Comparison of MCMC algorithms in Stochastic Volatility Models using Simulation Based Calibration

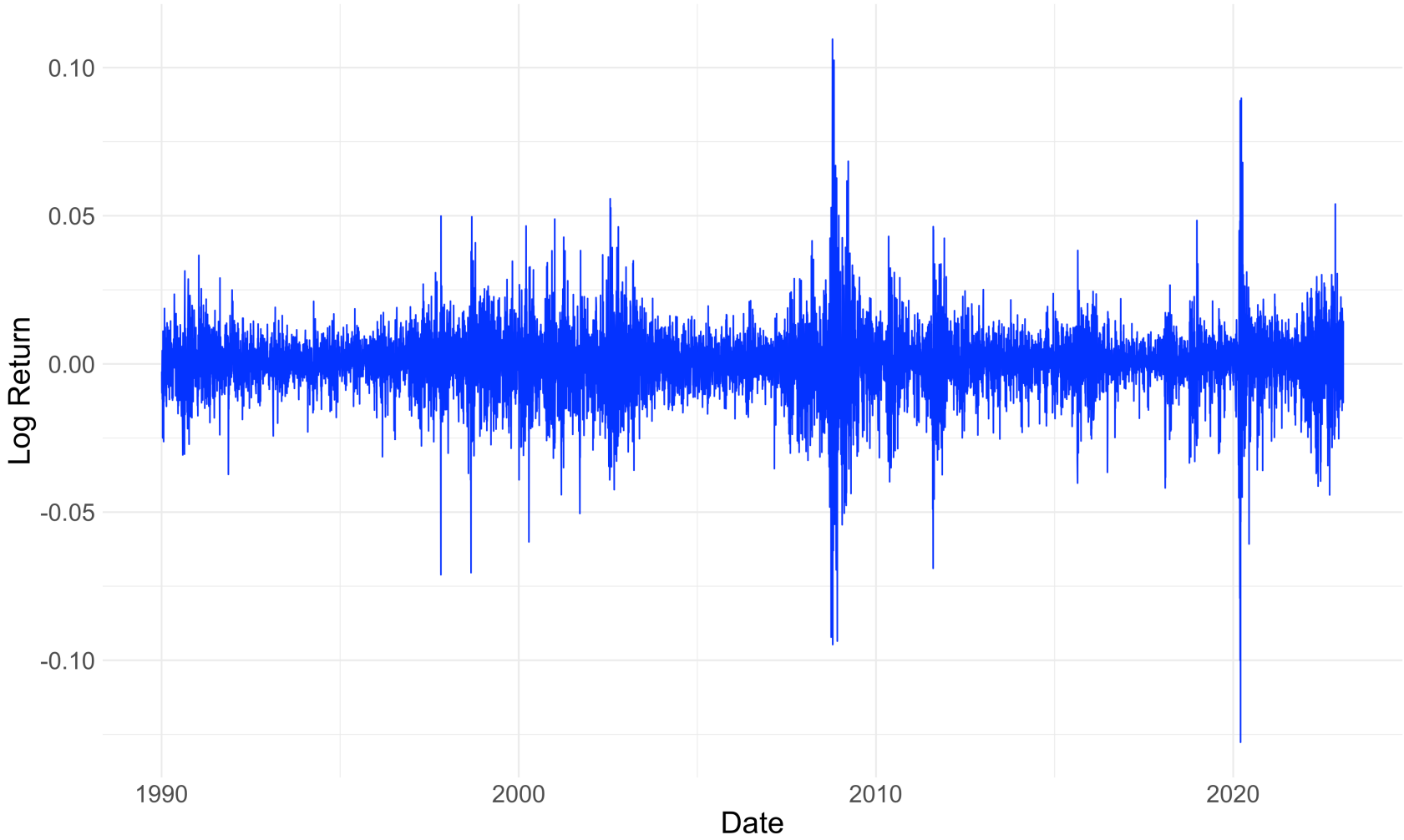
By Benjamin Wee

Supervised by Prof Catherine Forbes and Dr Lauren Kennedy

Research Outline and Literature Review

- Stochastic volatility model (Kim Shephard & Chib 1998)
- MCMC
  - Hamiltonian Monte Carlo, No-U-Turn Sampler (Hoffman and Gelman, 2014)
  - Kim Shephard & Chib (KSC) bespoke MCMC strategy
- Compare performance of MCMC
  - Simulation Based Calibration (Talts, Betancourt, Simpson, Vehtari, and Gelman 2020): check if MCMC algorithms are returning the correct posterior estimates on average
  - Efficiency: Effective sample size
- Model reparameterisation - improving MCMC performance (Strickland, Martin, and Forbes, 2008)
- KSC Importance Weights - correcting approximation error

Daily Log returns S&P 500



Stochastic Volatility model

KSC (1998) estimate a univariate discrete time SV model which models the variance as a latent stochastic process.

$$
\begin{aligned}
y_t &= e^{h_t/2}\epsilon_t \\
h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma_\eta \eta_t \\
h_1 &\sim \mathrm{N}\left(\mu, \frac{\sigma_\eta^2}{1-\phi^2}\right) \\
\epsilon_t &\sim \mathrm{N}(0,1) \quad \eta_t \sim \mathrm{N}(0,1)
\end{aligned}
$$

