Bayesian Modelling Problem Sheet 6

Radon is a natural radioactive gas that is usually found in rocks and soils. Radon gas, which is for most people the single largest source of radiation, is known to be a source of lung cancer when breathed in high concentration. In particular, according to the United States Environmental Protection Agency (EPA), radon is the most frequent cause of lung cancer among non-smokers. Due to local dierences in geology, the level of the radon gas hazard diers from location to location, and it is therefore important to identify areas with high radon exposures to help home-owners to make decisions about measuring or remediating the radon in their house. More information about radon and its health eects can be found here: www.radon.com/radon_facts. The dataset "radon.txt" contains radon measurements taken by the EPA in 917 randomly chosen houses located in 84 (out of 87) counties in the state of Minnesota.

```
radon <- read.table('DataSheet6/radon.txt', sep='')
set.seed(333)
library(MASS)
library(rjags)
```

```
## Warning: package 'rjags' was built under R version 3.5.3

## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs
```

```
library(coda)
load.module('glm')
```

```
## module glm loaded
```

The first column of the dataset contains the 917 measurements of radon (in log) while the second column indicates if a measurement was taken in a basement (value 1) or in a first floor (value 0). This information is important to understand the level of radon in a given house since, as the radon comes from the soil, it enters more easily in a house when it is built into the ground. Columns three and four provide respectively the county name and the county index of the sampled house and the last column of the dataset gives a measurement of soil uranium (the ultimate source of radon) taken at the county level.

The objective of this assignment is to use a Bayesian approach to solve the following two problems:

P1 Predict the log-radon level of a house with a basement for each of the 84 counties in the dataset. P2 Predict the log-radon level of a house with no basement and of a house with a basement in the county of Cook, for which no measurements have been taken.

2. Estimate your model. In particular:

a) (15 marks) Implement your model in JAGS.

```
#write model to radon.bug
cat('
model{
for(j in 1:M){
theta[j] ~ dnorm(alpha*uu[j], sigma2^(-2))
}
for(i in 1:n){
x[i] ~ dnorm( (theta[county.index[i]]) + beta*B[i], sigma^(-2))
}

alpha ~ dnorm(0,0.0001)
```

```
beta ~ dnorm(0,0.001)
sigma ~ dunif(0,100)
sigma2 ~ dunif(0,10000)
}', file='radon.bug')
```

```
#specify determined parts
radon_data<-list(n=nrow(radon), M=max(radon$county.index), x=radon$log.radon, county.index = radon$coun
B=radon$floor, uu = unique(radon$uu))
#specify prior points
radon_inits <- list(alpha=0,beta=0,sigma=1,sigma2=1)
```

```
#construct model in jags
radon_mu<-jags.model('radon.bug',inits = radon_inits,data=radon_data)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 917
##    Unobserved stochastic nodes: 88
##    Total graph size: 3164
##
## Initializing model
```

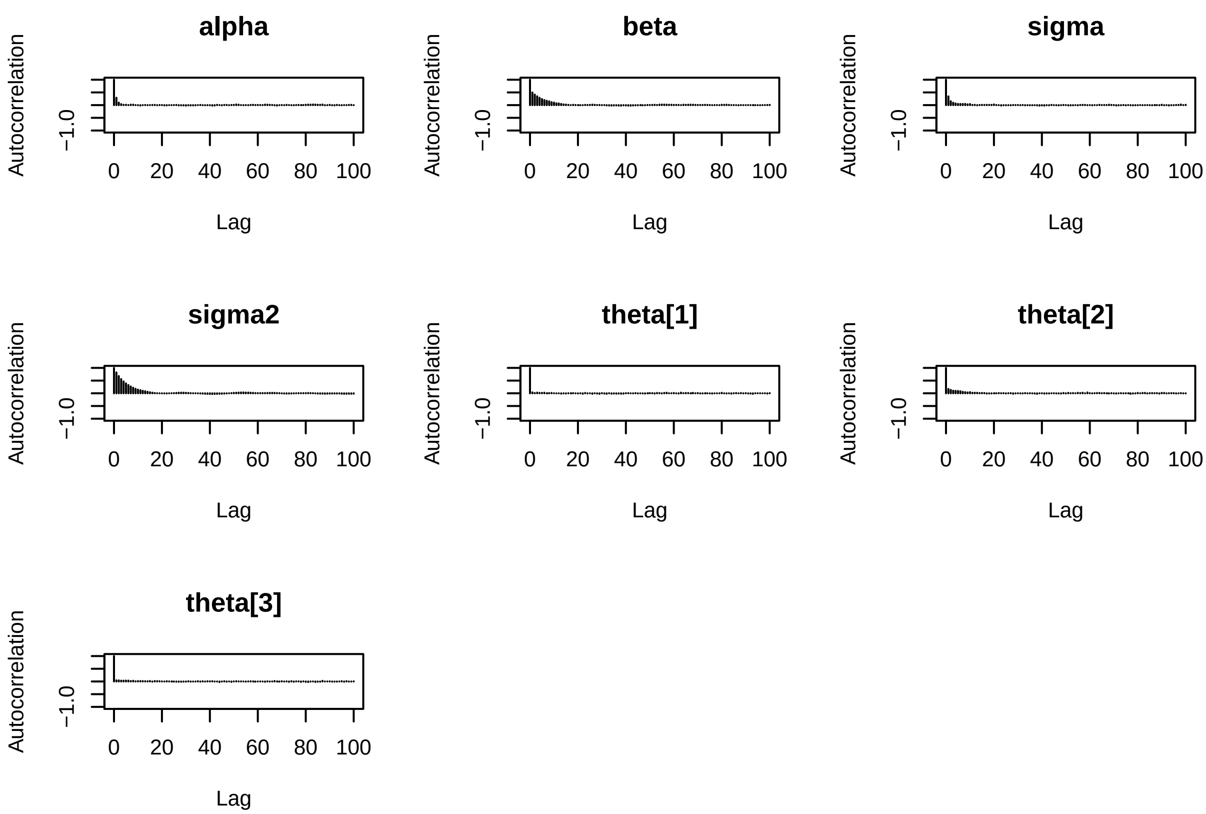
```
#verify we are sampling correct quantities, and using correct samplers
list.samplers(radon_mu)
```

```
## $`bugs::ConjugateNormal`
## [1] "alpha"
##
## $`glm::Generic`
##  [1] "beta"      "theta[1]"  "theta[2]"  "theta[3]"  "theta[4]"
##  [6] "theta[5]"  "theta[6]"  "theta[7]"  "theta[8]"  "theta[9]"
## [11] "theta[10]" "theta[11]" "theta[12]" "theta[13]" "theta[14]"
## [16] "theta[15]" "theta[16]" "theta[17]" "theta[18]" "theta[19]"
## [21] "theta[20]" "theta[21]" "theta[22]" "theta[23]" "theta[24]"
## [26] "theta[25]" "theta[26]" "theta[27]" "theta[28]" "theta[29]"
## [31] "theta[30]" "theta[31]" "theta[32]" "theta[33]" "theta[34]"
## [36] "theta[35]" "theta[36]" "theta[37]" "theta[38]" "theta[39]"
## [41] "theta[40]" "theta[41]" "theta[42]" "theta[43]" "theta[44]"
## [46] "theta[45]" "theta[46]" "theta[47]" "theta[48]" "theta[49]"
## [51] "theta[50]" "theta[51]" "theta[52]" "theta[53]" "theta[54]"
## [56] "theta[55]" "theta[56]" "theta[57]" "theta[58]" "theta[59]"
## [61] "theta[60]" "theta[61]" "theta[62]" "theta[63]" "theta[64]"
## [66] "theta[65]" "theta[66]" "theta[67]" "theta[68]" "theta[69]"
## [71] "theta[70]" "theta[71]" "theta[72]" "theta[73]" "theta[74]"
## [76] "theta[75]" "theta[76]" "theta[77]" "theta[78]" "theta[79]"
## [81] "theta[80]" "theta[81]" "theta[82]" "theta[83]" "theta[84]"
##
## $`base::RealSlicer`
## [1] "sigma"
##
## $`base::RealSlicer`
## [1] "sigma2"
```

