

Chapter 3

Adjusting for bias in long horizon regressions using R

Kenneth D. West^{a,*} and Zifeng Zhao^b

^a*Department of Economics, University of Wisconsin-Madison, Madison, WI, United States*

^b*Department of Information Technology, Analytics and Operations, Mendoza College of Business, University of Notre Dame, Notre Dame, IN, United States*

*Corresponding author: e-mail: kdwest@wisc.edu

Abstract

Long horizon regressions that rely on linear models are common in many applied fields. Examples from economics include forecasting inflation 12 quarters ahead (Crone et al., 2013) and relating 120 month ahead changes in exchange rates to current period variables (Snaith et al., 2013). We describe R code to implement recently developed procedures that adjust long horizon regressions to lessen bias in parameter estimates (West, 2016).

Keywords: Least squares bias, Small sample bias, Bias reduction, Multistep forecast, Direct forecast, VAR model

1 Introduction

In this chapter we consider small sample bias in long horizon least squares regressions in discrete time linear time series models. A leading application is to “direct” multistep forecasts. We describe R code that adjusts for small sample bias in such regressions. Such adjustments may be important because in some specifications such bias is arbitrarily large for an arbitrarily long horizon (West, 2016).

We begin by reviewing long horizon regressions and the direct method for making a multistep forecast. We then describe R functions to implement recently developed procedures that modify long horizon regression parameters to lessen bias. We close with the code for an empirical application.

Throughout, our topic is solely construction of bias adjusted regression estimates, taking as given a set of regressors or predictors. That is, we do not discuss selection of predictors nor many other topics that are important

in applied work such as construction of confidence intervals and forecast evaluation. See [West and Zhao \(2018\)](#) for how bias adjustment relates to forecast evaluation via mean squared prediction error.

The R code can be downloaded from <https://www.ssc.wisc.edu/~kwest/appendices/appendices.htm>.

2 Long horizon regressions

Let y_t be a scalar time series, with data running up to time T . We suppose that one wishes to model or forecast y_{t+q} for some horizon $q+1 > 0$. If the data are monthly, $q+1$ is measured in months, and similarly for frequencies other than monthly.

In economics applications, the relevant range for $q+1$ runs from one step ahead ($q+1=1$) to $q+1$ in excess of 100. One step ahead forecasts—business investment next quarter, employment next month, and so on—are ubiquitous. Multistep forecasts are also common. The Survey of Professional Forecasters asks participants to forecast GDP and inflation up to five quarters ahead; [Mark \(1995\)](#) forecasts exchange rates 16 quarters ahead; and [Crone et al. \(2013\)](#) forecast inflation 12 quarters ahead. Policy and academic work sometimes look at even longer horizons. [Lunsford and West \(2017\)](#) forecast interest rates 10 years ahead; [Snaith et al. \(2013\)](#) relate 120 month ahead changes in exchange rates to period t predictors; and [Hjalmarsson \(2011\)](#) relates 10 year ahead stock returns to period t predictors. These examples could be multiplied many times over.

Note that the final two examples used the verb “relates” and not “forecasts.” Such research evaluates the connection between a many step ahead variable and a set of regressors using in-sample analysis only. The R procedures we describe here are just as valuable for such in-sample analysis as it is for analysis that involves forecasting. However, to focus the discussion, we often shall describe our R functions in terms of forecasts.

Much though not all the relevant work—both in- and out-of-sample—relies on stationary linear models, which we maintain here. Specifically, we assume that the forecast or in-sample modeling of a stationary variable y_{t+q} relies on the projection of y_{t+q} onto a constant and a $(k \times 1)$ vector X_{t-1} . Write the population least squares projection as

$$y_{t+q} = \alpha + X'_{t-1}\beta + \eta_{t+q}. \quad (1)$$

The disturbance η_{t+q} is unobserved and is defined as the difference between y_{t+q} and the population projection of y_{t+q} onto a constant and X_{t-1} .

In economics, there are two broad classes of applications. In the first class, the elements of X_{t-1} are financial market or survey variables that are hypothesized to be good predictors of an economic variable y . A simple example occurs in the [Snaith et al. \(2013\)](#) paper cited above:

- Let s_t be the log of the end of month nominal exchange rate (say, dollars per British pound), so that

$$\Delta s_t \equiv s_t - s_{t-1}$$

is approximately percentage change in the exchange rate. Observe that with this definition,

$$y_{t+q} \equiv s_{t+q} - s_{t-1} = \Delta s_{t+q} + \Delta s_{t+q-1} + \dots + \Delta s_t$$

is approximately the percentage change in the exchange rate from month $t-1$ to month $t+q$.

