

## Effective Sample Size $\hat{n}_{eff}$

Garland Will & Leslie Gains-Germain & Matt Tyers

In Bayesian Data Analysis 3 by Gelman, et.al. effective sample size is calculated as

$$\hat{n}_{eff} = \frac{mn}{1 + 2 \sum_{t=1}^T \hat{\rho}_t}; \quad \hat{\rho}_t = 1 - \frac{V_t}{2\widehat{var}^+}$$

In the above formula,  $\widehat{var}^+$  is the marginal posterior variance, and is a weighted average of the between and within chain variance. M is the number of chains and n is the the number of draws from each chain.  $V_t$  is the variogram at each lag, where smaller  $V_t$  values indicate higher correlation.

Calculation of the effective sample size  $\hat{n}_{eff}$  in coda is done using a different (and much more code-efficient) technique than the algorithm presented in Chapter 11 of the Gelman, et.al. text. Function `effectiveSize()` first checks to see if it has been passed an MCMC list. If so, it breaks the list apart by variables and runs itself on the set of chains for each variable. Otherwise (or after this), it calls `spectrum0.ar()` to calculate the spectral density at frequency zero of each chain. `Spectrum0.ar()` fits an autoregressive model to the chain, treating the chain as a time series. The larger the variance left unexplained by the autoregressive model, the larger the spectral density at frequency zero, reflecting higher correlation within the chain.

If the spectral density is zero, i.e. there is no variance in the chain left unexplained by the autoregressive model, `effectiveSize` returns an effective sample size of  $\hat{n}_{eff} = 0$  for the chain. If not, it returns

$$\hat{n}_{eff} = \sum_{i=1}^K \frac{\text{number.of.iterations} * V(\text{chain}_i)}{\text{spec}_i}$$

Where  $V(\text{chain}_i)$  is the overall variance of chain  $i$ ,  $\text{spec}_i$  is the spectral density as described above for the  $i$ th chain, and  $K$  is the number of chains.

### Somewhat worrisome example

This example looks at simulating three chains of length 1,000 from a multivariate normal. All of the chains have a mean of 0, and correlation equal to 0.8. Looking at the traceplots, the three chains are very correlated with one another. Using the coda package to calculate effective sample size on each chain, we get values above 1,000 which is the number of draws for each chain. This serves as an illustration that n effective as calculated in coda is not a perfect diagnostic tool, as having an effective sample size greater than the number of draws is not reasonable. Also worth noting is that coda calculates n effective strictly within each chain. The method from the BDA3 on the other hand uses a variance that is a weighted average of the between and within chain variance.

```
set.seed(38)
require(coda, quietly = T)
require(mvtnorm, quietly = T)
d <- matrix(c(1, 0, 0, 0, 1, 0, 0, 0, 1), nrow = 3, ncol = 3)
```

```

r <- matrix(c(1, 0.8, 0.8, 0.8, 1, 0.8, 0.8, 0.8, 1), nrow = 3, ncol = 3)
v <- d %*% r %*% d
chains <- rmvnorm(1000, c(0, 0, 0), sigma = sqrt(v))

```

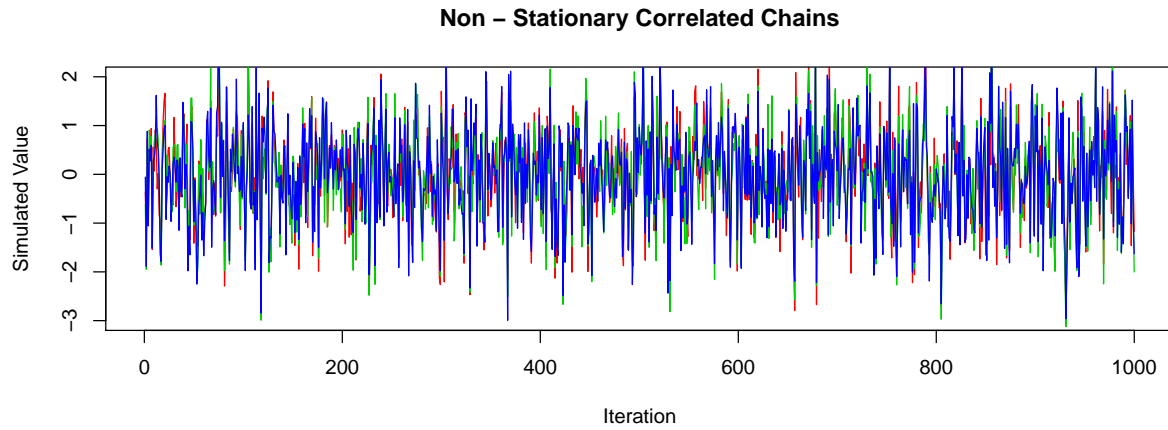


Table 1: Effective sample size	
	Effective Sample Size
Chain 1	1443.15
Chain 2	1210.35
Chain 3	1000.00