Bayesian hierarchical modeling

Jingchen (Monika) Hu

MATH 347 Bayesian Statistics

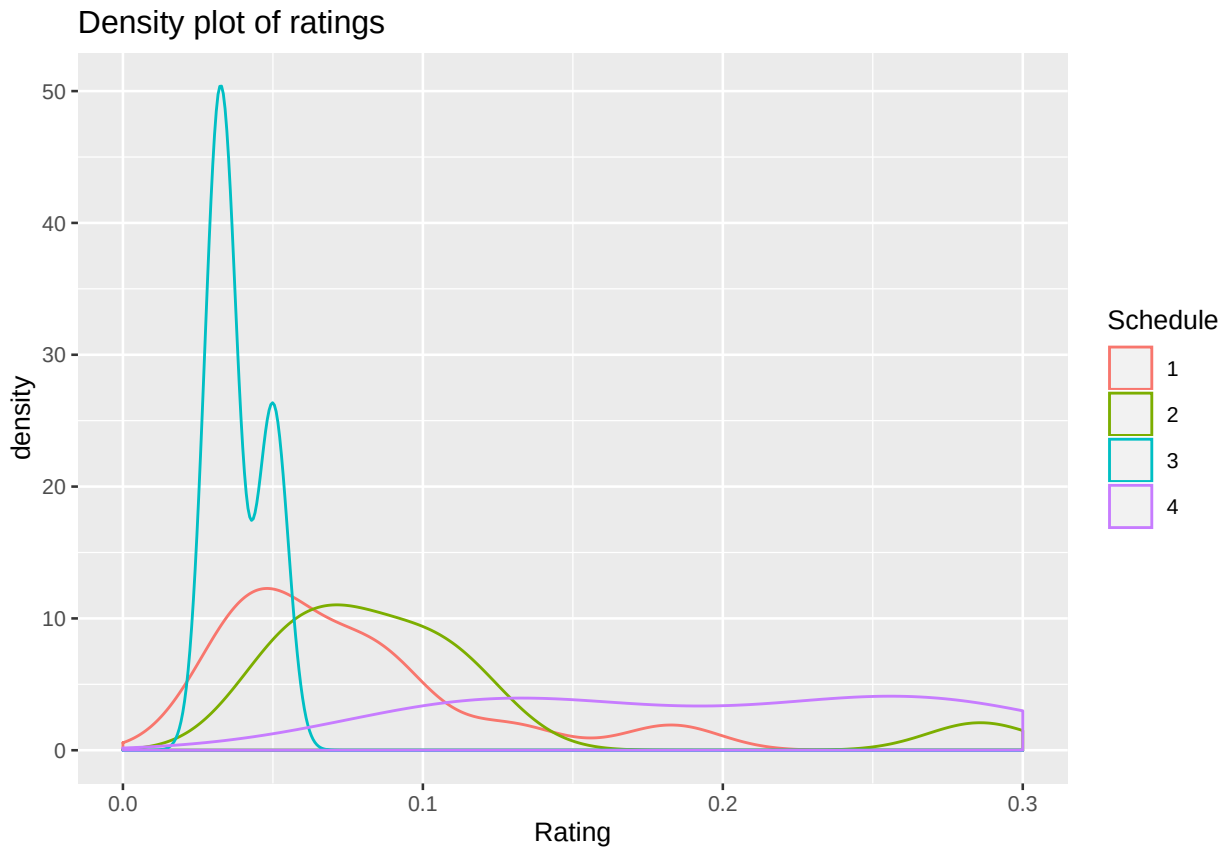
Installing the necessary packages

```
install.packages("devtools")
require(devtools)
devtools::install_github("bayesball/ProbBayes")

require(ggplot2)
require(gridExtra)
require(ProbBayes)
require(tidyverse)
crcblue <- "#2905a1"
```

Example: Korean Drama Ratings

Ratings by Schedule

```
dramadata = read.csv("KDramaData.csv", header=T)

KBSdrama = dramadata[dramadata$Producer==2,]
KBSdrama$Schedule = as.factor(KBSdrama$Schedule)
```




```
table(KBSdrama$Schedule)
```

```
##
##  1  2  3  4
## 13 11  3  6
```

```
tapply(KBSdrama$Rating, KBSdrama$Schedule, summary)
```

```
## $`1`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03310 0.04740 0.06310 0.07396 0.08740 0.18290
##
## $`2`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04830 0.06465 0.08000 0.09976 0.10645 0.28580
##
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0312  0.0326  0.0340  0.0384  0.0420  0.0500
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1120  0.1261  0.1974  0.1949  0.2632  0.2750
```

```r
tapply(KBSdrama$Rating, KBSdrama$Schedule, sd)
```

```
##          1          2          3          4
## 0.04264994 0.06608002 0.01014298 0.07570259
```

