Hiearchical Modelling in R with Example

Ozancan Özdemir

Middle East Technical University

ozancan@metu.edu.tr

25.12.2018

NYR
wb

Overview

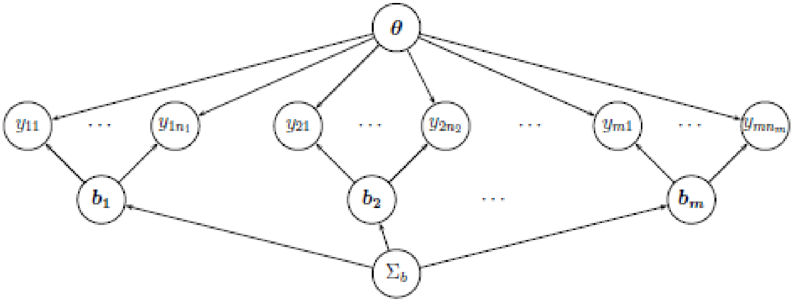
1. What is a hierarchical model?
   - What is a hierarchical model?
   - Hierarchical Bayes Estimation
2. Example for Hierarchical Bayes Estimation
   - Incidence of Tumors in Rodents, Gelman et al. (2014)
3. Implementation in R
4. Checking Convergence
5. Short Comparison with OpenBUGS

What is a hierarchical model?

- There is not a single authorative definition of a hierarchical model.
- Gelman, 2014
  Estimating the population distribution of unobserved parameters.
  Multiple parameters related by the structure of the problem.
- Junker, B., 2006
  "A model where there is some sort of hierarchical structure to the parameters."
- Kruschke, J. K. and Vanpaemel, W., 2015
  "Probability of one parameter can be conceived to depend on the value of another parameter".

What is a hierarchical model?

- Simple Hierarchical Model



Hierarchical Bayes Estimation

- In hierarchical Bayesian estimation, we not only specify a prior on the data models parameter(s), but specify a further prior (called a hyperprior) for the hyperparameters.
- This more complicated prior structure can be useful for modeling hierarchical data structures, also called multilevel data.
- Multilevel data involves a hierarchy of nested populations, in which data could be measured for several levels of aggregation.

Hierarchical Bayes Estimation

- Assume we have data x from density $f(x|\theta)$ with a parameter of interest θ .
- Typically we would choose a prior for θ that depends on some hyperparameter(s) ϕ .
- Instead of choosing fixed values for ϕ , we could place a hyperprior p(ϕ) on it.

Hierarchical Bayes Estimation

- Our posterior is then:
  $p(\theta, \phi | x) \propto L(\theta | x) p(\theta | \phi) p(\phi)$
- Posterior inference about θ is based on the marginal posterior for θ:
  $p(\theta | x) = \int_{\phi} p(\theta, \phi | x) d\phi$
- Except in simple situations, such analysis typically requires MCMC methods.

Incidence of Tumors in Rodents, Gelman et al. (2014)

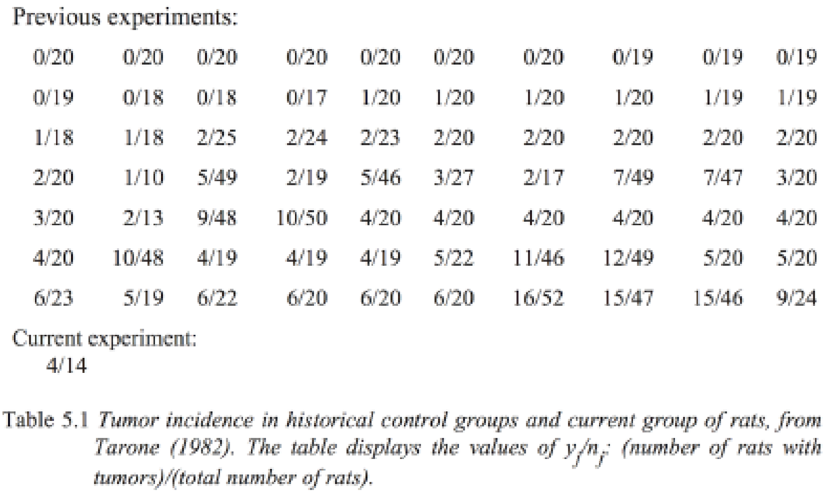
- Let's develop a Hierarchical model using information so far.