- Let i_t be the nominal return on a nominally safe $q+1$ month US bond. Since we are assuming for the sake of illustration that the data are monthly, if $q+1=120$, then i_t is the interest rate on a 10-year US Treasury bond. (“Nominally safe” means: the borrower [i.e., the US government] will not default, and the return is only guaranteed in nominal rather than real inflation adjusted terms.) Let i_t^* be the corresponding foreign interest rate (the comparable interest rate in the UK, in this example).

Then a certain economic model says that the interest differential on 10-year bonds $i_t - i_t^*$ well explains the cumulative change in the exchange rate over the following 10 years. So the regression run is

$$y_{t+q} \equiv s_{t+q} - s_{t-1} = \alpha + \beta_1 (i_t - i_t^*) + \eta_{t+q}; X'_{t-1} \equiv i_t - i_t^* \text{ and } k=1. \quad (2)$$

The second broad class of applications are ones where X_{t-1} consists of deterministic terms and lags of y_t and perhaps other variables. An example is [Marcellino et al. \(2006\)](#), who examine prediction of many monthly economic series at horizons up to $q+1=24$ months, using both univariate (X_{t-1} includes lags of y_t) and bivariate (X_{t-1} includes lags of y_t and of a second variable) models. In the simplest possible case, the model is univariate and the forecasting horizon is one step ahead ($q+1=1$). Then (1) is

$$y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_k y_{t-k} + \eta_t; X'_{t-1} = (y_{t-1} \dots y_{t-k}). \quad (3)$$

A specification such as (3) indicates that the researcher thinks an AR(k) model well approximates y_t . For this same set of predictors, a multistep direct prediction ($q+1 > 1$) relies on the projection

$$y_{t+q} = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_k y_{t-k} + \eta_{t+q}. \quad (4)$$

Eq. (4) relies on the fact that a multistep prediction of an AR(k) model is linear in k lags of the variable. Of course, the coefficients $\alpha, \beta_1, \dots, \beta_k$ in (4) are different from those in (3) (except when $q+1=1$).

A modest permutation of (4) involves predicting the average rather than point in time forecast. For example, we generally are less interested in inflation in the fourth quarter of next year than we are in average inflation over the next four quarters. This sort of application involves a regression of the form

$$\begin{aligned} y_{t+q} &\equiv (x_{t+q} + x_{t+q-1} + \cdots + x_t) / (q + 1) \\ &= \alpha + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \cdots + \beta_k x_{t-k} + \eta_{t+q}. \end{aligned} \quad (5)$$

Again, the coefficients in (5) are, in general, different from those in (4).

Each of (3)–(5) illustrated the direct multistep forecast of y_{t+q} . Note for future use that in (4) and (5), if, indeed, the hypothesized AR(k) model is correct, the disturbance follows a moving average process of order q :

$$\eta_{t+q} \sim \text{MA}(q). \quad (6)$$

In many though not all motivations for regressions in the first class of applications, illustrated by (2), it is also true that $\eta_{t+q} \sim \text{MA}(q)$.

A brief digression on methods for multistep forecasts: for forecasts such as those in (4) and (5), an alternative approach is to recursively generate j period ahead forecasts by using $j - 1$ period ahead forecasts. This is the approach of [Box and Jenkins \(1976\)](#), for example. Let “ $\hat{\cdot}$ ” denote a least squares estimate. For example, in the model (4), the Box and Jenkins method constructs one and two step ahead forecasts via

$$\text{one step ahead forecast} = \hat{\alpha} + \hat{\beta}_1 y_T + \cdots + \hat{\beta}_k y_{T-k+1}, \quad (7a)$$

$$\begin{aligned} \text{two step ahead forecast} &= \hat{\alpha} + \hat{\beta}_1 \times (\text{one step ahead forecast}) + \hat{\beta}_2 y_T + \\ &\quad \cdots + \hat{\beta}_k y_{T-k+2}. \end{aligned} \quad (7b)$$

This is sometimes called the *iterated* or *plug-in* method of forecasting. Our procedures for bias adjustment are trivially applicable for one step ahead forecasts such as (7a), when the direct and iterated methods are identical. But they are not directly applicable for multistep iterated forecasts such as (7b). For a theoretical comparison of iterated and direct forecasts, see [Ing \(2003\)](#). We focus on direct forecasts because they are dominant in economics.

3 Bias adjustment for long horizon regressions

3.1 Introduction

Least squares estimators of time series models are biased in finite samples. That is, even if we make (mild) assumptions so that estimates are consistent for underlying population quantities, in finite samples the expectation of the least squares estimator is not, in general, equal to the underlying population quantity. For the simple AR(1) model (in (3), $k=1$ and $\eta_t \sim \text{i.i.d.}$), [Kendall \(1954\)](#) suggested that

$$E\hat{\beta}_1 \approx \beta_1 - \frac{(1+3\beta_1)}{T}. \quad (8)$$

More generally, West (2016) shows that for a $k \times 1$ vector b that depends on own- and cross-moments of X_{t-1} and η_t , the least squares estimator of (8) satisfies

$$E\hat{\beta} = \beta + \frac{b}{T} + O\left(T^{-3/2}\right). \quad (9)$$

In the simple AR(1) model underlying (8), $k=1$, $\beta=\beta_1$ and $b=-(1+3\beta_1)$. The small sample bias we are concerned with in this chapter is the b/T term in (9). We describe R code to construct an estimate \hat{b} , yielding a bias adjusted estimate $\hat{\beta} - \hat{b}/T$.

