Potential Scale Reduction Factor ($\widehat{R}$) in coda

as summarized by

Matt Tyers, Doug Anderson, and Leslie Gaines-Germaine

1 Background:

Calculation of the potential scale reduction factor (psrf), or $\widehat{R}$, using `gelman.diag()` via the R package `coda` uses the within-chain variance (W) and between-chain variance B employed in the Gelman et. al text (though without splitting the chains), but uses them differently, and allows for multivariate chains.

The function `gelman.diag()` requires the input to be a mcmc object, and allows the user to specify whether to use a transformation on the chain, whether to accept an "auto-burnin", whether the chains are to be considered multivariate, and a confidence level for the upper CI limit reported.

If the "auto-burnin" is accepted, `gelman.diag()` discards the first half of the chains as burnin, otherwise, the full chains are retained. It then extracts the number of iterations, the number of chains, the number of variables used, and the variable names from the mcmc object, and then converts the mcmc object to a matrix. It then computes the variance of each chain s_j^2, average within-chain variance W, and between-chain variance B, using the same calculations as given in Gelman, et. al. However, s_j^2, W, and B ar stored as vectors with a value for each variable in the mcmc object the function is given. In the non-multivariate case, it generates diagonal matrixes $s2$, w, and b for each, and all subsequent calculations are done using these matrices, which cuts down on processing time.

2 The Steps (with R code):

The steps that we used on 4 chains of 1000 iterations where each chain is stored in a 1×2 matrix called `samp#`. (We focused on the second parameter.)

1. We will begin with calculating the between-chain variance B as outlined by Gelman et. al. in BDA3 (m is the number of chains, n is the number of iterations per chain):

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\overline{\psi}_{.j} - \overline{\psi}_{..})^2, \quad \text{where}$$

$$\overline{\psi}_{.j} = \frac{1}{n}\sum_{i=1}^{n}\psi_{ij}$$

$$\overline{\psi}_{..} = \frac{1}{m}\sum_{j=1}^{m}\overline{\psi}_{.j}$$

Using R, this is done by

```
n <- 1000 # one-thousand iterations in each chain
m <- 4 ## four chains
    psi_bar.1 <- (1/n) * sum(samp1[,2]) # chain 1, parameter of choice
    psi_bar.2 <- (1/n) * sum(samp2[,2]) # chain 2, parameter of choice
    psi_bar.3 <- (1/n) * sum(samp3[,2]) # chain 3, parameter of choice
    psi_bar.4 <- (1/n) * sum(samp4[,2]) # chain 4, parameter of choice
  psi_bar.j <- c(psi_bar.1, psi_bar.2, psi_bar.3, psi_bar.4)
## Compute psi_bar..
  psi_bar.. <- ( 1/m ) * sum( psi_bar.j )
## Compute B (between-chain variance)
  B <- ( n/(m-1) ) * sum( (psi_bar.j - psi_bar..)^2 )
```

2. Next, compute the the within-chain variance W as outlined in BDA3:

$$W = \frac{1}{m}\sum_{j=1}^{m} s_j^2, \quad \text{where}$$

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\psi_{ij} - \overline{\psi}_{.j})^2$$

```
## Compute s_j_sq
    s_1_sq <- ( 1/(n-1) ) * sum( (samp1[,2] - psi_bar.1)^2 )
    s_2_sq <- ( 1/(n-1) ) * sum( (samp2[,2] - psi_bar.2)^2 )
    s_3_sq <- ( 1/(n-1) ) * sum( (samp3[,2] - psi_bar.3)^2 )
    s_4_sq <- ( 1/(n-1) ) * sum( (samp4[,2] - psi_bar.4)^2 )
  s_j_sq <- c(s_1_sq, s_2_sq, s_3_sq, s_4_sq)
## Compute W (within-chain variance)
  W <- ( 1/m ) * sum( s_j_sq )
## Compute sigma_hat_sq
  sigma_hat_sq <- ( (n-1)/n ) * W + ( 1/n ) * B
```

3. Next is to calculate what Gelman et. al. call $\widehat{\text{var}}^+(\psi|y)$ or the `coda` package refers to as `sigma.hat^2` (or $\widehat{\sigma}^2$):

$$\widehat{\text{var}}^+(\psi|y) = \frac{n-1}{n}W + \frac{1}{n}B$$

```
  sigma_hat_sq <- ( (n-1)/n ) * W + ( 1/n ) * B
```

4. Calculate $\widehat{R}$:

   (a) In BDA3 there is only one more step required

   $$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^+(\psi|y)}{W}}$$

   ```
   Rhat_BDA3 <- sqrt( sigma_hat_sq / W )
   ```

   (b) The `coda` package does things slightly different

i. Compute what the authors call `V.hat`:

$$\widehat{V} = \widehat{\sigma}^2 + \frac{B}{mn}$$

```
V_hat_coda <- sigma_hat_sq + (B / ( m*n ))
```

ii. Calculate the degrees of freedom estimated by the method of moments. (Here I have followed mostly how SAS estimates d (similar to Matt.)