Example

- Suppose we have the results of a clinical study of a drug in which rodents were exposed to either a dose of the drug or a control treatment (no dose)
- 4 out of 14 rodents in the control group developed tumors
- We want to estimate θ, the probability that the rodents in the control group developed a tumor given no dose of the drug

Incidence of Tumors in Rodents, Gelman et al. (2014)

- We also have the following data about the incidence of this kind of tumor in the control groups of other studies:



Previous experiments:

0/20	0/20	0/20	0/20	0/20	0/20	0/20	0/19	0/19	0/19
0/19	0/18	0/18	0/17	1/20	1/20	1/20	1/20	1/19	1/19
1/18	1/18	2/25	2/24	2/23	2/20	2/20	2/20	2/20	2/20
2/20	1/10	5/49	2/19	5/46	3/27	2/17	7/49	7/47	3/20
3/20	2/13	9/48	10/50	4/20	4/20	4/20	4/20	4/20	4/20
4/20	10/48	4/19	4/19	4/19	5/22	11/46	12/49	5/20	5/20
6/23	5/19	6/22	6/20	6/20	6/20	16/52	15/47	15/46	9/24

Current experiment:

4/14

Table 5.1 *Tumor incidence in historical control groups and current group of rats, from Tarone (1982). The table displays the values of y_j/n_j: (number of rats with tumors)/(total number of rats).*

Incidence of Tumors in Rodents, Gelman et al. (2014)

- Including the current experimental results, we have information on 71 random variables $\theta_1, ...\theta_{71}$.
- We can model the current and historical proportions as a random sample from some unknown population distribution: each y_j is independent binomial data, given the sample sizes n_j and experiment-specific θ_j.
- Each θ_j is in turn generated by a random process governed by a population distribution that depends on the parameters α and β.

Incidence of Tumors in Rodents, Gelman et al. (2014)

- This relationship can be depicted as graphically as

$$\begin{tikzcd}
 & & & & \alpha, \beta \arrow[dllll] \arrow[dlll] \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] \arrow[drr] \arrow[drrr] \arrow[drrrr] & & & & \\
\theta_1 \arrow[d] & \theta_2 \arrow[d] & \theta_3 \arrow[d] & \cdots & \cdots & \cdots & \cdots & \theta_{70} \arrow[d] & \theta_{71} \arrow[d] \\
y_1 & y_2 & y_3 & \cdots & \cdots & \cdots & \cdots & y_{70} & y_{71}
\end{tikzcd}$$

Incidence of Tumors in Rodents, Gelman et al. (2014)

- Formally, posterior distribution is now of the vector (θ, α, β). The joint prior distribution is

$$p(\theta, \alpha, \beta) = p(\alpha, \beta)p(\theta|\alpha, \beta)$$

and the joint posterior distribution is

$$\begin{aligned} p(\theta, \alpha, \beta|y) &\propto p(\theta, \alpha, \beta)p(y|\theta, \alpha, \beta) \\ &= p(\alpha, \beta)p(\theta|\alpha, \beta)p(y|\theta, \alpha, \beta) \\ &= p(\alpha, \beta)p(\theta|\alpha, \beta)p(y|\theta) \end{aligned}$$

Incidence of Tumors in Rodents, Gelman et al. (2014)

- Since the beta prior is conjugate, we can derive the joint posterior distribution analytically
- Each y_j is conditionally independent of the hyperparameters α, β given θ_j. Hence, the likelihood function is still

$$\begin{aligned} p(y|\theta,\alpha,\beta) = p(y|\theta) &= p(y_1, y_2, ..., y_J|\theta_1, \theta_2, ..., \theta_J) \\ &= \prod_{j=1}^{J} p(y_j|\theta_j) = \prod_{j=1}^{J} \binom{n_j}{y_j} \theta_j^{y_j}(1-\theta_j)^{n_j-y_j} \end{aligned}$$

