



*The World's Largest Open Access Agricultural & Applied Economics Digital Library*

**This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.**

**Help ensure our sustainability.**

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

[aesearch@umn.edu](mailto:aesearch@umn.edu)

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

*No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.*

# kpsstest: A command that implements the Kwiatkowski, Phillips, Schmidt, and Shin test with sample-specific critical values and reports p-values

Ali Kagalwala  
Texas A&M University  
College Station, TX  
alikagalwala@tamu.edu

**Abstract.** Commonly used unit-root tests in time-series analysis—such as the Dickey–Fuller and Phillips–Perron tests—use a null hypothesis that the series contains a unit root. Such tests have low power against the alternative—when a time series is near integrated or highly autoregressive—implying that they do poorly in distinguishing such a series from having a unit root. Kwiatkowski et al. (1992, *Journal of Econometrics* 54: 159–178) introduced the Kwiatkowski, Phillips, Schmidt, and Shin test, in which the null hypothesis is that the series is stationary, to deal with this problem. One shortcoming of the presently available Kwiatkowski, Phillips, Schmidt, and Shin test in Stata is that it uses asymptotic critical values regardless of the sample size. This poses a problem in that researchers—especially social scientists—are often presented with short time series. I introduce **kpsstest**, a command that extends the previous implementation by including an option for a zero-mean-stationary null hypothesis, generating sample- and test-specific critical values, and reporting appropriate *p*-values.

**Keywords:** st0671, kpsstest, KPSS, GKPSS, unit-root tests, stationary null hypothesis, time series

## 1 Introduction

Commonly used unit-root tests in time-series analysis—such as the Dickey–Fuller and Phillips–Perron tests—are problematic because they perform poorly in distinguishing near-integrated or highly autoregressive series from unit roots. This implies that researchers, when presented with such series, would fail to conclude that these series are stationary when using the Dickey–Fuller and Phillips–Perron tests. That is, they would fail to reject the null hypothesis that the series contains a unit root. Failing to reject the null hypothesis does not necessarily imply that the null is true. It simply indicates that there exists an absence of strong evidence against the null hypothesis (Kwiatkowski et al. 1992).

To increase the information available to researchers about the time-series properties of their data, Kwiatkowski et al. (1992) introduced the Kwiatkowski, Phillips, Schmidt, and Shin (KPSS) test. The null hypothesis of the KPSS test is either that the series is

trend stationary or that the series is level stationary. Kwiatkowski et al. (1992) recommended that the test be used in conjunction with unit-root tests whose null hypotheses are that the series contains a unit root. In their application, using economic time-series variables, Kwiatkowski et al. (1992) demonstrated that their test fails to reject the null hypothesis of stationarity and the Dickey–Fuller test fails to detect that the series are stationary. Thus, while researchers may use unit-root tests such as the Dickey–Fuller test, they should confirm their results with the KPSS test. This is especially important with highly autoregressive series, for which the Dickey–Fuller test has low power.

The KPSS test and its current implementation in Stata are wanting. Specifically, the implementation of the KPSS procedure uses asymptotic critical values—presented by Kwiatkowski et al. (1992, 166, table 1)—to test for significance of the KPSS test statistic. However, researchers are frequently presented with short time series or, at best, those that are unlikely to be considered infinite. Consequently, if the finite-sample distribution of the KPSS test statistic differs from its asymptotic distribution, then using asymptotic critical values with a finite sample poses threats to inference. A second shortcoming of the KPSS test and the current implementation of the test in Stata is the absence of a no-constant or zero-mean-stationary null hypothesis (Hobijn, Franses, and Ooms 2004). Third, while the current version of the KPSS test in Stata allows for the generalized Kwiatkowski, Phillips, Schmidt, and Shin (GKPSS) test, it does not provide GKPSS-specific critical values (Baum 2000). Finally, Baum’s (2000) `kpss` command does not provide users with *p*-values.

I introduce `kpsstest` to deal with these shortcomings. `kpsstest` provides users with finite-sample critical values from response surface regressions for any sample size between 10 and 400 observations for the KPSS and GKPSS tests and for all three null hypotheses options. For sample sizes larger than 400, `kpsstest` uses asymptotic critical values. Additionally, `kpsstest` uses the distribution of the KPSS test statistic for a given sample size to calculate *p*-values. Similar to `kpss`, `kpsstest` provides users with the option to implement either the original KPSS test or the GKPSS test (Hobijn, Franses, and Ooms 2004). `kpsstest` extends the capabilities of `kpss` by allowing users to implement the test against a null hypothesis of no constant or zero mean stationarity. Hobijn, Franses, and Ooms (2004) demonstrate that the distribution of this test statistic differs from the distribution of the test statistic when the null hypothesis is level stationary.

## 2 An overview of the KPSS test

### 2.1 The original KPSS test

In this section, I provide a brief overview of the original KPSS test developed by Kwiatkowski et al. (1992). A more technical description of the test can be found in the appendix. Consider a time series,  $y_t$ , with trend ( $t$ ) and random-walk ( $r_t$ ) components, such that

$$\begin{aligned} y_t &= \alpha + \xi t + dr_t + \epsilon_t, & \text{where } \epsilon_t &\sim N(0, \sigma_\epsilon^2) \\ r_t &= r_{t-1} + u_t, & \text{where } u_t &\sim N(0, \sigma_u^2) \end{aligned} \quad (1)$$

where  $\epsilon_t$  are stationary errors. The KPSS test is essentially a Lagrange multiplier test with a null hypothesis,  $H_o: \sigma_u^2 = 0$ , which means the series is stationary. Specifically, 1) when  $\xi$  is restricted to 0, the null hypothesis is that the series is level stationary; 2) when  $\alpha$ ,  $\xi$ , and  $d$  are all restricted to 0, the null hypothesis is that the series is zero mean stationary; and 3) when  $\alpha$ ,  $\xi$ , and  $d$  are not restricted to 0, the null hypothesis is that the series is trend stationary. Rejecting the null hypothesis means that the series is nonstationary and, if deemed necessary by the researcher, appropriate corrections (for example, differencing, or removing the trend) need to be made to account for the nonstationarity of the series.

