**Which Unit Root Tests Are Better for Univariate Time Series Forecasts?**

Last Updated: November 27, 2021

**Adam Check**
Assistant Professor
Department of Economics
University of St. Thomas
Mail OEC 5029
2115 Summit Ave.
St. Paul, MN 55105
ajc@stthomas.edu

**Ming Chien Lo** (Corresponding Author)
Associate Professor
Department of Economics and Finance
College of Management (Minneapolis Campus)
Metropolitan State University
1300 Harmon Place
Minneapolis, MN 55403
Email: ming.lo@metrostate.edu

**Kwok Ping Tsang**
Associate Professor
Department of Economics
College of Science
Virginia Tech
3016 Pamplin Hall, Mail Code 0316
880 West Campus Drive
Blacksburg, VA 24061
Email: byront@vt.edu

**Abstract.** In this paper, we consider univariate forecasts made when using stationary, near unit root, and unit root data. Like Diebold and Kilian (2000), we conduct a Monte Carlo experiment investigating the usefulness of unit root tests prior to forming univariate forecasts. In our experiment, we consider more than one unit root test and also vary the order of integration in the time series. We find that unit root tests are indeed useful for forecasting, especially when the series has a large number of in-sample observations. However, the choice of unit test matters. Using root mean square error as a criterion for forecast performance, we find that the Philips-Perron test has an edge over the augmented Dickey-Fuller test and the Kwiatkowski–Phillips–Schmidt–Shin test. We recommend practitioners to be mindful of the choice of test, as the KPSS test is the default used in the *forecast* package in R, following Hyndman and Khandakar (2008), but the Philips-Perron test is available as an option in that package.

Key Words: Augmented Dickey-Fuller; KPSS; Philips-Perron; Forecasting Algorithm; Monte Carlo; Unit Root Test

## 1. Introduction

Unit root tests have been used for decades to determine if a given time-series is stationary (see Wolters and Hassler (2006) for an overview of the history). There are several reasons why econometricians want to identify whether a time series is stationary. Stationarity implies mean reversion, and as such the presence of stationarity can support certain theories. For example, distinguishing between stationary and unit root behavior in the real exchange rate can provide evidence for or against purchasing power parity. Distinguishing between stationary and non-stationary data is also important outside of theoretical models. For example, the use of non-stationary data in linear regression may result in spurious regressions (Ventosa-Santaulària (2009)), while spurious regression can be avoided if the regressors are stationary.

Distinguishing between stationary and non-stationary data is also important for forecasters. Forecasts made assuming stationary data will often be vastly different than those made under the assumption that the data is non-stationary, especially at long forecast horizons. Therefore, researchers like Diebold and Kilian (2000) have suggested that unit root tests are useful in forecasting. Their Monte Carlo experiment pitches a pre-test method, in which a series is tested for a unit root, and if found, differenced, against an $AR(1)$ model and a random walk with drift model. They find that the augmented Dickey-Fuller test is useful in informing forecasters when to first-difference the data, especially when the evidence is strongly against the presence of a unit root (i.e. the series does not have near unit root behavior).

In later work, Hyndman and Khandakar (2008, hereafter HK) develop an algorithm that automates a procedure that extends the method advocated by Diebold and Kilian (2000). It first uses a unit root test— Kwiatkowski–Phillips–Schmidt–Shin (KPSS) by default, or augmented Dickey-Fuller (ADF) and Philips-Perron (PP) as options—before applying the corresponding ARIMA model. This paper extends the idea of Diebold and Kilian (2000) to compare which unit root test results in better forecasting performance.[1] To be specific, we create a Monte Carlo exercise assuming certain linear data generating processes (DGP) ranging from $I(0)$ to $I(2)$, and use these simulated series to compare the HK approach against a model averaging approach (AVG) that assumes an I(0), I(1), or I(2) process with equal probability.[2] If a unit root test helps determine the order of

[1] Diebold and Kilian (2000) is more specific; they have the real GDP series in mind in their simulation. They also explicitly assume a linear time trend.
[2] Therefore, unit root tests are absent in AVG.

integration in HK, an AVG with equal weights on $I(0)$, $I(1)$ and $I(2)$ can be seen as one without utilizing such posterior information. Our choice to use HK is partially a matter of convenience, because Hyndman and Khandakar have developed and have been updating the popular and widely used *forecast* package in R, which includes the HK algorithm. Therefore, our results also provide guidance on how to use the package effectively.[3]

We find that unit root tests are useful, but not equally so. Not surprisingly, unit root tests are particularly useful when the sample size is large, since the tests each have well-documented issues that arise in small samples. We find that, in general, the Philips-Perron test is more helpful in improving forecast performance than the default KPSS used in the R package. Practitioners should be mindful of the choice of unit root test when dealing with real world data.