Observations in groups: approaches to modeling

A two-stage prior in a hierarchical model

The Hierarchical Normal Model

- The sampling density for group j, and $j = 1, \cdots, J$:

$$Y_{ij} \overset{i.i.d.}{\sim} \text{Normal}(\mu_j, \sigma), \tag{1}$$

  where $i = 1, \cdots, n_j$ and n_j is the number of observations in group j.

- The stage 1 prior distribution for μ_j:

$$\mu_j \sim \text{Normal}(\mu, \tau). \tag{2}$$

- The stage 2 prior distribution for μ_j:

$$\mu, \tau \sim \text{g}(\mu, \tau). \tag{3}$$

- The prior distribution for σ:

$$1/\sigma^2 \sim \text{Gamma}(\alpha_\sigma, \beta_\sigma). \tag{4}$$

Prior and Hyperprior Specifications

- The stage 1 prior distribution for μ_j:

$$\mu_j \sim \text{Normal}(\mu, \tau). \tag{5}$$

- The stage 2 prior distribution for μ_j:

$$\mu, \tau \sim \text{g}(\mu, \tau). \tag{6}$$

- Hyperpriors:

$$\mu \mid \mu_0, \gamma_0 \sim \text{Normal}(\mu_0, \gamma_0), \tag{7}$$
$$1/\tau^2 \mid \alpha_\tau, \beta_\tau \sim \text{Gamma}(\alpha_\tau, \beta_\tau). \tag{8}$$

- The prior distribution for σ:

$$1/\sigma^2 \sim \text{Gamma}(\alpha_\sigma, \beta_\sigma). \tag{9}$$

MCMC simulation by JAGS

Recap: Prior and Hyperprior Specifications

- The stage 1 prior distribution for μ_j:

$$\mu_j \sim \text{Normal}(\mu, \tau). \tag{10}$$

- The stage 2 prior distribution for μ_j:

$$\mu, \tau \sim \text{g}(\mu, \tau). \tag{11}$$

- Hyperpriors:

$$\begin{aligned} \mu \mid \mu_0, \gamma_0 &\sim \text{Normal}(0.1, 0.5), & (12)\\ 1/\tau^2 \mid \alpha_\tau, \beta_\tau &\sim \text{Gamma}(1, 1). & (13) \end{aligned}$$

- The prior distribution for σ:

$$1/\sigma^2 \sim \text{Gamma}(1, 1). \tag{14}$$

JAGS Script for the Hierarchical Model

```
modelString <-"
model {
## likelihood
for (i in 1:N){
y[i] ~ dnorm(mu_j[schedule[i]], invsigma2)
}

## priors
for (j in 1:J){
mu_j[j] ~ dnorm(mu, invtau2)
}
invsigma2 ~ dgamma(a_g, b_g)
sigma <- sqrt(pow(invsigma2, -1))

## hyperpriors
mu ~ dnorm(mu0, 1/g0^2)
invtau2 ~ dgamma(a_t, b_t)
tau <- sqrt(pow(invtau2, -1))
}
"
```

- Notes about the `modelString`
  1. Need a vector of `mu_j`, of length `J`.
  2. Need a vector of `schedule`, of length `N`.

3. `dnorm` takes mean and precision.
4. Work with `invsigma2`, can return `sigma`.
5. Work with `invtau2`, can return `tau`.