## 2.2 The GKPSS test

Before I describe the GKPSS test, note that one step in calculating the KPSS test statistic requires estimating  $\sigma_\epsilon^2$ . And a component of the estimator of  $\sigma_\epsilon^2$  is the temporal dependence in the residuals—which needs to be weighted—from a regression of the series on a constant or on the trend term. For a zero-mean-stationary null hypothesis, the temporal dependence in the series itself needs to be weighted. To weight this temporal dependence, one needs to specify a kernel,  $w()$ . Using a kernel in this test also requires that one must specify the maximum number of lags of the residuals or series to be accounted for in the temporal dependence of residuals or series. This quantity is known as the lag truncation parameter. The original KPSS test uses a Bartlett kernel and a long or short Schwert criterion calculation to determine the appropriate lag truncation parameter. Readers interested in the technical description of these quantities may refer to the appendix. However, a technical understanding beyond the description in this paragraph is not necessary to understand the differences between the KPSS and GKPSS tests and which test to implement when testing for stationarity.

The GKPSS test builds on the KPSS test by using a different kernel and algorithm for calculating the lag truncation parameter (Hobijn, Franses, and Ooms 2004). The GKPSS test uses a quadric spectral kernel with an automatic bandwidth selection algorithm in determining the lag truncation parameter. The most important difference between the KPSS and GKPSS tests is realized in the statistical properties of the tests: although both tests are consistent, the GKPSS test has lower type 1 error rates than the KPSS test in finite samples.

## 2.3 Implementing the tests

The main differences between the KPSS and GKPSS tests are in the choice of kernel and the method in which the lag truncation parameter is determined. This leads to the question: which test should a researcher use?

As I mentioned, the KPSS test is found to have higher type 1 error rates for highly autoregressive series, whereas the GKPSS test, by using a quadratic spectral kernel and an automatic bandwidth selection algorithm for the lag truncation parameter, has a lower incidence of type 1 errors in finite samples (Hobijn, Franses, and Ooms 2004). Despite the superiority of the GKPSS test over the KPSS test, **kpsstest** allows users to implement the original KPSS test with a Bartlett kernel in combination with a user-defined lag truncation parameter, a short or long Schwert criterion, or an automatic bandwidth selection for selecting the lag truncation parameter.<sup>1</sup> Kwiatkowski et al. (1992, 170) used the Schwert criterion to select the lag truncation parameter because these were the choices being commonly made at the time. As Hobijn, Franses, and Ooms (2004) note about Kwiatkowski et al.'s (1992, 494) choice of the lag truncation parameter, "the automatic data dependent bandwidth selection procedure yields better small sample results than when the bandwidth is chosen arbitrarily, as by KPSS." The user-defined lag truncation parameter option in **kpsstest** is for advanced practitioners who wish to implement the KPSS test with lags set manually. However, the literature demonstrates that the GKPSS test—that is, the KPSS test implemented with a quadratic spectral kernel and an automatic bandwidth selection algorithm for the lag truncation parameter—has the best statistical properties and should thus be preferred over other common implementations of the KPSS test.

With regard to the choice of the null hypothesis, **kpsstest** has three options: 1) zero mean stationary, 2) level stationary, and 3) trend stationary. A zero-mean-stationary null hypothesis [ $\alpha = \xi = d = 0$  in (1)] means that, under the null hypothesis, the series is stationary around a mean value of 0. A level-stationary null hypothesis implies that, under this null hypothesis, the series is stationary around a constant or drift term. Last, a trend-stationary null hypothesis means that a stationary series contains both a drift and a trend term. The Stata manual for **dfuller** (see [TS] **dfuller**) (Dickey–Fuller test) provides an excellent explanation for selecting which null hypothesis option to use. It suggests that the choice of null hypothesis must be theoretically or visually guided. If the user has a theory about the stationarity of a series, then the user may select the null hypothesis most suitable to that theory. However, more often than not a visual inspection of the series is preferred. If the user then finds a series to be trending upward or centered around zero, the user may select the trend-stationary or zero-mean-stationary null hypothesis, respectively.

To summarize the above discussion, the GKPSS is the preferred test, and the choice of null hypothesis should largely focus on the visual inspection of the series.

---

1. Readers interested in the technical discussion of these tests and options may refer to the appendix.

### 3 Calculating sample-specific critical values and p-values

#### 3.1 Response surface regressions and sample-specific critical values

Because response surface regressions easily provide accurate sample-specific critical values, I used this method to calculate critical values.<sup>2</sup> This involved the following steps:

1. For a given sample size, I simulated the distribution of the test statistic using 50,000 replications and calculated the 1, 2.5, 5, 7.5, 10, 12.5, ..., 90, 95, 97.5, 99, 99.9, and 99.99 percentiles. Doing so provides the critical values of the test statistic at various levels of significance. For example, the critical value at the 0.01 significance level is the critical value at the 99 percentile.<sup>3</sup>
2. For the same sample size, I repeated this procedure 100 times and calculated the averages and standard deviations of the test statistics over the 100 experiments for each percentile.
3. I repeated the above two steps for the following 14 sample sizes: 10, 25, 50, 60, 75, 100, 125, 150, 200, 250, 300, 400, 500, and 2,000.
4. To account for heteroskedasticity, I fit a weighted least-squares (WLS) regression model for each percentile in which the average test statistic for each percentile is a quadratic function of time and the standard deviations for each percentile are used as weights (Davidson and MacKinnon 1993; MacKinnon 1994). The quadratic functions depend on the test used and are shown in table 1.

Table 1. WLS regressions for the response surfaces

Test	Quadratic function
KPSS	$\frac{c}{\sigma} = \beta_{\infty} \left(\frac{1}{\sigma}\right) + \sum_{i=1}^2 \beta_i \left(\frac{1}{T}\right)^i \left(\frac{1}{\sigma}\right)$
GKPSS	$\frac{c}{\sigma} = \beta_{\infty} \left(\frac{1}{\sigma}\right) + \sum_{i=1}^4 \beta_i \left(\frac{l}{T}\right)^i \left(\frac{1}{\sigma}\right)$

NOTE:  $c$  is the average critical value across the 100 experiments for a given percentile,  $\sigma$  is the standard deviation of the critical value across the 100 experiments for a given percentile,  $\beta_{\infty}$  is the asymptotic critical value for a given percentile,  $T$  is the number of time periods, and  $l$  is the lag truncation parameter.