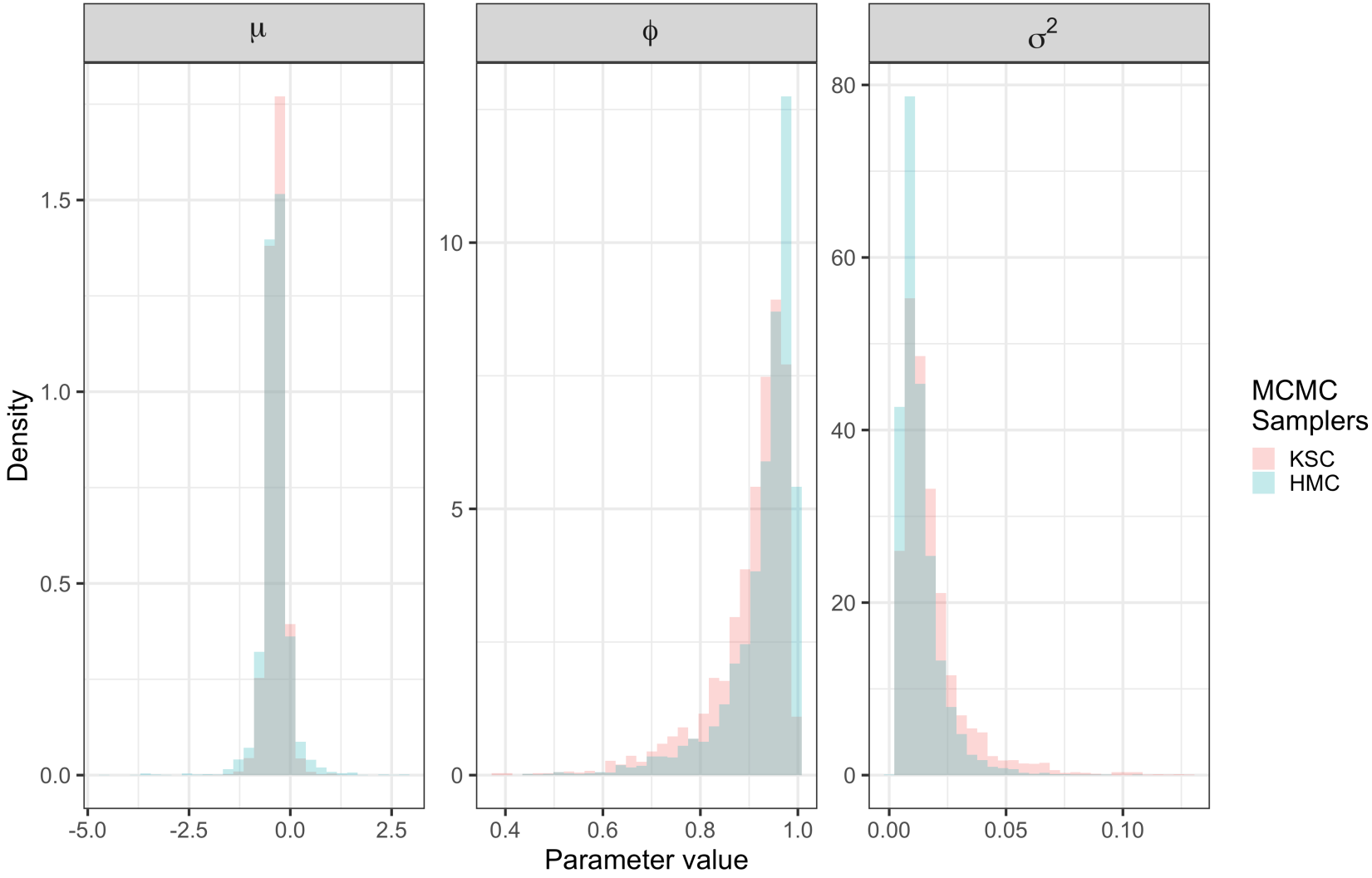
- State space model
- More parameters/unknowns then data points.
- Can estimate using Bayesian methods

Challenge 1: Comparing posteriors with no ground truth

Estimating the model on real data (S&P 500)

Challenge 1: Comparing posteriors with no ground truth

Estimating the model on real data (S&P 500)

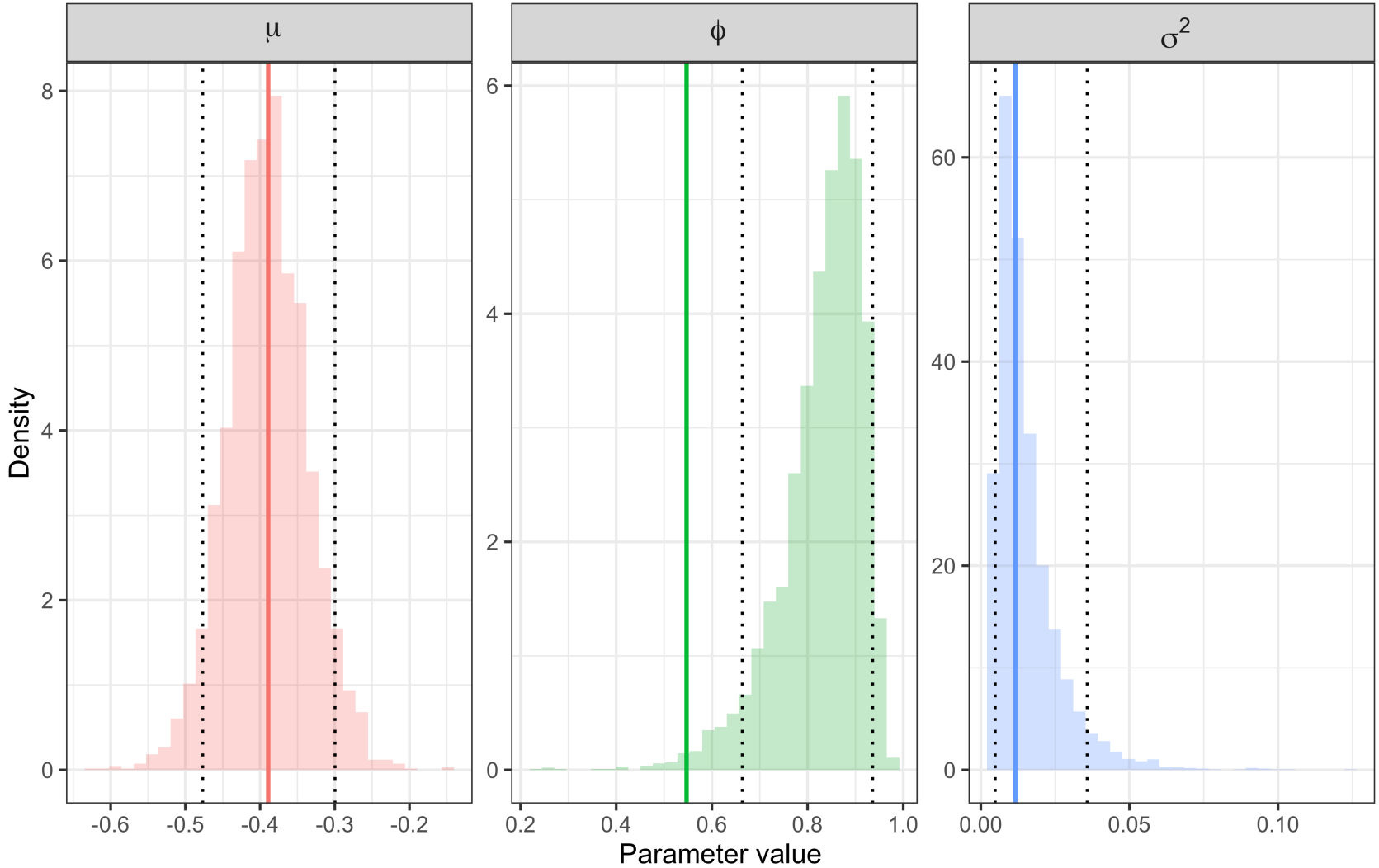


Challenge 2: Limitations of a single simulation

Estimate model on simulated data

Challenge 2: Limitations of a single simulation

Estimate model on simulated data



Research goal

- To validate the computation of different MCMC algorithms in the context of Bayesian stochastic volatility (SV) models
- Check whether our algorithm is returning the correct posteriors on average using simulation based calibration (Talts et al, 2020)
- Compare Hamiltonian Monte Carlo with KSC (1998)