$$\begin{aligned}
\widehat{d} &= \frac{2\widehat{V}^2}{\widehat{\text{var}}(\widehat{V})}, \quad \text{where} \\
\widehat{\text{var}}(\widehat{V}) &= \left(\frac{n-1}{n}\right)^2 \text{var}(W) + \left(\frac{1}{n}\right)^2 \text{var}(B) + 2\left(\frac{n-1}{n^2}\right) \text{cov}(W,B) \\
\text{var}(W) &= \frac{1}{m}\text{var}(s_j^2) \\
\text{var}(B) &= \frac{2B^2}{m-1} \\
\text{cov}(W,B) &= \left(\frac{n}{m}\right)\left[\text{cov}(s_j^2, \overline{\psi}_{.j}^2) - 2\overline{\psi}_{..}\text{cov}(s_j^2, \overline{\psi}_{.j})\right]
\end{aligned}$$

```
## compute variance of V_hat
var_V_hat_1_coda <- ((n-1)/n)^2 * (1/m) * var(s_j_sq)
var_V_hat_2_coda <- (1/n)^2 * (2/(m-1)) * B^2
var_V_hat_3_coda <- 2 * ((n-1)/n^2) * (n/m) *
                      ( cov(s_j_sq,psi_bar.j^2) -
                        2 * psi_bar.. * cov(s_j_sq,psi_bar.j) )
var_V_hat_coda <- var_V_hat_1_coda + var_V_hat_2_coda +
                      var_V_hat_3_coda
## compute the degrees estimated by the method of moments
d_coda <- 2 * V_hat_coda^2 / var_V_hat_coda
```

iii. Calculate $\widehat{R}$ as the `coda` package instructs:

$$\widehat{R} = \sqrt{\left(\frac{\widehat{d}+3}{\widehat{d}+1}\right)\frac{\widehat{V}}{W}}$$

```
Rhat_coda <- sqrt( ((d_coda+3) * V_hat_coda) / ((d_coda+1) * W) )
```

(c) The code above, if implemented, does not yield the exact results of the function `gelman.diag()` but is close. Perhaps, how Matt outlined previously (how SAS computes $\widehat{R}$) is how the package calculates $\widehat{R}$.

i. Compute $\widehat{V}$:

$$\widehat{V} = \frac{n-1}{n}W + \frac{m+1}{mn}B$$

```
V_hat_SAS <- ((n-1)/n) * W + ( (m+1)/(m*n) ) * B
```

ii. Compute $\widehat{d}$ (there are subtle differences between this method and the previous method):

$$\widehat{d} = \frac{2\widehat{V}^2}{\widehat{\mathrm{var}}(\widehat{V})}, \quad \text{where}$$

$$\widehat{\mathrm{var}}(\widehat{V}) = \left(\frac{n-1}{n}\right)^2 \mathrm{var}(W) + \left(\frac{m+1}{mn}\right)^2 \mathrm{var}(B) + 2\left(\frac{(m+1)(n-1)}{mn^2}\right)\mathrm{cov}(W,B)$$

$$\mathrm{var}(W) = \frac{1}{m}\mathrm{var}(s_j^2)$$

$$\mathrm{var}(B) = \frac{2B^2}{m-1}$$

$$\mathrm{cov}(W,B) = \left(\frac{n}{m}\right)\left[\mathrm{cov}(s_j^2, \overline{\psi}_{.j}^2) - 2\overline{\psi}_{..}\mathrm{cov}(s_j^2, \overline{\psi}_{.j})\right]$$

```
## compute variance of V_hat
var_V_hat_1_SAS <- ( (n-1)/n )^2 * (1/m) * var(s_j_sq)
var_V_hat_2_SAS <- ( (m+1)/(m*n) )^2 * (2/(m-1)) * B^2
var_V_hat_3_SAS <- 2 * ( (m+1)*(n-1)/(m*n^2) ) * (n/m) *
                     ( cov(s_j_sq,psi_bar.j^2) -
                       2 * psi_bar.. * cov(s_j_sq,psi_bar.j) )
var_V_hat_SAS <- var_V_hat_1_SAS + var_V_hat_2_SAS + var_V_hat_3_SAS
## compute the degrees estimated by the method of moments
d_SAS <- 2 * V_hat_SAS^2 / var_V_hat_SAS
```

iii. Calculate $\widehat{R}$:

$$\widehat{R} = \sqrt{\left(\frac{\widehat{d}+3}{\widehat{d}+1}\right)\frac{\widehat{V}}{W}}$$

```
Rhat_sas <- sqrt( ((d_SAS+3) * V_hat_SAS) / ((d_SAS+1) * W) )
```

3 Warning and Upper Bound:

Following any of these procedures does not yield results perfectly matching the output obtained from using `gelman.diag()` function in `coda`: `gelman.diag()`=1.033328, $\widehat{R}_{BDA3} = 1.023483$, $\widehat{R}_{coda} = 1.032407$, and $\widehat{R}_{SAS} = 1.032702$ were estimates we found using fake data and `coda` output. However, the method used by SAS yielded the closest results. An upper $100(1-\alpha/2)\%$ confidence limit for $\widehat{R}$ can be computed by ($\widehat{d}$ should be consistent from previous calculations):

1. SAS method (and Matt):

$$\sqrt{\left(\frac{n-1}{n} + \frac{m+1}{mn}\cdot F_{1-\alpha/2}\left(m-1, \frac{2W^2}{\widehat{\mathrm{var}}(s_j^2)/m}\right)\right)\left(\frac{\widehat{d}+3}{\widehat{d}+1}\right)}$$

2. `coda` method (we are guessing here):

$$\sqrt{\left(\frac{n-1}{n} + \frac{1}{n}\cdot F_{1-\alpha/2}\left(m-1, \frac{2W^2}{\widehat{\mathrm{var}}(s_j^2)/m}\right)\right)\left(\frac{\widehat{d}+3}{\widehat{d}+1}\right)}$$