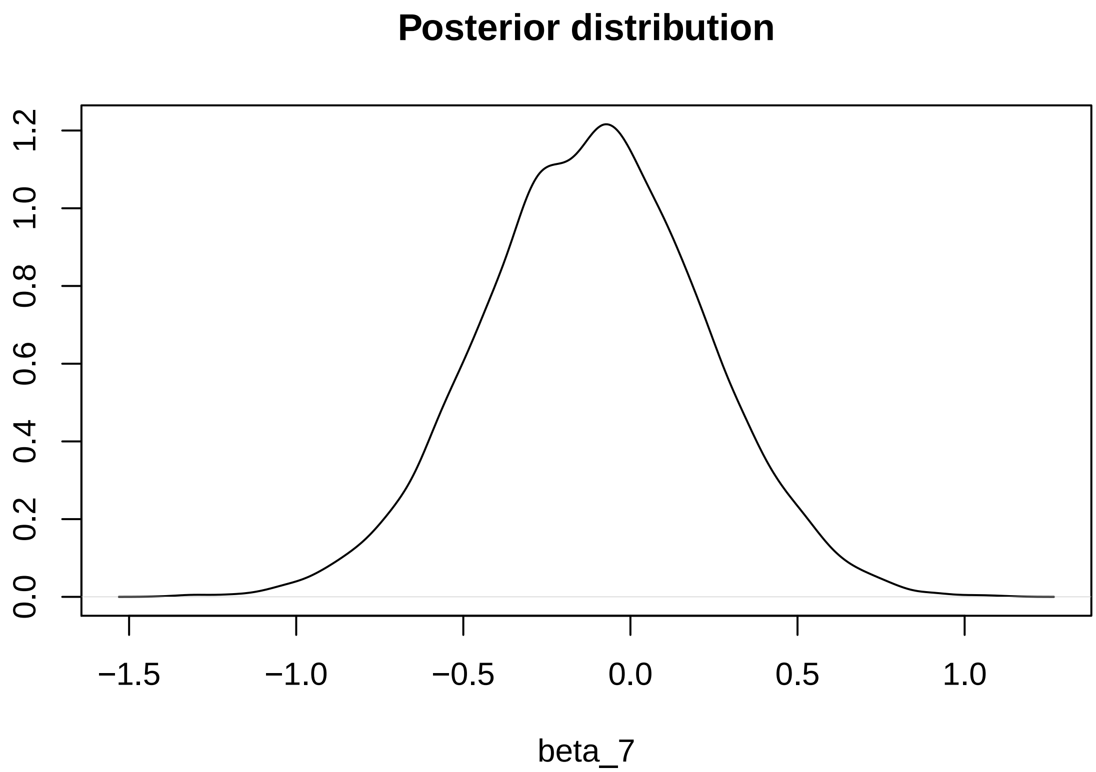
```
Diff_Exp_Price <- (Betas[,5]+Betas[,7])-(Betas[,6]+Betas[,8])
plot(density(Diff_Exp_Price),type="l",
     main="Posterior distribution for the difference in expected price",
     xlab="",ylab="")
```




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

```
##     2.5%    97.5%
## 1.087836 2.414187
```

```
plot(density(Betas[,8]),type="l",main="Posterior distribution",xlab="beta_7",ylab="")
```