Although the quadratic functions for these WLS models are determined through experimentation based on the best fit ( $R^2$  and sum of squared residuals), I use functional

2. These simulations were conducted across multiple computer cores using the `parallel` command (Vega Yon and Quistorff 2019).

3. It is important to calculate the critical values at these various percentiles to approximate where the test statistic of our realized sample lies and thus to calculate its  $p$ -value.

forms similar to those of Sephton (1995, 2017), who has demonstrated that such functional forms provide the most accurate estimates of the finite and asymptotic critical values for the KPSS and GKPSS tests. My results confirm that these functional forms provide the best-fitting regression models. The benefits of using response surface regressions are that—if the response surface is correctly specified—parameter estimates of the regressions can be used to calculate accurate sample-specific critical values with ease. The critical value ( $\hat{c}$ ) for a particular sample size ( $T$ ) can be approximated by substituting for  $T$  and the parameter estimates ( $\hat{\beta}_\infty$  and  $\sum_{i=1}^4 \hat{\beta}_i$ ) from the response surface regressions in the quadratic functions in table 1. Additionally, the fact that each data-generating process is subject to 50,000 replications and that this process is repeated 100 times reduces the experimental uncertainty involved in Monte Carlo experiments and increases the accuracy and precision of the estimates of quantities of interest (Davidson and MacKinnon 1993).<sup>4</sup>

### 3.2 Assessing the calculated critical values

The asymptotic critical values from my response surface regressions and those generated by Sephton (2017)—who also calculates KPSS and GKPSS critical values using response surface regressions—are compared in table 2. These results demonstrate that the asymptotic critical values, generated from response surface regressions for *kpsstest*, are similar to those generated by Sephton (2017). Similarly, the finite-sample critical values for the KPSS test with null hypotheses of level stationary and trend stationary are also similar to those generated by Sephton (2017).<sup>5</sup> However, note that the finite-sample critical values for the KPSS test with a null hypothesis of zero mean stationary and those for the GKPSS tests differ from Sephton (2017).

---

4. For some other applications of response surface regressions, see Hendry (1979), Engle, Hendry, and Trumble (1985), Ericsson (1991), MacKinnon (1994), and Sephton (1995, 2017).

5. The finite-sample critical values from *kpsstest* and Sephton (2017) are provided in table A2 in the appendix.

Table 2. Comparison of asymptotic critical values

	KPSS			GKPSS		
	10%	5%	1%	10%	5%	1%
Zero mean stationary						
<b>kpsstest</b>	1.197	1.655	2.788	1.198	1.658	2.797
Sephton (2017)	1.195	1.656	2.788	1.195	1.652	2.794
Level stationary						
<b>kpsstest</b>	0.347	0.461	0.744	0.347	0.461	0.742
Sephton (2017)	0.347	0.461	0.743	0.347	0.461	0.741
Trend stationary						
<b>kpsstest</b>	0.119	0.148	0.218	0.119	0.148	0.218
Sephton (2017)	0.119	0.148	0.218	0.118	0.146	0.218

While the finite-sample critical values generated by Sephton (2017) in smaller samples are sometimes four times as large as the asymptotic critical values, those used in **kpsstest**, for increases in sample size, demonstrate more gradual increments or decrements toward the asymptotic critical values. For example, while the asymptotic 1% critical value for the GKPSS test with a zero-mean-stationary null hypothesis is between 2.79 and 2.80 (table 2), Sephton's (2017) critical value is 9.477 for a sample size of 50. For the same sample size, the critical value I calculated from the response surface is 2.410. To confirm that the algorithm used in **kpsstest** generates the right critical values, I compared the test statistics, for a given time series, with those generated by Baum's (2000) **kpss** command. I found the test statistics to be similar (as seen in section 4 of this article).<sup>6</sup> Further, I conducted Monte Carlo experiments to analyze the type 1 error rates and power for tests in which our finite-sample critical values differed. I report results for the 5% significance level, at which a type 1 error rate of 0.05 and a power of 1 are expected.

6. At least two places where Sephton's (2017) and my simulations differ are that I include sample sizes of 10 and 25 in my simulations and that my simulations are conducted in Stata while his are conducted in MATLAB.



Table 3 shows the type 1 error and power for the finite-sample critical values in **kpsstest** and for those in Sephton (2017). The type 1 error rates for **kpsstest** are either the same as or slightly above the type 1 errors for Sephton (2017), and they are almost always under the 0.05 expected value. The largest difference in type 1 error rates is for the GKPSS test with a trend-stationary null hypothesis for a time series of length 50. However, such a large difference is unusual for any other null hypothesis and sample size. When we compare power, the power gains in **kpsstest** are significantly higher, especially for the GKPSS test with a zero-mean-stationary null hypothesis. Thus, overall with **kpsstest**, small losses in type 1 error rates are offset by large gains in power. However, with Sephton's (2017) critical values, large losses in power are not offset by small gains in type 1 error rates.

Table 3. Comparison of type 1 error rates and power between **kpsstest** and Sephton (2017)

		KPSS	GKPSS		
		Zero mean	Zero mean	Level	Trend
		Type 1 error			
$T = 50$	<b>kpsstest</b>	0.008	0.008	0.088	0.088
	Sephton (2017)	0.008	0.008	0.072	0.008
$T = 75$	<b>kpsstest</b>	0.024	0.024	0.048	0.048
	Sephton (2017)	0.016	0.000	0.040	0.024
$T = 100$	<b>kpsstest</b>	0.008	0.008	0.040	0.080
	Sephton (2017)	0.008	0.008	0.040	0.056
$T = 200$	<b>kpsstest</b>	0.032	0.032	0.072	0.016
	Sephton (2017)	0.024	0.024	0.072	0.016
		Power			
$T = 50$	<b>kpsstest</b>	0.552	0.728	1	1
	Sephton (2017)	0.528	0.432	1	1
$T = 75$	<b>kpsstest</b>	0.632	0.640	1	1
	Sephton (2017)	0.624	0.512	1	1
$T = 100$	<b>kpsstest</b>	0.768	0.800	1	1
	Sephton (2017)	0.744	0.744	1	1
$T = 200$	<b>kpsstest</b>	0.872	0.896	1	1
	Sephton (2017)	0.856	0.896	1	1

