# Markov chain Monte Carlo storage guide

When using Markov chain Monte Carlo (MCMC) methods to sample $\theta^{(1)}, \ldots, \theta^{(m)} \in \mathbb{R}^p$, sequentially, from the joint posterior $\pi(\theta)$, the samples should be stores in pre-allocated storage. Relevant options are using either R data frames or R matrices, and either storing the samples row-wise or column-wise. If storing the samples column-wise, the storage should be transposed after being filled, and if using matrices, they should be converted back to data frames at the end. The use of R data frames for this task is heavily discouraged, as the execution time scales poorly with the number of samples, $m$, as evident from fig. 1.

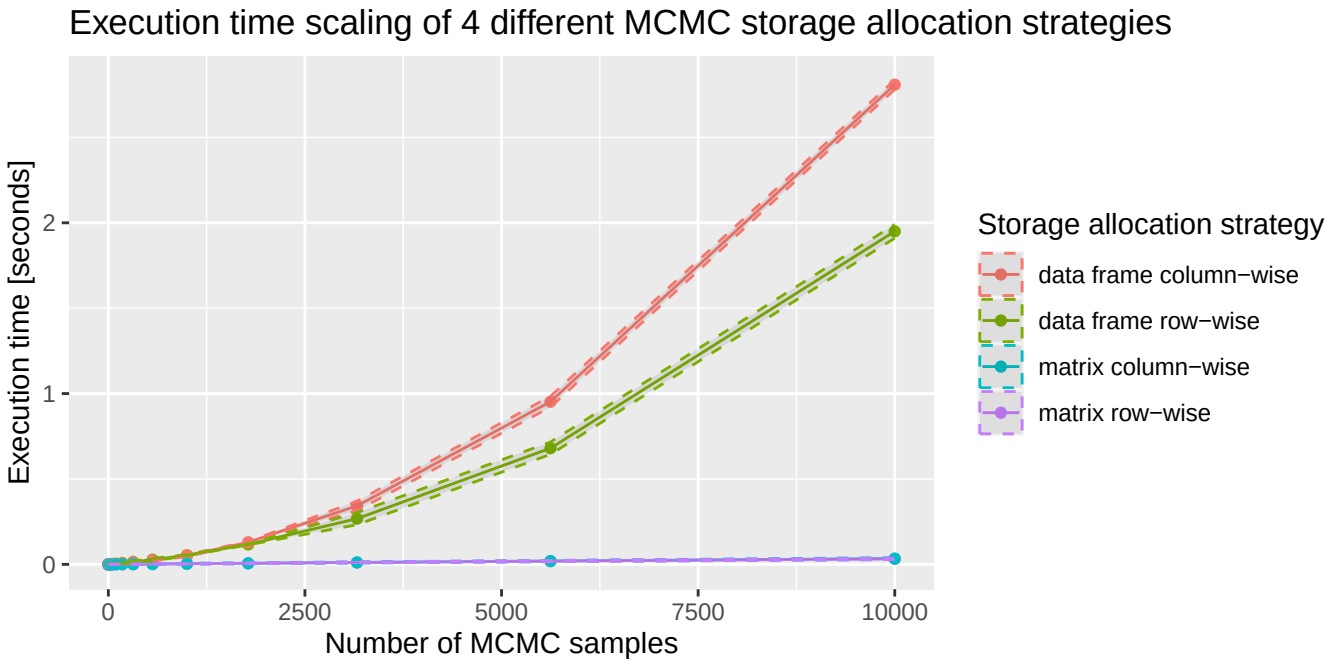


Figure 1: For each of the four storage allocation strategies described above, the matrix or dataframe was filled sequentially using a dummy MCMC sampler. Both the storage initialisation, the sampling and the transformation and conversion at the end, if applicable, was timed. This was done for $m$-values logarithmically spaced between 1 and $m_{max} = 10000$. For all the simulations, the dimensions of the posterior distribution was set to $p = 10$. The confidence intervals are asymptotic normal 95% intervals, based on repeating the simulation 8 times.

Instead, the storage should be pre-allocated using either a $p \times m$ or a $m \times p$ R matrix, and should then be filled column-wise or row-wise, respectively. The matrices should then be converted to a data frame after being filled, and if filling column-wise, the matrix should be transposed before converting to a data frame. There is no statistically significant difference between the two matrix approaches as their execution times for $m = 10000$ samples is very similar:

| **Storage strategy** | **95% asymptotic normal confidence interval for execution time** |
|---|---|
| Matrix row-wise | $0.0323 \pm 0.0039$ seconds |
| Matrix column-wise | $0.0337 \pm 0.0053$ seconds |

Therefore, either of the two matrix option are acceptable.