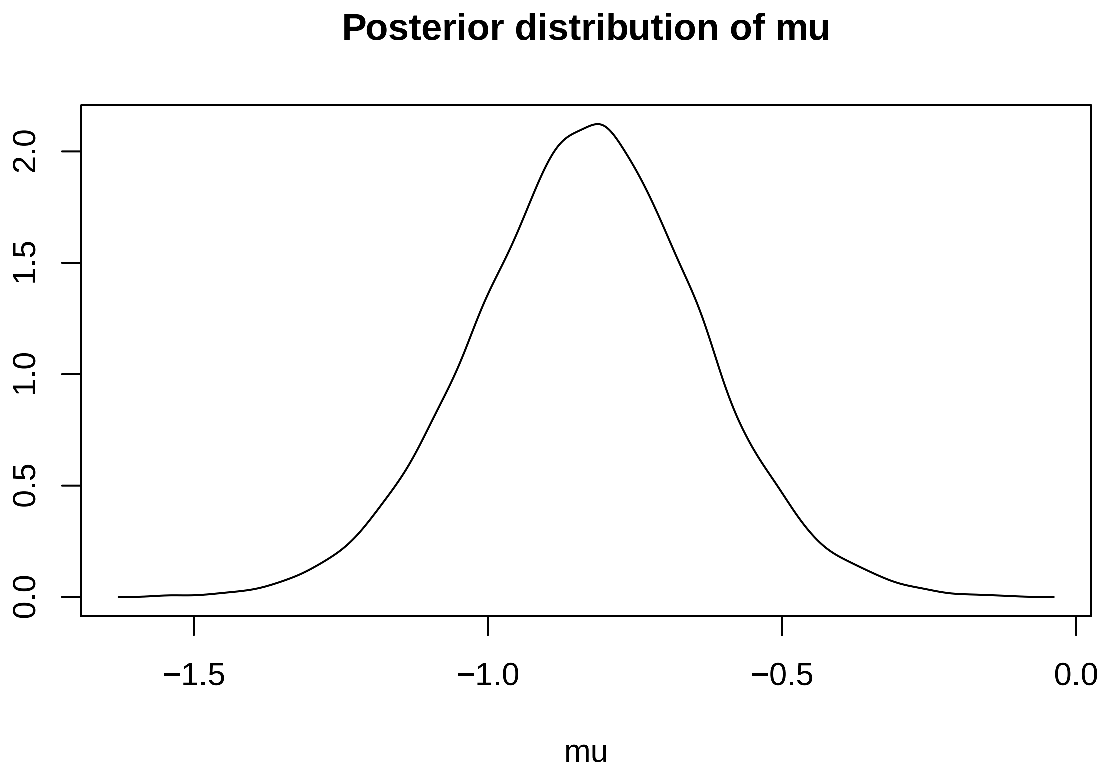


There is substantial probability mass that the expected price is higher for apartments in the inner city compared to the south side of the city. A 95 % equal tail credible interval for the difference in expected selling price when x1=1 is equal to [1.09,2.41], which further points in this direction.

The posterior distribution for the difference in slopes of x1 between the south side and neither inner city nor south side (beta_7) has substantial probability mass on both sides of 0, so that the effect from x1 on the selling price y is not likely to be different between the two regions.

2d

```
Mu_draws <- Betas%*%as.vector(c(1,-0.5,-0.5,0,0,1,0,-0.5))
plot(density(Mu_draws),type="l",main="Posterior distribution of mu",xlab="mu",ylab="")
```




```
mean(Mu_draws>0)
```

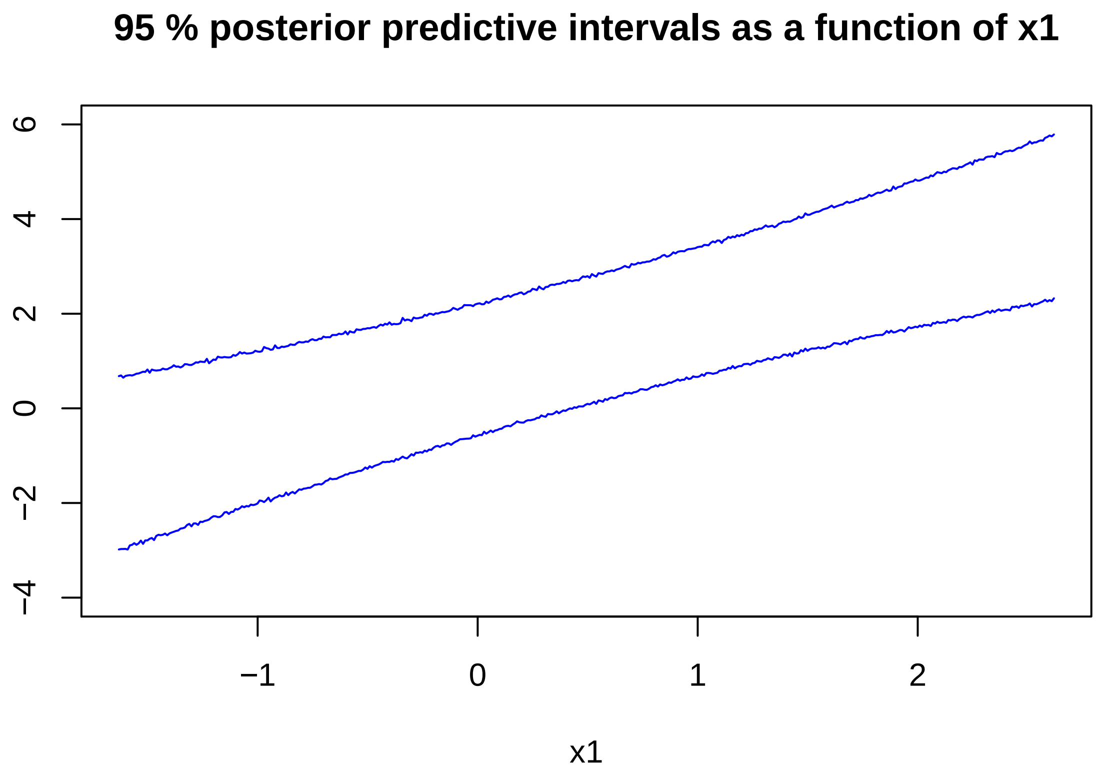
```
## [1] 0
```

The posterior distribution of mu is plotted above. The posterior probability is 0.