### 3.3 Reported p-values

The 1, 2.5, 5, 7.5, ..., 97.5, 99, 99.9, and 99.99 percentiles of the distributions of the test statistics correspond to  $p$ -values of 0.99, 0.975, 0.95, 0.925, ..., 0.025, 0.01, 0.001, and 0.0001, respectively. When the test statistic is equal to the critical value (up to four decimal places) for the calculated percentiles, **kpsstest** reports the corresponding  $p$ -value. When the test statistic is less than the 1 percentile value of the distribution, **kpsstest** reports a  $p$ -value of 1. When the test statistic lies between the 1 percentile and 99.99 percentile values of the distribution and is not equal to a critical value for the calculated percentiles, **kpsstest** linearly interpolates the  $p$ -value.<sup>7</sup> Finally, when the test statistic is greater than the 99.99 percentile value of the distributions, **kpsstest** reports a  $p$ -value of 0.00001.<sup>8</sup>

## 4 Syntax

```
kpsstest varname [if] [in], [[kpss|gkpss]
  [trunclag(string)|lag(#)|auto] [noconstant|level|trend]
```

*varname* is the variable to be tested for stationarity using the KPSS or GKPSS test. If no options are specified, the GKPSS test with a zero-mean-stationary null hypothesis is conducted by default.

## 5 Options

**kpss** indicates that a Bartlett kernel will be used to weight the serial dependence in the residuals when calculating the estimate of the long-run variance and will be used in conjunction with a Schwert criterion, a user-specified value, or an automatic bandwidth selection for the lag truncation parameter. **trunc****lag**(), **lag**(), or **auto** must be specified with the **kpss** option.

**gkpss** indicates that a quadratic spectral kernel will be used to weight the serial dependence in the residuals when calculating the estimate of the long-run variance in conjunction with automatic bandwidth selection to determine the lag truncation parameter. Except for the choice of the null hypothesis, no other option is allowed when specifying **gkpss**. If the user specifies neither **kpss** nor **gkpss**, **gkpss** is used by default.

**trunc****lag**(*string*) indicates that the Schwert criterion will be used to calculate the lag truncation parameter for the kernel. Users will specify either **short** or **long**. This option can be used only with the **kpss** option.

7. This is similar to the **tseries** package in R (Trapletti, Hornik, and LeBaron 2019), except that command is able only to linearly interpolate between the 10% and 1% significance levels.

8. Note that the distribution of the test statistic is not calculated every time **kpsstest** is run. Instead, the data have already been calculated, and **kpsstest** uses these data to report  $p$ -values.

**lag(#)** allows the user to specify the lag truncation parameter to be used in the Bartlett kernel. This option can be used only with the **kpss** option.

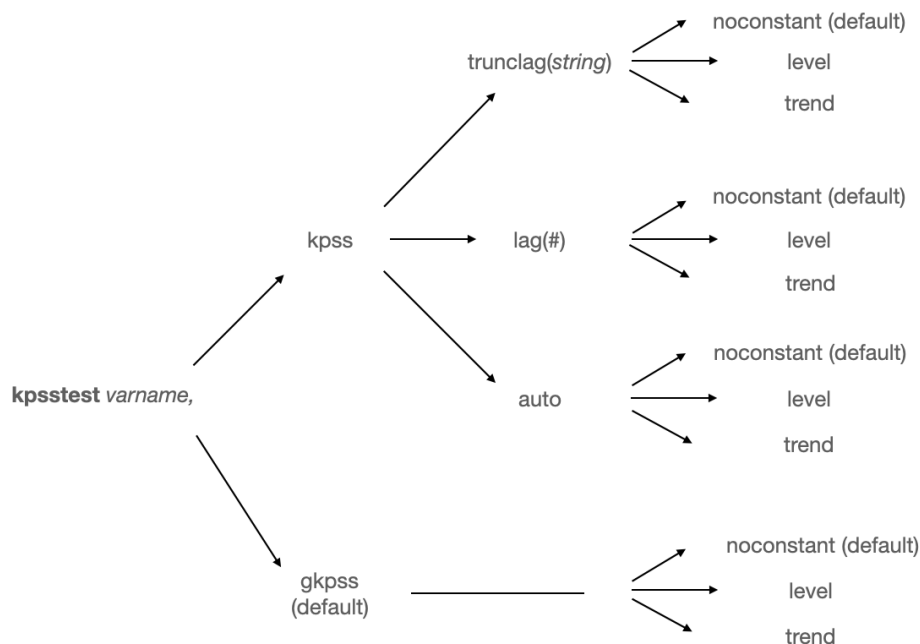
**auto** indicates automatic bandwidth selection for the lag truncation parameter is to be used in the Bartlett kernel. This option can be used only with the **kpss** option.

**noconstant** indicates that a zero-mean-stationary null hypothesis will be used. This is the default option if a null hypothesis is not specified. This option may not be used with **level** or **trend**.

**level** indicates that a level-stationary null hypothesis will be used. This option may not be used with **noconstant** or **trend**.

**trend** indicates that a trend-stationary null hypothesis will be used. This option may not be used with **noconstant** or **level**.

The two types of tests (KPSS and GKPSS) that **kpsstest** can implement, combined with the various lag truncation parameter options and the three null hypotheses options, can make implementing **kpsstest** confusing. To aid with implementing the command, figure 1 demonstrates how a practitioner can use **kpsstest**. First, the user must specify **kpsstest varname**, followed by the choice of whether the user wants to use the **kpss** or **gkpss** test. If the user chooses **kpss**, then the user must choose between **trunc lag()**, **lag()**, or **auto** for determining the lag truncation parameter. The user must then select the null hypothesis (for example, **kpsstest varname, kpss auto level**). If the user chooses **gkpss**, then the user needs to only select the null hypothesis (for example, **kpsstest varname, gkpss trend**). If the user chooses no option at all, a GKPSS test with a zero-mean-stationary null hypothesis is conducted by default (for example, **kpsstest varname**).

Figure 1. Using `kpsstest` with its various options

## 5.1 Stored results

`kpsstest` stores the following in `r()`:

### Scalars

<code>r(N)</code>	number of observations	<code>r(c10)</code>	10% critical value
<code>r(c1)</code>	1% critical value	<code>r(pval)</code>	<i>p</i> -value for the test statistic
<code>r(c2p5)</code>	2.5% critical value	<code>r(kpss)</code>	KPSS or GKPSS test statistic
<code>r(c5)</code>	5% critical value		

## 6 Examples

In this section, I demonstrate the utility of `kpsstest` using examples. First, I demonstrate its similarities to Baum's (2000) `kpss` command; second, how `kpsstest` extends the capabilities of `kpss` and provides more functionality to the user; and finally, how users can use a visual inspection of the series to appropriately select the null hypothesis and the implications of failing to do so.

