Gibbs sampler and MCMC

Jingchen (Monika) Hu

Vassar College

MATH 347 Bayesian Statistics

Outline

1. Example: Expenditures in the Consumer Expenditure Surveys
2. Prior and posterior distributions for mean AND standard deviation
3. Use JAGS (Just Another Gibbs Sampler) and Bayesian inferences
4. MCMC diagnostics
5. Recap

Outline

1. Example: Expenditures in the Consumer Expenditure Surveys
2. Prior and posterior distributions for mean AND standard deviation
3. Use JAGS (Just Another Gibbs Sampler) and Bayesian inferences
4. MCMC diagnostics
5. Recap

The TOTEXPPQ variable in the CE sample

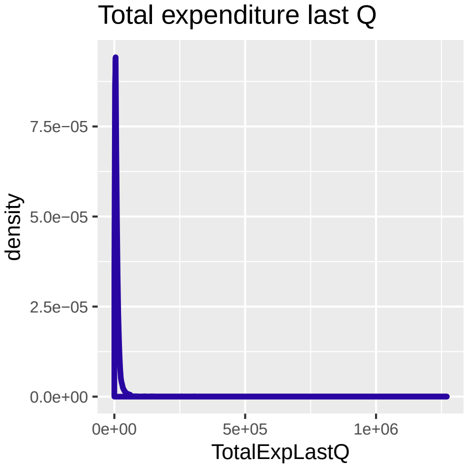
```r
CEsample <- read_csv("CEsample1.csv")

summary(CEsample$TotalExpLastQ)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##       30     3522     6417     9513    11450  1270598
```

```r
sd(CEsample$TotalExpLastQ)
```

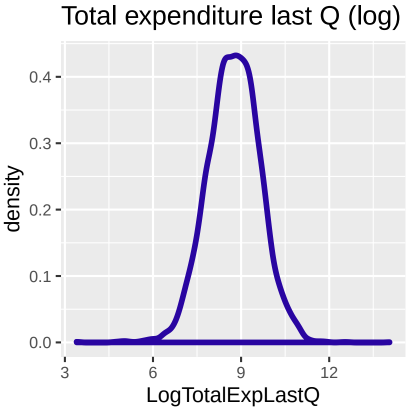
```
## [1] 19341.25
```

The TOTEXPPQ variable in the CE sample cont'd



Log transformation of the TOTEXPPQ variable

```
CEsample$LogTotalExpLastQ <- log(CEsample$TotalExpLastQ)
ggplot(data = CEsample, aes(LogTotalExpLastQ)) +
  geom_density(color = crcblue, size = 1) +
  labs(title = "Total expenditure last Q (log)") +
  theme_grey(base_size = 8, base_family = "")
```

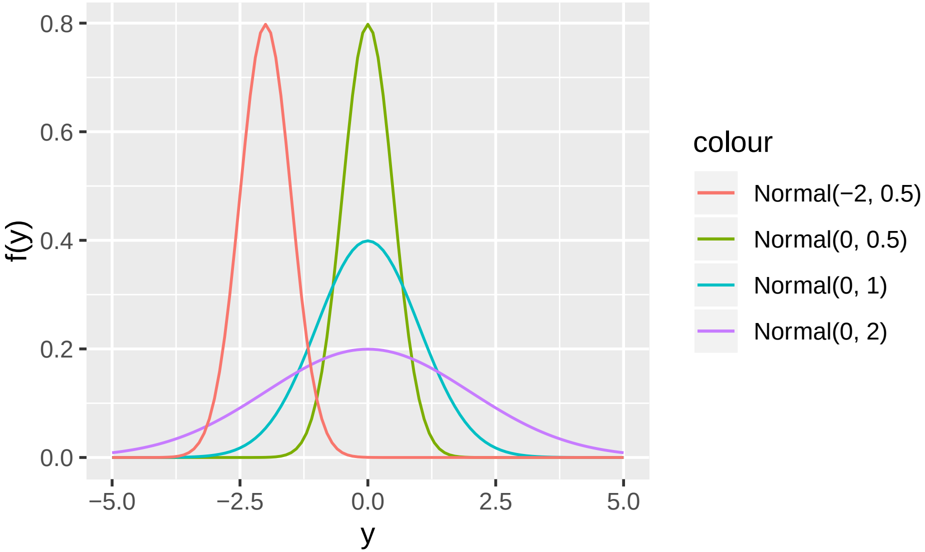


The Normal distribution

- The Normal distribution is a symmetric, bell-shaped distribution.
- It has two parameters: mean μ and standard deviation σ.
- The probability density function (pdf) of $\mathrm{Normal}(\mu, \sigma)$ is:

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y-\mu)^2}{2\sigma^2}\right), -\infty < y < \infty.$$

The Normal distribution cont'd



i.i.d. Normals

- Suppose there are a sequence of n responses: $Y_1, Y_2, \cdots, Y_n$.
- Further suppose each response independently and identically follows a Normal distribution:

$$Y_i \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma).$$

- Then the joint probability density function (joint pdf) of $y_1, \cdots, y_n$ is:

$$f(y_1, \cdots, y_n) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(y_i - \mu)^2}{2\sigma^2}\right), -\infty < y_i < \infty. \qquad (1)$$

Outline

1. Example: Expenditures in the Consumer Expenditure Surveys
2. Prior and posterior distributions for mean AND standard deviation
3. Use JAGS (Just Another Gibbs Sampler) and Bayesian inferences
4. MCMC diagnostics
5. Recap

Step 1: Prior distributions

- The data model/sampling density for n observations:
$$Y_i \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma).$$
- There are two parameters μ and σ in the Normal model.
- Therefore, the likelihood is in terms of both unknown parameters:
$$f(y_1, \cdots, y_n) = L(\mu, \sigma). \tag{2}$$

Step 1: Prior distributions

- The data model/sampling density for n observations:
$$Y_i \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma).$$
- There are two parameters μ and σ in the Normal model.
- Therefore, the likelihood is in terms of both unknown parameters:
$$f(y_1, \cdots, y_n) = L(\mu, \sigma). \tag{2}$$
- Need a joint prior distribution:
$$\pi(\mu, \sigma). \tag{3}$$
- Bayes' rule will help us derive a joint posterior:
$$\pi(\mu, \sigma \mid y_1, \cdots, y_n). \tag{4}$$

If only mean μ is unknown: Normal conjugate prior

- For this special case, Normal prior for μ is a conjugate prior:
  - The prior distribution:

$$\mu \mid \sigma \sim \text{Normal}(\mu_0, \sigma_0). \tag{5}$$

  - The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \text{Normal}(\mu, \sigma). \tag{6}$$

If only mean μ is unknown: Normal conjugate prior

- For this special case, Normal prior for μ is a conjugate prior:
  - The prior distribution:
  $$\mu \mid \sigma \sim \text{Normal}(\mu_0, \sigma_0). \tag{5}$$
  - The sampling density:
  $$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \text{Normal}(\mu, \sigma). \tag{6}$$
  - The posterior distribution:
  $$\mu \mid y_1, \cdots, y_n, \phi \sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \tag{7}$$

  where $\phi = \frac{1}{\sigma^2}$ (and $\phi_0 = \frac{1}{\sigma_0^2}$), the precision. Since σ (and σ_0) is known, ϕ (and ϕ_0) is known too.
- We can then use the `rnorm()` R function to sample posterior draws of μ from Equation (7). Known quantities: $\phi_0, \mu_0, n, \bar{y}, \phi$

If only standard deviation σ is unknown: Gamma conjugate prior for $1/\sigma^2$

- For this special case, Gamma prior for $1/\sigma^2$ is a conjugate prior:
  - The prior distribution:

$$1/\sigma^2 \mid \mu \sim \mathrm{Gamma}(\alpha, \beta). \tag{8}$$

  - The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma). \tag{9}$$

If only standard deviation σ is unknown: Gamma conjugate prior for $1/\sigma^2$

- For this special case, Gamma prior for $1/\sigma^2$ is a conjugate prior:
  - The prior distribution:

$$1/\sigma^2 \mid \mu \sim \mathrm{Gamma}(\alpha, \beta). \tag{8}$$

  - The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma). \tag{9}$$

  - The posterior distribution:

$$1/\sigma^2 \mid y_1, \cdots, y_n, \mu \sim \mathrm{Gamma}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right) \tag{10}$$

- We can then use `rgamma()` R function to sample posterior draws of σ from Equation (10). Known quantities: $\alpha, n, \beta, \{y_i\}, \mu$

Go back to: both μ and σ are unknown

- Given what we have seen, how about a joint prior distribution as:

$$\pi(\mu, \sigma) = \pi_1(\mu)\pi_2(\sigma), \tag{11}$$

and let priors be

$$\mu \sim \text{Normal}(\mu_0, \sigma_0), \tag{12}$$

$$1/\sigma^2 \sim \text{Gamma}(\alpha, \beta). \tag{13}$$

Go back to: both μ and σ are unknown

- Given what we have seen, how about a joint prior distribution as:

$$\pi(\mu, \sigma) = \pi_1(\mu)\pi_2(\sigma), \tag{11}$$

and let priors be

$$\begin{aligned} \mu &\sim \mathrm{Normal}(\mu_0, \sigma_0), \qquad (12) \\ 1/\sigma^2 &\sim \mathrm{Gamma}(\alpha, \beta). \qquad (13) \end{aligned}$$