Section 2 briefly explains the models. Section 3 discusses our Monte Carlo design, the data generating processes and our out-of-sample forecasting exercises. Section 4 reports and discusses the Monte Carlo results. We summarize this study and spell out both caveats and possible future research in Section 5.

## 2. The Models

The HK approach is standard, and details can be found in Hyndman and Athanasopoulos (2018). In order to approximate the experience of most practitioners, we keep the default HK settings from the R package, with a few exceptions.[4] In a nutshell, the HK algorithm first uses a unit root test – the KPSS test is the default – to determine the order of integration and how many times a series needs to be differenced to rendered stationary before applying ARIMA. The maximum order of integration is set at 2 by default. Then, based on the results of the unit root test, the HK algorithm fits one of the three following types of ARIMA models: $ARIMA(p,0,q)$, $ARIMA(p,1,q)$ , or $ARIMA(p,2,q)$. Conditioning on the result from the unit root test using the default 5% level of significance, HK commits fully—that is, a 100% weight—on one of these three levels of

[3] To get a sense of the popularity, albeit somewhat unscientifically, we conducted several Googles searches on June 15, 2021. Keywords "forecast package" generates 85.6 million results and "forecast package in R" 39.3 million. The specific function that executes the HK algorithm is auto.arima. Keywords "auto.arima" generates 2.7 million.
[4] When forecasters approach real world data that are not simulated, and the default setting is more likely to be deployed. Later research can further investigate whether results are robust against changes in the setting.

differencing. Next, an information criterion is used to select the order of autoregression ($p$) and moving average ($q$).

Differencing after pre-testing for a unit root is a convenient but not always appropriate method of rendering a non-stationary series stationary. Size distortion, lack of power, the presence of outliers or structural breaks, etc. can all affect the results of unit root tests. For example, consider a single simulated time-series used in an out-of-sample forecasting exercise. Even on a single time-series we find that the unit root tests can behave erratically as the sample length increases (i.e., as time progresses), even though our simulated data is a linear ARIMA process throughout the sample. One possible alternative to pre-testing, which we explore, is to not rely on any unit root test at all. Our AVG approach consists of first imposing each level of differencing (0, 1, or 2), then, conditional on the difference, using the second step of the HK algorithm to determine the order of autoregression ($p$) and moving average ($q$). Finally, we assign 1/3 weight to each of the forecasts from these three ARIMA models.

## 3. The Monte Carlo and Out-of-Sample Forecast Design

*Data Generating Processes*

For simplicity, we use an AR(1) model. The first DGP group assumes an autoregressive structure with the possibility of a random walk when the AR(1) parameter equals one.

$$y_t = a + by_{t-1} + \varepsilon_t. \tag{1}$$

For simplicity, we restrict $a = 0$. We allow $b \in \{0.90, 0.95, 0.975, 0.99, 1.00\}$. Focusing on $b \geq 0.9$ covers the area in which a unit root test may have either size distortion or lack of power especially when the sample is small. Many macroeconomic series that exhibit slow mean reversion behavior, such as the unemployment rate or the real exchange rate, can be represented by this DGP. Other than the case $b = 1$, DGP (1) can be described as a process that is persistent yet stationary. When $b = 0.975$ or $0.99$, we have a near unit root process.

The second DGP assumes a similar structure, but for the first difference,

$$\Delta y_t = a + (b-1)\Delta y_{t-1} + \eta_t \tag{2}$$

which can be written as

$$y_t = a + by_{t-1} - (b-1)y_{t-2} + \eta_t. \quad (3)$$

We restrict $a = 0$ and allow $b \in \{1.50, 2.00\}$. When $b = 2$, the series is $I(2)$. When $b = 1.5$, the series is fractionally integrated between $I(0)$ and $I(2)$. Note that if $b = 1$, both (1) and (3) result in a random walk model without drift.

DGP's (1) and (2) with the set of possible values of $b$ cover a plausible range for many financial and economic data sets. Particularly relevant for DGP (2) is that Caporale, Gil-Alana and Plastun (2019) and Hartl, Tschernig and Weber (2021) both find that series with an order of integration above one are more common than previously thought.

To simulate data from either DGP, given a number of observations $N$, we simulate $1.1 \times N$ observations $y_t$ according to (1) and (2) respectively and trim off the first 10%. We fix the conditional standard deviation of the processes, by setting the standard deviations of the shocks $\varepsilon_t$ and $\eta_t$ to 1. For both DGPs, we consider $N \in \{50, 200, 500\}$. For each choice of $N$, we perform an expanding window forecasting exercise and consider various $k$-period-ahead point-forecasts, where $k \in \{1,3,6,12\}$. Because the number of observations is small when $N = 50$, we do not consider $k = 12$ for that case.

