DE GRUYTER Journal of Time Series Econometrics. 2018; 20170011

David Ardia[1] / Keven Bluteau[2,4] / Lennart F. Hoogerheide[3]

Methods for Computing Numerical Standard Errors: Review and Application to Value-at-Risk Estimation

[1] Institute of Financial Analysis, University of Neuchâtel, Neuchâtel, Switzerland; Department of Finance, Insurance and Real Estate, Laval University, Québec City, Canada; University of Neuchâtel, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland, E-mail: david.ardia@unine.ch
[2] Institute of Financial Analysis, University of Neuchâtel, Neuchâtel, Switzerland, E-mail: keven.bluteau@unine.ch
[3] Department of Econometrics and Tinbergen Institute, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, E-mail: l.f.hoogerheide@vu.nl
[4] Vrije Universiteit Brussel, Solvay Business School, Brussel, Belgium, E-mail: keven.bluteau@unine.ch

Abstract:
Numerical standard error (NSE) is an estimate of the standard deviation of a simulation result if the simulation experiment were to be repeated many times. We review standard methods for computing NSE and perform a Monte Carlo experiments to compare their performance in the case of high/extreme autocorrelation. In particular, we propose an application to risk management where we assess the precision of the value-at-risk measure when the underlying risk model is estimated by simulation-based methods. Overall, heteroscedasticity and autocorrelation estimators with prewhitening perform best in the presence of large/extreme autocorrelation.

Keywords: bootstrap, GARCH, HAC kernel, numerical standard error (NSE), Monte Carlo, Markov chain Monte Carlo (MCMC), spectral density, value-at-risk, Welch

DOI: 10.1515/jtse-2017-0011

1 Introduction

The numerical standard error (NSE) is an estimate of the standard deviation of a simulation result, if the simulation experiment were to be repeated many times. In this note, we review existing methods for NSE estimation and compare their accuracy in a Monte Carlo study.

We consider the standard situation in which the expectation $\mu \equiv \mathbb{E}[g(X)]$ of a certain scalar function $g(X)$ is estimated by generating pseudo–random draws X_i $(i=1,2,\ldots,n)$ and computing the sample average $\hat{\mu} \equiv \sum_{i=1}^{n} g(X_i)$. If the X_i $(i=1,2,\ldots,n)$ would be independent, then the NSE would simply be obtained as the sample standard deviation of $g(X)$ divided by $\sqrt{n}$. However, many time-series models or simulation methods, such as Markov chain Monte Carlo (MCMC) popular in Bayesian statistics, exhibit (high) autocorrelation. Hence the usual *naive* method overstates the precision of the estimate. To overcome the problem, we could obviously repeat the simulation experiment many times and compute the standard deviation of the simulation result. However, the computing time could be enormous. Therefore, a reliable method is required that provides an accurate NSE based upon the serially correlated draws from a single time series or a single run of a simulation method.

Several approaches for computing the NSE have been proposed in the literature over the last decades. In this note, we review the most common methods to give an overview to non-experts. We then perform a Monte Carlo study to compare their performance in the context of high/extreme autocorrelation. First, we consider time-series models. Second, we propose an application to risk management where we assess the precision of the value-at-risk (VaR) measure when the underlying risk model is estimated via MCMC techniques.

Overall, our simulations studies indicate to rely on heteroscedasticity and autocorrelation (HAC)-type estimators with prewhitening when dealing with time-series or MCMC estimation outputs exhibiting large/extreme autocorrelation.

The contents are organized as follows. In Section 2, we review existing methods for computing the NSE. In Section 3, we present the setup of the Monte Carlo studies. Section 4 discusses the results and concludes.

David Ardia is the corresponding author.

2 Numerical Standard Error Estimators

In this section, we describe the methods used to compute the NSE. Section 2.1 describes batch means estimators, Section 2.2 outlines initial sequence estimators, and Section 2.3 discusses spectrum–at–zero estimators. Section 2.5 discusses bootstrap estimators. Overall, we consider 33 methods summarized in Table 1.

Table 1: NSE methods investigated in our study
References in the table are: FJ10 (Flegal and Jones 2010), G92 (Geyer 1992), HW91 (Heidelberger and Welch 1981), NW87 (Newey and West 1987), NW94 (Newey and West 1994), A91 (Andrews 1991), AM92 (Andrews and Monahan 1992), H10 (Hirukawa 2010), PC06 (Percival and Constantine 2006), PR92 (Politis and Romano 1992), PR94 (Politis and Romano 1994), PW04 (Politis and White 2004) and W67 (Welch 1967). All methods are available in the `R` package **nse** (Ardia and Bluteau 2017).