Simulation Based Calibration

Simulation Based Calibration (SBC)

Let θ be a scalar parameter and y represent the dataset. Start with a draw from the prior distribution:

$$\theta^{sim} \sim \pi(\theta)$$

Simulation Based Calibration (SBC)

Let θ be a scalar parameter and y represent the dataset. Start with a draw from the prior distribution:

$$\theta^{sim} \sim \pi(\theta)$$

Generate a dataset given by the prior draws.

$$y^{sim} \sim \pi(y|\theta^{sim})$$

Simulation Based Calibration (SBC)

Let θ be a scalar parameter and y represent the dataset. Start with a draw from the prior distribution:

$$\theta^{sim} \sim \pi(\theta)$$

Generate a dataset given by the prior draws.

$$y^{sim} \sim \pi(y|\theta^{sim})$$

Then take draws from the posterior distribution generated by a MCMC algorithm or estimation strategy (HMC or KSC in our case).

$$\{\theta_1, ..., \theta_B\} \sim \pi(\theta|y^{sim})$$

Simulation Based Calibration (SBC)

The rank statistic for a given parameter and simulation is given by:

$$r = \text{rank}(\{\theta_1, \ldots, \theta_B\}, \theta^{sim}) = \sum_{b=1}^{B} 1[\theta_b < \theta^{sim}]$$

This completes one iteration of SBC.

Simulation Based Calibration (SBC)

The rank statistic for a given parameter and simulation is given by:

$$r = \text{rank}(\{\theta_1, \dots, \theta_B\}, \theta^{sim}) = \sum_{b=1}^{B} 1[\theta_b < \theta^{sim}]$$

This completes one iteration of SBC.

Talts et al. (2020) prove if the algorithm is calibrated (that is, returning correct posterior estimates on average), then the rank statistics is drawn from a uniform distribution.

Simulation Based Calibration (SBC)

The rank statistic for a given parameter and simulation is given by:

$$r = \text{rank}(\{\theta_1, \dots, \theta_B\}, \theta^{sim}) = \sum_{b=1}^{B} 1[\theta_b < \theta^{sim}]$$

This completes one iteration of SBC.

Talts et al. (2020) prove if the algorithm is calibrated (that is, returning correct posterior estimates on average), then the rank statistics is drawn from a uniform distribution.

Therefore, if we run multiple iterations of SBC, we should check whether the distribution of rank statistics is uniform.

Simulation Based Calibration (SBC)

Another key result for calibration is that the posterior averaged over the data and true parameters is equal to the prior distribution.

$$\pi(\theta) = \int \int \pi(\theta|y^{sim})\pi(y^{sim}, \theta^{sim}) d\theta^{sim} dy^{sim}$$

Simulation Based Calibration (SBC)

Another key result for calibration is that the posterior averaged over the data and true parameters is equal to the prior distribution.

$$\pi(\theta) = \int \int \pi(\theta|y^{sim})\pi(y^{sim}, \theta^{sim}) d\theta^{sim} dy^{sim}$$

Two key takeaways, if the MCMC calibrated then:

1) Rank statistics should be uniformly distributed
2) The average posterior distribution over SBC iterations should equal the prior distribution

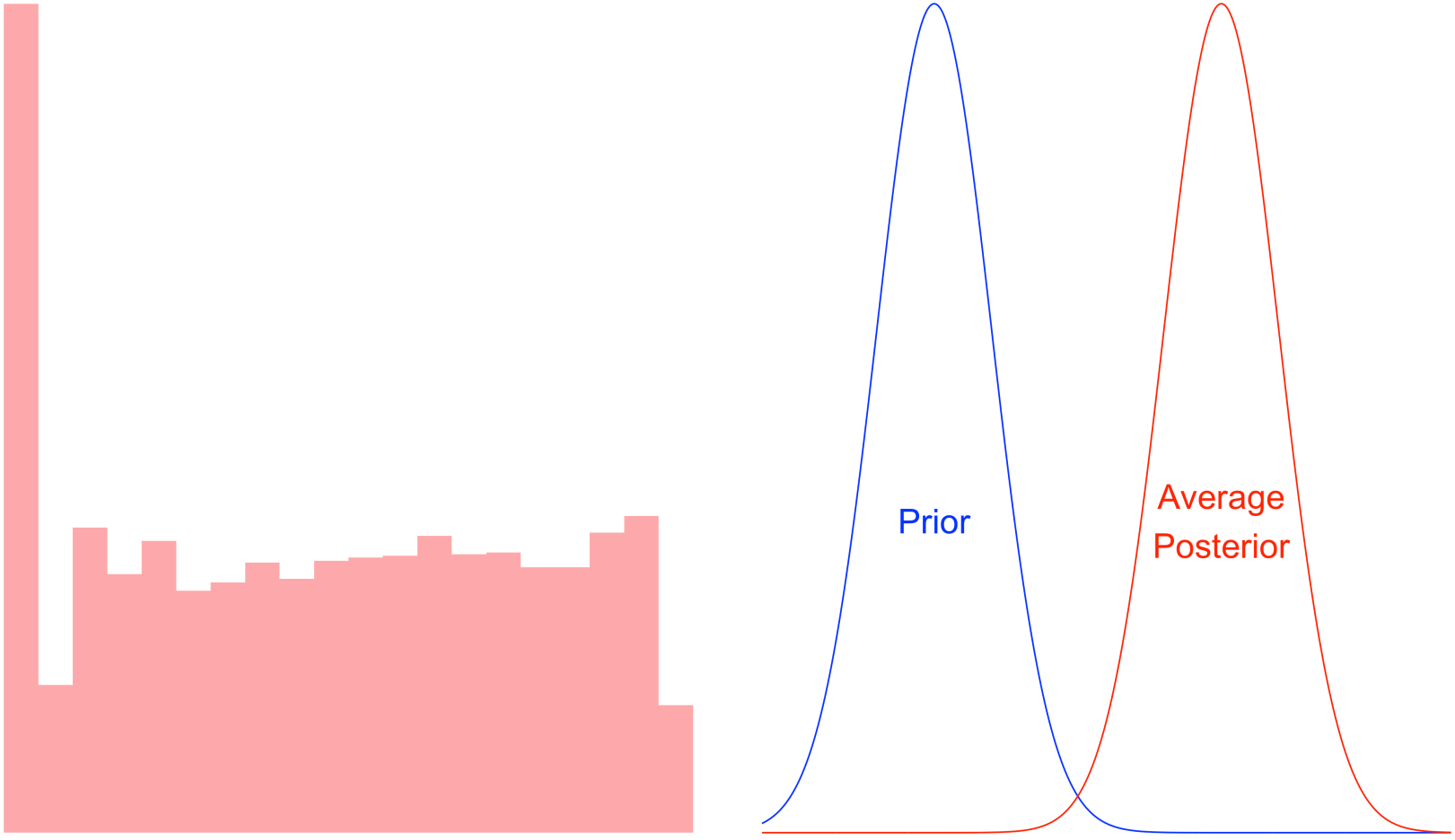
Interpretation of miscalibration

Interpretation of miscalibration

Reproduced from Talts et al. (2020)