## 6.1 Similarities between `kpss` and `kpsstest` for the GKPSS test

The first example is a stationarity test for the log of consumption from the German macroeconomic data (Lütkepohl 1993) available in Stata. I implement the GKPSS test (KPSS test with a quadratic spectral and automatic bandwidth selection procedure) using both `kpss` and `kpsstest` for a trend-stationary null hypothesis.

```
. webuse lutkepohl2
(Quarterly SA West German macro data, Bil DM, from Lutkepohl 1993 Table E.1)
. kpss ln_consump, auto qs

KPSS test for ln_consump

Automatic bandwidth selection (maxlag) = 3
Autocovariances weighted by Quadratic Spectral kernel

Critical values for H0: ln_consump is trend stationary

10%: 0.119  5% : 0.146  2.5%: 0.176  1% : 0.216

Lag order    Test statistic
   3          .232
```

As seen above, Baum's (2000) `kpss` command finds that the automatic bandwidth selection algorithm selects a lag truncation parameter of 3 and finds the test statistic to be 0.232.

```
. kpsstest ln_consump, gkpss trend

KPSS test for stationarity

Quadratic Spectral Kernel, Automatic Bandwidth Lag Truncation with 3 lags

H0: ln_consump is trend-stationary

Number of obs = 92
```

Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
0.232	0.200	0.170	0.146	0.121

```
Linearly-interpolated p-value = 0.0061
```

Similar to `kpss`, `kpsstest` determines the lag truncation parameter to be 3 and displays a test statistic of 0.232. However, `kpsstest` displays sample-specific GKPSS test critical values, while `kpss` displays asymptotic KPSS test critical values. The output of `kpsstest` is also more user friendly. This output should be familiar to Stata users because it mimics that of the Dickey–Fuller test (`dfuller`). It neatly displays the number of observations, the test statistic, and the critical values. In contrast to `kpss`, `kpsstest` displays the  $p$ -value for the given test statistic, saving users time from having to identify the significance of the test statistic by comparing the test statistic with the critical values.

## 6.2 Similarities between `kpss` and `kpsstest` for the KPSS test

In this example, using the same macroeconomic data as the previous example, I compare the functionality of `kpss` and `kpsstest` for the original KPSS test—with a Bartlett kernel and Schwert criterion for a short lag truncation parameter—with a level-stationary null hypothesis.

```
. kpss ln_consump, notrend

KPSS test for ln_consump

Maxlag = 3 chosen by Schwert criterion
Autocovariances weighted by Bartlett kernel

Critical values for H0: ln_consump is level stationary
10%: 0.347  5% : 0.463  2.5%: 0.574  1% : 0.739

Lag order    Test statistic
  0           9.26
  1           4.7
  2           3.17
  3           2.4
```

As seen above, Baum's (2000) `kpss` command determines the short lag truncation parameter to be 3 and finds the test statistic at the third lag to be 2.4.

```
. kpsstest ln_consump, kpss trunc lag(short) level

KPSS test for stationarity

Bartlett Kernel, Short Lag Truncation with 3 lags

H0: ln_consump is level-stationary

Number of obs = 92
```

Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
2.404	0.733	0.578	0.462	0.350

```
Linearly-interpolated p-value = 0.0000
```

`kpsstest` concludes similarly to `kpss` with two differences. First, `kpss` returns the test statistic for every lag up to the third lag. While there is no *ex ante* reason to do so, users of `kpsstest` may recover similar output using the `lag()` option. Second, the test statistic from `kpss` at the third lag is 2.4, while that from `kpsstest` is 2.404. This is because the test statistic from `kpss` is rounded to one decimal place and that from `kpsstest` is rounded to three decimal places. The test statistics without rounding for `kpss` and `kpsstest` are similar up to six decimal places.

### 6.3 Similarities between `kpss` and `kpsstest` for a user-defined lag truncation parameter

Continuing with the same German macroeconomic data as in the previous examples, I now compare the results from `kpss` and `kpsstest` for the original KPSS test with a user-defined lag truncation parameter (4 for this example).

```
. kpss ln_consump, notrend maxlag(4)

KPSS test for ln_consump

Maxlag = 4
Autocovariances weighted by Bartlett kernel

Critical values for H0: ln_consump is level stationary

10%: 0.347   5% : 0.463   2.5%: 0.574   1% : 0.739

Lag order   Test statistic
  0           9.26
  1           4.7
  2          3.17
  3           2.4
  4           1.94
```

As shown in the above output, Baum's (2000) `kpss` command returns a test statistic of 1.94 at the fourth lag.

```
. kpsstest ln_consump, kpss lag(4) level

KPSS test for stationarity

Bartlett Kernel, User-Specified Lag Truncation with 4 lags

H0: ln_consump is level-stationary

Number of obs = 92

Test Statistic      1% Critical Value   2.5% Critical Value   5% Critical Value   10% Critical Value
-----
      1.944           0.733           0.578           0.462           0.350

Linearly-interpolated p-value = 0.0000
```

`kpsstest` also returns a similar test statistic (1.944) but rounded to the three decimal places instead of two as in `kpss`. The test statistics are identical up to six decimal places.

## 6.4 Similarities between `kpss` and `kpsstest` for automatic bandwidth-selected lag truncation parameter

Using the same time series as above, the examples below compare the results from `kpss` and `kpsstest` for the original KPSS test with a Bartlett kernel used in conjunction with the automatic bandwidth selection criterion in determining the lag truncation parameter.

```
. kpss ln_consump, notrend auto

KPSS test for ln_consump

Automatic bandwidth selection (maxlag) = 6
Autocovariances weighted by Bartlett kernel

Critical values for H0: ln_consump is level stationary

10%: 0.347   5% : 0.463   2.5%: 0.574   1% : 0.739

Lag order      Test statistic
     6              1.42
```

As seen above, Baum's (2000) `kpss` command calculates the lag truncation parameter to be equal to 6, which results in a test statistic of 1.42.

```
. kpsstest ln_consump, kpss auto level

KPSS test for stationarity

Bartlett Kernel, Automatic Bandwidth Lag Truncation with 6 lags

H0: ln_consump is level-stationary

Number of obs = 92
```

Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
1.419	0.550	0.508	0.440	0.354

Linearly-interpolated p-value = 0.0000

`kpsstest` recovers a similar test statistic equal to 1.419 after determining the lag truncation parameter is 6.<sup>9</sup> The test statistics from both `kpsstest` and `kpss` are identical up to 6 decimal places.