Bias in estimate of the constant term α (defined in (1)) follows from:

$$(\text{bias to order } T^{-1} \text{ in } \hat{\alpha}) = (EX'_t)b.$$

Given an estimate \hat{b} supplied by the R code we are about to describe, and a sample average \bar{X} , one can adjust for such bias via:

$$(\text{bias adjusted estimate of } \alpha) = \hat{\alpha} - \bar{X}'\hat{b}/T.$$

Because such an adjustment follows directly from adjustment for bias in the slope coefficient vector β , we shall not further discuss bias adjustment of α .

A number of papers have derived b/T when forecasts are one step ahead ($q+1=1$) and η_t is a conditionally homoskedastic martingale difference. See Shaman and Stine (1988) for the univariate AR (Eq. 4) and the summary in Engsted and Pedersen (2014) when the one step ahead forecast comes from an equation from a vector autoregression. West (2016) derives b/T for arbitrary horizons $q+1$ and allowing time varying second moments in η_{t+q} . Our R procedures implement a subset of the results allowed in West (2016). In particular, our code coheres with a special case of the theory in West (2016). This special case requires that (a) a certain cumulant condition holds that rules out time varying second moments and (b) η_{t+q} is uncorrelated not just with X_{t-1} itself but also with all lags of X_{t-1} (i.e., $E\eta_{t+q}X'_{t-j} = 0$ for $j=2, 3, \dots$). (This last condition is generally assumed under the null of the model but may fail under misspecification. For example, in (4), if $y_t \sim \text{AR}(k)$, then $E\eta_{t+q}X'_{t-j} = 0$ for all $j \geq 1$ and the condition holds. But if $y_t \sim \text{AR}(m)$ for some $m > k$, then in (4) the condition fails and $E\eta_{t+q}X'_{t-j} \neq 0$ for at least one $j > 1$.) In any application in which (a) or (b) fail, the estimate \hat{b} that our code delivers should be taken with a larger than usual grain of salt.

See West (2016) for details. One important theoretical result from West (2016): in regressions such as (2) or (5) where the left hand side variable is a long horizon sum or average of a stationary variable, bias b is arbitrarily big in absolute value for an arbitrarily long horizon q . This is a general result for

such left hand side variables, and is not specific to the examples such as (2) and (5). Hence a bias adjustment is especially appealing for such regressions.

To understand the parameters and/or moments that must be passed to our functions that estimate b , it may be helpful to note that b depends on own- and cross-covariances of the right hand side variables X_{t-1} and the disturbance η_{t+q} . The first pair of functions that we are about to describe (**longhor1**, **longhor**) construct estimates of the relevant second moments and the user needs only to pass a parameter specifying a certain lag length. The second pair of functions that we describe (**proc_vb_ma0**, **prov_vb_maq**) rely partially on the user to construct the relevant second moments; prior to invoking the functions, the user is required to have estimated a certain autoregression or vector autoregression, with the results of that estimation passed to our R functions.

The first pair of functions (**longhor1**, **longhor**) are high-level, but require that X_{t-1} consist solely of lags of a single variable such as in (2)–(5). This single variable may or may not be lags of the left hand side variable; the “may” case is illustrated in (3) and (4), the “may not” in (2) and (5). The second pair of functions (**proc_vb_ma0**, **prov_vb_maq**) are lower level, requiring more work from the user. But they do not restrict the specification of X_{t-1} .

3.2 R function longhor1

One of our R functions is most easily motivated with reference to any of (3)–(5). The user passes a vector **yseries** with data on the left hand side variable and a second vector **xseries** with data on the right hand side variable. The two vectors are different in the case of (5) but are the same in the cases of (3) or (4). The user also specifies the integer order of the lead of the left hand side relative to the right hand side **nq** ($=q$ in (1)) the integer number of lags **nk** ($=k$ in (1)) on the right hand side, integer pointers **first** and **last** to the sample period and an integer **narlag** that should be set to **nk** if (3)–(5) is of interest and whose presence is explained below. The function returns three $\mathbf{nk} \times 1$ vectors:

- **vbias** ($= b$, as defined in (9));
- **betahat** ($=$ least squares $\hat{\beta}$);
- **betahat_adj** ($=$ bias adjusted $\hat{\beta} = \text{betahat} - (\text{vbias}/T)$, $T = \text{last} - \text{first} + 1$).