- Pass the data and hyperparameter values to JAGS:

```
y = KBSdrama$Rating
schedule = KBSdrama$Schedule
N = length(y)
J = length(unique(schedule))

initsfunction <- function(chain){
  .RNG.seed <- c(1,2)[chain]
  .RNG.name <- c("base::Super-Duper",
                 "base::Wichmann-Hill")[chain]
  return(list(.RNG.seed=.RNG.seed,
              .RNG.name=.RNG.name))
}

the_data <- list("y" = y, "schedule" = schedule, "N" = N, "J" = J,
                 "mu0" = 0.1, "g0" = 0.5,
                 "a_t" = 1, "b_t" = 1,
                 "a_g" = 1, "b_g" = 1)
```

- Run the JAGS code for this model:

```
posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("mu", "tau", "mu_j", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000,
                      thin = 1,
                      inits = initsfunction)
```

```
## Calling the simulation...
## Welcome to JAGS 4.3.0 on Mon Nov 25 15:01:15 2019
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 33
##    Unobserved stochastic nodes: 7
##    Total graph size: 90
## . Reading parameter file inits1.txt
```

```
## . Initializing model
## . Adaptation skipped: model is not in adaptive mode.
## . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . . Updating 5000
## -------------------------------------------------| 5000
## ************************************************** 100%
## . . . . Updating 0
## . Deleting model
## .
## Note: the model did not require adaptation
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Calculating summary statistics...
## Finished running the simulation
```

JAGS Output of the Hierarchical Model

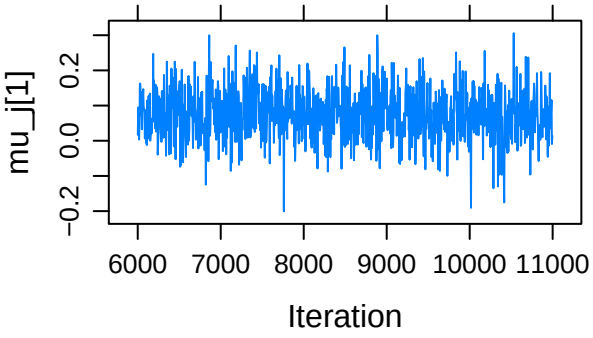
- Obtain posterior summaries of all parameters:

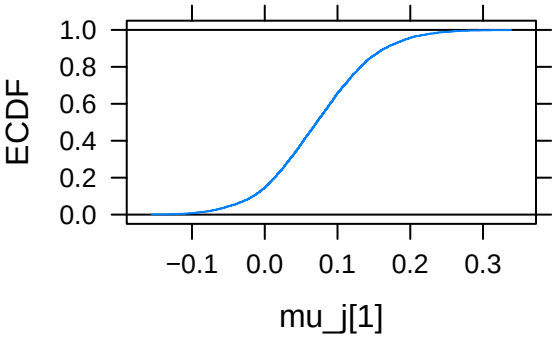
```
summary(posterior)
```

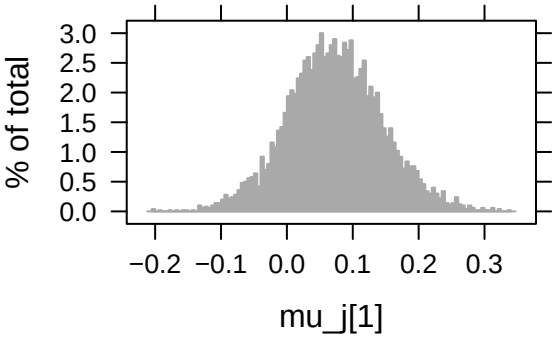
```
##            Lower95     Median  Upper95       Mean         SD Mode
## mu      -0.4904750 0.10801400 0.668230 0.10472246 0.28838371   NA
## tau      0.3526710 0.65852600 1.249940 0.72061723 0.27488876   NA
## mu_j[1] -0.0719655 0.07132615 0.216751 0.07266281 0.07238920   NA
## mu_j[2] -0.0568824 0.09944590 0.253260 0.09928626 0.07995014   NA
## mu_j[3] -0.2414660 0.04479395 0.349108 0.04271370 0.15109887   NA
## mu_j[4] -0.0248182 0.19140950 0.399113 0.19242820 0.10745634   NA
## sigma    0.2010700 0.26158400 0.332643 0.26503794 0.03460243   NA
##                MCerr MC%ofSD SSeff        AC.10 psrf
## mu      0.0041809869     1.4  4758 -0.010518447   NA
## tau     0.0041977729     1.5  4288 -0.015113474   NA
## mu_j[1] 0.0010237378     1.4  5000  0.002230454   NA
## mu_j[2] 0.0011306657     1.4  5000  0.018813934   NA
## mu_j[3] 0.0020734838     1.4  5310 -0.007974236   NA
## mu_j[4] 0.0015196622     1.4  5000 -0.017268736   NA
## sigma   0.0005535074     1.6  3908  0.009879833   NA
```