Bayes' rule will produce two full conditional posterior distributions:

$$\mu \mid y_1, \cdots, y_n, \phi \sim \mathrm{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \tag{14}$$

$$1/\sigma^2 = \phi \mid y_1, \cdots, y_n, \mu \sim \mathrm{Gamma}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right). \tag{15}$$

Go back to: both μ and σ are unknown

- Given what we have seen, how about a joint prior distribution as:

$$\pi(\mu, \sigma) = \pi_1(\mu)\pi_2(\sigma), \tag{11}$$

and let priors be

$$\begin{aligned} \mu &\sim \mathrm{Normal}(\mu_0, \sigma_0), \quad (12)\\ 1/\sigma^2 &\sim \mathrm{Gamma}(\alpha, \beta). \quad (13) \end{aligned}$$

Bayes' rule will produce two full conditional posterior distributions:

$$\mu \mid y_1, \cdots, y_n, \phi \sim \mathrm{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \tag{14}$$

$$1/\sigma^2 = \phi \mid y_1, \cdots, y_n, \mu \sim \mathrm{Gamma}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right). \tag{15}$$

- Why and how to derive them? Key: independent priors.

Sampling scheme: A Gibbs sampler

$$
\begin{aligned}
\mu \mid y_1, \cdots, y_n, \phi &\sim \mathrm{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \\
1/\sigma^2 = \phi \mid y_1, \cdots, y_n, \mu &\sim \mathrm{Gamma}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right).
\end{aligned}
$$

Sampling scheme: A Gibbs sampler

$$\mu \mid y_1, \cdots, y_n, \phi \sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\phi\bar{y}}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right),$$

$$1/\sigma^2 = \phi \mid y_1, \cdots, y_n, \mu \sim \text{Gamma}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right).$$

Start with initial values for parameters, $(\mu^{(0)}, \phi^{(0)})$. For $s = 1, \ldots, S$, generate from the following sequence of full conditional posterior distributions:

- $\mu^{(s)} \sim p(\mu \mid \phi^{(s-1)}, Y)$
- $\phi^{(s)} \sim p(\phi \mid \mu^{(s)}, Y)$
- Set $\theta^{(s)} = (\mu^{(s)}, \phi^{(s)})$

The sequence $\{\theta^{(s)}\colon\ s = 1, \ldots, S\}$ may be viewed (but is not necessarily...yet) as a dependent sample from the joint posterior distribution of (μ, ϕ).

Use R/RStudio to run a Gibbs sampler

```r
gibbs_normal <- function(input, S, seed){
  set.seed(seed)
  ybar <- mean(input$y)
  n <- length(input$y)
  para <- matrix(0, S, 2)
  phi <- input$phi_init
  for(s in 1:S){
    mu1 <- (input$mu_0/input$sigma_0^2 + n*phi*ybar)/
    (1/input$sigma_0^2 + n*phi)
    sigma1 <- sqrt(1/(1/input$sigma_0^2 + n*phi))
    mu <- rnorm(1, mean = mu1, sd = sigma1)
    alpha1 <- input$alpha + n/2
    beta1 <- input$beta + sum((input$y - mu)^2)/2
    phi <- rgamma(1, shape = alpha1, rate = beta1)
    para[s, ] <- c(mu, phi)
  }
  para }
```

Use R/RStudio to run a Gibbs sampler cont'd

- Run the Gibbs sampler:

```
input <- list(y = CEsample$LogTotalExpLastQ, mu_0 = 5,sigma_0 = 1,
alpha = 1, beta = 1,phi_init = 1)
output <- gibbs_normal(input, S = 10000, seed = 123)
```

Use R/RStudio to run a Gibbs sampler cont'd

- Run the Gibbs sampler:

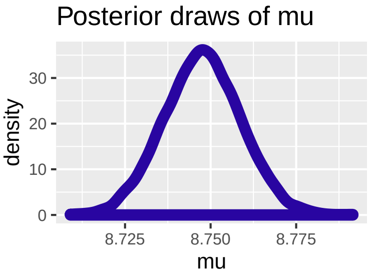
```
input <- list(y = CEsample$LogTotalExpLastQ, mu_0 = 5,sigma_0 = 1,
alpha = 1, beta = 1,phi_init = 1)
output <- gibbs_normal(input, S = 10000, seed = 123)
```

- Extract posterior draws of `mu` and `phi` from the Gibbs sampler output:

```
para_post <- as.data.frame(output)
names(para_post) <- c("mu", "phi")
```