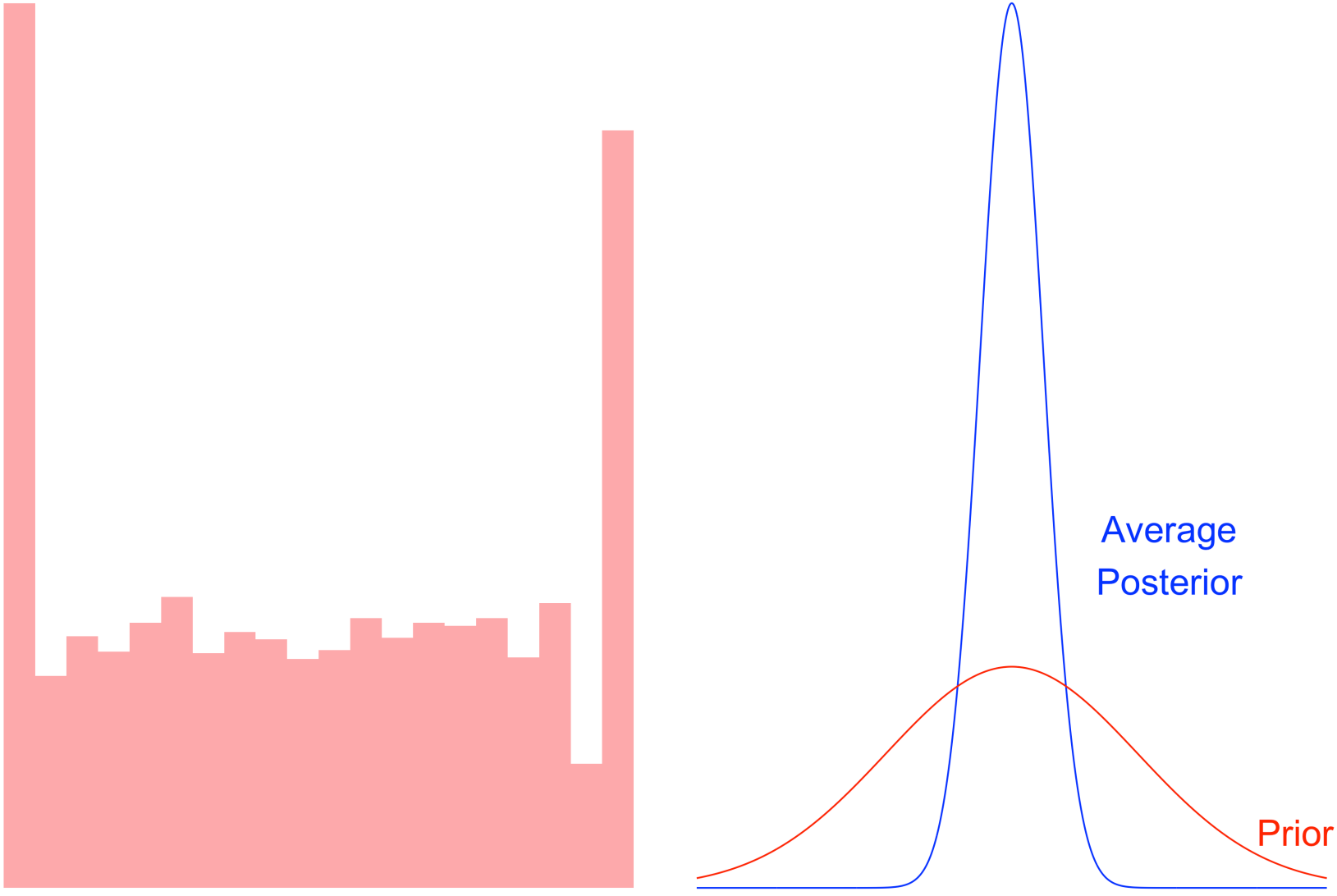
Interpretation of miscalibration

Reproduced from Talts et al. (2020)



Interpretation of miscalibration

Reproduced from Talts et al. (2020)



Markov Chain Monte Carlo (MCMC) algorithms

Stochastic Volatility model

KSC (1998) estimate a univariate discrete time SV model which models the variance as a latent stochastic process.

$$
\begin{aligned}
y_t &= e^{h_t/2}\epsilon_t \\
h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma_\eta \eta_t \\
h_1 &\sim \mathrm{N}\left(\mu, \frac{\sigma_\eta^2}{1-\phi^2}\right) \\
\epsilon_t &\sim \mathrm{N}(0,1) \quad \eta_t \sim \mathrm{N}(0,1)
\end{aligned}
$$

Stochastic Volatility model

KSC (1998) estimate a univariate discrete time SV model which models the variance as a latent stochastic process.

$$
\begin{aligned}
y_t &= e^{h_t/2}\epsilon_t \\
h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma_\eta \eta_t \\
h_1 &\sim \mathrm{N}\left(\mu, \frac{\sigma_\eta^2}{1-\phi^2}\right) \\
\epsilon_t &\sim \mathrm{N}(0,1) \quad \eta_t \sim \mathrm{N}(0,1)
\end{aligned}
$$

With priors:

$$
\begin{aligned}
\mu &\sim \mathrm{N}(0, 10) \\
\sigma_\eta^2 &\sim \mathrm{IG}(5/2, (0.01 \times 5)/2) \\
\phi^* &\sim \mathrm{Beta}(20, 1.5) \\
\phi &= 2\phi^* - 1.
\end{aligned}
$$

Kim Shephard Chib (1998)

- Estimate states using a standard Kalman filter, however, this requires the model to be linear and Gaussian.