b) (15 marks) Simulate a trajectory of length T and justify your choice for T

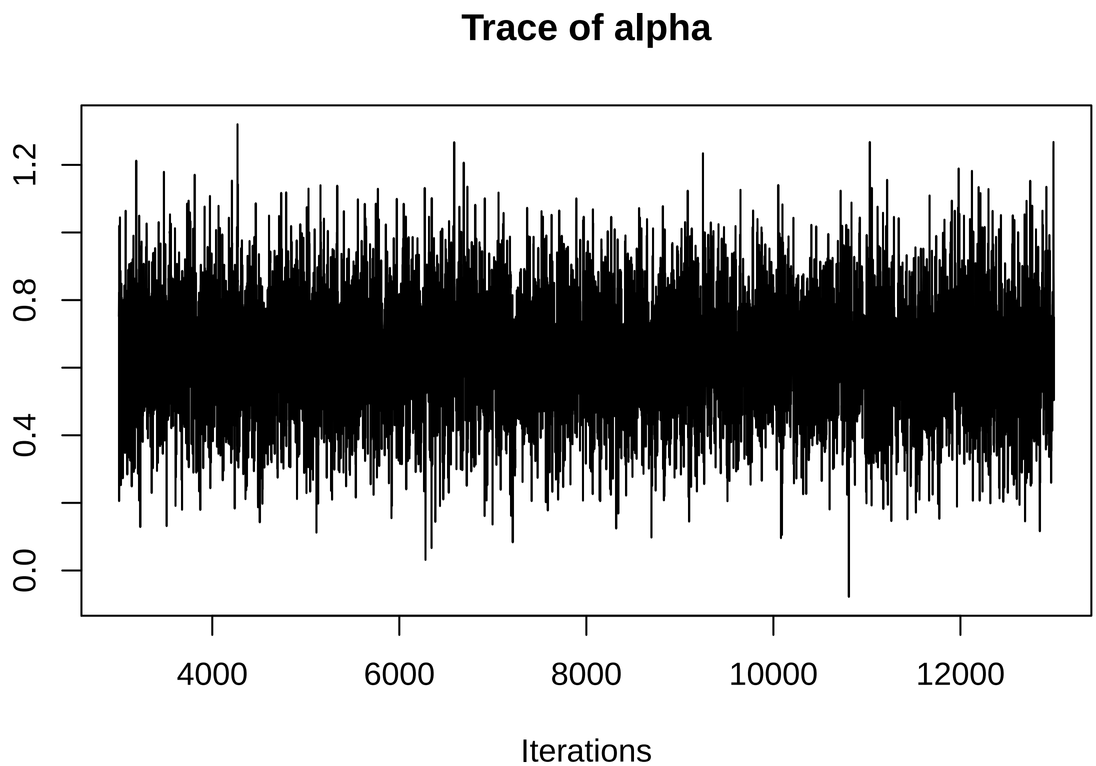
```r
#set burn-in length
Burn <- 2000
#set trajectory length
Tr <- 10000
#update chain by burn-in length
update(radon_mu, n.iter = Burn)
#sample from distribution with correct quantities
sample<-coda.samples(radon_mu, variable.names = c("alpha","theta","beta","sigma","sigma2"), n.iter=Tr)
```

```r
#plot autocorrelations for parameters to verify convergence
autocorr.plot(sample[,1:7], 100)
```