Acronym	Description	Reference
`Nv`	*Naive* empirical variance	
`Bm`	Batch means	G92
`BmO`	Overlapping bach means	G92
`IsPo`	Initial sequence, nonnegative constraint	G92
`IsDe`	Initial sequence, nonnegative, nonincreasing constraints	G92
`IsCo`	Initial sequence, nonnegative, nonincreasing,convex constraints	G92
`IsPoBm`	`IsPo` applied to batch means	G92
`IsDeBm`	`IsDe` applied to batch means	G92
`IsCoBm`	`IsCo` applied to batch means	G92
`SdAr`	Parametric spectral density at zero, AR(q) fit	HW91
`SdGl`	Parametric spectral density at zero, GLM fit	HW91
`SdBa`	Nonparametric spectral density at zero, Bartlett kernel	FJ10
`SdPa`	Nonparametric spectral density at zero, Parzen kernel	FJ10
`SdQs`	Nonparametric spectral density at zero, Quadratic Spectral kernel	FJ10
`SdBaWe`	`SdBa` with Welch's smoothing	W67/PC06
`SdPaWe`	`SdPa` with Welch's smoothing	W67/PC06
`SdQsWe`	`SdQs` with Welch's smoothing	W67/PC06
`Nw`	Newey West	NW87
`NwPr`	`Nw` with prewhitening	NW94
`AnBa`	Andrews bandwidth, Bartlett kernel	A91
`AnBaPr`	`AnBa` with prewhitening	A91+AM92
`AnPa`	Andrews bandwidth, Parzen kernel	A91
`AnPaPr`	`AnPa` with prewhitening	A91+AM92
`AnQs`	Andrews bandwidth, Quadratic Spectral kernel	A91
`AnQsPr`	`AnQs` with prewhitening	A91+AM92
`HiBa`	Hirukawa bandwidth, Bartlett kernel	H10
`HiBaPr`	`HiBa` with prewhitening	H10+AM92
`HiPa`	Hirukawa bandwidth, Parzen kernel	H10
`HiPaPr`	`HiPa` with prewhitening	H10+AM92
`BsSt`	Stationary bootstrap, automatic block length	PR94+PW04
`BsCi`	Circular bootstrap, automatic block length	PR92+PW04
`BsStFx`	Stationary bootstrap, ad-hoc block length	PR94
`BsCiFx`	Circular bootstrap, ad-hoc block length	PR92

2.1 Batch Means Estimators

The first type of NSE is called (overlapping) batch means estimator (see, e.g., Geyer 1992, Section 3.2). The batches are simply subsequences of consecutive iterates of the time series. Let m be a fixed small integer, where n is a multiple of m. Divide the time series into m batches of equal size. Then, the batch means are given by:

$$Z_{n,k} \equiv \frac{m}{n} \sum_{i=(k-1)n/m+1}^{kn/m} g(X_i)\,, \quad \text{for } k = 1, \ldots, m\,.$$

If the batch size is large enough, the batch means should be approximately uncorrelated and NSE can be computed as the sample standard deviation of the $Z_{n,k}$ divided by $\sqrt{m}$. If the length of batches is too short (in case of a small sample and/or huge serial correlation), then the initial sequence estimator (see Section 2.2) can be applied to the series of batch means $Z_{n,k}$. Typically, at least 20 batches are used to obtain a reliable NSE. In this note, we use 30 batches (`Bm`). We also consider overlapping batch means (`BmO`).

2.2 Initial Sequence Estimators

Let γ_k be the kth order autocovariance of $g(X_i)$ $(i=1,\ldots,n)$ and $\Gamma_m \equiv \gamma_{2m} + \gamma_{2m+1}$ be the sums of adjacent pairs of autocovariances. Geyer (1992), Theorem 3.1 shows that if $g(X_i)$ stems from a stationary, irreducible, and reversible Markov chain, then Γ_m is a strictly positive, strictly decreasing, and strictly convex function of m. Using this property of the autocovariances, Geyer (1992), Equation 3.3 derives the following estimator computed from sample autocovariances $\widehat{\gamma}_k$ and $\widehat{\Gamma}_m \equiv \widehat{\gamma}_{2m} + \widehat{\gamma}_{2m+1}$:

$$\widehat{\mathbb{V}}[\widehat{\mu}] \equiv \frac{-\widehat{\gamma}_0 + 2\sum_{m=0}^{p}\widehat{\Gamma}_m}{n},$$

where the summation runs to p such that $\widehat{\Gamma}_0, \widehat{\Gamma}_1, \ldots, \widehat{\Gamma}_p$ form the longest series that is (i) non-negative; (ii) non-negative and nonincreasing; or (iii) non-negative, nonincreasing, and convex. In this note, we consider the three estimators (`IsPo`, `IsDe`, `IsCo`) together with initial sequence estimators applied to batch means series (`IsPoBm`, `IsDeBm`, `IsCoBm`).

2.3 Spectrum-at-Zero Estimators

Tools from spectral analysis can be used to estimate the autocovariance function and from that, the variance. Suppose γ_k is the autocovariance function of a stationary process and that $f(\omega)$ is the spectral density for the same process where k is the time lag and ω is the frequency. If a time series $g(X_i)$ has autocovariance satisfying $\sum_{k=-\infty}^{\infty}|\gamma_k| < \infty$, the spectral density and the autocovariance function are Fourier transform pairs. Hence, $\gamma_k = \int_{-1/2}^{1/2}\exp(2\pi i\omega k)f(\omega)d\omega$ for $-\infty < k < \infty$ and $f(\omega) = \sum_{k=-\infty}^{\infty}\gamma_k\exp(-2\pi i\omega k)$ from which we can estimate the variance by $\gamma_0 = \int_{-1/2}^{1/2}f(\omega)d\omega$. In this note, we consider parametric and nonparametric approaches for estimating the spectral density.

