Exercise 2: Problem 1 f)

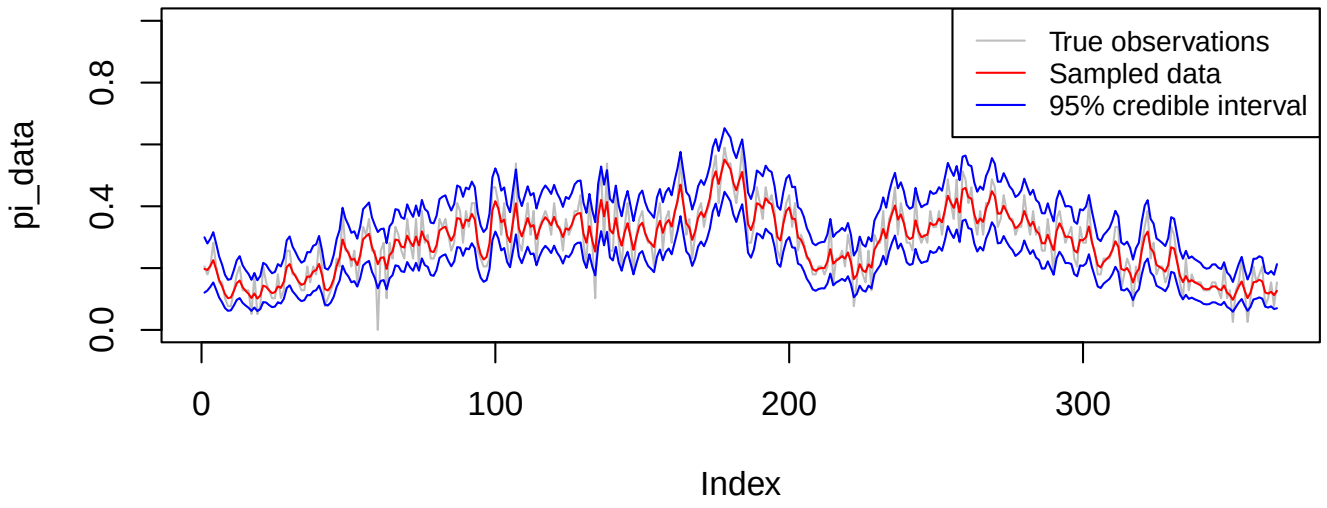
Jostein Aastebøl Aanes, Frederick Nilsen

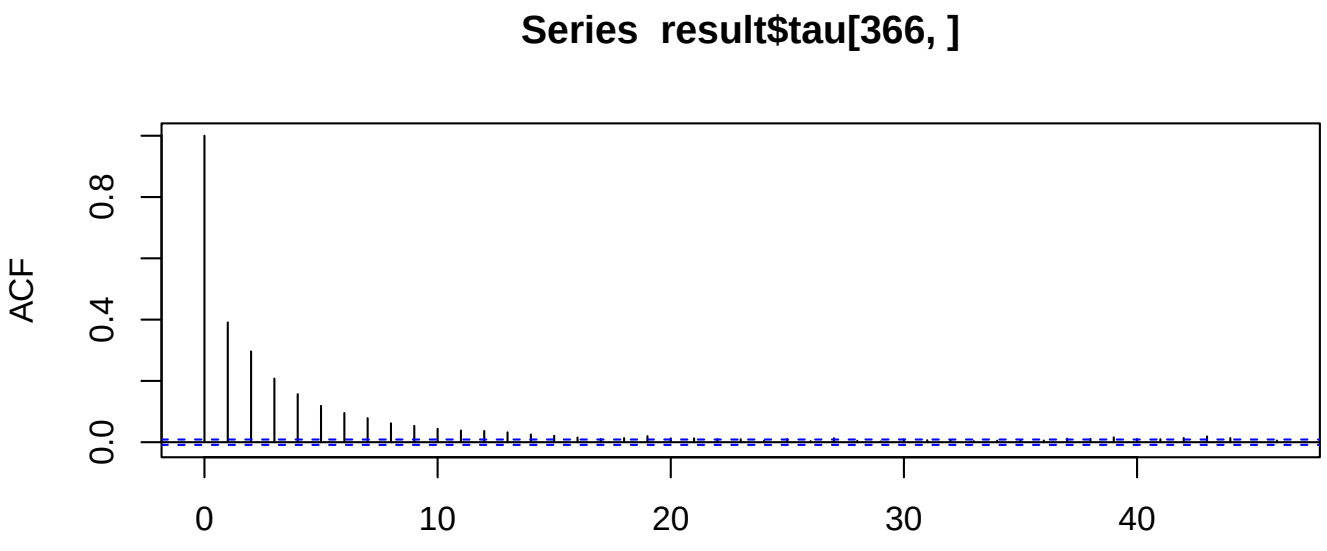
22.03.2022

Problem 1 f) - What does the task entail?

- Change MCMC sampler to allow for block updates
- Explore results for different choices of blocking parameter M.
- Compare runtime and predictions to the single-site sampler

Results from 1 e) - single site MCMC





Expected results from 1 f)

- Updating $\boldsymbol{\tau}$ as block should improve performance.
- Should get less variation and "more" stationarity
- Choice of M should be determining factor for runtime and performance
  - Runtime should decrease as M increases. Performance should improve for increasing M up to a certain threshold, in which the acceptance probability becomes too small.

Precomputations

- Before iterating, we precompute Q given by

$$Q = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$$

- Precompute for 3 different steps
  - First block is the first $M \times M$ block of Q, the second block is the block $(2 : M+1) \times (2 : M+1)$ and the last block is the last block from $M \cdot \lfloor 366/M \rfloor + 1$ to the end.
- Precompute also its inverse and Cholesky decomposition of its inverse
- Total of 9 precomputations

Results

- Incredibly high runtime, $\approx$ 300 seconds for 20 000 iterations
- Analysis shows that memory allocation dominate the time taken
- Results hard to extract when runtime is this long

Code	File	Memory (MB)			Time (ms)	
▼ profvis		-8432...		84102...	331450	
tau[, k] = MH_block_new(M, current_tau, sigma_u, alpha, beta, mu_factor_1, mu_factor_2, mu_factor_3, chol_11, chol_22, chol_33, blockList,...	<expr>	-7720...		71729...	265700	
▶ MCMC_block_sampler	<expr>	-6470...		11491...	62960	
.classEnv		-1519.9		899.5	780	
getClass		-506.5		561.0	370	
current_tau = tau[, k-1]	<expr>	-281.7		279.3	130	
setTxtProgressBar(pb,k) # Update progress bar	<expr>	-112.4		335.6	120	
variances[k-1] = get_variance(tau, alpha, beta, k)	<expr>	0		279.2	70	
▶ compiler:::tryCmpfun	<expr>	0		1.5	60	
sigma_u = sqrt(variances[k-1])	<expr>	-112.3		55.9	20	
for(k in 2:nsteps){	<expr>	0		0.0	10	
<Anonymous>		-0.2		0	10	

Provisory prediction plot

- Here, $M = 10, n = 5000$.