See Table 1, which summarizes this information.

To clarify dating and variable definitions: the regression of interest is **yseries**, **nq** periods ahead, on lags 1 to **nk** of **xseries**:

$$\begin{aligned} \text{yseries}(t + \mathbf{nq}) &= \alpha + \beta_1 \text{xseries}(t - 1) + \dots + \beta_k \text{xseries}(t - \mathbf{nk}) \\ &\quad + \text{disturbance}(t + \mathbf{nq}), \end{aligned} \tag{10}$$

$$t + \mathbf{nq} = \mathbf{first}, \dots, t + \mathbf{nq} = \mathbf{last}.$$

Since the regression is run with **yseries** dates running from **first** to **last**, the dates on **xseries** $_{t-1}$ go from $t - 1 = \mathbf{first_nq} - 1$ to $t - 1 = \mathbf{last_nq} - 1$.

TABLE 1 Function `longhor1`

```
result <- longhor1(yseries, xseries, first, last, nq, nk, narlags)
vbias <- result[[1]]; betahat <- result[[2]]; betahat_adj
<- result[[3]]
```

The right hand side variables in the regression of interest consist of a constant and lags of a single variable, such as in (2)–(6). The left hand side may or may not be a lead of the same variable.

Passed by user

yseries	vector for the left hand side variable
xseries	vector for the right hand side variable
first	integer start date for the left hand side variable in the regression
last	integer end date for the left hand side variable in the regression
nq	q : integer horizon, $nq \geq 0$
nk	k : integer number of lags k of xseries to include on the r.h.s. of the regression (1)
narlag	integer number of lags to include in estimating an AR model for xseries (needed to compute the bias). The user should insure that narlag is sufficient to produce a white noise residual in this autoregression

Returned to user

result[[1]]	$nk \times 1$ vector: \hat{b} = estimate of $k \times 1$ numerator of bias to order T^{-1} (T = sample size)
result[[2]]	$nk \times 1$ vector: $\hat{\beta}$ = ordinary least squares estimate of $k \times 1$ β
result[[3]]	$nk \times 1$ vector, bias adjusted $\hat{\beta}$, result[[3]] = result[[2]] – (result[[1]])/ T , $T = \text{last} - \text{first} + 1$

Functions invoked directly or indirectly by **longhor1**: **proc_vb_ma0**, **proc_vb_maq**, and **proc_vbias**.

To further clarify dating, consider a concrete example. Suppose that **yseries** includes 99 observations. For simplicity of exposition, assume data are annual and run from 1901 to 1999. Thus **yseries**(4) is data from 1904 and **xseries**(11) is data from 1911, for example. Suppose further that $nq = 7$ and $nk = 2 = narlag$. Suppose, finally, that one wishes to run the regression with left hand side data running from 1910 to 1990 (thus not using some of the data):

$$\mathbf{yseries}_{t+7} = \text{const.} + \beta_1 \mathbf{xseries}_{t-1} + \beta_2 \mathbf{xseries}_{t-2} + \text{disturbance}, \quad (11)$$

$$t+7 = 1910, \dots, 1990$$

Thus the vector of the left hand side variable and the matrix of stochastic right hand side variables are

$$\begin{pmatrix} \mathbf{yseries}_{1910} \\ \mathbf{yseries}_{1911} \\ \vdots \\ \mathbf{yseries}_{1990} \end{pmatrix}, \begin{pmatrix} \mathbf{xseries}_{1902} & \mathbf{xseries}_{1901} \\ \mathbf{xseries}_{1903} & \mathbf{xseries}_{1902} \\ \vdots & \vdots \\ \mathbf{xseries}_{1982} & \mathbf{xseries}_{1981} \end{pmatrix}.$$

Then one invokes **longhor1** via

```
result<-longhor1(yseries,xseries,10,90,7,2,2),
vbias<-result[[1]];betahat<-result[[2]];betahat_adj<-result[[3]].
```

 (12)

In this example, the procedure returns three 2×1 vectors: **betahat** $\equiv \hat{\beta} \equiv (\hat{\beta}_1 \hat{\beta}_2)'$, **vbias** $\equiv \hat{b}$, **betahat_adj** $\equiv \hat{\beta} - \hat{b}/T$, where $T = 81$.

More generally, the procedure is invoked via

```
result<-longhor1(yseries,xseries,first,last,nq,nk,narlag)
vbias<-result[[1]];betahat<-result[[2]];betahat_adj<-result[[3]].
```

 (13)

Note: The code does not do error checks for missing data. So, suppose in the concrete example just given, where available data run from 1901 to 1999, that the user passes **first**=5 along with **nq**=7. Then the function would assume the first observation on the left hand side variable is 1905 and the first observation on **xseries** is 8 years earlier ($8 = \mathbf{nq} + 1$), i.e., 1897—a date that is not in the sample. Results are unpredictable if, as in this illustration, parameters point to data that are not available.