On the parametric side, we use the spectral density estimator derived from a fitted AR(q) model (`SdAr`). We also use the method described in Heidelberger and Welch (1981), where the spectral density at frequency zero is estimated by fitting a generalized linear model (GLM) to the low-frequency end of the periodogram (`SdGl`).

For nonparametric approaches, we estimate the entire spectrum using the cross-validation methods described in Hurvich (1985) and average in the frequency domain using a Bartlett (`SdBa`), Parzen (`SdPa`) and quadratic spectral (`SdTu`) window; see Section 2.4.

Welch

A way to improve spectrum-at-zero estimators is to reduce the noise in the periodogram using the Welch's overlapped segment approach (Welch 1967) described in Percival and Constantine (2006). It consists of splitting the time series into overlapping segments, calculating the spectral density estimator on each of the segments, and then taking the average of the spectral densities as the final estimator. In our case, we use a 50% overlap and eight segments (which is the setup recommended by Welch (1967)). We apply this smoothing to the three spectrum-at-zero estimators (and denote them by `SdBaWe`, `SdPaWe` and `SdTuWe`, respectively).

2.4 HAC Kernel Estimators

We can think of estimating the mean of serially correlated series $g(X_i)$ $(i=1,2,\ldots,n)$ as fitting a linear regression model $y = \beta_0 + \epsilon$ with $y = g(X)$ where the sample mean $\widehat{\mu} = \widehat{\beta}_0$ is our ordinary least squares (OLS) estimate of the mean $\mu = \mathbb{E}[g(X)] = \beta_0$. Since there is autocorrelation, the OLS standard error is not valid (in the sense that it is inconsistent). Newey and West (1987) proposed a standard error that is consistent in case of HAC:

$$\widehat{\mathbb{V}}[\widehat{\mu}] \equiv \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{|i-j|}(g(X_i)-\widehat{\mu})(g(X_j)-\widehat{\mu})}{n^2},$$

where the weights $w_{|i-j|}$ can be specified in several ways, depending on the so-called bandwidth and kernel. The bandwidth h indicates for which large values of $|i-j|$ the weights $w_{|i-j|}$ are nonzero, while the kernel determines the way observations are aggregated. Although Newey and West (1987) determined the range of bandwidth choices that make the covariance matrix estimator consistent and positive semidefinite, they did not formulate a guideline for an appropriate bandwidth choice in finite samples, nor did they mention which of the kernels to use.

Note that, as pointed out in Andrews (1991), Newey–West (type of) estimator corresponds to the spectral density evaluated at zero and depends on the class of spectral density estimator considered by Parzen (1957).

Kernel. Let us define $x \equiv \frac{|i-j|}{h+1}$. In this note, we use the following kernels:

$$\text{Bartlett:} \quad w_x \equiv \begin{cases} 1-x & \text{if } x \leq 1 \\ 0 & \text{otherwise}, \end{cases}$$

$$\text{Parzen:} \quad w_x \equiv \begin{cases} 1-6x^2+6x^3 & \text{if } \quad 0 \leq x \leq \frac{1}{2} \\ 2(1-x)^3 & \text{if } \quad \frac{1}{2} < x \leq 1 \\ 0 & \text{otherwise}, \end{cases}$$

$$\text{Quadratic spectral:} \quad w_x \equiv \frac{25}{12\pi^2 x^2}\left(\frac{\sin(6\pi x/5)}{6\pi x/5}\right) - \cos(6\pi x/5)\,.$$

Bandwidth. Andrews (1991) formulates for each kernel its optimal finite sample bandwidth value which depends on the true distribution of the data generating process (DGP) which is unknown. He comes up with a plug-in type of estimator: the automatic bandwidth estimator. This automatic bandwidth depends on a model which has to be estimated to determine the optimal bandwidth before the kernel estimation can be performed. This ensures that the bandwidth procedure takes the level of autocorrelation into account.

Newey and West (1994) came up later with a more convenient way to calculate the optimal bandwidth for the Bartlett, Parzen, and Quadratic spectral kernels. They compare via a Monte Carlo study their bandwidth to those from Andrews (1991) and the results support their procedure and highlight the importance of bandwidth selection.

More recently, Hirukawa (2010) proposes an alternative approach that estimates the unknown quantity in the optimal bandwidth for the HAC estimator (referred to as normalized curvature) using a general class of kernels and derives the optimal bandwidth that minimizes the asymptotic means squared error of the estimator of normalized curvature. The normalized curvature was estimated in Andrews (1991) by a simple AR(1) model and Newey and West (1994) estimate it nonparametrically using the truncated kernel. The procedure used in Hirukawa (2010) is called the solve-the-equation plug-in (SP) rule. The SP rule and the implementation are done for the Bartlett and Parzen kernels. Monte Carlo results indicate that for a variety of DGPs, the HAC estimator based on the SP rule can estimate the variance more accurately than the quadratic spectral estimator of Andrews (1991) or the Bartlett estimator of Newey and West (1994). Details on the implementation are presented in Hirukawa (2010), Section 2.2.