- Now we also have a population distribution $p(\theta|\alpha,\beta)$:

$$\begin{aligned} p(\theta|\alpha,\beta) &= p(\theta_1, \theta_2, ..., \theta_J|\alpha,\beta) \\ &= \prod_{j=1}^{J} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1}(1-\theta_j)^{\beta-1} \end{aligned}$$

Incidence of Tumors in Rodents, Gelman et al. (2014)

- Then, the unnormalized joint posterior distribution $p(\theta, \alpha, \beta | y)$ is

$$p(\alpha, \beta) \prod_{j=1}^{J} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha - 1} (1 - \theta_j)^{\beta - 1} \prod_{j=1}^{J} \theta_j^{y_j} (1 - \theta_j)^{n_j - y_j}.$$

- We can also determine analytically the conditional posterior density of $\theta = (\theta_1, \theta_2, ...\theta_j)$:

$$p(\theta | \alpha, \beta, y) = \prod_{j=1}^{J} \frac{\Gamma(\alpha + \beta + n_j)}{\Gamma(\alpha + y_j)\Gamma(\beta + n_j - y_j)} \theta_j^{\alpha + y_j - 1} (1 - \theta_j)^{\beta + n_j - y_j - 1}.$$

- Note that equation $p(\theta, \alpha, \beta | y)$, the conditional posterior, is now a function of (α, β). Each θ_j depends on the hyperparameters of the hyperprior $p(\alpha, \beta)$.

Incidence of Tumors in Rodents, Gelman et al. (2014)

- To compute the marginal posterior density, observe that if we condition on y, we have

$$p(\alpha, \beta | y) = \frac{p(\theta, \alpha, \beta | y)}{p(\theta | \alpha, \beta, y)}$$

- If we put the equations on the previous slides, we see

$$p(\alpha, \beta | y) = p(\alpha, \beta) \frac{\prod_{j=1}^{J} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1} (1-\theta_j)^{\beta-1} \prod_{j=1}^{J} \theta_j^{y_j} (1-\theta_j)^{n_j - y_j}}{\prod_{j=1}^{J} \frac{\Gamma(\alpha+\beta+n_j)}{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)} \theta_j^{\alpha+y_j-1} (1-\theta_j)^{\beta+n_j-y_j-1}}$$

$$= p(\alpha, \beta) \prod_{j=1}^{J} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha+y_j)\Gamma(\beta+n_j-y_j)}{\Gamma(\alpha+\beta+n_j)},$$

which is computationally tractable, given a prior for (α, β).

Incidence of Tumors in Rodents, Gelman et al. (2014)

- From the full model,

  $$p(\alpha, \beta) \prod_{j=1}^{J} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta_j^{\alpha-1}(1-\theta_j)^{\beta-1} \prod_{j=1}^{J} \theta_j^{y_j}(1-\theta_j)^{n_j-y_j}.$$

  the complete conditionals can be obtained.
  $P(\theta_i|rest) = Beta(\alpha + y_i, \beta + n_i - y_i)$
  $P(\alpha|rest) \propto [\frac{\Gamma\alpha+\beta}{\Gamma\alpha}]^J \prod_{j=1}^{J} \theta_i^{\alpha} p(\alpha, \beta)$
  $P(\beta|rest) \propto [\frac{\Gamma\alpha+\beta}{\Gamma\beta}]^J \prod_{j=1}^{J} (1-\theta_i)^{\beta} p(\alpha, \beta)$
  This suggests:
- Gibbs steps for θ_i 's: $\theta_i \sim Beta(...)$
- Metropolis steps for α and β using Normal proposal draws which is random walk M-H. Normal variances are "tuning parameters."