Use R/RStudio to run a Gibbs sampler cont'd

```
ggplot(para_post, aes(mu)) +
  geom_density(size = 2, color = crcblue) +
  labs(title = "Posterior draws of mu") +
  theme_grey(base_size = 8,
  base_family = "")
```

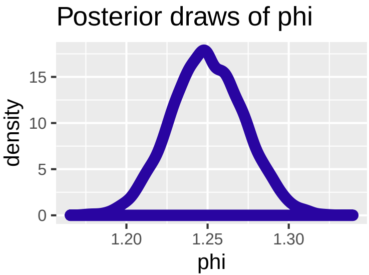


```
quantile(para_post$mu, c(0.025,0.975))
```

```
##     2.5%    97.5%
## 8.725482 8.769543
```

Use R/RStudio to run a Gibbs sampler cont'd

```
ggplot(para_post, aes(phi)) +
  geom_density(size = 2, color = crcblue) +
  labs(title = "Posterior draws of phi") +
  theme_grey(base_size = 8,
  base_family = "")
```



```
quantile(para_post$phi, c(0.025,0.975))
```

```
##      2.5%    97.5% 
## 1.206482 1.294191
```

Exercise: Gibbs sampler

- Exercise 1: Update the following Gibbs sampler to initiate it with `mu_init`. Compare your results to the existing results.
- Exercise 2: Update the following Gibbs sampler to initiate it with `mu_init` and `phi_init`. Compare your results to the existing results.

Extends to more than 2 parameters

Suppose the full conditional posterior distributions of $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$ are all tractable.

- Start the Gibbs sampler at the initial value $(\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_m^{(0)})$, or a subset of them.
- For the $(s+1)$-th iteration: generate from the following full conditionals sequentially:
  - $\theta_1^{(s+1)} \sim p(\theta_1 \mid \theta_2^{(s)}, \theta_3^{(s)}, \theta_4^{(s)}, \ldots, \theta_m^{(s)}, Y)$
  - $\theta_2^{(s+1)} \sim p(\theta_2 \mid \theta_1^{(s+1)}, \theta_3^{(s)}, \theta_4^{(s)}, \ldots, \theta_m^{(s)}, Y)$
  - $\theta_3^{(s+1)} \sim p(\theta_3 \mid \theta_1^{(s+1)}, \theta_2^{(s+1)}, \theta_4^{(s)}, \ldots, \theta_m^{(s)}, Y)$
  - $\vdots$
  - $\theta_m^{(s+1)} \sim p(\theta_m \mid \theta_1^{(s+1)}, \theta_2^{(s+1)}, \theta_3^{(s+1)}, \ldots, \theta_{m-1}^{(s+1)}, Y)$

Outline

1. Example: Expenditures in the Consumer Expenditure Surveys

2. Prior and posterior distributions for mean AND standard deviation

3. Use JAGS (Just Another Gibbs Sampler) and Bayesian inferences

4. MCMC diagnostics

5. Recap

JAGS for unknown mean and standard deviation case

- R package `runjags` to run Markov chain Monte Carlo simulations.
- Descriptive of the sampling model and the prior.

JAGS for unknown mean and standard deviation case

- R package runjags to run Markov chain Monte Carlo simulations.
- Descriptive of the sampling model and the prior.
- Installing JAGS software and runjags R package
  - Download JAGS at this link
  - Install and load runjags R package

```
install.packages("runjags")
```

```
library(runjags)
```

JAGS for unknown mean and standard deviation case cont'd

- Example: Normal model with unknown mean and standard deviation.
  - The prior distributions:

$$\begin{aligned} \mu &\sim \text{Normal}(\mu_0, \sigma_0), \\ 1/\sigma^2 = \phi &\sim \text{Gamma}(\alpha, \beta). \end{aligned}$$

  - The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \text{Normal}(\mu, \sigma).$$

  - Bayes' rule will produce two full conditional posterior distributions:

$$\begin{aligned} \mu \mid y_1, \cdots, y_n, \phi &\sim \text{Normal}\left(\frac{\phi_0\mu_0 + n\bar{y}\phi}{\phi_0 + n\phi}, \sqrt{\frac{1}{\phi_0 + n\phi}}\right), \\ 1/\sigma^2 = \phi \mid y_1, \cdots, y_n, \mu &\sim \text{Gamma}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right). \end{aligned}$$

JAGS for unknown mean and standard deviation case cont'd

- Only need to focus on the sampling density and the prior:
  - The sampling density:

$$y_1, \cdots, y_n \mid \mu, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu, \sigma).$$