Prewhitening. Even though many HAC estimators have been developed to obtain better overall variance estimates, performance was still lacking, especially when there was high autocorrelation. This has led to the introduction of prewhitening for HAC estimator by Andrews and Monahan (1992) which was inspired by prewhitening methods used in spectral density estimation (see, e.g., Press and Tukey 1956). The prewhitening procedure aims at filtering (part of) the autocorrelation in the data before applying a kernel estimator. In the univariate case, OLS estimation of an AR(q) model is first conducted on the data. Second, a kernel estimator is applied to the residuals. Third, the estimated variance is transformed to the variance of the original sample mean by making use of the unconditional variance in an AR(q) model. In this note, we test several HAC and spectral estimators, for various bandwidth kernels and with/without AR(q) prewhitening, where the lag q is selected using the AIC criterion (Akaike 1974).

2.5 Bootstrap Estimators

An alternative is to make use of a bootstrap method where we generate blocks of consecutive draws from the actual series $g(X_i)$ $(i=1,2,\ldots,n)$ to form multiple new series. The NSE is then the standard deviation of the results from the different generated time series. If we simulate the block length from a geometric distribution with mean b, then we have the stationary bootstrap of Politis and Romano (1994). Using a fixed (non-random) block length b results in the circular bootstrap of Politis and Romano (1992).

The block length, like the bandwidth size in the kernel method, plays an important role in the accuracy of the resulting estimator. Politis and White (2004) develop an automatic block length selection procedure for the stationary and circular bootstrap which depends on the smallest lag where the autocorrelation appears negligible. Nordman (2009) discovered an error in the calculation of Lahiri (1999) of the variance associated with the stationary bootstrap of Politis and White (2004). Since the theoretical results of Politis and White (2004) are built on Lahiri (1999) calculations, a correction is given in Patton, Politis, and White (2009). In this note, we consider the stationary bootstrap with automatic block length (`BsSt`), *ad-hoc* block length (`BsStFx`), as well as the circular bootstrap with automatic block length (`BsCi`) and *ad-hoc* block length (`BsCiFx`). The *ad-hoc* block length is set to the closest integer of $3.15 \times n^{1/3}$.

3 Experiment Design and Estimation Methods

Section 3.1 presents the two experiments on simulated time series. Section 3.2 presents the experiment on the VaR measure.

3.1 Serially Correlated Observations from Time-Series Models

In the time-series experiments, 1,000 datasets of length n ($n = 100$ and $n = 1{,}000$) are simulated from a model. For each dataset, the estimation of the NSE is compared with the *true* NSE. We then compare the accuracy of the NSE to the *true* standard deviation by calculating the root mean squared error (RMSE) and the bias of the estimates.[1]

3.1.1 AR(1) Model

The first experiment consists of simulating data from the AR(1) model:

$$y_t = \alpha + \rho y_{t-1} + \epsilon_t \,, \quad \epsilon_t \sim iid\,\mathcal{N}(0,1)\,, \tag{1}$$

for $t = 2,\ldots,n$, $y_1 \sim \mathcal{N}\left(0, \frac{1}{1-\rho^2}\right)$ and $\alpha = 0$. We have $\mathbb{E}[y_t] = 0$, $\mathbb{V}[y_t] = \frac{1}{1-\rho^2}$ and $\mathbb{C}or[y_t, y_{t-1}] = \rho$. Two different values are used for the autocorrelations: high autocorrelation ($\rho = 0.9$) and extreme autocorrelation ($\rho = 0.99$). Under process eq. (1), the *true* variance of the sample mean (i.e. squared NSE) is known in closed form and is given by:

$$\mathbb{V}\left[\frac{\sum_{t=1}^{n} y_t}{n}\right] = \frac{1}{(1-\rho^2)n^2}\left(n + \sum_{i=1}^{n-1} 2(n-i)\rho^i\right),$$

for $|\rho| < 1$.

3.1.2 Regime-Switching Model

The second experiment consists in simulating data from a regime-switching model with two regimes of normally distributed observations: one regime where $y_t \sim \mathcal{N}(5,1)$, one regime where $y_t \sim \mathcal{N}(-5,1)$, and a probability p of staying in the same regime (so that the probability of switching is $1-p$). We change p to create datasets with high ($p = 0.9$) and extreme ($p = 0.99$) autocorrelations. The first observation has a probability of one half to be in either one of the regimes, which implies that $\mathbb{E}[y_t] = 0$. Under this process, the *true* variance of the sample mean of the regime-switching model is not known in closed form but can be obtained by simulation (i.e. by computing the standard deviation of the averages over 10^5 Monte Carlo replications).

3.2 Precision in Value-at-Risk Prediction

We also investigate the performance of NSE estimators when assessing the numerical precision of financial risk measures obtained with sophisticated econometric models. More specifically, we consider the one-day ahead VaR measure, which is nothing else than a quantile of the distribution of financial log-returns over a one-day horizon. The estimation of this measure is crucial for asset and risk managers, as it focuses on the left tail of their profit and loss (P&L) distribution, and therefore provides an assessment of the financial risk they face (Hoogerheide and van Dijk 2010).