Incidence of Tumors in Rodents, Gelman et al. (2014)

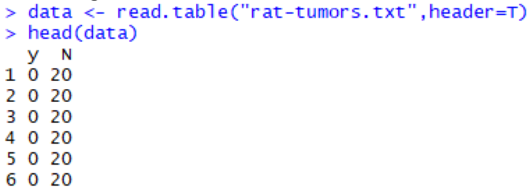
- Recall:$\theta \sim Beta(\alpha, \beta)$, so $E[\theta] = \frac{\alpha}{\alpha+\beta}$ and $Var[\theta] = \sqrt{\frac{1}{\alpha+\beta}}$
- What is a reasonable prior distibution of (α, β)?
- One reasonable way for prior distibution of (α, β) is as follows,
  Let's consider the new parameters η_1, η_2;
  $\eta_1 = \frac{\alpha}{\alpha+\beta}$ where $0< \eta_1 <1$
  $\eta_2 = \sqrt{\frac{1}{\alpha+\beta}}$where $0< \eta_2 <1$
  Consider a hyperprior for (η_1, η_2). That is,
  $p(\eta_1, \eta_2) = U(0,1)U(0,1)$
  $p(\eta_1, \eta_2) = 1$
- Then, transforming back to (α, β) using Jacobian, we get
  $p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2}$

Incidence of Tumors in Rodents, Gelman et al. (2014)

- Kalaylioglu, 2018:
  "Why do we waste our time on analtyic derivations?
  Because computational algorithm is created using the analytical derivations."

Implementation in R

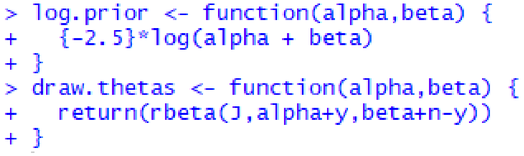
- Reading the data set.



```
> data <- read.table("rat-tumors.txt",header=T)
> head(data)
  y  N
1 0 20
2 0 20
3 0 20
4 0 20
5 0 20
6 0 20
```

Implementation in R

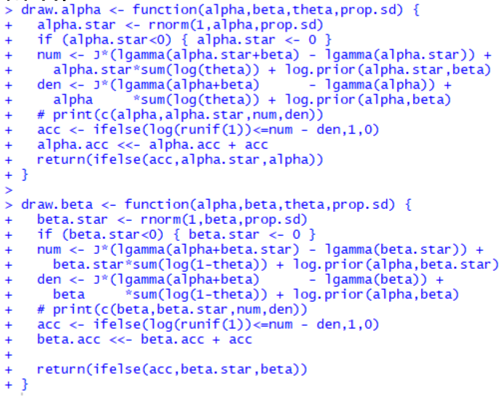
- Then, we write the following function for drawing θ_j.



```
> log.prior <- function(alpha,beta) {
+   {-2.5}*log(alpha + beta)
+ }
> draw.thetas <- function(alpha,beta) {
+   return(rbeta(J,alpha+y,beta+n-y))
+ }
```