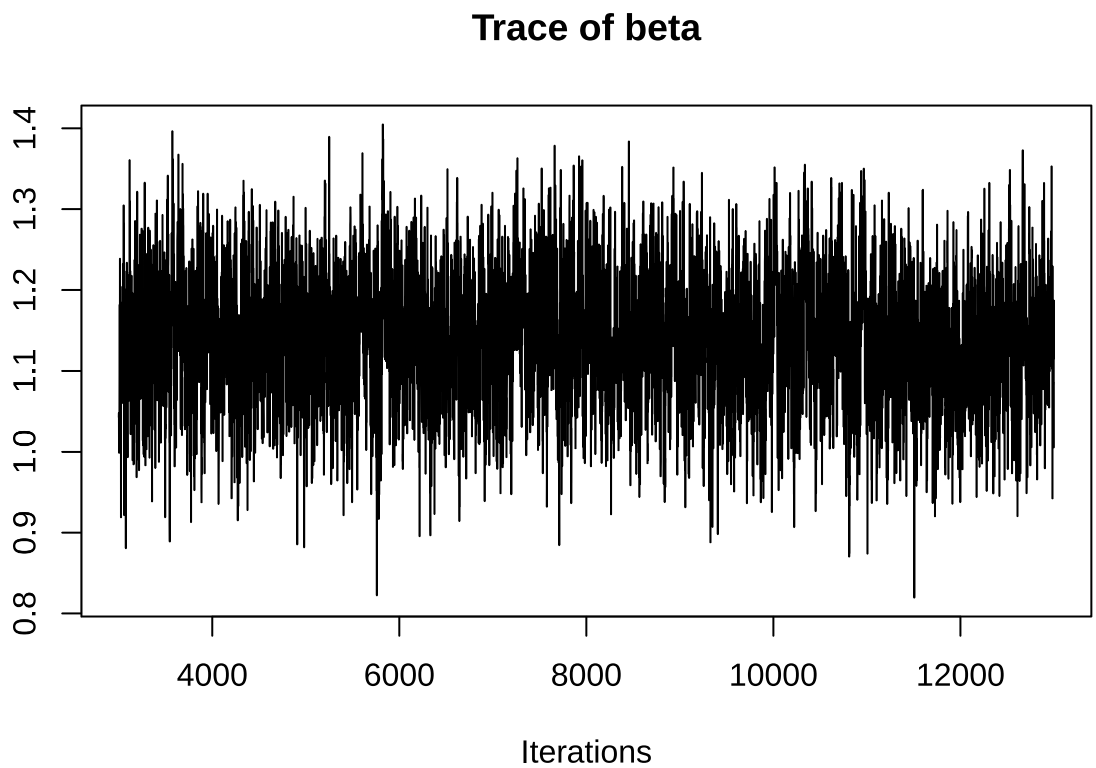



```r
#These have rapidly-declining autocorrelation, indicating the chain has converged for all parameters
```