Procedure **longhor1** can also handle applications such as (2), at the additional cost of the user specifying a lag length for an autoregression in the right hand side variable. Let x_t be a generic right hand side variable in a regression, with $x_t = i_t - i_t^*$ in (2) as an example. As noted above, b depends on own- and cross-covariances of the right hand side variables X_{t-1} —in this case, simply x_{t-1} —and the disturbance η_{t+q} . In **longhor1**, to compute the necessary second moments, the code relies in part on the presumption that the dynamics of x_t can be approximated by a finite order autoregression. The user must specify the order of this autoregression, i.e., the order of an autoregression in x_t that produces an approximately white noise disturbance. That is the purpose of the parameter **narlag**. In (3)–(5), it will normally be the case that **narlag** = **nk**—one chooses to use k lags in the regression because use of k lags produces an approximately white noise disturbance. But in (2), the theory that leads to the regression does not constrain the order of an approximating autoregression for $x_t (= i_t - i_t^*)$.

To illustrate: suppose that the user decides that an AR(4) produces an approximately white noise disturbance in $i_t - i_t^*$. Then for (2), one invokes **longhor1** with

- **nk**=1 (because there is only one stochastic right hand side variable), and
- **narlag**=4 (on the user's conclusion that an AR(4) in $i_t - i_t^*$ produces an approximately white noise residual).

3.3 R function **longhor**

This is a generalization of **longhor1** in which a vector autoregression rather than an autoregression is used to compute autocovariances of the variables whose lags are in X_{t-1} . In the exchange rate example (2), one might suppose that sharper estimates of the moments of $i_t - i_t^*$ will result from use of the time series of exchange rates in addition to the time series of $i_t - i_t^*$.

Again let x_t be the variable whose lags are in X_{t-1} . Let w_{1t}, \dots, w_{nt} be n additional variables thought to have useful information about the autocovariances of x_t . Put these n additional variables in a matrix **Wseries**. Then to compute autocovariances from a VAR in $(x_t, w_{1t}, \dots, w_{nt})$ one invokes **longhor** passing **Wseries** and setting **nWseries**= n . See Table 2.

3.4 R functions **proc_vb_ma0** and **proc_vb_maq**

These are low level functions invoked by **longhor** and **longhor1**. They are flexible enough to allow computation of bias to order T in any least squares regression. An example of a regression covered by these procedures but not allowed by **longhor** or **longhor1** is a direct forecast that relies on a bivariate information set

TABLE 2 Function **longhor**

```
result <- longhor(yseries, xseries, Wseries, nWseries, first,
                last, nq, nk, narlag)
vbias <- result[[1]]; betahat <- result[[2]]; betahat_adj
<- result[[3]]
```

The right hand side variables in the regression of interest consist of a constant and lags of a single variable, such as in (2)–(6). The left hand side may or may not be a lead of the same variable.

Parameters are as for **longhor1**, with the two additional parameters defined as

Wseries	matrix containing variables in addition to <i>xseries</i> to be used in the VAR that will be used to compute autocovariances of x_t
nWseries	the number of series or columns in Wseries . If nWseries =0, the procedure calls longhor1 to compute b

Functions invoked directly or indirectly by **longhor**: **longhor1**, **proc_vb_ma0**, **proc_vb_maq**, and **proc_vbias**.

$$\begin{aligned}
y_{t+q} &= \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_m y_{t-m} + \beta_{m+1} x_{t-1} + \beta_{m+2} x_{t-2} + \cdots + \beta_k x_{t-m} + \eta_{t+q}; \\
X'_{t-1} &= (y_{t-1}, y_{t-2}, \dots, y_{t-m}, x_{t-1}, x_{t-2}, \dots, x_{t-m}).
\end{aligned}
\tag{14}$$

In (14), $k=2m$. Of course our functions also allow the left hand side variable to be not point in time as in (14) but an average or cumulated sum as in (2) or (5). The key difference between (14) and specifications covered by **longhor** and **longhor1** is that the latter require that X_{t-1} consist of lags of a single variable, whereas there are lags of two different variables on the right hand side of (14). The functions **proc_vb_ma0** and **proc_vb_maq** accommodate not only two but any number of different variables on the right hand side of the regression of interest.

In contrast to **longhor** and **longhor1**, **proc_vb_ma0** and **proc_vb_maq** require the user to do preliminary calculations before being invoked. First, the user, and not these functions, is required to estimate $\hat{\beta}$. These functions will compute \hat{b} but not $\hat{\beta}$. Second, the user, and not these functions, must estimate a vector autoregression whose variables include those in X_{t-1} , passing certain results from this vector autoregression to these two functions.