9. Note that the critical values used in this implementation of `kpsstest`—a Bartlett kernel with automatic bandwidth selection—are calculated from response surface regressions specific to this test specification.



So far, while we have looked at the similarities between the two commands in terms of test statistics and functionality, we have also noted differences: 1) `kpsstest`'s output is user friendly, resembling that of `dfuller`; and 2) `kpsstest` displays the number of observations, the sample-specific critical values, and the  $p$ -value for the test statistic. The next few examples will demonstrate other ways in which `kpsstest` differs from `kpss`.

## 6.5 kpsstest with a zero-mean-stationary null hypothesis

`kpss` lacks the feature that allows for a zero-mean-stationary null hypothesis for the original KPSS and GKPSS tests. Using the same German macroeconomic data as in the above examples, I demonstrate how users can implement the GKPSS test with a zero-mean-stationary null hypothesis using `kpsstest`.

```
. kpsstest ln_consump, gkpss

KPSS test for stationarity

Quadratic Spectral Kernel, Automatic Bandwidth Lag Truncation with 3 lags

H0: ln_consump is zero-mean-stationary

                                     Number of obs = 92

      Test Statistic      1% Critical Value      2.5% Critical Value      5% Critical Value      10% Critical Value
-----
              7.412              2.501              1.992              1.594              1.186

Linearly-interpolated p-value = 0.0000
```

The above output demonstrates that the GKPSS test for log of consumption rejects the null hypothesis that the series is zero mean stationary. This means that the test rejects that the series is stationary when we do not account for constant and trend terms.

## 6.6 kpsstest with a Schwert criterion and long lag truncation parameter

In calculating the lag truncation parameter using a Schwert criterion for the Bartlett kernel in the original KPSS test in addition to the `trunc lag(short)` and `lag()` options, users may specify the `trunc lag(long)` option when using `kpsstest`. This was one of the options used by Kwiatkowski et al. (1992). `kpss` does not allow for this option. I demonstrate how users may do so using the log of consumption from the German macroeconomic dataset used above for a level-stationary null hypothesis.

```
. kpsstest ln_consump, kpss trunclag(long) level
```

KPSS test for stationarity

Bartlett Kernel, Long Lag Truncation with 11 lags

H0: ln\_consump is level-stationary

Number of obs = 92				
Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
0.876	0.733	0.578	0.462	0.350

Linearly-interpolated p-value = 0.0067

The results demonstrate that the lag truncation parameter is determined to be 11 using a long Schwert criterion. The test rejects the null hypothesis that the series is level stationary.

6.7 Choosing a null hypothesis for kpsstest

As mentioned earlier in the article, `kpsstest` provides users with the choice of a zero-mean-stationary, level-stationary, or trend-stationary null hypothesis. This choice should be guided by theory, if possible, but should largely be a consequence of visual inspection of the time series. Consider the log of investment, plotted in figure 2, using the same German macroeconomic data as in the previous examples.

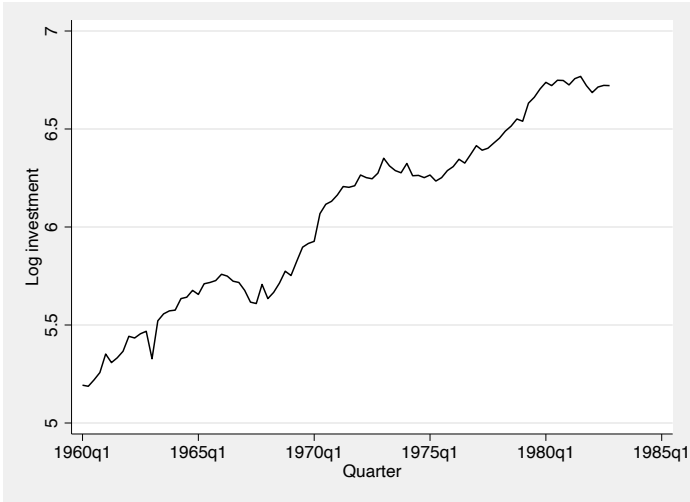


Figure 2. Time-series line plot for the log of investment

The above figure clearly demonstrates that the log of investment is trending upward over time. This suggests that a trend-stationary null hypothesis is appropriate. Failing to use a trend-stationary hypothesis could lead to the conclusion that the series is nonstationary and may contain a unit root when the series is actually trend stationary.

```
. kpsstest ln_inv, gkpss trend
```

```
KPSS test for stationarity
```

```
Quadratic Spectral Kernel, Automatic Bandwidth Lag Truncation with 3 lags
```

```
H0: ln_inv is trend-stationary
```

Number of obs = 92

Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
0.107	0.200	0.170	0.146	0.121

Linearly-interpolated p-value = 0.1485

As seen in the output above, the choice of a GKPSS test with a trend-stationary null hypothesis finds that the series is indeed trend stationary. While the GKPSS tests with the other null hypotheses, as seen in the two examples below, may correctly provide insight that the series is nonstationary, they can lead to the false inference that this nonstationarity is due to a unit root.

```
. kpsstest ln_inv, gkpss
```

```
KPSS test for stationarity
```

```
Quadratic Spectral Kernel, Automatic Bandwidth Lag Truncation with 3 lags
```

```
H0: ln_inv is zero-mean-stationary
```

Number of obs = 92

Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
7.414	2.501	1.992	1.594	1.186

Linearly-interpolated p-value = 0.0000

```
. kpsstest ln_inv, gkpss level
```

```
KPSS test for stationarity
```

```
Quadratic Spectral Kernel, Automatic Bandwidth Lag Truncation with 3 lags
```

```
H0: ln_inv is level-stationary
```

```
Number of obs = 92
```

Test Statistic	1% Critical Value	2.5% Critical Value	5% Critical Value	10% Critical Value
2.337	0.679	0.552	0.452	0.350

```
Linearly-interpolated p-value = 0.0000
```

This example demonstrates that a visual inspection of a series is insightful and necessary to determine the choice of null hypothesis. It also demonstrates that using the incorrect null hypothesis can lead to incorrect inferences about the stationarity of the series.

