mysol

Isak Berntsson

8/15/2022

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

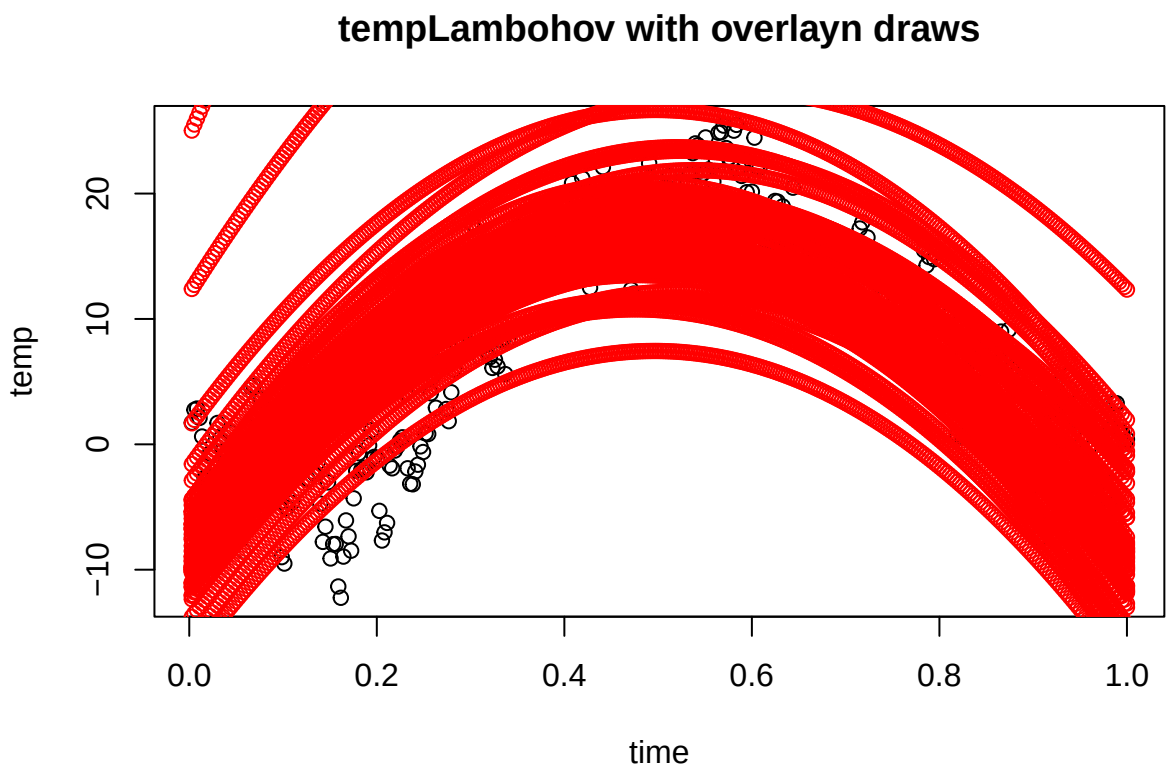
```
library(mvtnorm)
library(geoR)
```

```
## Warning: package 'geoR' was built under R version 4.1.3
```

```
## ----------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.9-2 (built on 2022-08-09) is now loaded
## ----------------------------------------------------------
```

```
#loading data
dat = read.table("TempLambohov.txt", header = T)
#head(dat)
df = data.frame(const = 1, time = dat$time,time2 = dat$time^2, temp = dat$temp)
#head(df)
#hyperparams
mu_0 = c(-10, 100, -100)
omega_0 = .3 * diag(3)
v0 = 3
sigma2_0 = 2
#drawing from prior
numdraws = 50
#draw sigmas
Xdraws = rchisq(numdraws,v0)
sigma2 = v0*sigma2_0/Xdraws
#draw betas
betas = matrix(nrow=numdraws, ncol=length(mu_0))

for (i in 1:numdraws){
  betas[i,] = rmvnorm(1,mean=mu_0, sigma = sigma2[i]*solve(omega_0))

}
```

```
betas = as.matrix(betas)

X = as.matrix(df[,c("const","time","time2")])
y = df$temp
plot(x=df$time,y=df$temp, main="tempLambohov with overlayn draws", ylab="temp", xlab="time")


linpreds = matrix(nrow=numdraws ,ncol=length(y))

for (i in 1:numdraws){
  linpreds[i,] = t(X%*%betas[i,])
  points(x=df$time,y=linpreds[i,],col="red")

}
```