*Out-of-Sample Forecasting*

As mentioned above, we conduct an expanding window pseudo-out-of-sample forecasting exercise similar to that of Meese and Rogoff (1983). For any simulated series, we start with an estimation period that starts at the first observation and ends at $0.6Nth$ observation (i.e., the initial estimation period covers the first 60% of the sample). Then, we use a statistical model to form the $k$-period-ahead point-forecast. Next, we add one additional observation, and repeat this process. We continue to expand the number of observations for estimation by one at each iteration until all data points are exhausted. The set of $k$-horizon point-forecasts are then compared against the actual data using root mean squared errors (RMSE).

*The Monte Carlo Design*

All together, given the varying values of the AR(1) parameter, $b$; the sample length, $N$; and forecast horizon, $k$, we have 77 sets of experiments. For each set, 1,000 series are simulated. Appendix Table 1 shows the numbering of our sets and the varying values of the parameters. To get a sense of our simulated data, Figure 1 shows the series generated from DGP (1) and (2) for each possible value of $b$ when $N = 500$.

*Settings in auto.arima() and Arima()*

- We use the following default options:
  - The maximum possible order of integration is 2.
  - The level of significance for the unit root tests in auto.arima() is five percent.[5]
  - The information criterion used when selecting the ARMA(p,q) order is the corrected Akaike Information Criterion (AICc).
- Since we know we are using a (possibly integrated) AR(1) model, we set:
  - The combined maximum order of p and q to 3 for auto.arima(); i.e. max.order=3.
  - We do not consider seasonal ARIMA models; i.e. we set seasonal=FALSE.

## 4. Results

After simulating data and forming forecasts for each of the 77 sets, we compare forecast accuracy across four models: (1) HK with the KPSS test (default), (2) HK with the ADF test, (3) HK with the PP test, (4) the model average ("AVG") described in section 2.

To measure forecast accuracy, we compute the Monte Carlo mean RMSE from the 1,000 simulations for each model:

$$\overline{RMSE}_m = \frac{\sum_{i=1}^{1000} RMSE_{m,i}}{1000} \tag{4}$$

and rank them to determine the winner in each set, where $i$ is the $i$-th simulation in the set and $m$ denotes the model. Models resulting in a lower mean RMSE have better forecast performance.

[5] Also assumed in Hyndman and Athanasopoulos (2018). In recent years, such a default has been questioned (e.g. Ziliak and McCloskey, 2007; Wasserstein and Lazar, 2016; Amrhein, Greenland and McShane, 2019).

To get a sense of relative differences in RMSEs, we compute a few additional measures. Since we fix the conditional SD of the error terms in the simulations, the unconditional SD, which also depends on the value of b, differs across different generated time-series. To make it easier to compare *relative* performance across different simulations and forecast horizons, we first compute the relative mean RMSE for each simulation which is given by:

$$Relative\ RMSE = \frac{\overline{RMSE}_m}{\overline{RMSE}_{Winner}} \tag{5}$$

where *Winner* denotes the model with the lowest absolute RMSE (given in equation (4)). Next , for a given experiment, we compute the average of the relative RMSE across each of the 1,000 simulations:

$$Mean\ Ratio = \sum_{i=1}^{1000} \frac{RMSE_{m,i}}{RMSE_{Winner,i}} \tag{6}$$

Further, we can compute a $p$-value based using $RMSE_{m,i}/RMSE_{Winner,i}$,

$$p - value = \sum_{i=1}^{1000} I\left(\frac{RMSE_{m,i}}{RMSE_{Winner,i}} \leq 1\right)/1000 \tag{7}$$

Depending on the mean ratio and the p-value may result in different rankings that the mean absolute RMSE given in (4).

The ranking based on (4) and other summary statistics are reported in Appendix Table 2. From the table, we observe that:

(i) PP has the lowest mean RMSE in 37 out of the 77 sets, followed by KPSS with 23 and ADF with 14. AVG only has 3. In fact, each of the three HK models considered (i.e., HK with any unit root test) beats AVG in 64 of the 77 sets. Using this criterion, we can conclude that the unit root tests are useful. Surprisingly, these results also suggest that in many cases, using the PP test generates better forecast performance than does using the default KPSS test.

(ii) The $p$-values among the top three models in any given set is in general high. When $N = 500$, the RMSEs from the last place model is consistently larger than that from the winner.