Recent advances in econometrics models applied to risk management suggest the use of flexible Markov-switching GARCH models (Haas, Mittnik, and Paollela 2004), as these are able to capture several stylized facts observed in financial time series (i.e. volatility clustering, leverage effect, conditional skewness and kurtosis). For these models, simulation-based MCMC techniques should be preferred for their estimation (see, e.g., Ardia 2008). In this context, precision in the estimation of the risk measures is an important aspect for risk management. Indeed, if changes of risk measures over time are merely caused by simulation noise (from the MCMC estimation), this leads to useless fluctuations in financial positions, leading to extra costs (e.g. transactions costs). In particular, MCMC schemes used for the Bayesian estimation of GARCH models yield a set of posterior draws that are serially correlated.[2] Hence, the possibly high persistence and complex structure of the series of draws can make it difficult to accurately assess the precision of simulation results used in risk management, such as the VaR.

To assess the performance of the various NSE estimators in a risk management context, we design the following experiment. Based on a time series of S&P index log-returns, we estimate a Markov-switching GARCH(1,1) model with normal disturbances. The S&P 500 data range from January 2011 to December 2014 for a total of 1,005 observations. The econometric model is estimated 1,000 times (Monte Carlo replications). For each replication, we rely on the simulation sampler proposed by Vihola (2012), which guarantees an automatic tuning of the MCMC estimation strategy.[3] The sampler is used to generate a MCMC sample of length n ($n = 100$ and $n = 1{,}000$), after a burn-in of 2,500 draws. For each of the n posterior draws in each chain, we calculate the one–step ahead VaR95, that is, the 5th percentile of the P&L distribution. This yields an autocorrelated sample of n VaR estimates from which we compute the average. We can then compare the accuracy of the NSE to the *true* standard deviation, by calculating the RMSE and the bias of the estimate. The *true* standard deviation of the average VaR is obtained from a large number of independent MCMC runs (in our case 10^5 runs).[4]

4 Discussion of Results

Table 2 and Table 3 report results for the time-series models and Table 4 reports the results for the precision of the VaR measure obtained by MCMC. In bold, we highlight the five best estimators per setup and per performance measure (RMSE and bias). From these results, we draw the following conclusions. First, as expected, the naive estimator is the worst in all experiments. Second, the batch means estimator does not perform well, and performance is sometimes even worse when combined with the initial sequence estimator. Third, it is difficult to choose between the initial sequence estimators, the spectral density estimator, and the HAC kernel estimators (without prewhitening). The differences in performance can be small, and sometimes different methods perform better. Fourth, prewhitening makes the HAC estimators substantially better in each experiment. They are systematically outperforming the other estimators in terms of RMSE and bias. Fifth, the considered bootstrap methods are never best, because they are always beaten by the initial sequence estimators and/or the spectral density estimators and/or the HAC-type estimators. The extra computing time required for the bootstrap methods does not payoff. Sixth, among the spectral density estimators, the parametric AR method performs better than the alternative methods in each experiment. Seventh, Welch's overlapping approach worsens the results in the cases of extreme autocorrelation.

Table 2: Monte Carlo results for the AR model.
Root mean squared error (RMSE) and bias of the various NSE estimators in the case of n simulated data from the AR(1) model. The number of Monte Carlo replications is set to 1,000. In bold, we report the five best performers per column. See Table 1 for the description of the methods and Section 3.1 for the description of the Monte Carlo study.

	RMSE (× 10)				**Bias (× 10)**			
	$\rho = 0.90$		$\rho = 0.99$		$\rho = 0.90$		$\rho = 0.99$	
Method	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$
`Nv`	7.50	2.43	57.57	28.06	-7.49	-2.43	-57.56	-28.06
`Bm`	6.02	0.59	54.95	19.34	-5.97	-0.48	-54.91	-19.18
`BmO`	4.48	0.64	51.50	19.90	-4.22	-0.55	-51.34	-19.76
`IsPo`	3.78	0.55	47.17	12.24	-2.95	-0.13	-46.54	-9.65
`IsDe`	3.81	0.52	47.17	12.32	-3.00	-0.16	-46.54	-9.82
`IsCo`	3.87	0.50	47.44	12.59	-3.17	-0.20	-46.84	-10.35
`IsPoBm`	3.83	0.72	47.17	12.37	-2.98	-0.16	-46.55	-9.80
`IsDeBm`	3.84	0.71	47.18	12.45	-3.02	-0.16	-46.56	-9.90
`IsCoBm`	3.87	0.70	47.28	12.42	-3.08	-0.16	-46.68	-9.96
`SdAr`	**3.67**	**0.46**	44.80	11.38	-2.38	**-0.11**	-43.56	-7.40
`SdGl`	5.80	**0.39**	55.19	22.05	-5.75	-0.26	-55.16	-22.00