Kim Shephard Chib (1998)

- Estimate states using a standard Kalman filter, however, this requires the model to be linear and Gaussian.

$$y_t = e^{h_t/2}\epsilon_t$$
$$y_t^* = log(y_t^2) = h_t + z_t$$

- $z_t = log(\epsilon_t^2)$ follows a log chi-squared distribution

Kim Shephard Chib (1998)

- Estimate states using a standard Kalman filter, however, this requires the model to be linear and Gaussian.

$$y_t = e^{h_t/2}\epsilon_t$$
$$y_t^* = log(y_t^2) = h_t + z_t$$

- $z_t = log(\epsilon_t^2)$ follows a log chi-squared distribution
- Approximate log chi squared error using Gaussian mixture model
  - Model is now linear in with respect to the states and conditionally Gaussian

Kim Shephard Chib (1998)

- Estimate states using a standard Kalman filter, however, this requires the model to be linear and Gaussian.

$$y_t = e^{h_t/2}\epsilon_t$$
$$y_t^* = log(y_t^2) = h_t + z_t$$

- $z_t = log(\epsilon_t^2)$ follows a log chi-squared distribution
- Approximate log chi squared error using Gaussian mixture model
  - Model is now linear in with respect to the states and conditionally Gaussian
- Kalman Filter to jointly sample states

Kim Shephard Chib (1998)

- Estimate states using a standard Kalman filter, however, this requires the model to be linear and Gaussian.

$$y_t = e^{h_t/2}\epsilon_t$$
$$y_t^* = log(y_t^2) = h_t + z_t$$

- $z_t = log(\epsilon_t^2)$ follows a log chi-squared distribution
- Approximate log chi squared error using Gaussian mixture model
  - Model is now linear in with respect to the states and conditionally Gaussian
- Kalman Filter to jointly sample states
- Sample μ and σ^2 directly from conjugate posterior dsitributions

Kim Shephard Chib (1998)

- Estimate states using a standard Kalman filter, however, this requires the model to be linear and Gaussian.

$$y_t = e^{h_t/2}\epsilon_t$$
$$y_t^* = log(y_t^2) = h_t + z_t$$

- $z_t = log(\epsilon_t^2)$ follows a log chi-squared distribution
- Approximate log chi squared error using Gaussian mixture model
  - Model is now linear in with respect to the states and conditionally Gaussian
- Kalman Filter to jointly sample states
- Sample μ and σ^2 directly from conjugate posterior dsitributions
- Metropolis Hastings to sample ϕ

Hamiltonian Monte Carlo (HMC)

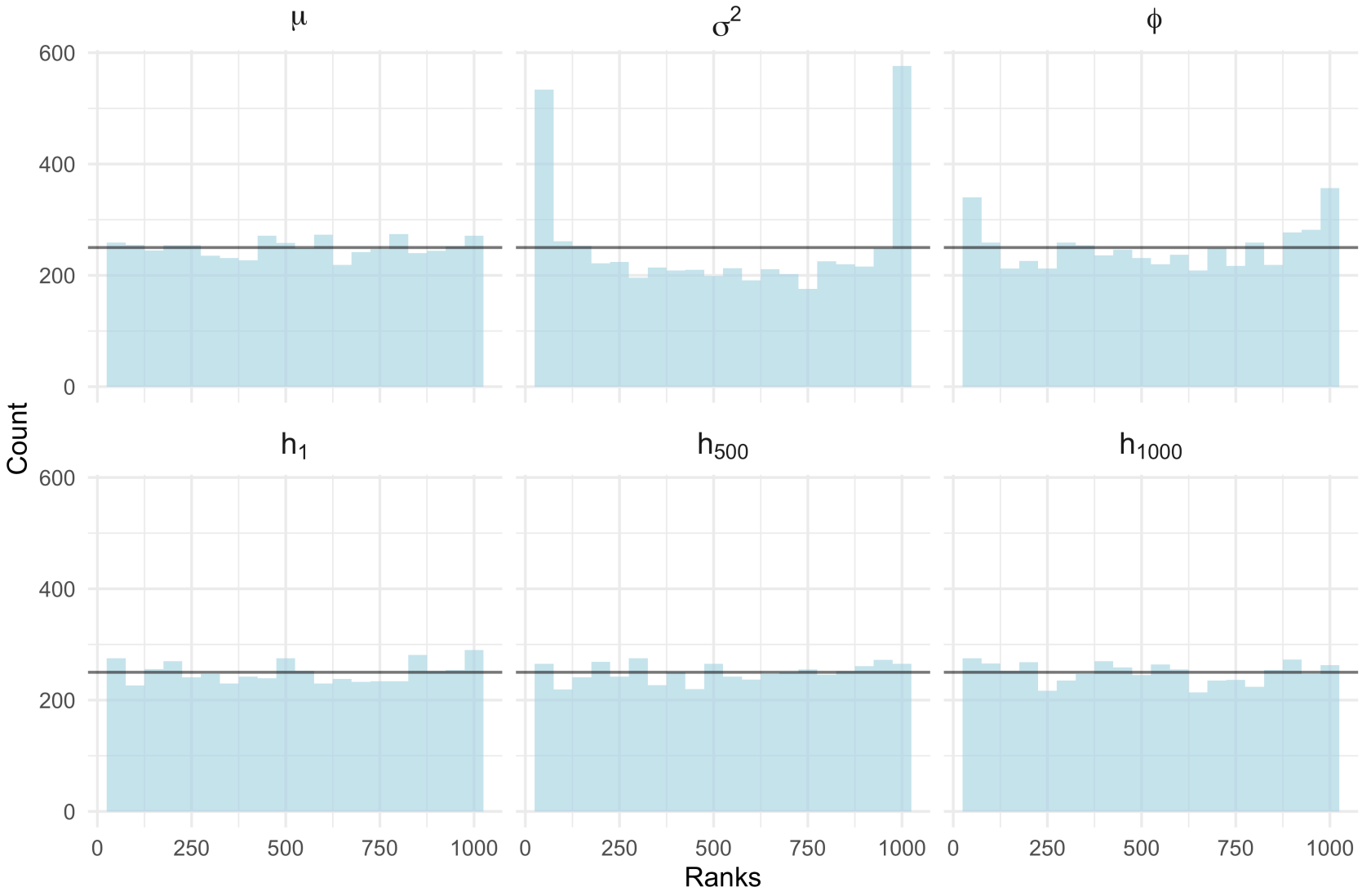
- The Stan programming language's implementation of Hamiltonian Monte Carlo, the No-U-Turn Sampler (Hoffman & Gelman 2014), will be compared with KSC's strategy.
- Key innovations of HMC:
  - Uses the gradients of the target posterior distribution to generate an efficient path for the sampler to explore.
  - Own programming language - can flexibly estimate wide range of models
  - Allows for direct sampling of the specified stochastic volatility model