Implementation in R

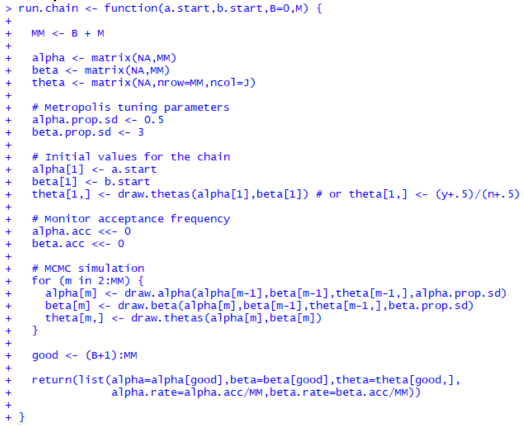
- Then, we write the following functions for drawing α and β using M-H.



```
> draw.alpha <- function(alpha,beta,theta,prop.sd) {
+   alpha.star <- rnorm(1,alpha,prop.sd)
+   if (alpha.star<0) { alpha.star <- 0 }
+   num <- J*(lgamma(alpha.star+beta) - lgamma(alpha.star)) +
+     alpha.star*sum(log(theta)) + log.prior(alpha.star,beta)
+   den <- J*(lgamma(alpha+beta)      - lgamma(alpha)) +
+     alpha     *sum(log(theta)) + log.prior(alpha,beta)
+   # print(c(alpha,alpha.star,num,den))
+   acc <- ifelse(log(runif(1))<=num - den,1,0)
+   alpha.acc <<- alpha.acc + acc
+   return(ifelse(acc,alpha.star,alpha))
+ }
>
> draw.beta <- function(alpha,beta,theta,prop.sd) {
+   beta.star <- rnorm(1,beta,prop.sd)
+   if (beta.star<0) { beta.star <- 0 }
+   num <- J*(lgamma(alpha+beta.star) - lgamma(beta.star)) +
+     beta.star*sum(log(1-theta)) + log.prior(alpha,beta.star)
+   den <- J*(lgamma(alpha+beta)      - lgamma(beta)) +
+     beta     *sum(log(1-theta)) + log.prior(alpha,beta)
+   # print(c(beta,beta.star,num,den))
+   acc <- ifelse(log(runif(1))<=num - den,1,0)
+   beta.acc <<- beta.acc + acc
+
+   return(ifelse(acc,beta.star,beta))
+ }
```

Implementation in R

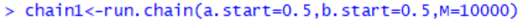
- After this, the following function that includes MCMC algorithm for the problem is written.



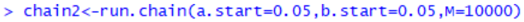
```
> run.chain <- function(a.start,b.start,B=0,M) {
+
+   MM <- B + M
+
+   alpha <- matrix(NA,MM)
+   beta <- matrix(NA,MM)
+   theta <- matrix(NA,nrow=MM,ncol=J)
+
+   # Metropolis tuning parameters
+   alpha.prop.sd <- 0.5
+   beta.prop.sd <- 3
+
+   # Initial values for the chain
+   alpha[1] <- a.start
+   beta[1] <- b.start
+   theta[1,] <- draw.thetas(alpha[1],beta[1]) # or theta[1,] <- (y+.5)/(n+.5)
+
+   # Monitor acceptance frequency
+   alpha.acc <<- 0
+   beta.acc <<- 0
+
+   # MCMC simulation
+   for (m in 2:MM) {
+     alpha[m] <- draw.alpha(alpha[m-1],beta[m-1],theta[m-1,],alpha.prop.sd)
+     beta[m] <- draw.beta(alpha[m],beta[m-1],theta[m-1,],beta.prop.sd)
+     theta[m,] <- draw.thetas(alpha[m],beta[m])
+   }
+
+   good <- (B+1):MM
+
+   return(list(alpha=alpha[good],beta=beta[good],theta=theta[good,],
+               alpha.rate=alpha.acc/MM,beta.rate=beta.acc/MM))
+
+ }
```

Implementation in R

- Then, we run the function on the previous slide for 2 different initial values with 10000 iterations.



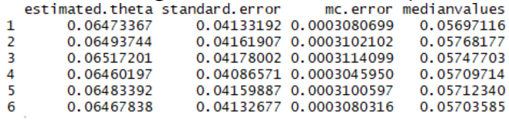
```
> chain1<-run.chain(a.start=0.5,b.start=0.5,M=10000)
```



```
> chain2<-run.chain(a.start=0.05,b.start=0.05,M=10000)
```

Implementation in R

- After running the chain, a 1000 update burn in followed by a further 10000 updates gave the parameter estimates and related statistics. The following table shows first 6 parameters.



```
  estimated.theta standard.error     mc.error medianvalues
1      0.06473367     0.04133192 0.0003080699   0.05697116
2      0.06493744     0.04161907 0.0003102102   0.05768177
3      0.06517201     0.04178002 0.0003114099   0.05747703
4      0.06460197     0.04086571 0.0003045950   0.05709714
5      0.06483392     0.04159887 0.0003100597   0.05712340
6      0.06467838     0.04132677 0.0003080316   0.05703585
```

- As a rule of thumb, the simulation should be run until the Monte Carlo error for each parameter of interest is less than about 5% of the sample standard deviation.