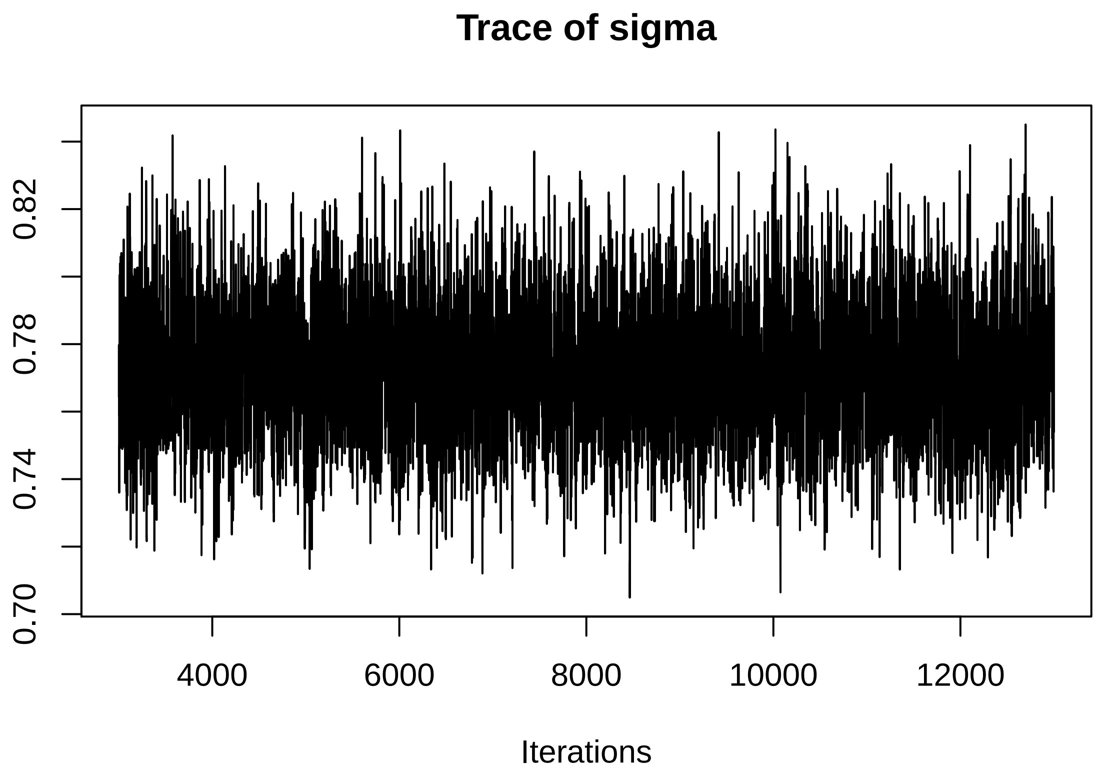
```r
traceplot(sample[,1:7])
```

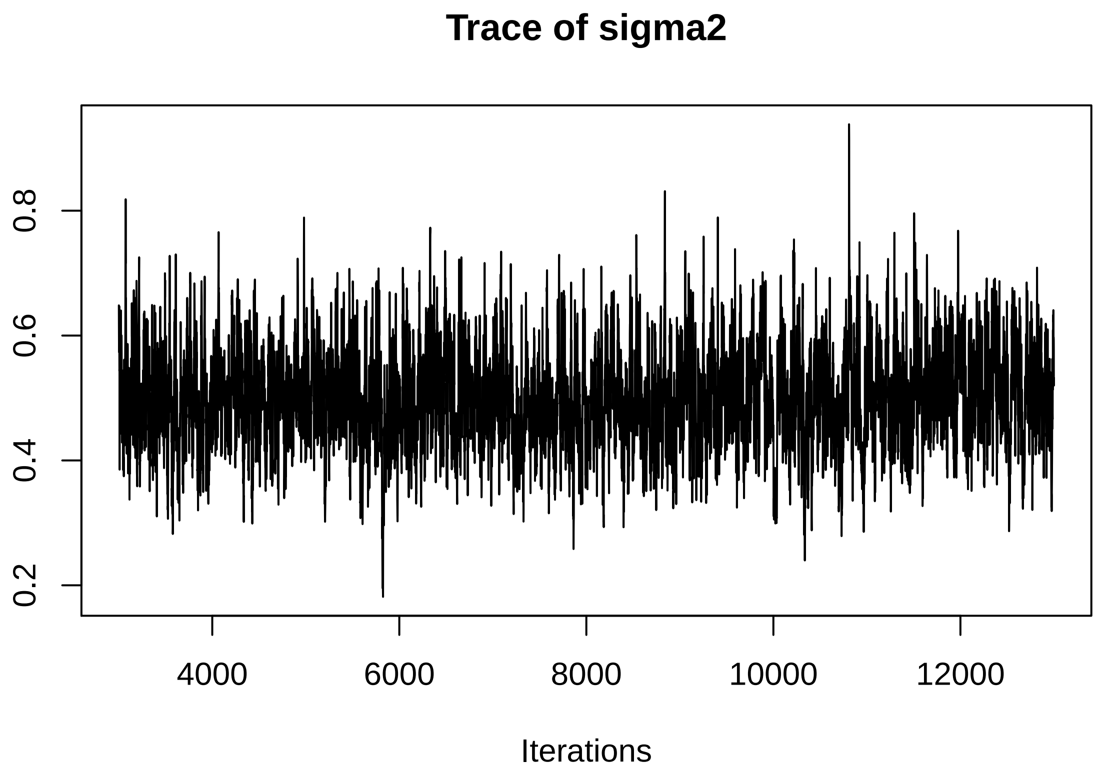


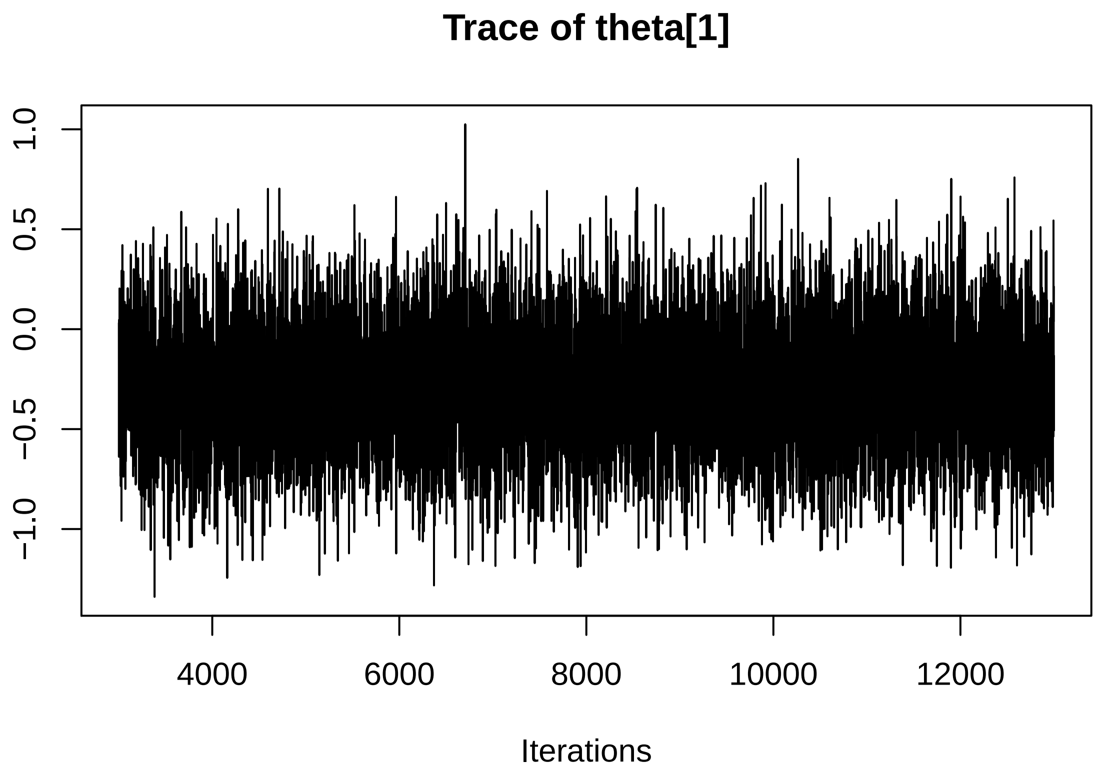
Trace of alpha
0.0
0.4
0.8
1.2
4000
6000
8000
10000
12000
Iterations