1.b

```
posterior <- function(numdraws, X, y, mu_0, sigma2_0, v_0, omega_0){
  numrows = length(y) #n
  numcovs = length(mu_0) #k


  XtX = t(X)%*%X #helper var
```

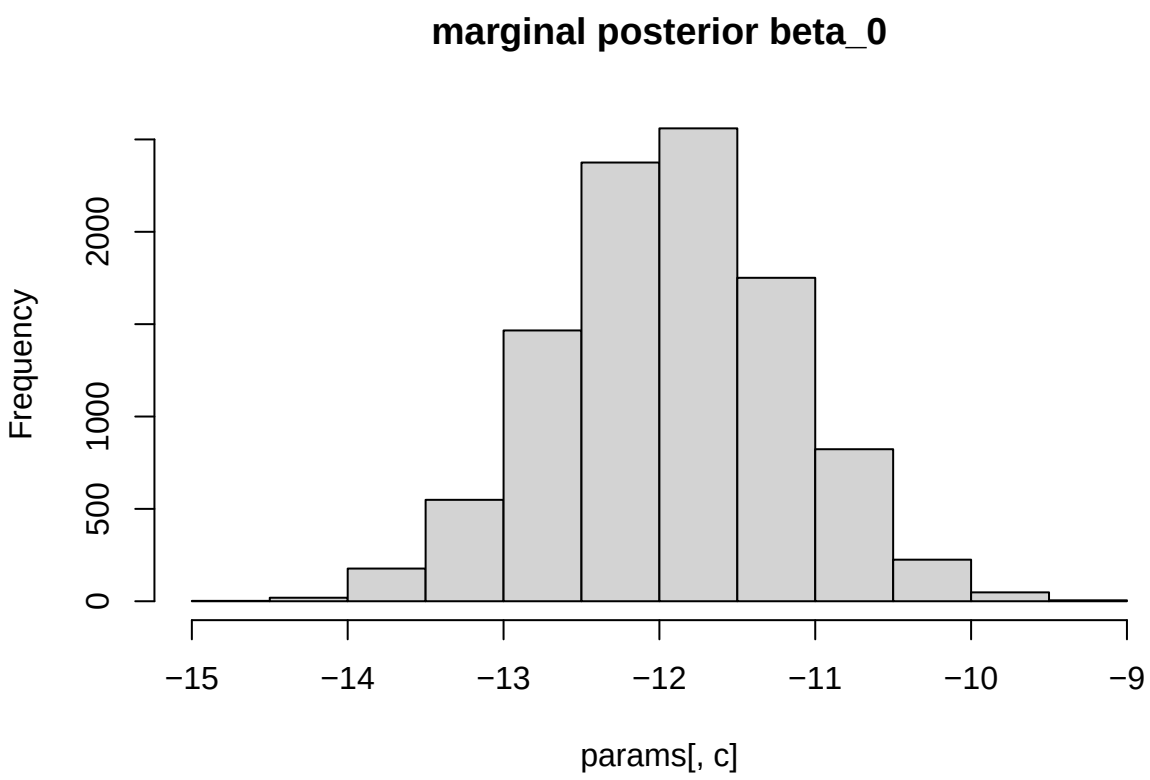
```
  beta_hat = solve(XtX,t(X)%*%y)

  mu_n = solve(XtX + omega_0) %*% (XtX%*%beta_hat + omega_0%*%mu_0)
  omega_n = XtX + omega_0
  v_n = v_0 + numrows

  vsn_2 = v_0*sigma2_0 +(t(y)%*%y + t(mu_0)%*%omega_0%*%mu_0 - t(mu_n)%*%omega_n%*%mu_n )


  Xdraws = rchisq(numdraws,v_n)
  sigma2 =vsn_2/Xdraws

  omega_n_inv = solve(omega_n)

  param_draws = matrix(nrow = numdraws, ncol = numcovs + 1 )
  colnames(param_draws) = c("beta_0","beta_1", "beta_2", "sigma2")
  for (i in 1:numdraws){
    draw = rmvnorm(1,mean = mu_n , sigma2[i]*omega_n_inv)

    param_draws[i,1:3] = draw
    }
  param_draws[,4] = sigma2

  print(beta_hat)


  return(param_draws)


}
```