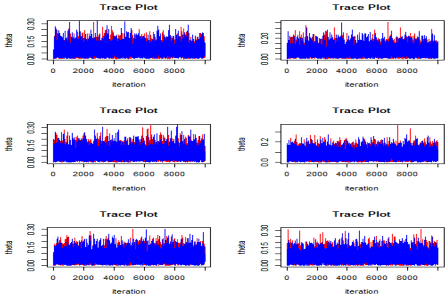
Checking Convergence

There are three ways to check the convergence.

- Trace Plot
- ACF plot of Samples
- Gelman-Rubin Statistic

Trace Plot

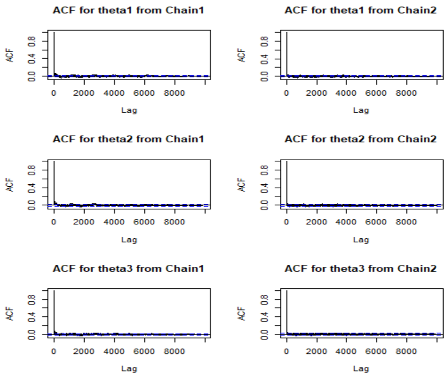
- Trace plot shows the variable value against the iteration number.
- If you are running more than one chain simultaneously, the trace plot will show each chain in a different color. In this case, we can be reasonably confident that convergence has been achieved if all the chains appear to be overlapping one another.



$\therefore$ Convergence is achieved.

ACF Plot

- The ACF shows that there is only one significant lag as we expected.



$\therefore$ Convergence is achieved.

German Rubin Statistic

- Generate R replicate in M chains from well-dispersed starting values.
- Compute

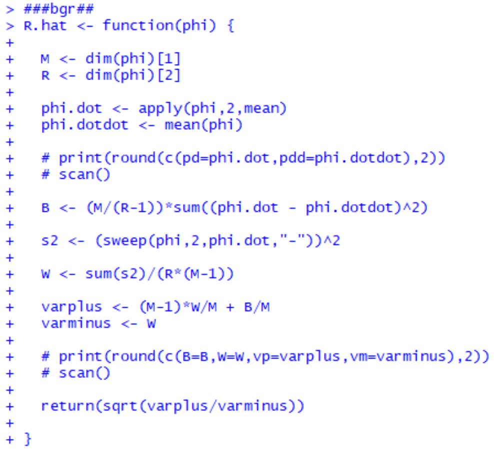
$$B = \frac{M}{R-1} \sum_{r=1}^{R} (\overline{\phi}_{\cdot r} - \overline{\phi}_{\cdot\cdot})^2 \quad W = \frac{1}{R} \sum_{r=1}^{R} \left[\frac{1}{M-1} \sum_{m=1}^{M} (\phi_{mr} - \overline{\phi}_{\cdot r})^2 \right]$$

$$\hat{R} = \sqrt{\frac{\mathrm{Var}^{+}(\phi|data)}{\mathrm{Var}^{-}(\phi|data)}} = \sqrt{\frac{\frac{M-1}{M} W + \frac{1}{M} B}{W}}$$

- If $\hat{R} < 1+\epsilon$, convergence is assessed.

German Rubin Statistic

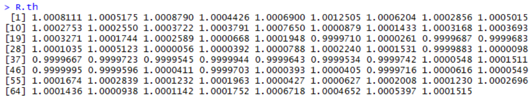
- The following function is helping us to calculate the $\hat{R}$.



```
> ###bgr##
> R.hat <- function(phi) {
+
+   M <- dim(phi)[1]
+   R <- dim(phi)[2]
+
+   phi.dot <- apply(phi,2,mean)
+   phi.dotdot <- mean(phi)
+
+   # print(round(c(pd=phi.dot,pdd=phi.dotdot),2))
+   # scan()
+
+   B <- (M/(R-1))*sum((phi.dot - phi.dotdot)^2)
+
+   s2 <- (sweep(phi,2,phi.dot,"-"))^2
+
+   W <- sum(s2)/(R*(M-1))
+
+   varplus <- (M-1)*W/M + B/M
+   varminus <- W
+
+   # print(round(c(B=B,W=W,vp=varplus,vm=varminus),2))
+   # scan()
+
+   return(sqrt(varplus/varminus))
+
+ }
```