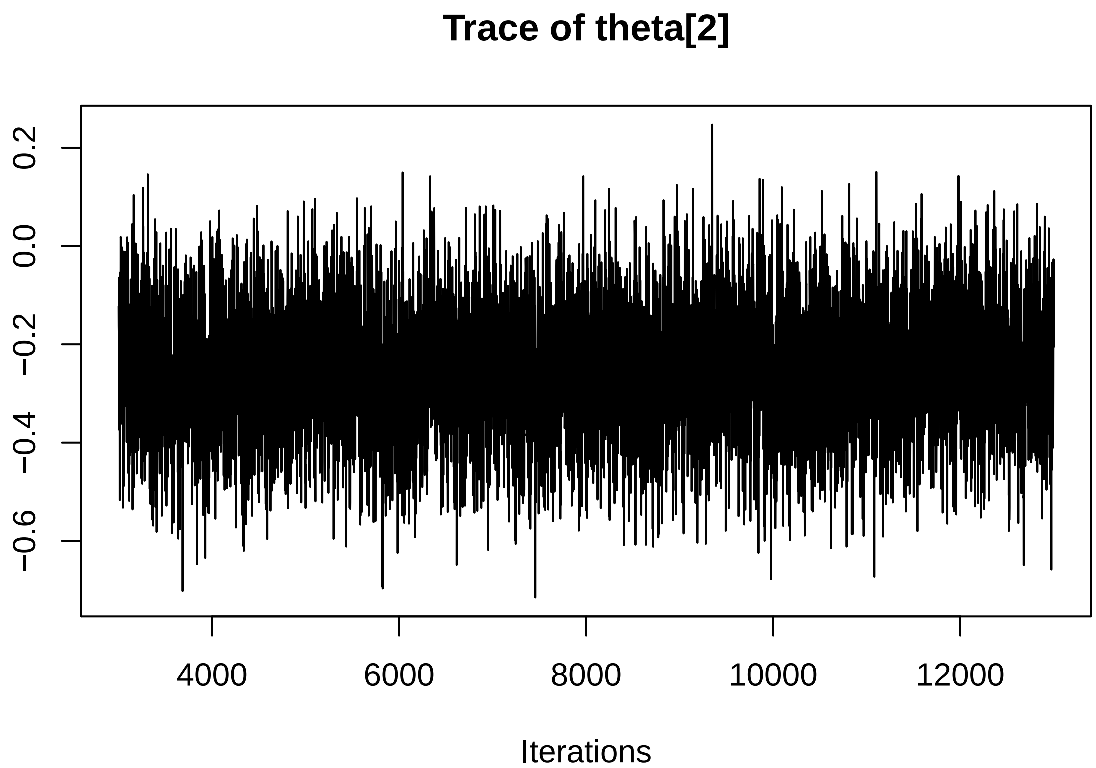


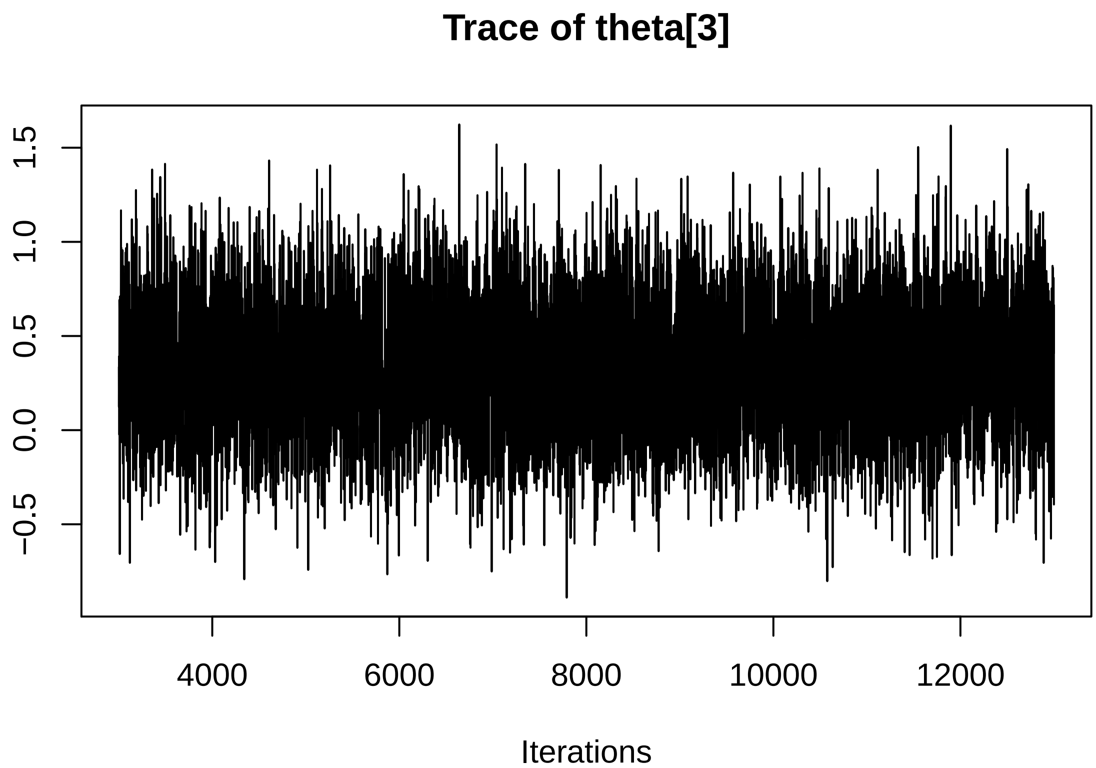
Trace of beta
1.4
1.3
1.2
1.1
1.0
0.9
0.8
4000
6000
8000
10000
12000
Iterations