To illustrate how to invoke these functions, let us use (14), setting $m=2$ for concreteness:

$$y_{t+q} = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \beta_3 x_{t-1} + \beta_4 x_{t-2} + \eta_{t+q}; \quad k=4; \quad X'_{t-1} = (y_{t-1}, y_{t-2}, x_{t-1}, x_{t-2}).
\tag{15}$$

The user needs to specify a vector autoregressive model for the right hand side variables in (15) that can be used by our R functions to deliver accurate estimates of the autocovariances of X_{t-1} . The fact that there are two lags on the right hand side of (15) suggests that a vector autoregression of order 2 will suffice. Let $Y_t(2 \times 1) = (y_t, x_t)'$. Write the VAR(2) as

$$Y_t = \text{const.} + \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + V_t; \quad V_t \sim \text{i.i.d.}; \quad \Omega_V \equiv EV_t V_t'.
\tag{16}$$

In (16), Φ_1 , Φ_2 , and Ω_V are 2×2 ; V_t is 2×1 ; here and in subsequent equations “const.” is an inessential vector of constants whose dimension may be different in different equations.

The user must estimate a VAR such as (16) and pass to our R functions the estimates of the autoregressive coefficients (Φ_1 and Φ_2 in example (16)) and the variance–covariance matrix of the disturbance to the VAR (Ω_V in example (16)) to our R functions. These estimates are passed after rewriting the VAR in the VAR(1) companion form. For the VAR (16), the companion form is

$$\begin{pmatrix} Y_t \\ Y_{t-1} \end{pmatrix} = \text{const.} + \begin{pmatrix} \Phi_1 & \Phi_2 \\ I_2 & 0_{2 \times 2} \end{pmatrix} \begin{pmatrix} Y_{t-1} \\ Y_{t-2} \end{pmatrix} + \begin{pmatrix} V_t \\ 0_{2 \times 1} \end{pmatrix}, \text{ written compactly as}$$

$$Z_t = \text{const.} + \Phi Z_{t-1} + U_t.$$

(17)

The general setup: let Z_t be the $n_Z \times 1$ vector of variables in the VAR used to compute moments related to X_t , with the VAR written in companion form. That is,

$$Z_t - EZ_t = \Phi (Z_{t-1} - EZ_{t-1}) + U_t, \quad \Omega_U = EU_t U_t', \quad X_t = P_X Z_t$$

(18)

Note that the elements of X_t are elements of Z_t .

The user must compute and pass to the code: the dimension k of X_t (called **nk** in the code), the dimension of n_Z of Z_t (called **nZtwid** in the code), P_X and estimates $\hat{\Phi}$ and $\hat{\Omega}_U$ (called **PX**, **phitwid**, and **omegaUtwid** in the code). In the example (16) and (17), **nk** = 4, **nZtwid** = 4,

$$\mathbf{PX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and, letting “ $\hat{}$ ” denote a least squares estimate or residual,

$$\mathbf{phitwid} = \begin{pmatrix} \hat{\Phi}_1 & \hat{\Phi}_2 \\ I_2 & 0_{2 \times 2} \end{pmatrix}, \mathbf{omegaUtwid} = \begin{pmatrix} \hat{\Omega}_V & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} \end{pmatrix}, \hat{\Omega}_V = T^{-1} \sum \hat{V}_t \hat{V}_t'.$$

The user must also pass some moments related to the cross-covariances between Z_t or X_t on the one hand and η_t on the other. We supply separate function calls for (1) η_{t+q} i.i.d., and (2) $\eta_{t+q} \sim \text{MA}(q)$. The first is a special case of the second.

Let $\hat{\eta}_{t+q}$ be the least squares residuals. Let Z_t be the vector of variables in the companion form VAR. The two separate function calls are:

- (1) $q=0$ and $\eta_t \sim \text{i.i.d.}$: The user needs to compute and pass an estimate of $E\eta_t Z_t'$, called **EetaZtwid0**. In example (16) and (17), in which $Z_t' = (y_t, y_{t-1}, x_t, x_{t-1})$,

$$\mathbf{EetaZtwid0} = \left(T^{-1} \sum \hat{\eta}_t y_t, T^{-1} \sum \hat{\eta}_t y_{t-1}, T^{-1} \sum \hat{\eta}_t x_t, T^{-1} \sum \hat{\eta}_t x_{t-1} \right).$$

(To prevent misunderstanding: yes, in this example $T^{-1} \sum \hat{\eta}_t y_{t-1} = 0$ and $T^{-1} \sum \hat{\eta}_t x_{t-1} = 0$ by construction, since least square residuals are orthogonal to the regressors.)