German Rubin Statistic

- After running the function for each parameter, we get,

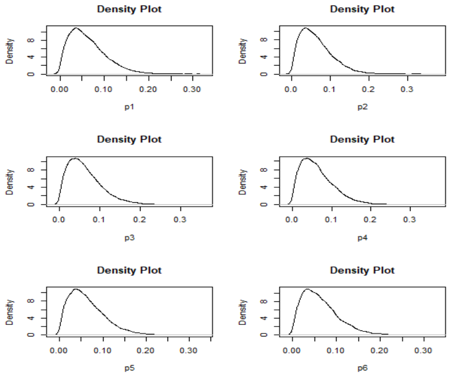


```
> R.th
 [1] 1.0008111 1.0005175 1.0008790 1.0004426 1.0006900 1.0012505 1.0006204 1.0002856 1.0005015
[10] 1.0002753 1.0002550 1.0003722 1.0003791 1.0007650 1.0000879 1.0001433 1.0003168 1.0003693
[19] 1.0003271 1.0001744 1.0002589 1.0000668 1.0001948 0.9999710 1.0000261 0.9999687 0.9999683
[28] 1.0001035 1.0005123 1.0000056 1.0000392 1.0000788 1.0002240 1.0001531 0.9999883 1.0000098
[37] 0.9999667 0.9999723 0.9999545 0.9999944 0.9999643 0.9999534 0.9999742 1.0000548 1.0001511
[46] 0.9999995 0.9999596 1.0000411 0.9999703 1.0000393 1.0000405 0.9999716 1.0000616 1.0000549
[55] 1.0001674 1.0002839 1.0001232 1.0001963 1.0000427 1.0000627 1.0002008 1.0001230 1.0002696
[64] 1.0001436 1.0000938 1.0001142 1.0001752 1.0006718 1.0004652 1.0005397 1.0001515
```

- Less than 1.

$\therefore$ Convergence is achieved.

Density Plot



- For first 6 estimates.
- Right skewed distributions.

Short Comparison with OpenBUGS

- We achieved convergence in both softwares.
- Corr(R,OpenBugs)=0.9999664.
- OpenBUGS run the chains with 10000 iterations in 6 seconds.
- R run the chains with 10000 iterations in 7.2 seconds.
- OpenBUGS has 12 lines codes
- R has more than 100 lines code.
- Therefore, OpenBUGS is faster and easier than R in hierarchical parameter estimation.

Conclusion

THAT CONCLUDES OUR PRESENTATION
ARE THERE ANY QUESTIONS?

References

- Gelman, A., Carlin, J. B., Stern, H., Dunson, D. B., Vehtari, A., Rubin, D. B. (2014). Bayesian data analysis. Boca Raton: CRC Press.
- Junker, B. (2006). Applied Bayesian and Computational Statistics.
- Kruschke, J. K. and Vanpaemel, W. (2015). Bayesian estimation in hierarchical models. In: J. R. Busemeyer, Z. Wang, J. T. Townsend, and A. Eidels (Eds.), The Oxford Handbook of Computational and Mathematical Psychology, pp. 279-299. Oxford, UK: Oxford University Press.
- Albert, J. (2009). Bayesian Computation with R. Dordrecht: Springer.
- Rossi, P. E., Allenby, G. M., McCulloch, R. (2009). Bayesian statistics and marketing. Chichester: Wiley.
- Yildirim, I. (2012) Bayesian Inference: Metropolis-Hastings Sampling, University of Rochester: Department of Brain and Cognitive Sciences.