Trace of sigma
0.70
0.74
0.78
0.82
4000
6000
8000
10000
12000
Iterations


Trace of sigma2
0.2
0.4
0.6
0.8
4000
6000
8000
10000
12000
Iterations


Trace of theta[1]
1.0
0.5
0.0
−0.5
−1.0
4000
6000
8000
10000
12000
Iterations


Trace of theta[2]
0.2
0.0
−0.2
−0.4
−0.6
4000
6000
8000
10000
12000
Iterations




```r
#These appear as white noise, centred on the parameter estimate, indicating the chain has converged to
summary(sample)$statistics
```

```
##                   Mean         SD     Naive SE Time-series SE
## alpha       0.6363128861 0.16952111 0.0016952111   0.0023224664
## beta        1.1390501925 0.07755525 0.0007755525   0.0020037190
## sigma       0.7730282358 0.01984379 0.0001984379   0.0003358169
## sigma2      0.4967976750 0.08060569 0.0008060569   0.0025802403
## theta[1]   -0.2810828123 0.30538463 0.0030538463   0.0034598993
## theta[2]   -0.2531120865 0.12587969 0.0012587969   0.0021159043
## theta[3]    0.3358838594 0.33970753 0.0033970753   0.0043040317
## theta[4]    0.3804111939 0.26389885 0.0026389885   0.0039871264
## theta[5]    0.2094975656 0.30934430 0.0030934430   0.0036719558
## theta[6]    0.3237815421 0.33522127 0.0033522127   0.0035070886
## theta[7]    0.8210702611 0.20759664 0.0020759664   0.0032968125
## theta[8]    0.7260519784 0.31645085 0.0031645085   0.0041438803
## theta[9]   -0.1149211070 0.22651484 0.0022651484   0.0025054849
## theta[10]   0.4642595575 0.27400686 0.0027400686   0.0033083743
## theta[11]   0.0484979764 0.29528336 0.0029528336   0.0042108871
## theta[12]   0.4313690196 0.31561353 0.0031561353   0.0038551867
## theta[13]  -0.2050654855 0.27734337 0.0027734337   0.0036254253
## theta[14]   0.8521832188 0.20347427 0.0020347427   0.0028122188
## theta[15]   0.2395291630 0.30878693 0.0030878693   0.0034929329
## theta[16]   0.3658891155 0.30459547 0.0030459547   0.0030459547
## theta[17]   0.0008745033 0.21554882 0.0021554882   0.0027787772
## theta[18]   0.2166070246 0.12111859 0.0012111859   0.0022037624
```

```
## theta[19]  0.4474768326 0.33819385 0.0033819385   0.0040720186
## theta[20]  0.5276475873 0.23920314 0.0023920314   0.0033312165
## theta[21] -0.2402221036 0.26721557 0.0026721557   0.0027585057
## theta[22]  0.3533369022 0.36903566 0.0036903566   0.0036903566
## theta[23]  0.8563645620 0.24708749 0.0024708749   0.0035755482
## theta[24]  0.7547971976 0.20764716 0.0020764716   0.0032857579
## theta[25]  0.2576315347 0.10251622 0.0010251622   0.0020293354
## theta[26]  0.6414368692 0.27266177 0.0027266177   0.0033558262
## theta[27]  0.1500662296 0.28521275 0.0028521275   0.0032670588
## theta[28] -0.2575741257 0.34257540 0.0034257540   0.0039238448
## theta[29] -0.2479170296 0.22059966 0.0022059966   0.0028269949
## theta[30]  0.6542298501 0.29710513 0.0029710513   0.0042024113
## theta[31]  0.0542193377 0.30597140 0.0030597140   0.0034429514
## theta[32]  0.5935390493 0.32323943 0.0032323943   0.0045854320
## theta[33]  0.4001478541 0.33532879 0.0033532879   0.0039841817
## theta[34] -0.2067414975 0.25829857 0.0025829857   0.0029955442
## theta[35]  1.0174364064 0.40005914 0.0040005914   0.0067787254
## theta[36] -0.6320144660 0.23163254 0.0023163254   0.0023623379
## theta[37]  0.4114787626 0.32656162 0.0032656162   0.0053588078
## theta[38]  0.5070708979 0.29029877 0.0029029877   0.0035230564
## theta[39]  0.8968665981 0.32150192 0.0032150192   0.0050126148
## theta[40]  0.7321340117 0.25498210 0.0025498210   0.0037500749
## theta[41]  0.1335598440 0.41817454 0.0041817454   0.0043269716
## theta[42]  0.5445280800 0.23430646 0.0023430646   0.0030479553
## theta[43]  0.0011997289 0.25550703 0.0025550703   0.0024996995
## theta[44]  0.2516966009 0.20280185 0.0020280185   0.0023850286
## theta[45]  0.0592003311 0.28525289 0.0028525289   0.0031321868
## theta[46] -0.0912010632 0.36490583 0.0036490583   0.0036490583
## theta[47]  0.0308585134 0.23392535 0.0023392535   0.0026328481
## theta[48]  0.5514219803 0.20479907 0.0020479907   0.0028155883
## theta[49]  0.5850035574 0.42601844 0.0042601844   0.0050530997
## theta[50]  0.6929042636 0.31613069 0.0031613069   0.0045146340
## theta[51]  0.5470747066 0.33854052 0.0033854052   0.0041054719
## theta[52]  0.2151916263 0.33331334 0.0033331334   0.0033331334
## theta[53]  0.2182974196 0.16547964 0.0016547964   0.0021105869
## theta[54]  0.4549958517 0.25019991 0.0025019991   0.0033773134
## theta[55]  0.1085392535 0.33028680 0.0033028680   0.0033028680
## theta[56] -0.2459311445 0.26884965 0.0026884965   0.0026884965
## theta[57]  0.6337471776 0.31306631 0.0031306631   0.0039619505
## theta[58]  0.5540744556 0.31014481 0.0031014481   0.0036219597
## theta[59]  0.1690016588 0.36938670 0.0036938670   0.0036938670
## theta[60]  0.0353465366 0.14975078 0.0014975078   0.0022668693
## theta[61]  0.6981756235 0.29482906 0.0029482906   0.0042558953
## theta[62]  0.4822576133 0.33790347 0.0033790347   0.0035234136
## theta[63]  0.6271370753 0.22933014 0.0022933014   0.0033941147
## theta[64]  0.2655952775 0.36988116 0.0036988116   0.0038486385
## theta[65]  0.6255981216 0.20013459 0.0020013459   0.0026815659
## theta[66]  0.6694515057 0.20898249 0.0020898249   0.0029176616
## theta[67] -0.1382104806 0.25166730 0.0025166730   0.0030858000
## theta[68]  0.1225880704 0.30737400 0.0030737400   0.0031752112
## theta[69] -0.2141776487 0.09636345 0.0009636345   0.0018774317
## theta[70]  0.3907312060 0.16376746 0.0016376746   0.0023535755
## theta[71]  0.3894289000 0.23089327 0.0023089327   0.0029132932
## theta[72]  0.4613850330 0.37296612 0.0037296612   0.0041325369
```

```
## theta[73] -0.0075914699 0.30731935 0.0030731935   0.0030731935
## theta[74]  0.3880284285 0.33996658 0.0033996658   0.0043094724
## theta[75]  0.7287051583 0.30905242 0.0030905242   0.0038498367
## theta[76]  0.5723549266 0.26360579 0.0026360579   0.0037646661
## theta[77]  0.0695784659 0.29704676 0.0029704676   0.0038036993
## theta[78] -0.2010448726 0.30545770 0.0030545770   0.0031933269
## theta[79]  0.2195468967 0.13000111 0.0013000111   0.0021593832
## theta[80]  1.0643734540 0.36602415 0.0036602415   0.0063152711
## theta[81]  0.4228769735 0.42570432 0.0042570432   0.0050954138
## theta[82]  0.5314314588 0.20192554 0.0020192554   0.0025195162
## theta[83]  0.4421782354 0.21126876 0.0021126876   0.0030782715
## theta[84]  0.1449490276 0.36976865 0.0036976865   0.0036976865
```