Because AVG is placed last 64 out of 77 times, we can conclude – perhaps unsurprisingly – that the unit root tests are especially useful in large samples.

(iii) We observe larger gaps in relative RMSE (5) and mean ratio (6) between models when the order of integration increases. Therefore, it seems that as the order of integration increases, it becomes more important to accurately identify the presence of integration.

As an alternative form of summary, we analyze the 77 outcomes using a multinomial logit model where the dependent variable is the winning model (KPSS, ADF, PP, and model averaging) and the independent variables are the sample size, horizon, and the integration order. Without loss of generality, we use KPSS as the baseline model.

The results are reported in Table 1, and they are consistent with our findings above. First, when we control for the sample size, the forecast horizon and the order of integration, PP is frequently superior to KPSS.[6] In the second row, we see that the estimate on the order of integration is relatively large, at -1.051. While the p-value is also fairly high (above 10%), the large point estimate suggests that the advantage from using the PP test (rather than the KPSS test) may decline somewhat when the order of integration is high.

Second, at first glance, the ADF test also appears to out-perform the KPSS test (since, like the PP test, it also has a large and significant constant term). However, it does not fare well when the order of integration is one or higher. With an estimate of -3.766, it offsets the positive constant term (3.987) when the order of integration is one, and more than offsets it when the order of integration is 1.5 or 2.

Finally, our model average of I(0), I(1), and I(2) models, which we call AVG, is clearly inferior to the default KPSS test. Even though the constant term is estimated at a positive value of 3.715, it is offset by the negative estimate for the parameter for $N$—even when $N = 50$ which is the smallest in our Monte Carlo, the product of -0.111 and 50 is -5.55, more than offsetting the positive constant. When $N$ is 200 or 500, the negative impact would become even larger. This result once again implies that unit root tests are useful and become increasingly so as the sample size increases.

**5. Conclusion**

[6] This can be seen in first column of the second row, since there is a positive constant (and small p-value) associated with the PP test.

We use the *forecast* package in R, specifically the auto.arima() function, to examine the forecasting performance of various approaches to unit-root testing in a Monte Carlo exercise. We are particularly interested in the Hyndman-Khandakar algorithm as it makes use of unit root tests to determine the appropriate number of differences to ensure stationarity before selecting an ARIMA model and forecasting. We have shown that both the HK algorithm and the unit root tests are useful for forecasting.

Our results constitute an important contribution to this literature. Previously, Diebold and Kilian (2000) examined the performance of the ADF test for forecasting in a relatively narrow context. Our study includes other unit root tests as well as a model-averaging alternative that represents ignorance of the test results. We also generalize the Monte Carlo exercise by considering different orders of integration, as well as near unit root processes. Our results imply that the Philips-Perron test may generate better forecast than the KPSS or ADF tests when the DGP is linear and the errors are normally distributed.

For forecast practitioners, our results suggest that relying solely on the the KPSS test may result in worse forecast performance for near-unit root, I(1), fractionally integrated, or I(2) processes, and that using the PP test for these types of series would lead to an increase in expected forecast accuracy. Of course, while the patterns of integration listed above capture many real-world time-series, many other real-world series feature nonlinearities or outliers in addition to (or in lieu of) these patterns of integration. We leave the study of the performance of unit root tests and the HK algorithm in these more complicated environments to future research.

**References**

Amrhein, V., Greenland, S. & McShane, B. (2019). Scientists Rise Up Against Statistical Significance. Nature, 567, 305-307.

Caporale, G.M., Gil-Alana, L. & Plastun, A. (2019). Long Memory and Data Frequency in Financial Markets. Journal of Statistical Computation and Simulation, 89, 1763-1779.

Diebold, F.X. & Kilian, L. (2000). Unit Root Tests Are Useful For Selecting Forecasting Models. Journal of Business and Economic Statistics, 18, 265-273.

Hartl, T., Tschernig, R. & Weber, E. (2021). Solving the Unobserved Components Puzzle: A Fractional Approach to Measuring the Business Cycle. University of Regensburg Working Paper.

Hyndman, R.J. & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd Edition). https://otexts.com/fpp2/.

Hyndman, R.J. & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, 27. https://www.jstatsoft.org

Meese, R.A. & Rogoff, K. (1983). Empirical Exchange Rate Models of the Seventies: Do They Fit Out of Sample? Journal of International Economics, 14, 3-24.

Wasserstein, R.L. & Lazar, N.L. (2016). The ASA Statement on p-Values: Context, Process, and Purpose. The American Statistician, p.129-133.