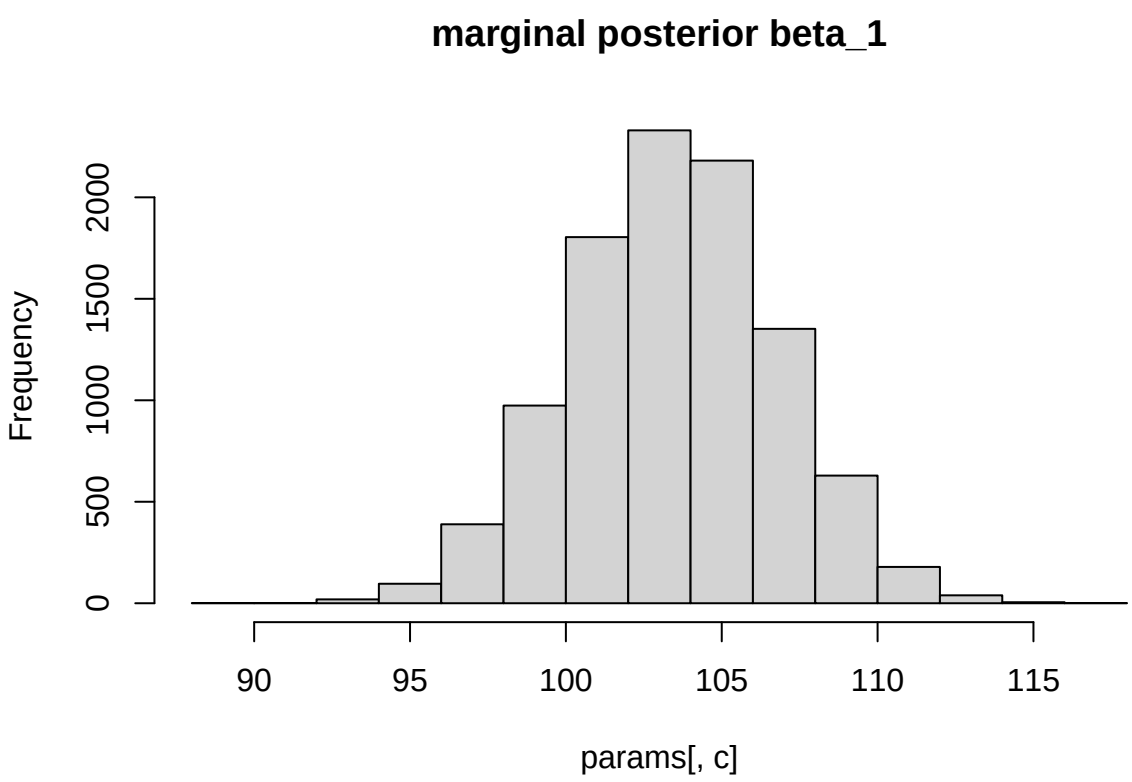
#1.b.i

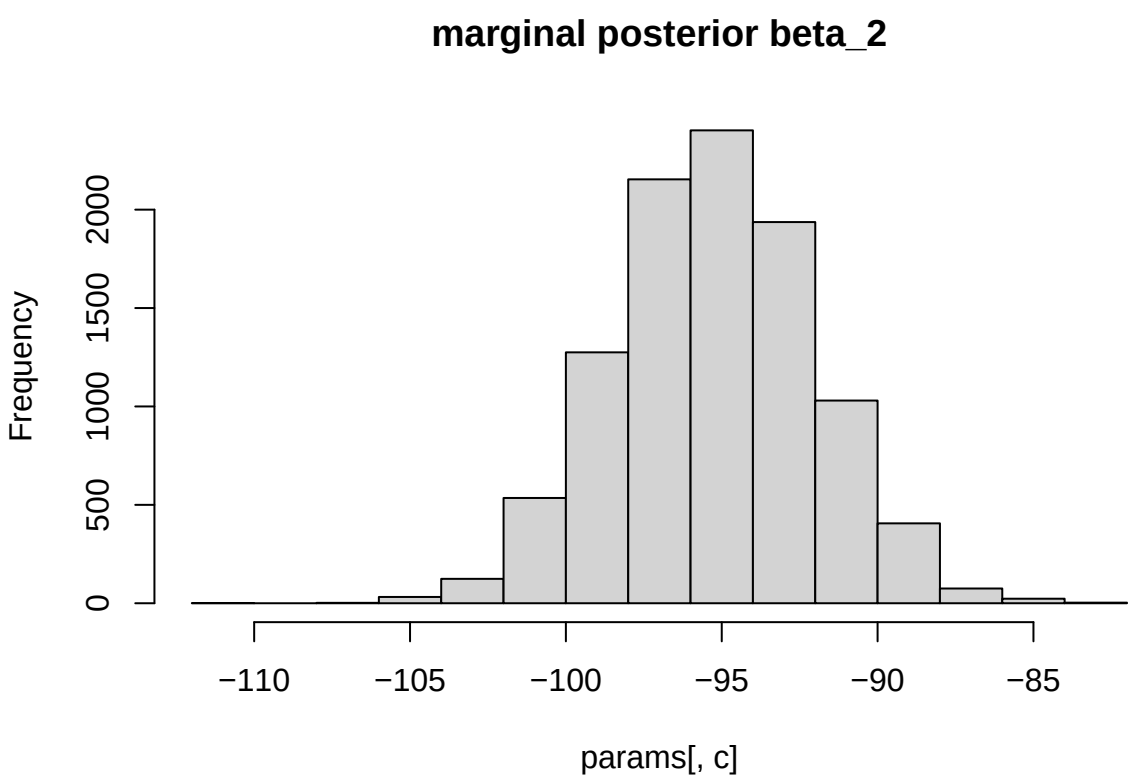
```
params = posterior(1e4, X, y, mu_0, sigma2_0, v0, omega_0)
```

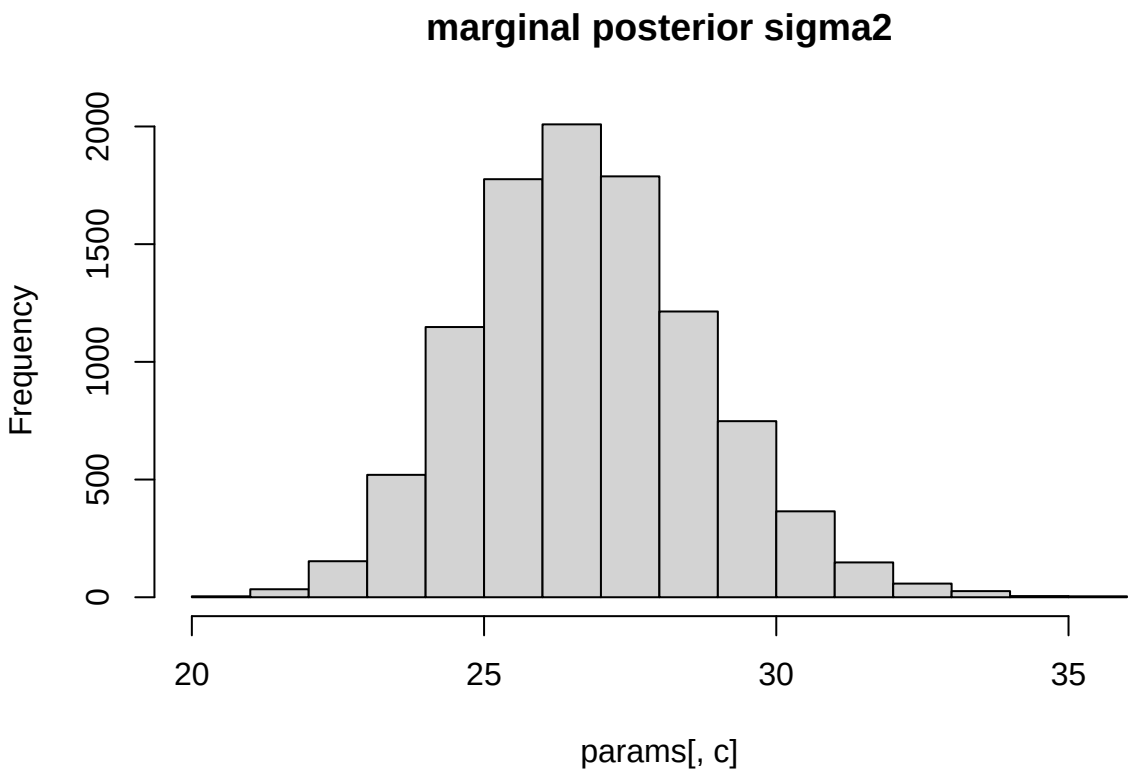
```
## Warning in vsn_2/Xdraws: Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.

##             [,1]
## const -11.92736
## time  103.41789
## time2 -95.20653
```

```
for (c in colnames(params)){
  hist(params[,c], main=paste("marginal posterior ",c,sep=""))
}
```