2e

```
x1_grid <- seq(min(X[,2]),max(X[,2]),0.01)
ypred_draws <- matrix(0,length(x1_grid),2)
for (ii in 1:length(x1_grid)){
  CurrMu <- Betas%*%as.vector(c(1,x1_grid[ii],1,0.5,1,0,x1_grid[ii],0))
  ypred_draws[ii,] <- quantile(rnorm(nIter,CurrMu,Sigma_draws),probs=c(0.025,0.975))
}

plot(x1_grid,ypred_draws[,1],"n",
     main="95 % posterior predictive intervals as a function of x1",
     xlab="x1", ylab="",ylim=c(-4,6))
lines(x1_grid,ypred_draws[,1],col="blue")
lines(x1_grid,ypred_draws[,2],col="blue")
```



The posterior predictive intervals as a function of x1 are plotted above.

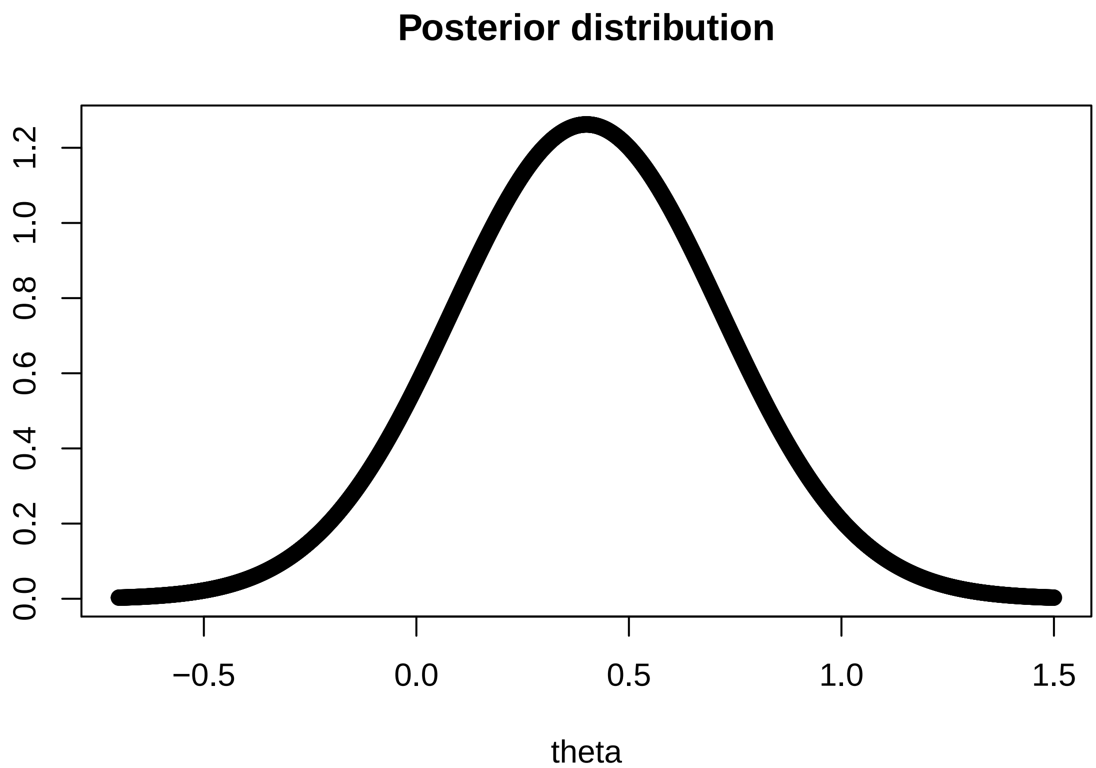
Problem 3

3d

```
LogPost <- function(theta,n,SumLogx){

  logLik <- -n*theta**2 + 2*theta*SumLogx;

  return(logLik)
}
theta_grid <- seq(-0.7,1.5,0.001)
PostDens_propto <- exp(LogPost(theta_grid,5,2))
PostDens <- PostDens_propto/(0.001*sum(PostDens_propto))
plot(theta_grid,PostDens,main="Posterior distribution",xlab="theta", ylab="")
```



The posterior distribution is given above.