Wolters, J. & Hassler, U. (2006). Unit Root Testing. Allgemeines Statistisches Archiv, 50, 43-58.

Ventosa-Santaulària, D. (2009). Spurious Regression. Journal of Probability and Statistics. DOI:10.1155/2009/802975

Ziliiak, S.T. & McCloskey, D.N. (2007). The Cult of Statistical Significance: How the Standard Error Costs Us Jobs, Justice, and Lives. The University of Michigan Press.

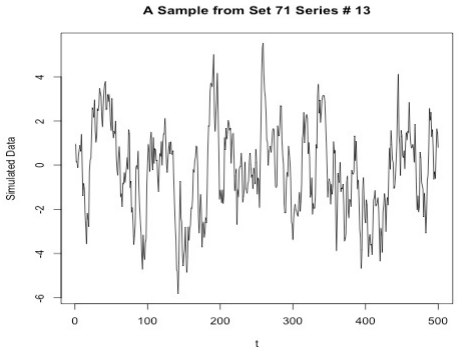
**Table 1: Summary Results from a Multinomial LOGIT Regression**

| | Constant | Sample Size | Horizon | Integration |
|---|---|---|---|---|
| ADF | 3.987*** | -0.002 | 0.047 | -3.766*** |
| | (1.200) | (0.019) | (0.095) | (0.656) |
| PP | 2.266** | -0.002 | 0.030 | -1.051 |
| | (1.081) | (0.002) | (0.071) | (0.676) |
| AVG | 3.715** | -0.111*** | 0.024 | 1.076 |
| | (1.638) | (0.007) | (0.336) | (0.971) |
| Pseudo $R^2$ | 0.118 | | | |

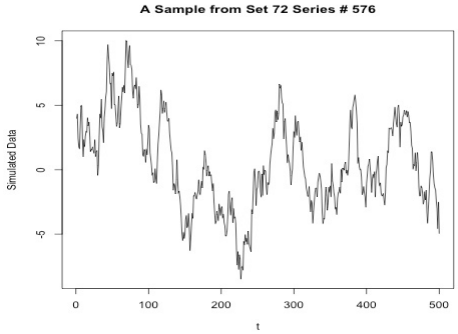
Robust standard errors reported in parentheses. *** p-value below 1% with null hypothesis that the parameter is equal to zero.

**Figure 1: Sample Simulated Series**

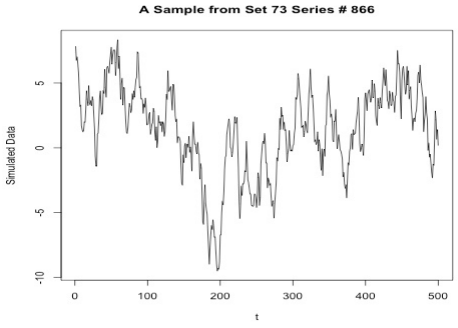
b = 0.900



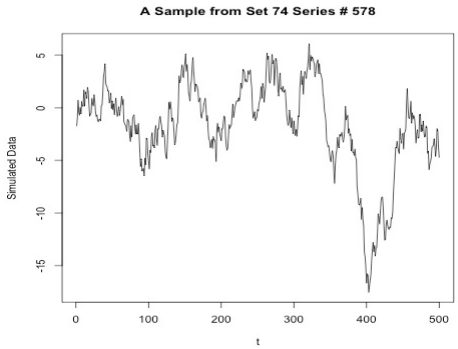

b = 0.950



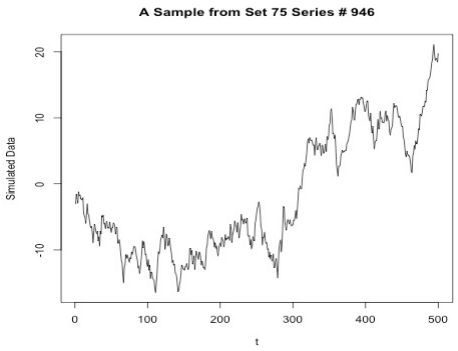

b = 0.975



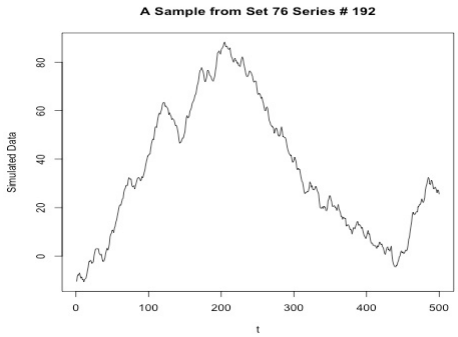

b = 0.990




b = 1.000




b = 1.500




b = 2.000