marginal posterior beta_0
Frequency
0
500
1000
2000
−15
−14
−13
−12
−11
−10
−9
params[, c]

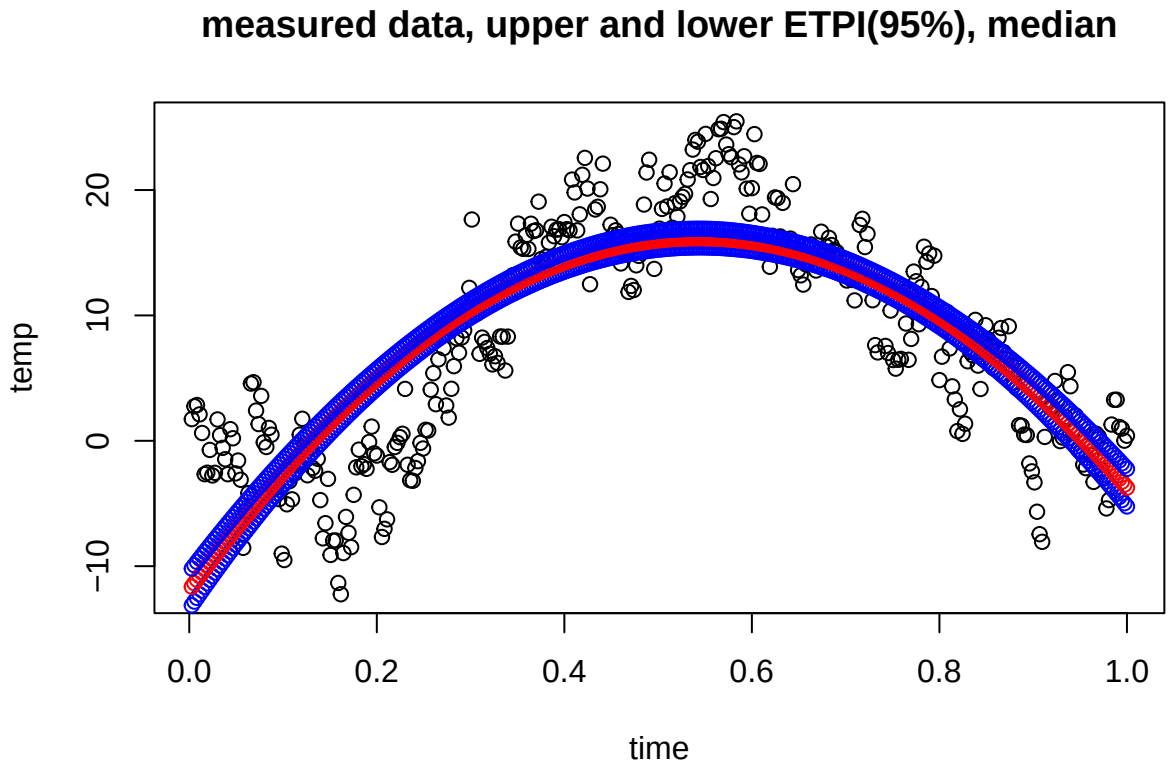

marginal posterior beta_1
Frequency
0
500
1000
1500
2000
90
95
100
105
110
115
params[, c]


marginal posterior beta_2
Frequency
0
500
1000
1500
2000
−110
−105
−100
−95
−90
−85
params[, c]