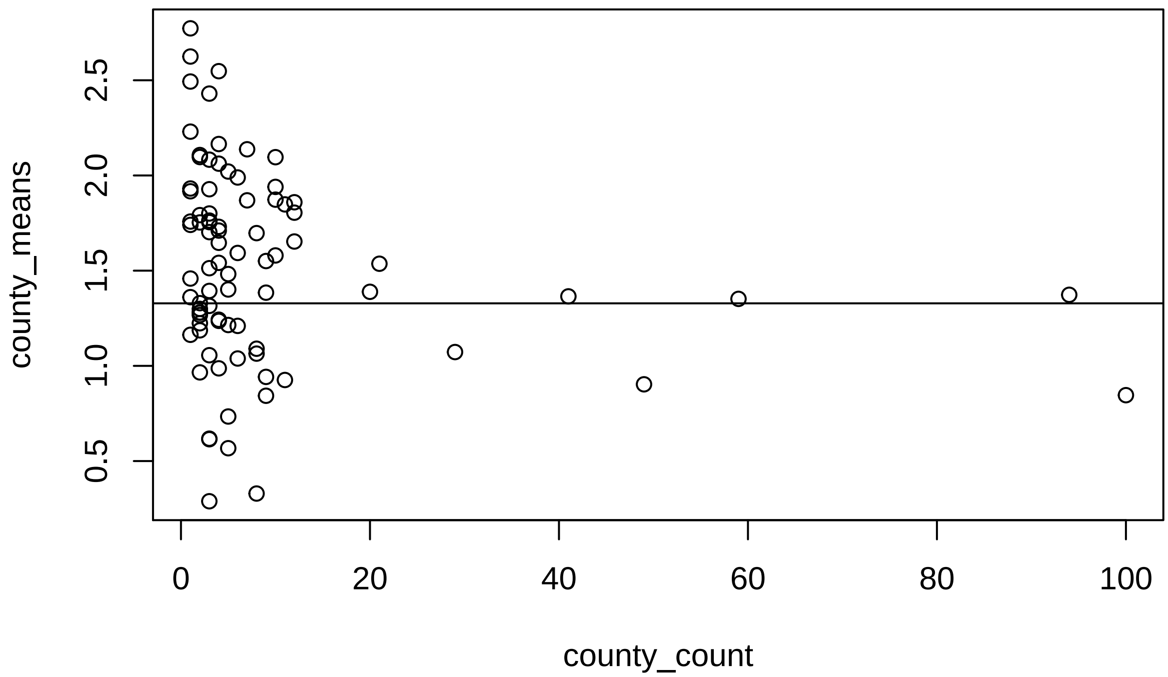
```
#All the standard deviations are small, supporting convergence
```

3. Provide an answer to the above problem P1. In particular,

a) (5 marks) For each county in the dataset, compute the average log-radon level of a house with a basement. Make a plot with these estimates on the y-axis and the number of observations per county on the x-axis. Add in the plot an horizontal line that represents the average log-radon level of a house with a basement for Minnesota.

```
#select houses with basements
radon_bm <- radon[radon$floor==1,]
#find average and counts by county
county_means <- rep(0, 84)
county_count <- rep(0, 84)
for(i in 1:84){
  county_means[i] <- mean(radon_bm$log.radon[radon_bm$county.index == i])
  county_count[i] <- nrow(radon_bm[radon_bm$county.index == i,])
}
```

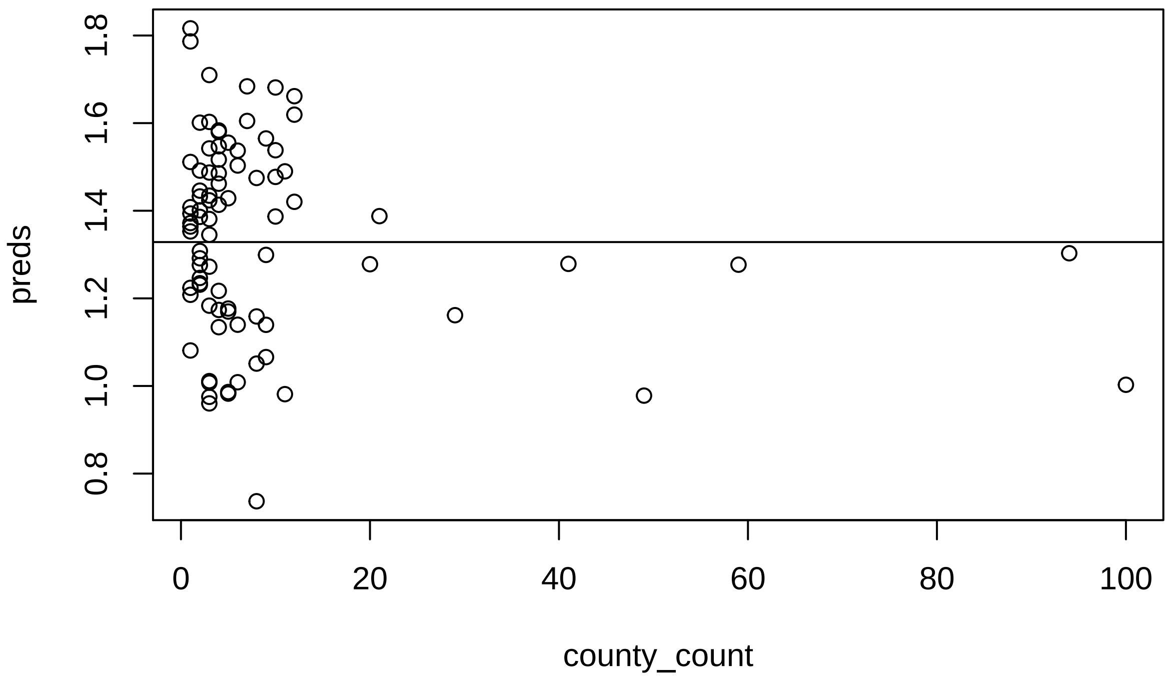
```
#plot means against counts
plot(county_count,county_means)
#find overall mean and overlay as line
state_mean <- mean(radon_bm$log.radon)
abline(a=state_mean, b=0)
```