`SdBa`	4.49	1.26	51.14	14.74	-3.30	-0.54	-50.84	-11.34
`SdPa`	4.41	1.27	50.78	14.52	-3.02	-0.48	-50.46	-10.51
`SdQs`	4.59	1.36	51.43	15.30	-3.47	-0.73	-51.14	-12.09
`SdBaWe`	6.65	0.69	57.71	21.48	-6.61	-0.39	-57.71	-21.40
`SdPaWe`	6.55	0.67	57.62	21.20	-6.52	-0.28	-57.61	-21.11
`SdQsWe`	6.75	0.78	57.83	21.80	-6.72	-0.54	-57.82	-21.72
`Nw`	4.83	0.78	52.25	20.91	-4.64	-0.73	-52.13	-20.81
`NwPw`	**3.70**	**0.46**	**43.21**	**11.37**	**-2.08**	**-0.10**	**-40.95**	**-6.46**
`AnBa`	4.45	0.63	49.53	14.32	-4.01	-0.47	-49.19	-12.92
`AnBaPw`	**3.74**	**0.46**	43.40	**11.23**	**-2.08**	**-0.09**	-41.28	**-6.31**
`AnPa`	4.45	0.57	51.26	14.37	-3.96	-0.36	-51.01	-12.83
`AnPaPw`	**3.74**	**0.46**	**43.28**	**11.24**	**-2.06**	**-0.10**	**-41.07**	**-6.40**
`AnQs`	4.32	0.56	50.72	13.95	-3.76	-0.36	-50.43	-12.20
`AnQsPw`	3.77	0.46	**43.35**	**11.27**	**-2.07**	**-0.09**	**-41.20**	**-6.34**
`HiBa`	4.82	0.65	50.17	15.44	-4.52	-0.55	-49.87	-14.58
`HiBaPw`	**3.73**	0.47	**43.28**	**11.31**	**-2.04**	-0.11	**-41.06**	**-6.43**
`HiPa`	4.92	0.98	49.80	15.95	-4.34	-0.64	-49.40	-14.18
`HiPaPw`	3.82	0.47	**43.32**	11.41	-2.10	-0.11	**-40.98**	-6.55
`BsSt`	4.81	0.70	50.91	16.34	-4.49	-0.55	-50.68	-15.81
`BsCi`	4.66	0.62	50.86	17.21	-4.35	-0.49	-50.63	-16.87
`BsStFx`	4.85	0.69	50.90	17.92	-4.54	-0.58	-50.67	-17.61
`BsCiFx`	4.51	0.66	51.00	19.98	-4.20	-0.58	-50.79	-19.84

Table 3: Monte Carlo results for the regime–switching model.
Root mean squared error (RMSE) and bias of the various NSE estimators in the case of n simulated data from the regime-switching model. The number of Monte Carlo replications is set to 1,000. In bold, we report the five best performers per column. See Table 1 for the description of the NSE estimators and Section 3.1 for the description of the Monte Carlo study.

	RMSE ($\times$ 10)				**Bias ($\times$ 10)**			
	p=0.9		p=0.99		p=0.9		p=0.99	
Method	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$
`Nv`	9.95	3.12	53.26	13.95	-9.95	-3.12	-53.21	-13.95
`Bm`	6.74	0.56	50.69	7.41	-6.71	-0.26	-50.49	-7.39
`BmO`	3.99	0.53	48.75	7.78	-3.64	-0.35	-47.75	-7.76
`IsPo`	3.63	0.52	47.79	3.80	-2.36	**-0.02**	-45.26	-2.43
`IsDe`	3.58	0.48	47.80	3.78	-2.41	**-0.06**	-45.26	-2.60
`IsCo`	3.61	0.44	47.82	3.88	-2.57	-0.10	-45.31	-2.85
`IsPoBm`	3.80	1.12	47.79	3.97	-2.57	-0.09	-45.28	-2.56
`IsDeBm`	3.84	1.11	47.80	4.00	-2.63	-0.11	-45.28	-2.66
`IsCoBm`	3.79	1.10	47.79	3.99	-2.65	-0.11	-45.28	-2.70
`SdAr`	3.14	**0.40**	47.01	**2.94**	-1.57	-0.08	-43.19	**-1.63**
`SdGl`	6.07	0.53	50.90	9.15	-6.04	0.41	-50.74	-9.14
`SdBa`	6.39	2.05	49.06	6.26	-2.93	-0.83	-47.95	-3.22
`SdPa`	6.67	2.04	49.04	6.47	-2.51	-0.77	-47.84	-2.88
`SdQs`	6.78	2.15	49.29	6.81	-3.62	-1.08	-48.26	-3.97
`SdBaWe`	7.30	0.91	53.85	8.32	-7.11	-0.30	-53.80	-8.15
`SdPaWe`	7.00	0.92	53.70	8.03	-6.78	-0.15	-53.64	-7.85
`SdQsWe`	7.63	1.00	54.00	8.62	-7.45	-0.56	-53.95	-8.46
`Nw`	4.85	0.64	48.40	8.53	-4.65	-0.57	-47.81	-8.52
`NwPw`	**3.03**	**0.39**	**46.58**	2.97	**-1.34**	**-0.07**	**-41.70**	-1.71
`AnBa`	4.50	0.59	48.43	5.65	-4.07	-0.46	-46.47	-5.50
`AnBaPw`	**3.04**	**0.40**	**46.60**	**2.87**	**-1.49**	**-0.07**	**-41.73**	**-1.48**
`AnPa`	4.30	0.49	48.22	4.90	-3.59	-0.31	-46.08	-4.64
`AnPaPw`	**2.99**	0.41	**46.62**	**2.88**	**-1.46**	-0.08	**-41.80**	**-1.48**
`AnQs`	4.08	0.48	48.09	5.10	-3.36	-0.33	-45.80	-4.91
`AnQsPw`	**3.04**	0.41	**46.63**	**2.87**	**-1.45**	-0.08	**-41.82**	**-1.48**
`HiBa`	5.11	0.63	48.61	6.00	-4.88	-0.54	-47.12	-5.88
`HiBaPw`	**3.00**	**0.39**	**46.61**	2.96	**-1.33**	**-0.07**	**-41.74**	**-1.58**
`HiPa`	5.85	1.16	48.79	4.95	-4.73	-0.58	-47.23	-3.93
`HiPaPw`	3.18	**0.39**	46.65	**2.93**	-1.49	-0.07	-41.88	-1.64
`BsSt`	5.42	0.73	49.34	5.80	-5.04	-0.54	-48.01	-5.60
`BsCi`	4.84	0.62	48.92	6.10	-4.44	-0.47	-47.52	-6.01
`BsStFx`	5.55	0.72	49.40	6.61	-5.04	-0.49	-48.09	-6.53