  - The prior distributions:

$$\begin{aligned} \mu &\sim \mathrm{Normal}(\mu_0, \sigma_0), \\ 1/\sigma^2 = \phi &\sim \mathrm{Gamma}(\alpha, \beta). \end{aligned}$$

```
modelString <- "
model{
for (i in 1:N) {
y[i] ~ dnorm(mu, phi)
}
mu ~ dnorm(mu_0, phi_0)
phi ~ dgamma(alpha, beta)
}
"
```

JAGS for unknown mean and standard deviation case cont'd

- Pass the data and hyperparameter values to JAGS:

```
y <- CEsample$LogTotalExpLastQ
N <- length(y)
the_data <- list("y" = y, "N" = N, "mu_0" = 5, "phi_0" = 1/1^2,
"alpha" = 1,"beta" = 1)
```

JAGS for unknown mean and standard deviation case cont'd

- Pass the data and hyperparameter values to JAGS:

```
y <- CEsample$LogTotalExpLastQ
N <- length(y)
the_data <- list("y" = y, "N" = N, "mu_0" = 5, "phi_0" = 1/1^2,
"alpha" = 1,"beta" = 1)
```

- Run the JAGS code for this model:

```
posterior <- run.jags(modelString,
                  data = the_data,
                  monitor = c("mu", "phi"),
                  n.chains = 1,
                  adapt = 1000,
                  burnin = 2000,
                  sample = 5000,
                  thin = 1)
```

JAGS for unknown mean and standard deviation case cont'd

- Obtain posterior summaries of mu and phi:

```
summary(posterior)
```

```
##     Lower95   Median Upper95     Mean         SD Mode        MCerr MC%ofSD
## mu  8.72529 8.747810 8.76868 8.747781 0.01111673   NA 0.0001572144     1.4
## phi 1.20652 1.251015 1.29268 1.250759 0.02206280   NA 0.0003091955     1.4
##     SSeff        AC.10 psrf
## mu   5000  0.030116334   NA
## phi  5092 -0.002062701   NA
```

Outline

1. Example: Expenditures in the Consumer Expenditure Surveys
2. Prior and posterior distributions for mean AND standard deviation
3. Use JAGS (Just Another Gibbs Sampler) and Bayesian inferences
4. MCMC diagnostics
5. Recap

MCMC Convergence to posterior distribution

Theory proves that if a Gibbs sampler iterates enough, the draws will be from the joint posterior distribution (called the target or stationary distribution).

MCMC Convergence to posterior distribution

Theory proves that if a Gibbs sampler iterates enough, the draws will be from the joint posterior distribution (called the target or stationary distribution).

- Do initial values matter? Should they matter?
- "Markov chain" indicate dependence of draws. How to create independent parameter draws?
- How long do we need to run the MCMC to adequately explore the posterior distribution?
- How can we tell if the chain is not converging?

Discard initial steps: burn-in

- Do not want pre-convergence values to influence the summary of the posterior distribution (the draws of the parameters from the MCMC) too much.
- Ideal is to throw away all draws before convergence to target distribution, and use only draws after convergence.
- Hard to know exactly when convergence has happened. Thus, it is standard to throw out the first $p\%$ of draws (default is 50%) as burn-in after you are satisfied that convergence has been achieved.
- Inference done with remaining $(100 - p\%)$ of draws.

Discard initial steps: burn-in cont'd

```r
posterior <- run.jags(modelString,
                 data = the_data,
                 monitor = c("mu", "phi"),
                 n.chains = 1,
                 adapt = 1000,
                 burnin = 2000,
                 sample = 5000,
                 thin = 1)
```