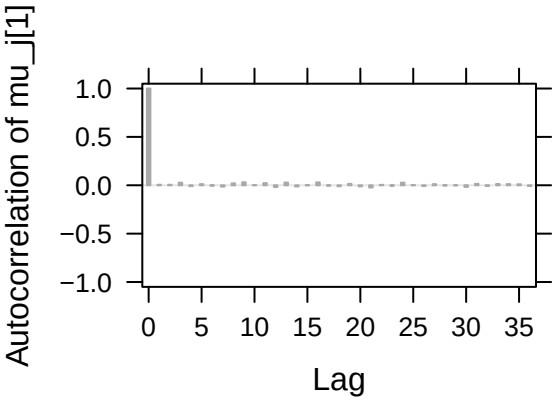
```
plot(posterior, vars = "mu_j[1]")
```

```
## Generating plots...
```



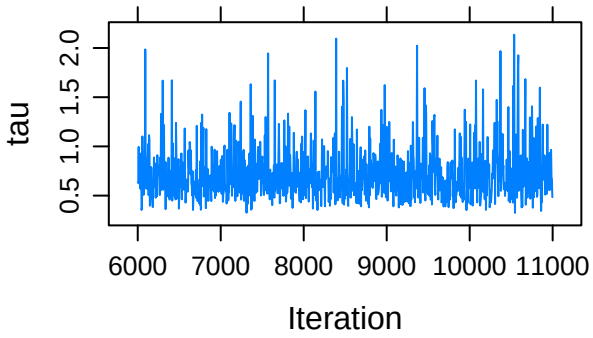


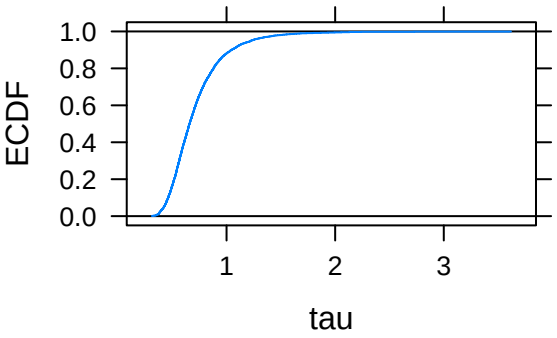


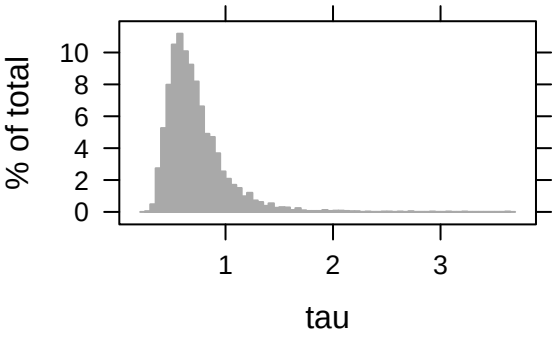


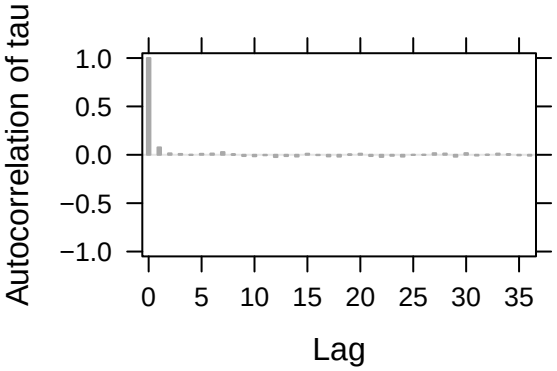

```
plot(posterior, vars = "tau")
```

```
## Generating plots...
```