`BsCiFx`	4.44	0.59	48.74	7.86	-3.92	-0.42	-47.49	-7.85

Table 4: Monte Carlo results for the value-at-risk precision.
Root mean squared error (RMSE) and bias of the various NSE estimators in the case of MCMC chains of length n generated for the estimation of a Markov–switching GARCH model for value-at-risk estimation. The number of Monte Carlo replications is set to 1,000. In bold, we report the 10 best performers per column. See Table 1 for the description of the methods and Section 3.2 for the description of the Monte Carlo study.

	RMSE ($\times 10^2$)		**Bias** ($\times 10^2$)	
	$n = 100$	$n = 1{,}000$	$n = 100$	$n = 1{,}000$
`Nv`	6.88	7.54	-6.88	-7.54
`Bm`	6.84	7.38	-6.84	-7.38
`BmO`	6.80	7.40	-6.80	-7.39
`IsPo`	6.74	7.17	-6.74	-7.16
`IsDe`	6.74	7.18	-6.74	-7.17
`IsCo`	6.74	7.19	-6.74	-7.17
`IsPoBm`	6.74	7.18	-6.74	-7.17
`IsDeBm`	6.74	7.18	-6.74	-7.17
`IsCoBm`	6.74	7.18	-6.74	-7.17
`SdAr`	6.70	7.11	-6.69	-7.09
`SdGl`	6.85	7.43	-6.85	-7.43
`SdBa`	6.81	7.29	-6.81	-7.28
`SdPa`	6.80	7.29	-6.80	-7.28
`SdQs`	6.81	7.31	-6.81	-7.30
`SdBaWe`	6.88	7.46	-6.88	-7.46
`SdPaWe`	6.88	7.46	-6.88	-7.46
`SdQsWe`	6.88	7.47	-6.88	-7.47
`Nw`	6.81	7.41	-6.81	-7.41
`NwPw`	6.64	7.08	-6.63	-7.06
`AnBa`	6.78	7.25	-6.77	-7.24
`AnBaPw`	**6.64**	**7.08**	**-6.63**	**-7.05**
`AnPa`	6.80	7.28	-6.80	-7.27
`AnPaPw`	**6.64**	**7.08**	**-6.63**	**-7.06**
`AnQs`	6.79	7.27	-6.79	-7.27
`AnQsPw`	**6.64**	**7.08**	**-6.63**	**-7.05**
`HiBa`	6.77	7.25	-6.77	-7.24
`HiBaPw`	**6.64**	**7.07**	**-6.63**	**-7.04**
`HiPa`	6.78	7.26	-6.77	-7.25
`HiPaPw`	**6.63**	**7.07**	**-6.62**	**-7.05**
`BsSt`	6.80	7.31	-6.79	-7.31
`BsCi`	6.79	7.34	-6.79	-7.33
`BsStFx`	6.80	7.35	-6.79	-7.35
`BsCiFx`	6.80	7.39	-6.79	-7.39

Overall, we recommend using HAC estimators in conjunction with prewhitening when dealing with (time or MCMC output) series exhibiting large/extreme autocorrelations.

Acknowledgements

All analyses have been performed in the R statistical language (Core Team 2015) with the package **nse** (Ardia and Bluteau 2017) available at https://github.com/keblu/nse. We thank the Fonds de Recherche du Québec – Société et Culture and Industrielle–Alliance for their financial support. All errors and omissions are the sole responsibility of the authors.

Notes

1 In a former version of this note, the performance of the estimators was evaluated in terms of coverage accuracy of corresponding t ratios. However, what is optimal for estimation may not be equally optimal for inference. In particular, when autocorrelation is high (in the case

of an almost unit root process) in small samples, measuring accuracy via inference can be biased as asymptotic arguments do not apply. In this case, methods for accounting to local-to-one unit roots and long memory could be employed.

[2] The series of draws may be similar to an AR process with a persistence that is close to a unit root. The series of draws may also have a different structure similar to a Markov regime-switching process (with possibly low probabilities of transition between different regimes).

[3] Estimation is carried out with the `R` package **MSGARCH** (Ardia et al. 2016a, 2016b).

[4] Obviously, such a strategy is computationally extremely costly, and infeasible in practical applications, where banks and risk managers need to assess the risk of several hundreds of positions in the portfolios. Therefore, an adequate NSE estimator, able to provide a clear picture of the precision of the risk measure, is essential.

References

Akaike, H. 1974. “A New Look at the Statistical Model Identification.” *IEEE Transactions on Automatic Control* 19: 716–23. doi:10.1007/978-1-4612-1694-0_16.