## 7 Conclusion

In this article, I briefly described the KPSS test and later extensions to the test. I introduced a command—`kpsstest`—that can implement the original KPSS and the GKPSS tests. `kpsstest` also provides users with sample- and test-specific critical values and  $p$ -values for the test statistic. Thus, `kpsstest` offers users more appropriate inferences and the ability to easily interpret the test’s result through a  $p$ -value. Users would otherwise have to check for the test statistic’s statistical significance by determining whether the test statistic lies above, between, or below the asymptotic critical values provided by Kwiatkowski et al. (1992). As Sephton (2017) demonstrates, using asymptotic critical values for finite samples is inappropriate and can lead to incorrect inferences. Additionally, `kpsstest` allows for a null hypothesis that the series has no constant or is zero mean stationary.

## 8 Acknowledgments

Portions of this research were conducted with the advanced computing resources and consultation provided by Texas A&M High Performance Research Computing. The author thanks the editor, the anonymous reviewer, Peter Sephton, and Guy D. Whitten for their useful comments and suggestions.

## 9 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 22-2
. net install st0671      (to install program files, if available)
. net get st0671          (to install ancillary files, if available)
```

## 10 References

- Andrews, D. W. K. 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59: 817–858. <https://doi.org/10.2307/2938229>.
- Baum, C. F. 2000. kps: Stata module to compute Kwiatkowski–Phillips–Schmidt–Shin test for stationarity. Statistical Software Components S410401, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s410401.html>.
- Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics*. New York: Oxford University Press.
- Engle, R. F., D. F. Hendry, and D. Trumble. 1985. Small-sample properties of ARCH estimators and tests. *Canadian Journal of Economics* 18: 66–93. <https://doi.org/10.2307/135114>.
- Ericsson, N. R. 1991. Monte Carlo methodology and the finite sample properties of instrumental variables statistics for testing nested and non-nested hypotheses. *Econometrica* 59: 1249–1277. <https://doi.org/10.2307/2938367>.
- Hendry, D. F. 1979. The behavior of inconsistent instrumental variables estimators in dynamic systems with autocorrelated errors. *Journal of Econometrics* 9: 295–314. [https://doi.org/10.1016/0304-4076\(79\)90076-9](https://doi.org/10.1016/0304-4076(79)90076-9).
- Hobijn, B., P. H. Franses, and M. Ooms. 2004. Generalizations of the KPSS-test for stationarity. *Statistica Neerlandica* 58: 483–502. <https://doi.org/10.1111/j.1467-9574.2004.00272.x>.
- Kwiatkowski, D., P. C. B. Phillips, P. Schmidt, and Y. Shin. 1992. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics* 54: 159–178. [https://doi.org/10.1016/0304-4076\(92\)90104-Y](https://doi.org/10.1016/0304-4076(92)90104-Y).
- Lütkepohl, H. 1993. *Introduction to Multiple Time Series Analysis*. 2nd ed. New York: Springer.
- MacKinnon, J. G. 1994. Approximate asymptotic distribution functions for unit-root and cointegration tests. *Journal of Business and Economic Statistics* 12: 167–176. <https://doi.org/10.2307/1391481>.

- Newey, W. K., and K. D. West. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653. <https://doi.org/10.2307/2297912>.
- Sephton, P. S. 1995. Response surface estimates of the KPSS stationarity test. *Economics Letters* 47: 255–261. [https://doi.org/10.1016/0165-1765\(94\)00561-F](https://doi.org/10.1016/0165-1765(94)00561-F).
- . 2017. Finite sample critical values of the generalized KPSS stationarity test. *Computational Economics* 50: 161–172. <https://doi.org/10.1007/s10614-016-9586-z>.
- Trapletti, A., K. Hornik, and B. LeBaron. 2019. tseries: Time series analysis and computational finance. R package version 0.10-49. <http://cran.r-project.org/web/packages/tseries/>.
- Vega Yon, G. G., and B. Quistorff. 2019. parallel: A command for parallel computing. *Stata Journal* 19: 667–684. <https://doi.org/10.1177/1536867X19874242>.

### About the author

Ali Kagalwala is a PhD candidate in political science at Texas A&M University.

## A Appendix

### A.1 Technical description of the KPSS test

The KPSS test decomposes a series—say,  $y_t$ —into three components: a deterministic trend ( $t$ ), a random walk ( $r_t$ ), and a stationary error ( $\epsilon_t$ ) (Kwiatkowski et al. 1992; Hobijn, Franses, and Ooms 2004):

$$\begin{aligned} y_t &= \alpha + \xi t + dr_t + \epsilon_t, & \text{where } \epsilon_t &\sim N(0, \sigma_\epsilon^2) \\ r_t &= r_{t-1} + u_t, & \text{where } u_t &\sim N(0, \sigma_u^2) \end{aligned}$$

Under the null hypothesis, the series is trend stationary and is mathematically depicted as  $H_0 : \sigma_u^2 = 0$ . The test statistic is a one-sided Lagrange multiplier statistic and also allows for a level-stationary null hypothesis when  $\xi$  is restricted to 0 and a zero-mean-stationary null hypothesis when  $\alpha = \xi = d = 0$ . When none of the parameters are restricted to 0, the null hypothesis is that the series is trend stationary. The test statistic is

$$\text{KPSS} = \frac{1}{T^2} \sum_{t=1}^T \frac{S_t^2}{\hat{\sigma}^2} \quad (\text{A1})$$

where  $\hat{\sigma}^2$  is a consistent estimator of  $\sigma_\epsilon^2$ ,  $S_t = \sum_{i=1}^t e_i$ , and  $e_t$  are the residuals from the regression of a series on its random walk, stationary errors, or deterministic trend components. That is, for a trend-stationary null hypothesis, the residuals are from a regression of the series on its intercept and trend term; for a level-stationary null hypothesis, the residuals are from a regression of the series only on its intercept; and