(2) $\eta_{t+q} \sim \text{MA}(q)$: The user passes the integer parameter **nq** (the value of q) as well as (i) a matrix **EetaZtwid** and (ii) a vector **EXeta**. (i) and (ii) are defined as follows:

(i) **EetaZtwid** is a matrix of dimension $(q+1) \times n_Z$. In this matrix, for $i=0, \dots, q$, the $(i+1)$ st row is a $1 \times \mathbf{nZtwid}$ estimate of $E\eta_{t+q} Z'_{t+i}$. The estimate can be computed by the user as

$$\text{estimate of } E\eta_{t+q} Z'_{t+i} = T^{-1} \sum \hat{\eta}_{t+q} Z'_{t+i}.$$

(ii) **EXeta** is an estimate of the $\mathbf{nk} \times 1$ vector $E(X_t + X_{t+1} + \dots + X_{t+q-1})\eta_{t+q}$. The estimate can be computed by the user as

$$\mathbf{EXeta} = T^{-1} \sum (X_t + X_{t+1} + \dots + X_{t+q-1}) \hat{\eta}_{t+q}.$$

Here is the syntax to invoke the functions. The functions return the $\mathbf{nk} \times 1$ estimate of \hat{b} , called **vbias**.

• $\eta_t \sim \text{i.i.d.}$: (19a)
`vbias <- proc_vb_ma0(nZtwid, phitwid, omegaUtwid, nk, PX, EetaZtwid0)`

• $\eta_t \sim \text{MA}(q)$: (19b)
`vbias <- proc_vb_maq(nZtwid, phitwid, omegaUtwid, nk, PX, EetaZtwid, EXeta, nq).`

Relative to the procedure used when $\eta_t \sim \text{i.i.d.}$, the procedure for $\eta_t \sim \text{MA}(q)$ requires that the user change one parameter (**EetaZtwid0** \rightarrow **EetaZtwid**) and include two additional parameters (**EXeta** and **nq**). See [Tables 3](#) and [4](#).

After obtaining **vbias** $\equiv \hat{b}$ from either procedure, the user must divide by sample size T , to obtain bias adjusted estimate $= \hat{\beta} - \frac{\hat{b}}{T}$.

4 R code for an empirical application

[Table 5](#) has R code for an application using **longhor1**. It estimates (4) with lag length $k=2$ and horizon $q+1=12$.

TABLE 3 Function `proc_vb_ma0`

```
vbias <- proc_vb_ma0(nZtwid, phitwid, omegaUtwid, nk, PX,
EetaZtwid0)
```

- In (1), $q=0$ and the regression disturbance $\eta_t \sim i. d.$ The right hand side variables in the regression are not restricted to be a constant and lags of a single variable.
- The user has estimated an auxiliary regression, written in companion form as

$$Z_t - EZ_t = \Phi (Z_{t-1} - EZ_{t-1}) + U_t, \quad \Omega_U = EU_t U_t', \quad X_t = P_X Z_t.$$

$n_Z \times 1$ $n_Z \times n_Z$ $n_Z \times 1$ $n_Z \times n_Z$ $k \times 1$ $k \times n_Z$ $n_Z \times 1$

This regression produces an approximately white noise disturbance U_t .

- The user has also estimated (1), yielding least squares residuals $\{\hat{\eta}_t\}$.

Passed by user

nZtwid	n_Z : integer number of variables in the auxiliary VAR (13)
phitwid	$\hat{\Phi}$: matrix of autoregressive estimates in (13)
omegaUtwid	$\hat{\Omega}_U$: matrix of estimates of the variance–covariance matrix in (13)
nk	k : integer number of variables in regression (1)
PX	P_X : matrix selecting from Z_t the right hand side variables in the regression of interest in (13) ($P_X = I$ is possible)
EetaZtwid0	vector estimate of $E\eta_t Z_t'$, i.e., $T^{-1} \sum \hat{\eta}_t Z_t'$

Returned to user

vbias	$nk \times 1$ vector: \hat{b} = estimate of $k \times 1$ numerator of bias to order T^{-1}
--------------	--

Functions invoked by **proc_vb_ma0**: **proc_vbias**.

TABLE 4 Function `proc_vb_maq`

```
vbias <- proc_vb_maq(nZtwid, phitwid, omegaUtwid, nk, PX,
EetaZtwid, EXeta, nq)
```

- In (1), q may be any integer and the regression disturbance $\eta_{t+q} \sim MA(q)$. The right hand side variables in the regression are not restricted to be a constant and lags of a single variable.
- The user has estimated an auxiliary regression, written in companion form as

$$Z_t - EZ_t = \Phi (Z_{t-1} - EZ_{t-1}) + U_t, \quad \Omega_U = EU_t U_t', \quad X_t = P_X Z_t.$$

$n_Z \times 1$ $n_Z \times n_Z$ $n_Z \times 1$ $n_Z \times n_Z$ $k \times 1$ $k \times n_Z$ $n_Z \times 1$

This regression produces an approximately white noise disturbance U_t .