b) (15 marks) Using your model, predict for each county in the dataset the log-radon level of a house with a basement. Explain how these predictions have been computed from your model and make a plot with these estimates on the y-axis and the number of observations per county on the x-axis. Add in the plot an horizontal line that represents the average log-radon level of a house with a basement for Minnesota.

```
#select estimates for parameters as a new vector
pars <- summary(sample)$statistics[,1]
#predict log radon for each county, assuming house has a basement, using model equation
preds <- rep(0, 84)
for(i in 1:84){
  preds[i] <-  pars[1]*pars[i+4] + pars[2]
}
```

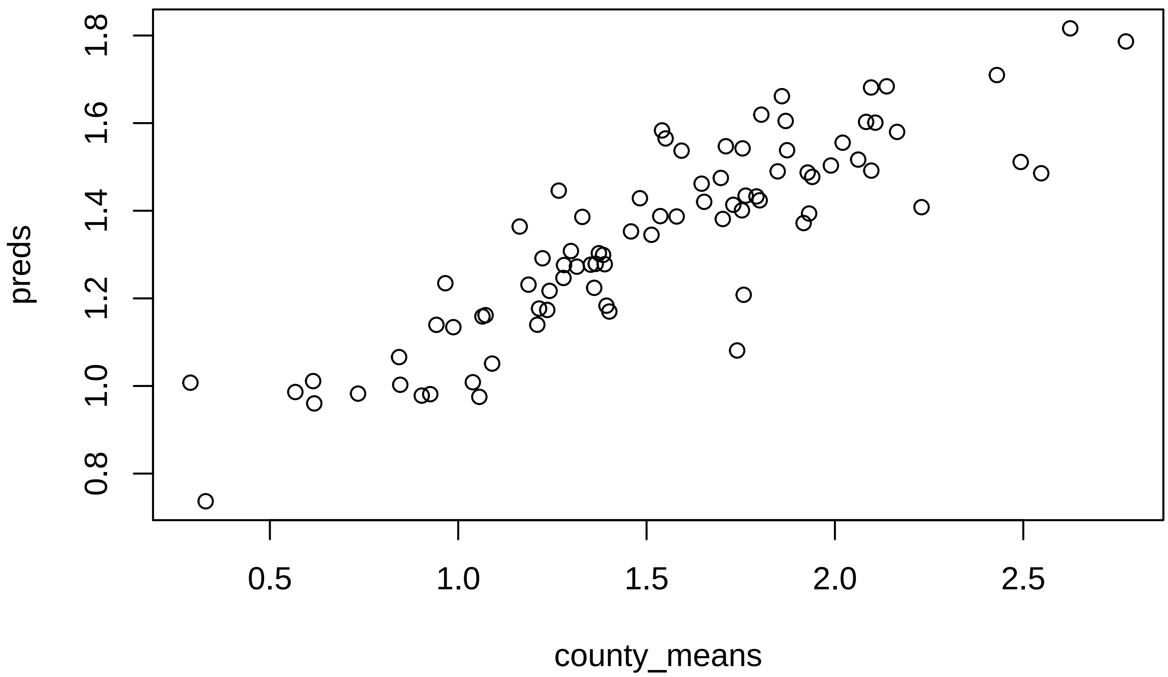
```
#plot predictions against counts by county
plot(county_count, preds)
abline(a=state_mean, b=0)
```



Values for estimated parameters are substituted into the model equation, drawn from the "pars" vector, to obtain a prediction for each county the log-radon level of a house with a basement.

c) (5 marks) Compare the 'naive' estimates obtained in part 3.a) with the model based estimates obtained in part 3.b). Comment your results.

```
#plot estimates against each other
plot(county_means, preds)
```




```r
#find mean square error of both estimates
mean((county_means-preds)^2)
```

```
## [1] 0.1435519
```

From the plot, we see from the linear relationship present that the predictions and means are strongly correlated; this is supported by the small value for the MSE. The predictions underestimate the log radon value on average, as can be seen from the prediction average being lower than the observed average, though this is only by a small amount.

4. (15 marks) Provide an answer to the above problem P2, that is use your model to predict the log-radon level of a house with no basement and of a house with a basement located in Cook, given that the soil uranium measurement for this county is equal to -0.504995982525783. Explain how your prediction has been computed from your model.

```r
u_cook <- -0.504995982525783
#Predict the log-radon level of a house with no basement in the county of Cook
pars[1]*u_cook
```

```
##      alpha
## -0.3213355
```

```r
#Predict the log-radon level of a house with a basement in the county of Cook
pars[1]*u_cook + pars[2]
```

```
##     alpha
## 0.8177147
```

Values for estimated parameters are substituted into the model equation, drawn from the “pars” vector, to obtain a prediction for the log-radon level of a house with no basement and of a house with a basement located in Cook.