Andrews, D. W. K. 1991. “Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation.” *Econometrica* 59: 817–58. doi:10.2307/2938229.

Andrews, D. W. K., and J. C. Monahan. 1992. “An Improved Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimator.” *Econometrica* 60: 953–66. doi:10.2307/2951574.

Ardia, D. 2008. *Financial Risk Management with Bayesian Estimation of GARCH Models*. Berlin/Heidelberg: Springer. doi:10.1007/978-3-540-78657-3.

Ardia, D., and K. Bluteau. 2017. “nse: Computation of Numerical Standard Errors in R.” *Journal of Open Source Software* 10. doi:10.21105/joss.00172.

Ardia, D., K. Bluteau, K. Boudt, and D.-A. Trottier. Markov-Switching GARCH Models in R: The MSGARCH package. *Working paper* 2016a https://ssrn.com/abstract=2845809. Forthcoming in Journal of Statistical Software.

Ardia, D., K. Bluteau, K. Bout, B. Peterson, and D.-A. Trottier. 2016b. “MSGARCH: Markov Switching GARCH Models in R.” https://cran.r-project.org/package=MSGARCH.

Flegal, J. M., and G. L. Jones. 2010. “Batch Means and Spectral Variance Estimators in Markov Chain Monte Carlo.” *Annals of Statistics* 38: 1034–70. doi:10.1214/09-aos735.

Geyer, C. J. 1992. “Practical Markov Chain Monte Carlo.” *Statistical Science* 7: 473–83. doi:10.1214/ss/1177011137.

Haas, M., S. Mittnik, and M. Paollela. 2004. “A New Approach to Markov-Switching GARCH Models.” *Journal of Financial Econometrics* 2: 493–530. doi:10.1093/jjfinec/nbh020.

Heidelberger, P., and P. D. Welch. 1981. “A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations.” *Communications of the ACM* 24: 233–45. doi:10.1145/358598.358630.

Hirukawa, M. 2010. A Two-Stage Plug-in Bandwidth Selection and Its Implementation for Covariance Estimation.” *Econometric Theory* 26: 710–43. doi:10.1017/s0266466609990089.

Hoogerheide, L., and H. K. van Dijk. 2010. “Bayesian Forecasting of Value at Risk and Expected Shortfall Using Adaptive Importance Sampling.” *International Journal of Forecasting* 26: 231–47. doi:10.1016/j.ijforecast.2010.01.007.

Hurvich, C. M. 1985. “Data-Driven Choice of a Spectrum Estimate: Extending the Applicability of Cross-Validation Methods.” *Journal of the American Statistical Association* 80: 933–40. doi:10.1080/01621459.1985.10478207.

Lahiri, S. N. 1999. “Theoretical Comparisons of Block Bootstrap Methods.” *Annals of Statistics* 27: 386–404. doi:10.1214/aos/1018031117.

Newey, W. K., and K. D. West. 1987. “A Simple, Positive Semi-definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix.” *Econometrica* 55: 703–8. doi:10.2307/1913610.

Newey, W. K., and K. D. West. 1994. “ Automatic Lag Selection in Covariance Matrix Estimation.” *Review of Economic Studies* 61: 631–53. doi:10.3386/t0144.

Nordman, D. J. 2009. “A Note on the Stationary Bootstrap’s Variance.” *Annals of Statistics* 37: 359–70. doi:10.1214/07-aos567.

Parzen, E. 1957. “On Consistent Estimates of the Spectrum of a Stationary Time Series.” *The Annals of Mathematical Statistics* 28: 329–48.

Patton, A., D. N. Politis, and H. White. 2009. “Correction to ‘Automatic Block-length Selection for the Dependent Bootstrap’ by D. Politis and H. White.” *Econometric Reviews* 28: 372–75. doi:10.1080/07474930802459016.

Percival, D., and W. Constantine. 2006. “Exact Simulation of Gaussian Time Series from Nonparametric Spectral Estimates with Application to Bootstrapping.” *Statistics and Computing* 16: 25–35. doi:10.1007/s11222-006-5198-0.

Politis, D. N., and J. P. Romano. 1992. “A Circular Block-Resampling Procedure for Stationary Data.” In *Exploring the Limits of Bootstrap*, 263–70. John Wiley & Sons.

Politis, D. N., and J. P. Romano. 1994. “The Stationary Bootstrap.” *Journal of the American Statistical Association* 89: 1303–13. doi:10.2307/2290993.

Politis, D. N., and H. White. 2004. “Automatic Block-Length Selection for the Dependent Bootstrap.” *Econometric Reviews* 23: 53–70. doi:10.1081/etc-120028836.

Press, H., and J. Tukey. 1956. “Power Spectral Methods of Analysis and Application in Airplane Dynamics.” In *Vol. IV of AGARD Flight Test Manual*, 1–41. Paris, Ch. C.

R Core Team. 2015. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/

Vihola, M. 2012. “Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate.” *Statistics and Computing* 22: 997–1008. doi:10.1007/s11222-011-9269-5.

Welch, P. D. 1967. “The Use of fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms.” *IEEE Transactions on Audio and Electroacoustics* 15: 70–73. doi:10.1109/TAU.1967.1161901.