- The user has also estimated (1), yielding least squares residuals $\{\hat{\eta}_{t+q}\}$.

Continued

TABLE 4 Function `proc_vb_maq`—Cont'd

Passed by user	
nZtwid	n_Z : integer number of variables in the auxiliary VAR (13)
phitwid	$\hat{\Phi}$: matrix of autoregressive estimates in (13)
omegaUtwid	$\hat{\Omega}_U$: matrix of estimates of the variance–covariance matrix in (13)
nk	k : integer number of variables in regression (1)
PX	P_X : matrix selecting from Z_t the right hand side variables in the regression of interest in (13) ($P_X = I$ is possible)
EetaZtwid	estimates of $E\eta_{t+q}Z'_{t+i-1}$, $i=1$ to $q+1$. Matrix of dimension $(nq+1) \times n_Z$ (i.e., of dimension $(q+1) \times n_Z$). Row $i+1$ has the estimate of $E\eta_{t+q}Z'_{t+i}$, e.g., the first row of EetaZtwid is $T^{-1}\sum\hat{\eta}_{t+q}Z'_t$
EXeta	vector estimate of $E(X_t + X_{t+1} + \dots + X_{t+q-1})\eta_{t+q}$, i.e., $T^{-1}\sum(X_t + X_{t+1} + \dots + X_{t+q-1})\hat{\eta}_{t+q}$
nq	integer value of q in (1)
Returned to user	
vbias	$nk \times 1$ vector: \hat{b} = estimate of $k \times 1$ numerator of bias to order T^{-1}
Functions invoked by <code>proc_vb_maq</code> : <code>proc_vbias</code> .	

TABLE 5 R code to illustrating use of `longhor1`

```

rm(list=ls())
library(MASS)
library(expm)
source("lagmatrix.R")
source("proc_vbias.R")
source("proc_vb_ma0.R")
source("proc_vb_maq.R")
source("UtilFunc_OLS.R")
source("longhor.R")
source("longhor1.R")

# Parameterwe Setting
T <- 240 # sample size
ARp <- 2 # Order of AR
q_horizon <- 11 # Forecast horizon is q+1

```

TABLE 5 R code to illustrating use of longhor1—Cont'd

```
#####
#insert code here to read data into "test data" from 1 to N, with
N>=T+q_horizon+Arp
# <code to read in data>
#####

# OLS AR coefficients bias correction
result <- longhor1(yseries=test_data, xseries=test_data,
first=Arp+q_horizon+1,
                    last=Arp+q_horizon+T, nq=q_horizon, nk=Arp,
                    narlag=Arp)
vbias <- result[[1]]; betahat <- result[[2]]; betahat_adj
  <- result[[3]]
print(vbias)
print(betahat)
print(betahat_adj)
```

Acknowledgment

We thank Ziqi Chen for assistance in preparing the manuscript.

References

- Box, G.E.P., Jenkins, G.M., 1976. *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco.
- Crone, T.M., Khettry, N.K., Mester, L.J., Novak, J.A., 2013. Core measures of inflation as predictors of total inflation. *J. Money Credit Bank.* 45 (2), 505–519.
- Engsted, T., Pedersen, T.Q., 2014. Bias-correction in vector autoregressive models: a simulation study. *Econometrics* 2, 45–71.
- Hjalmarsson, E., 2011. New methods for inference in long-horizon regressions. *J. Financ. Quant. Anal.* 46, 815–839.
- Ing, C., 2003. Multistep prediction in autoregressive processes. *Economet. Theor.* 254–279.
- Kendall, M.G., 1954. Note on bias in the estimation of autocorrelation. *Biometrika* 41, 403–404.
- Lunsford, K.G., West, K.D., 2017. Some evidence on secular drivers of U.S. safe real rates. In: *Federal Reserve Bank of Cleveland Working Paper*, pp. 17–23.
- Marcellino, M., Stock, J.H., Watson, M.W., 2006. A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *J. Econ.* 135, 499–526.
- Mark, N.C., 1995. Exchange rates and fundamentals: evidence on long-horizon predictability. *Am. Econ. Rev.* 85 (1), 201–218.

- Shaman, P., Stine, R.A., 1988. The bias of autoregressive coefficient estimator. *J. Am. Stat. Assoc.* 83, 842–848.
- Snaith, S., Coakley, J., Kellard, N., 2013. Does the forward premium puzzle disappear over the horizon? *J. Bank. Financ.* 37, 3681–3693.
- West, K.D., 2016. Approximate Bias in time series regression. Manuscript. In: University of Wisconsin.
- West, K.D., Zhao, Z., 2018. Improving Forecasts with Bias Adjustment. Manuscript in preparation. University of Wisconsin and University of Notre Dame.