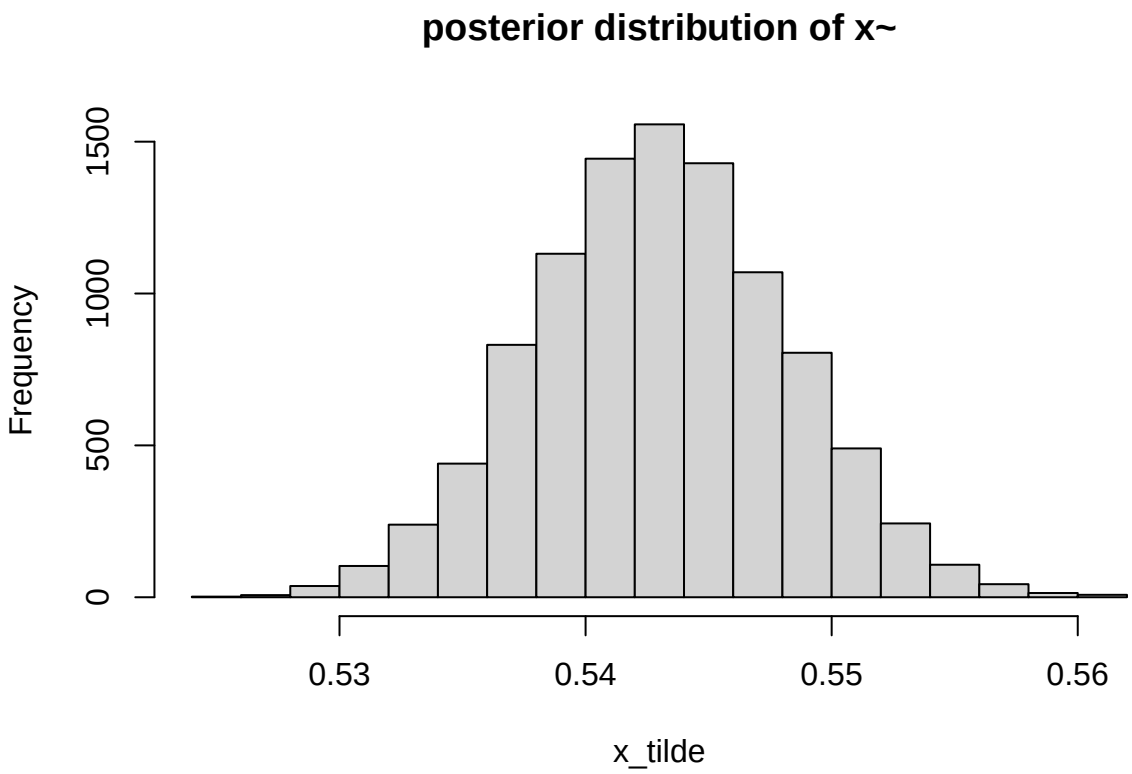
#1.b.ii

```
predicts = X %*% t(params[,1:3])


intervals = matrix(nrow=dim(predicts)[1],ncol=3)
colnames(intervals) = c("2.5%","median","97.5%")
for (i in 1:dim(predicts)[1]){
  sample = predicts[i,]
  intervals[i,c(1,3)] = quantile(sample, probs=c(0.025,0.975))
  intervals[i,2] = median(sample)
}

plot(x=df$time,y=df$temp, main="measured data, upper and lower ETPI(95%), median", ylab="temp",xlab = "
points(x=df$time,y=intervals[,1], col="blue")
points(x=df$time,y=intervals[,2], col="red")
points(x=df$time,y=intervals[,3], col="blue")
```



#1.c

```
x_tilde = -params[,2]/(2*params[,3])

hist(x_tilde,main="posterior distribution of x~")
```



#1.d great text stuff here

#2.a