Results

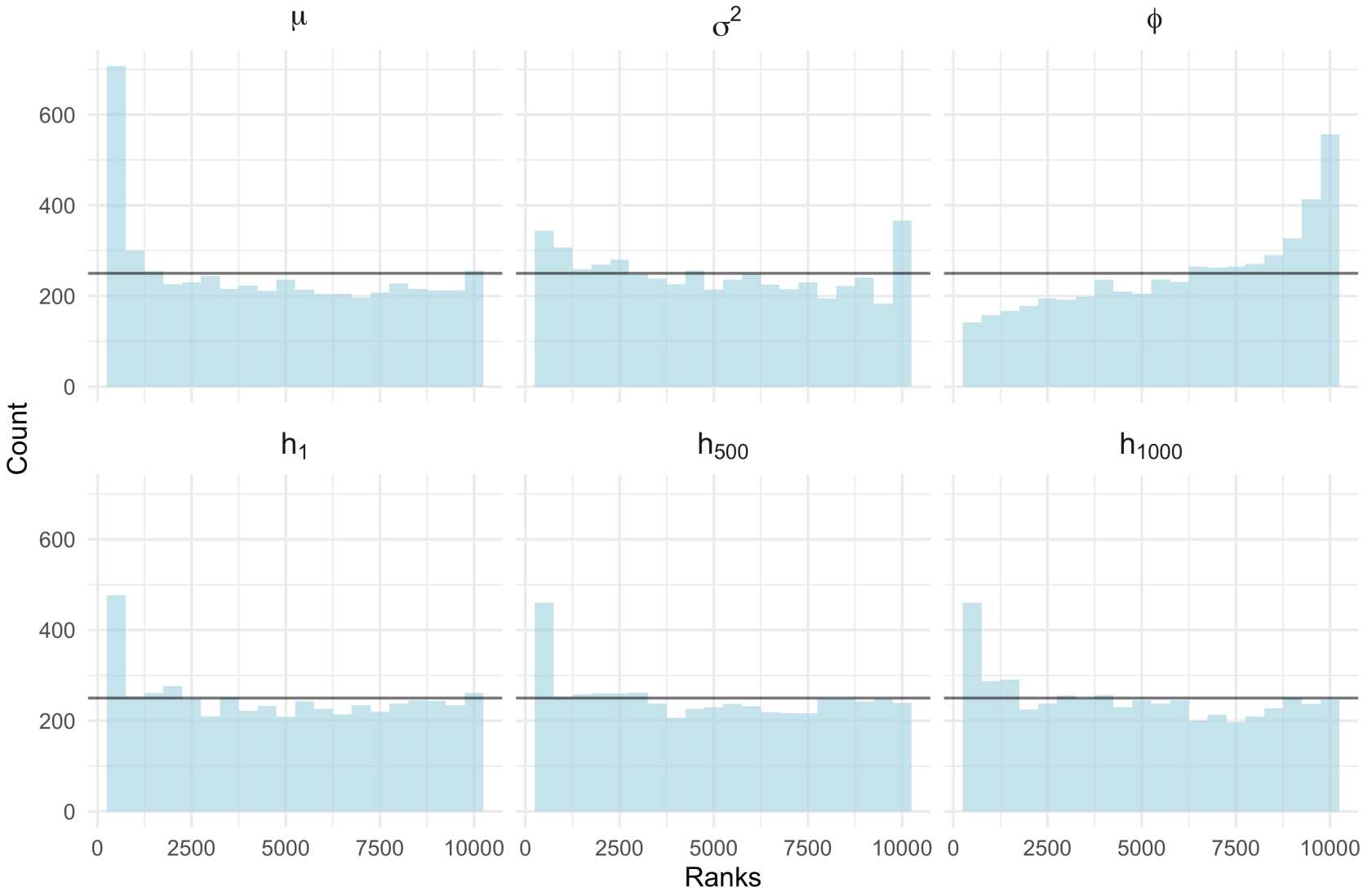
Results

- 5000 SBC iterations
- For each SBC iteration:
  - Simulate datasets of size T=1000
  - 999 post burn-in/warmup posterior draws for HMC
  - 9,999 post burn-in/warmup posterior draws for KSC

HMC (5000 Iterations)



KSC (5000 Iterations)



Model reparameterisation

Model reparameterisation

- Modify the model specification to (hopefully) improve MCMC performance
- Reparameterised model is mathematically equivalent to original model
    - Express the same model in a different way
- HMC: Sample states from standard normal distribution (then transform samples according to state equation)
- KSC: "Non centered parametersation" (Strickland, Martin, and Forbes, 2008)

Reparameterised HMC

First sample from a standard normal distribution and multiply by the variance of the log volatility. The state vector is sampled with mean centered on zero and variance equal to σ_η^2

$$
\begin{aligned}
h_{std} &\sim \mathrm{N}(0, 1) \\
h &= h_{std} \times \sigma_\eta
\end{aligned}
$$

Reparameterised HMC

First sample from a standard normal distribution and multiply by the variance of the log volatility. The state vector is sampled with mean centered on zero and variance equal to σ_η^2

$$\begin{aligned} h_{std} \sim & \mathrm{N}(0,1) \\ h = & h_{std} \times \sigma_\eta \end{aligned}$$

Then apply the appropriate re-scaling to get samples from log volatility.

$$\begin{aligned} h_1 = & \frac{h_{std,1} \times \sigma_\eta}{\sqrt{1-\phi^2}} + \mu \\ h_{t+1} = & h_{std,t+1} \times \sigma_\eta + \mu + \phi(h_t - \mu), t \neq 1 \end{aligned}$$

This returns log volatility as desired.

Reparameterised KSC

The following parameterisation is described as non centered in location. This follows the methodology outlined in Strickland, Martin, Forbes (2008).

Starting with the log chi squared model:

$$\begin{aligned} y_t^* &= h_t + z_t \\ h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma_\eta \eta_t \end{aligned}$$

Reparameterised KSC

The following parameterisation is described as non centered in location. This follows the methodology outlined in Strickland, Martin, Forbes (2008).