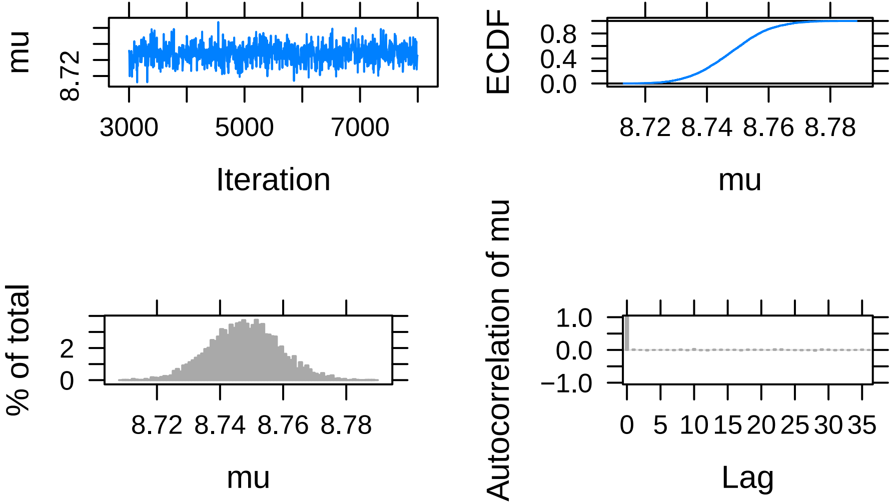
Trace plots to check MCMC mixing

- Trace plots show the progress of the chain over time. The ideal trace plot has a lot of movement around the mode from iteration to iteration and not a lot of "stickiness."
- Trace plots with a clear trend are evidence that MCMC sampler has not yet converged, and it needs to run longer (or there is an error in your code).
- Trace plots with extreme stickiness suggest that the chain may not have explored the entire parameter space yet, and it needs to run longer.
- Hard to tell convergence from trace plots alone, because you might be stuck in one region of the parameter space in all of the iterations.

Trace plots example

```r
plot(posterior, vars = "mu")
```

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



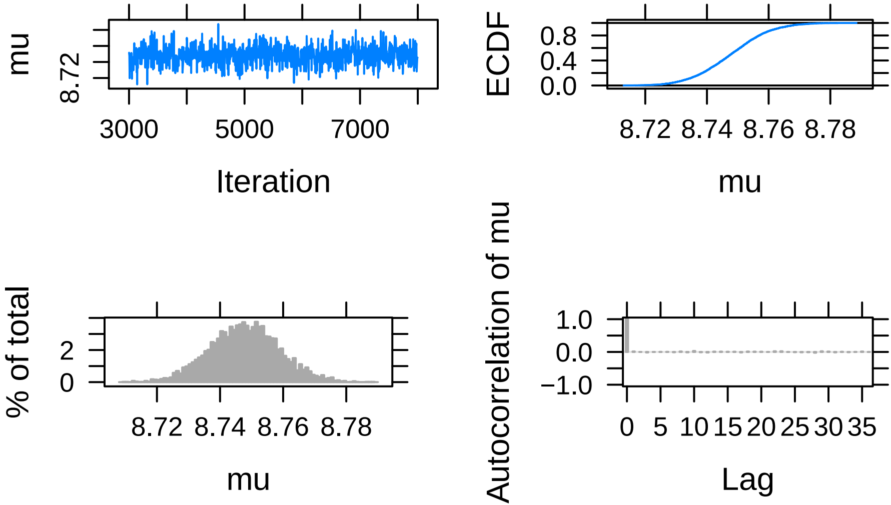
ACF plots and effective sample size to check correlation diagnostics

- Independent (Monte Carlo) draws have zero autocorrelations. Dependent draws (MCMC) have non-zero, usually positive, autocorrelations.
- Large autocorrelations indicate that chain is not mixing well, i.e., the chain possibly has not explored the full space of the posterior distribution (has not converged).
- Large autocorrelations often arise in multi-parameter MCMC algorithms when the parameters are highly correlated.
- This usually means you need to run the chains longer to feel comfortable about convergence. It also implies a smaller effective sample size. . . .
- ACF plots (auto correlation function) & effective sample size.

ACF plots example

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

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



Effective sample size example

- The column of `SSeff`; recall `sample` is 5000.

```
summary(posterior)
```

```
##     Lower95   Median Upper95     Mean         SD Mode        MCerr MC%ofSD
## mu  8.72529 8.747810 8.76868 8.747781 0.01111673   NA 0.0001572144     1.4
## phi 1.20652 1.251015 1.29268 1.250759 0.02206280   NA 0.0003091955     1.4
##     SSeff        AC.10 psrf
## mu   5000  0.030116334   NA
## phi  5092 -0.002062701   NA
```

Create independent samples: thin

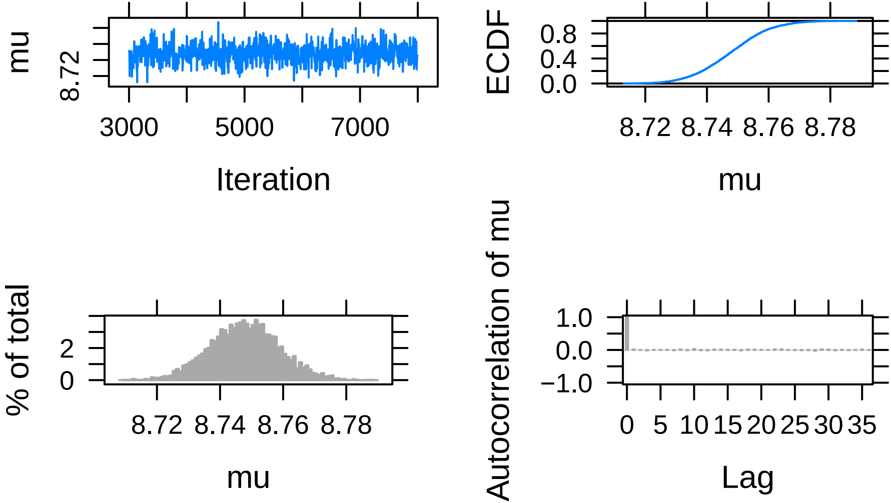
- Monte Carlo samples come from independent parameter draws, thus they move around the parameter space freely.
- Markov chain properties could create dependence among adjacent parameter draws.
- Use thin to save every x draws.

```
posterior <- run.jags(modelString,
                  data = the_data,
                  monitor = c("mu", "phi"),
                  n.chains = 1,
                  adapt = 1000,
                  burnin = 2000,
                  sample = 5000,
                  thin = 1)
```