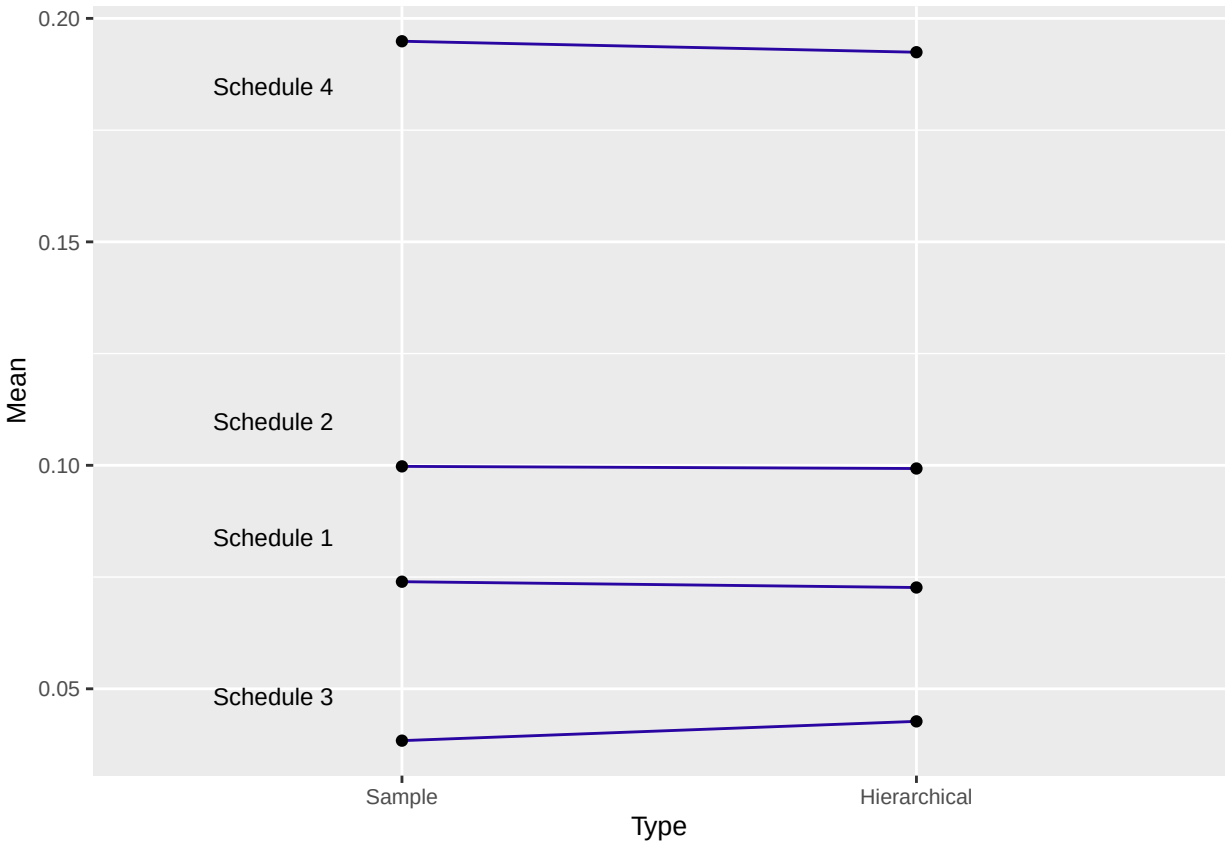




Shrinkage/Pooling Effects

```
Ind_Stats = as.data.frame(matrix(NA, J, 2))
names(Ind_Stats) = c("mean", "sd")
for (j in 1:J){
  Ind_Stats[j, ] = c(mean(KBSdrama$Rating[KBSdrama$Schedule == j]),
                     sd(KBSdrama$Rating[KBSdrama$Schedule == j]))
}

Post_Means <- summary(posterior)[, 4]

Means1 <- data.frame(Type = "Sample", Mean = Ind_Stats$mean)
Means2 <- data.frame(Type = "Hierarchical", Mean =
                       Post_Means[3:(4 + J - 2)])

Means1$Title <- c("Schedule 1", "Schedule 2", "Schedule 3",
                  "Schedule 4")
Means2$Title <- c("Schedule 1", "Schedule 2", "Schedule 3",
                  "Schedule 4")
```

```
ggplot(rbind(Means1, Means2), aes(Type, Mean, group=Title)) +
  geom_line(color = crcblue) + geom_point() +
  annotate(geom = "text", x = 0.75,
           y = Means1$Mean + c(0.01, 0.01, 0.01, -0.01),
           size = 3, label = Means1$Title) + increasefont(Size = 10)
```



Sources of Variability

- Two sources of variability in Y_{ij}:

$$Y_{ij} \overset{i.i.d.}{\sim} \text{Normal}(\mu_j, \sigma) \text{ [within-group variability]} \tag{15}$$

$$\mu_j \mid \mu, \tau \sim \text{Normal}(\mu, \tau) \text{ [between-group variability]} \tag{16}$$

- To compare these two sources of variability, one can compute the fraction

$$R = \frac{\tau^2}{\tau^2 + \sigma^2}, \tag{17}$$

  from the posterior draws of τ and σ.