Starting with the log chi squared model:

$$\begin{aligned} y_t^* &= h_t + z_t \\ h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma_\eta \eta_t \end{aligned}$$

Let g be the demeaned state variable. Rewrite the demeaned state equation g_{t+1} to be non centered inlocation by subtracting average volatility μ.

$$\begin{aligned} g_t &= h_t - \mu \\ g_{t+1} &= \phi g_t + \sigma_\eta \eta_t \end{aligned}$$

Reparameterised KSC

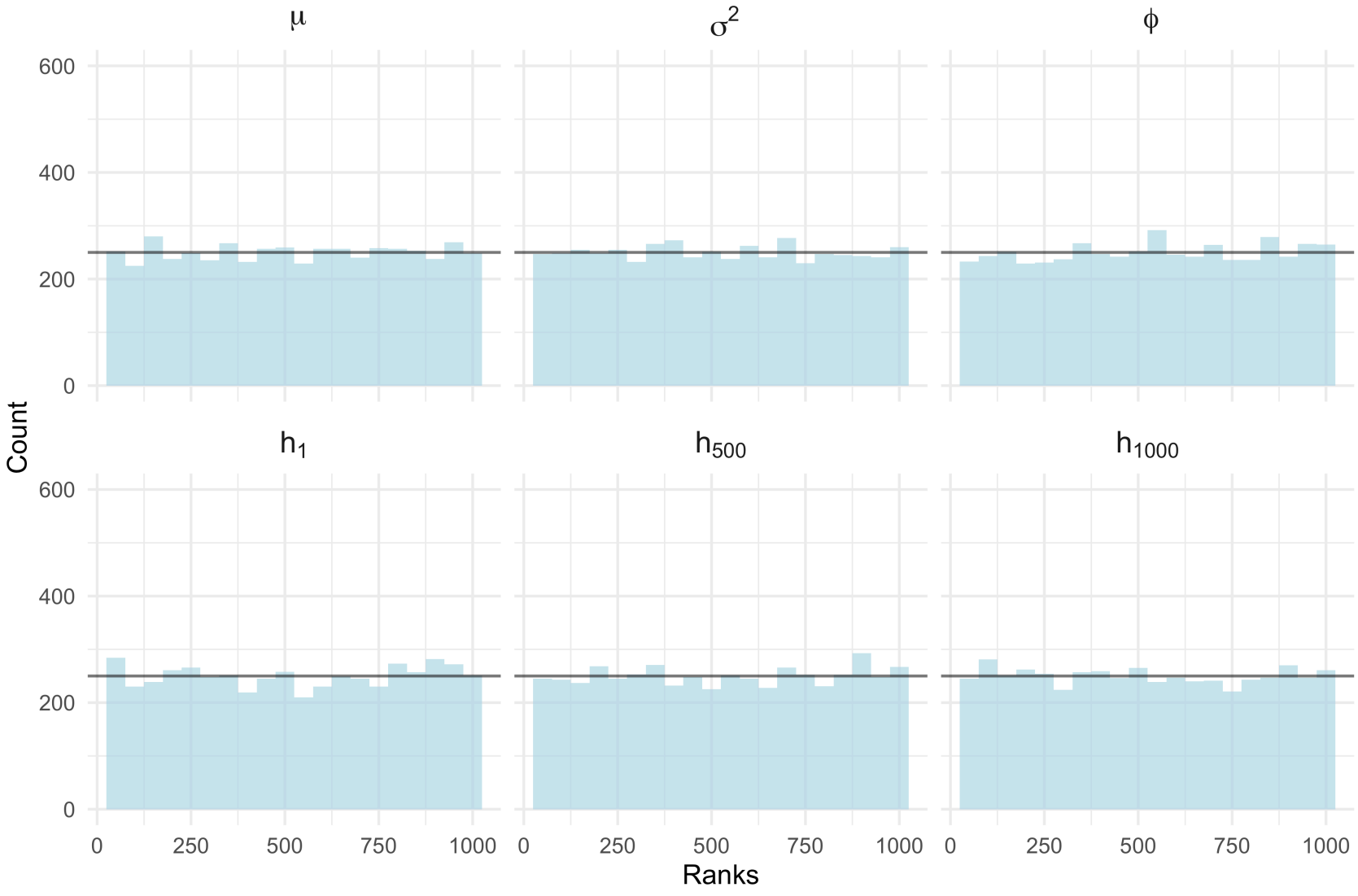
Return average volatility into measurement equation and rewrite as a function of the non centered state equation:

$$\begin{aligned} y_t^* &= g_t + \mu + z_t \\ g_{t+1} &= \phi g_t + \sigma_\eta \eta_t \end{aligned}$$

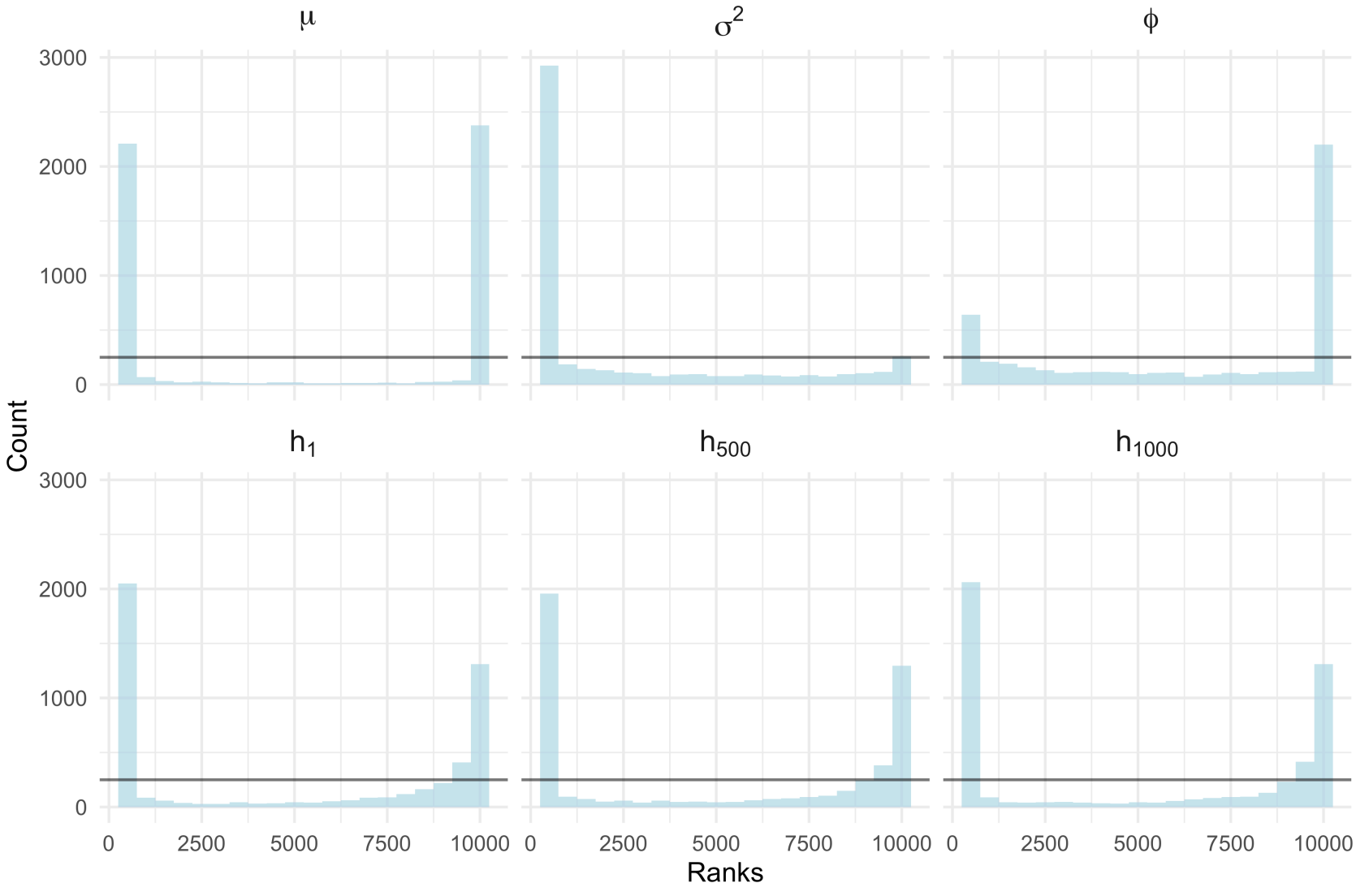
The mean of the log volatility is now inside the measurement equation and de-meaned from the state equation, where the initial state is drawn from