MCMC diagnostics for the CE example

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

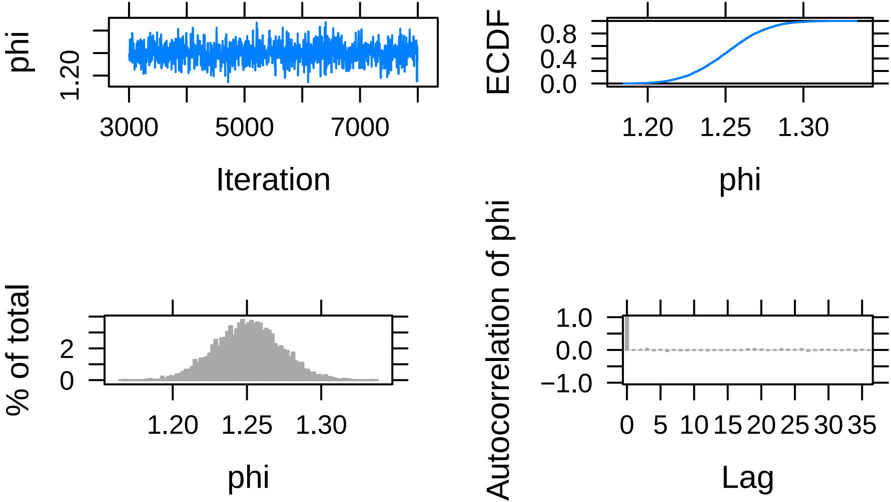
```
## Generating plots...
```



MCMC diagnostics for the CE example cont'd

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

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



Gelman-Rubin diagnostics

- Gelman and Rubin (1992) propose running $m > 1$ separate Markov chains each of length S. When convergence is reached, the draws from the chains are all from the target distribution. Hence, at convergence, the output from all chains is indistinguishable.
- Diagnostic based on the contrapositive: if output from chains is not indistinguishable, convergence has not been reached. Diagnostic applied to one variable at a time.
- Use different starting values that are overdispersed (really spread out) relative to the posterior distribution. This helps ensure all the chains are not stuck in the same region.
- Gelman-Rubin diagnostic based a comparison of within-chain and between-chain variances, which is similar to a classical analysis of variance.

Gelman-Rubin diagnostics example

- Create intinial values of mu and phi:

```
inits1 <- dump.format(list(mu = 1, phi = 1,
            .RNG.name="base::Super-Duper", .RNG.seed = 1))
inits2 <- dump.format(list(mu = 10, phi = 10,
            .RNG.name="base::Wichmann-Hill", .RNG.seed = 2))
```

Gelman-Rubin diagnostics example

- Create intinial values of mu and phi:

```
inits1 <- dump.format(list(mu = 1, phi = 1,
            .RNG.name="base::Super-Duper", .RNG.seed = 1))
inits2 <- dump.format(list(mu = 10, phi = 10,
            .RNG.name="base::Wichmann-Hill", .RNG.seed = 2))
```

- Feed in inits1 and inits2, and let n.chains = 2:

```
posterior_2chains <- run.jags(modelString,
                      data = the_data,
                      monitor = c("mu", "phi"),
                      n.chains = 2,
                      inits=c(inits1, inits2),
                      adapt = 1000,
                      burnin = 2000,
                      sample = 5000,
                      thin = 1)
```

Gelman-Rubin diagnostics example cont'd

- Return psrf from the output, as Gelman-Rubin diagnostic results:

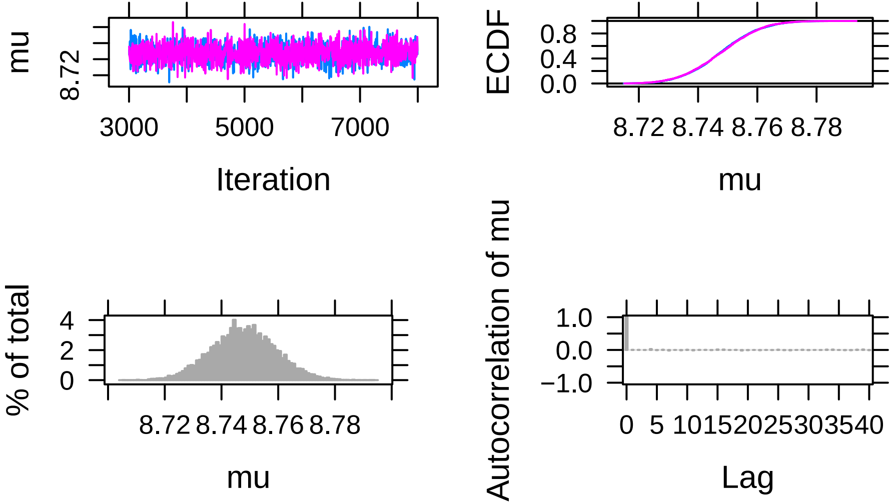
```
posterior_2chains$psrf
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## mu           1          1
## phi          1          1
##
## Multivariate psrf (for all monitored variables):
##
## 1
##
## Target psrf
##
## 1.05
```

MCMC diagnostics for the CE example, 2 chains

```
plot(posterior_2chains, vars = "mu")
```

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



Useful diagnostics/functions in coda package

The `coda` R package provides many popular diagnostics for assessing convergence of MCMC output from self-coded Gibbs sampler.

- Trace plots: `traceplot()`
- Autocorrelation: `autocorr.plot()`
- Effective Sample Size: `effectiveSize()`
- Gelman-Rubin: `gelman.diag()`

Useful diagnostics/functions in coda package

- One needs to convert parameter draws into an MCMC object. For example:

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

```r
library(coda)
out <- gibbs_normal(input, S = 10000, seed = 123)
para_post = as.data.frame(out)
names(para_post) = c("mu", "phi")
```

Useful diagnostics/functions in coda package

- One needs to convert parameter draws into an MCMC object. For example:

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

```r
library(coda)
out <- gibbs_normal(input, S = 10000, seed = 123)
para_post = as.data.frame(out)
names(para_post) = c("mu", "phi")
```

- Then one can perform MCMC diagnostics. For example:

```r
mu.mcmc = as.mcmc(para_post$mu)
```

```r
traceplot(mu.mcmc)
autocorr.plot(mu.mcmc)
effectiveSize(mu.mcmc)
gelman.diag(mu.mcmc)
```

Note: `gelman.diag()` needs at least 2 chains.

Outline

1. Example: Expenditures in the Consumer Expenditure Surveys

2. Prior and posterior distributions for mean AND standard deviation

3. Use JAGS (Just Another Gibbs Sampler) and Bayesian inferences

4. MCMC diagnostics

5. Recap

Recap

- Bayesian inference procedure:
  - Step 1: express an opinion about the location of mean μ and standard deviation σ (or precision ϕ) before sampling (prior).
  - Step 2: take the sample (data/likelihood).
  - Step 3: use Bayes' rule to sharpen and update the previous opinion about μ and σ (or precision ϕ) given the information from the sample (posterior).

Recap

- Bayesian inference procedure:
  - Step 1: express an opinion about the location of mean μ and standard deviation σ (or precision ϕ) before sampling (prior).
  - Step 2: take the sample (data/likelihood).
  - Step 3: use Bayes' rule to sharpen and update the previous opinion about μ and σ (or precision ϕ) given the information from the sample (posterior).
- For Normal data/likelihood, Normal distributions are conjugate priors for μ, and Gamma distributions are conjugate priors for ϕ. The same goes for the full conditional posterior distributions.

Recap

- Bayesian inference procedure:
  - Step 1: express an opinion about the location of mean μ and standard deviation σ (or precision ϕ) before sampling (prior).
  - Step 2: take the sample (data/likelihood).
  - Step 3: use Bayes' rule to sharpen and update the previous opinion about μ and σ (or precision ϕ) given the information from the sample (posterior).
- For Normal data/likelihood, Normal distributions are conjugate priors for μ, and Gamma distributions are conjugate priors for ϕ. The same goes for the full conditional posterior distributions.
- One can either write a loop function to run a Gibbs sampler, or use JAGS.

Recap cont'd

- MCMC diagnostics!
    - Diagnostics cannot guarantee that chain has converged.
    - Diagnostics can indicate that it has not converged.
    - Solutions: run longer, thinning, standardize variables in multi-parameter models (can reduce auto correlations).