- The closer the value of R to 1, the higher the between-group variability. ## Compute and Graph Sources of Variability
- We need the `coda` R package

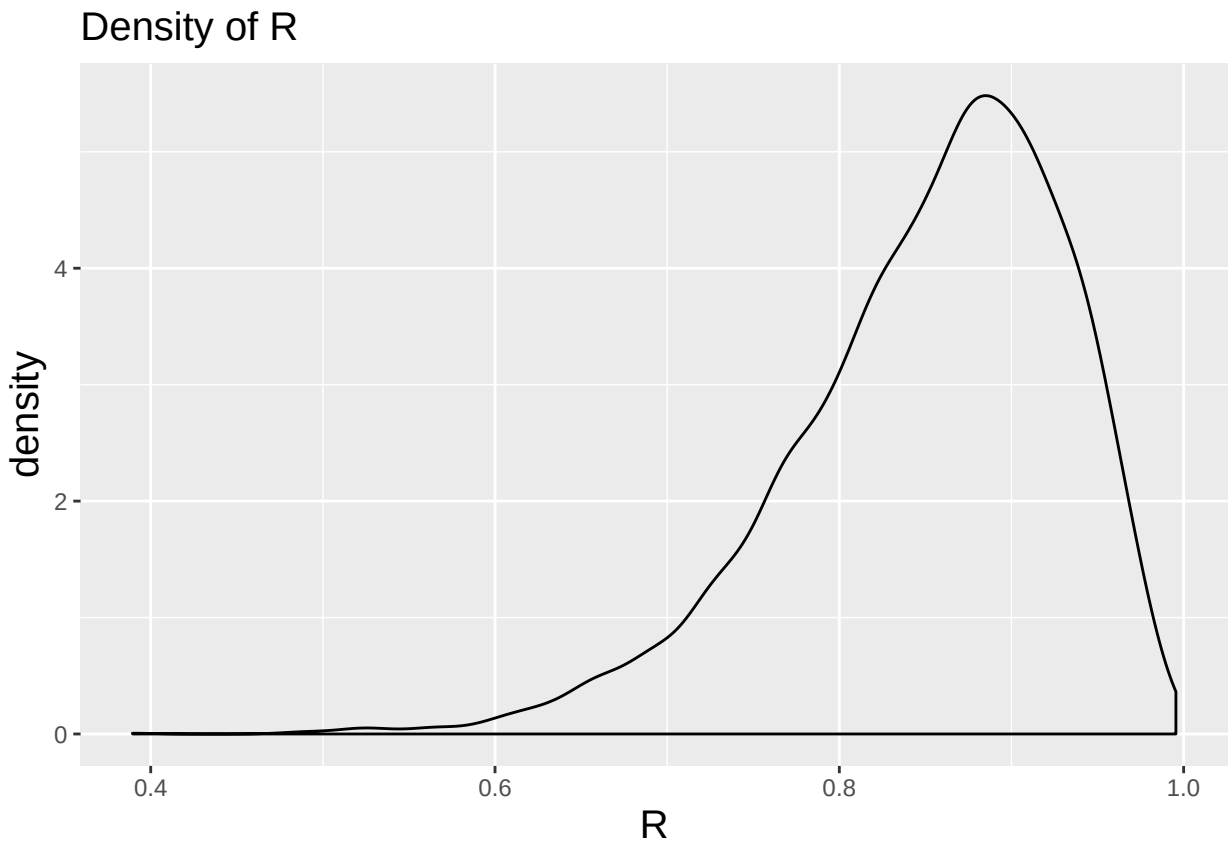
```
install.packages("coda")
```

```
require(coda)
tau_draws <- as.mcmc(posterior, vars = "tau")
sigma_draws <- as.mcmc(posterior, vars = "sigma")
R <- tau_draws^2/(tau_draws^2 + sigma_draws^2)
```

```r
df <- as.data.frame(R)

quantile(R, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.6620552 0.9687804
```

```r
ggplot(df, aes(x=R)) + geom_density() +
  labs(title="Density of R") +
  theme(plot.title = element_text(size=15)) +
  theme(axis.title = element_text(size=15))
```



Exercise: Hierarchical model with schedule-specific μ_j and σ_j

Recap