$$g_1 \sim \mathrm{N}\left(0, \frac{\sigma_\eta^2}{1-\phi^2}\right)$$

Reparameterised HMC (5000 Iterations)



Reparameterised KSC (5000 Iterations)



Conclusion

Limitations

- SBC is computationally intensive (required use of Monash HPC Cluster)
- Difficult to check histograms for many parameters
  - Can summarise rank statistic histograms into a single metric using chi-squared statistic - can show more details during Q&A
- Unclear what is considered a "fair" comparison

Conclusion

Limitations

- SBC is computationally intensive (required use of Monash HPC Cluster)
- Difficult to check histograms for many parameters
  - Can summarise rank statistic histograms into a single metric using chi-squared statistic - can show more details during Q&A
- Unclear what is considered a "fair" comparison

What do we learn

- Reparameterised model with HMC performs best based on calibration
- Not all reparameterisations improve MCMC performance
- SBC can help you understand that your algorithms are behaving the way they should

Questions

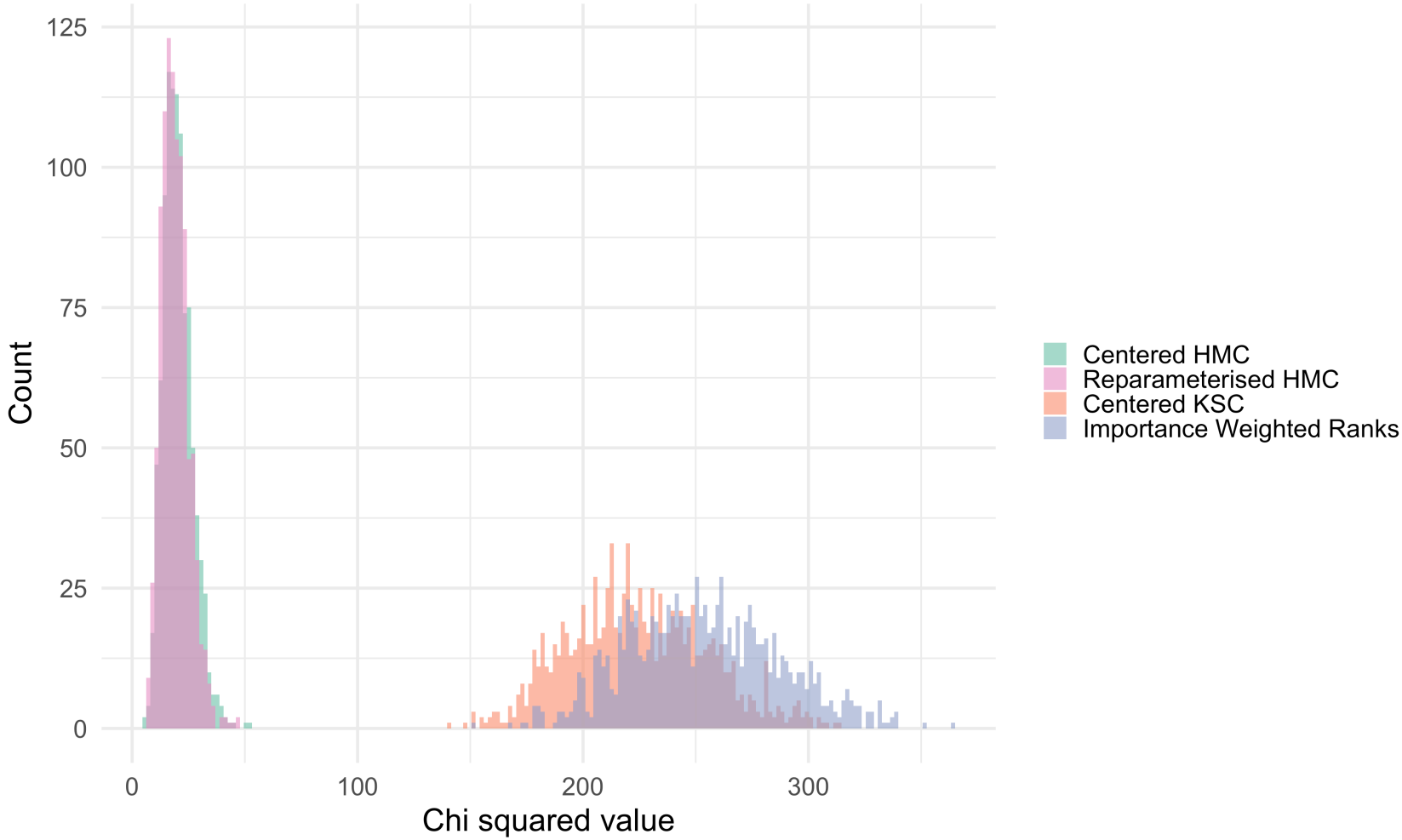
Chi squared statistic

Let b_j be the number of counts and e_j the expected count in bin j (where the expected count is a function of the number of bins and data points under a uniform distribution). Then the chi squared statistic is given by:

$$\chi^2 = \sum_{j=1}^{J} \frac{(b_j - e_j)^2}{e_j}$$

A perfectly uniform distribution will return a chi squared statistic of zero.

Summarising state variables



Kim Shephard Chib (1998)

Model is now linear but not Gaussian. KSC use a mixture of Gaussians to **approximate** the first 4 moments of the log chi squared distribution through moment matching. This is defined by:

$$f(z_t) = \sum_{i=1}^{K} q_i f_N(z_i | m_i - 1.2704, \nu_i^2)$$

K is the mixture of 7 normal densities f_N, component probabilities q_i, mean $m_i - 1.2704$ and variance ν_i^2.