for a zero-mean-stationary null hypothesis, the residuals are equal to the series itself ( $e_t = y_t$ ).<sup>10</sup>

The assumption that  $\sigma_\epsilon^2$  is independently and identically distributed is unlikely to hold in observed data. Thus, Kwiatkowski et al. (1992) derive the asymptotic distribution of the KPSS test statistic by relaxing the independent and identically distributed assumption and allowing for serial correlation. The long-run variance of  $\epsilon_t$  can be written as

$$\sigma_\epsilon^2 = \lim_{T \rightarrow \infty} \frac{1}{T} E(S_T^2)$$

The residuals in the regression ( $e_t$ ) are used to derive a consistent estimator  $\{s^2(l) = \hat{\sigma}^2\}$  of  $\sigma_\epsilon^2$ , such that

$$s^2(l) = \frac{1}{T} \sum_{t=1}^T e_t^2 + \frac{2}{T} \sum_{s=1}^l w(s, l) \sum_{t=s+1}^T e_t e_{t-s} \quad (\text{A2})$$

where  $w$  is a kernel that weights the temporal dependence in the residuals. Kwiatkowski et al. (1992) use a Bartlett kernel such that

$$w(s, l) = 1 - \frac{s}{l+1}, \quad \text{where } l \text{ is the lag truncation parameter}$$

Later developments demonstrate that a quadratic spectral kernel (Andrews 1991; Newey and West 1994) used in conjunction with automatic bandwidth lag truncation parameter ( $l$ ) selection (developed by Newey and West [1994]) produces a consistent test statistic that has better type 1 error and power properties in finite samples (Hobijn, Franses, and Ooms 2004). The quadratic spectral kernel is given by

$$w(s, l) = \frac{25}{12\pi^2(s/l)^2} \left[ \frac{\sin\{6\pi(s/l)/5\}}{6\pi(s/l)/5} - \cos\{6\pi(s/l)/5\} \right]$$

Kwiatkowski et al. (1992) use the Schwert criterion to select a lag truncation parameter for the Bartlett kernel, which can be defined as short—integer $\{4 \times (T/100)^{1/4}\}$ —or long—integer $\{12 \times (T/100)^{1/4}\}$ . To determine the lag truncation parameter for the quadratic spectral kernel, Hobijn, Franses, and Ooms (2004) use an automatic bandwidth selection procedure as follows:

- Select a bandwidth parameter such that  $n_t = \text{integer}(T^{2/25})$ .
- Compute  $\hat{a}^0 = \sum_{t=1}^T e_t^2 + 2 \sum_{i=1}^{n_t} \sum_{t=s+1}^T e_t e_{t-s}$ .
- Compute  $\hat{a}^2 = 2 \sum_{i=1}^{n_t} i^2 \sum_{t=s+1}^T e_t e_{t-s}$ .

---

10. Because the regression of the variable is only on either a trend or a constant, the number of lags specified or used as the lag truncation parameter in `kpsstest` does not affect the number of observations used to calculate the test statistic in (A1). As demonstrated below, the lag truncation parameter ( $l$ ) is used only in a kernel ( $w$ ) that weights the temporal dependence in (A2).

- Compute  $\hat{\gamma} = 1.3221[\{(\hat{a}^2/\hat{a}^0)\}^2]^{1/5}$ .
- Calculate the bandwidth parameter such that  $l = \min\{T, \text{integer}(\hat{\gamma}T^{1/5})\}$ .

As summarized in table 1, a Bartlett kernel used in conjunction with the Schwert criterion is referred to as the KPSS test, and the quadratic spectral kernel used in conjunction with automatic bandwidth selection is known as a GKPSS test (Hobijn, Franses, and Ooms 2004). After one selects a kernel and a method for calculating the lag truncation parameter, the KPSS and GKPSS test statistics can be computed by calculating the estimate of the long-run variance, as shown in (A2). This estimate is then substituted as the denominator in (A1). Note that using a quadratic spectral kernel with automatic bandwidth selection (GKPSS) reduces the type 1 error rate of the test when the series is highly autoregressive (Hobijn, Franses, and Ooms 2004).

Table A1. Comparison of the KPSS and GKPSS tests

Test	Kernel	Lag-truncation parameter	Properties
KPSS	Bartlett	Short/long Schwert criterion	
GKPSS	Quadratic spectral	Automatic bandwidth selection	Lower type 1 error rate in finite samples

`kpsstest` additionally allows for the test statistic to be a function of a Bartlett kernel used in conjunction with automatic bandwidth selection for the lag truncation parameter. In this case, the lag truncation parameter ( $l$ ) is determined as follows (Hobijn, Franses, and Ooms 2004):

- Select a bandwidth parameter such that  $n_t = \text{integer}(T^{2/9})$ .
- Compute  $\hat{a}^0 = \sum_{t=1}^T e_t^2 + 2 \sum_{i=1}^{n_t} \sum_{t=s+1}^T e_t e_{t-s}$ .
- Compute  $\hat{a}^1 = 2 \sum_{i=1}^{n_t} i \sum_{t=s+1}^T e_t e_{t-s}$ .
- Compute  $\hat{\gamma} = 1.1447[\{(\hat{a}^1/\hat{a}^0)\}^2]^{1/3}$ .
- Calculate the bandwidth parameter such that  $l = \min\{T, \text{integer}(\hat{\gamma}T^{1/3})\}$ .

## A.2 Comparison of finite-sample critical values between `kpsstest` and Sephton (2017)

Table A2 compares the finite-sample critical values from `kpsstest` and Sephton (2017). While the finite-sample critical values for the KPSS test with level-stationary and trend-stationary null hypotheses are similar, those for the KPSS test with zero-mean-stationary null hypothesis and the GKPSS test are less similar.



Table A2. Comparison of the KPSS and GKPSS tests critical values from *kpsstest* and Sephton (2017)

Null hypothesis	$T$	KPSS			GKPSS		
		10%	5%	1%	10%	5%	1%
Zero mean stationarity	50	1.218 (1.283)	1.668 (1.789)	2.731 (3.073)	1.190 (1.970)	1.578 (3.123)	2.410 (9.477)
	75	1.211 (1.252)	1.664 (1.742)	2.751 (2.971)	1.187 (1.600)	1.588 (2.354)	2.469 (4.993)
	100	1.207 (1.237)	1.662 (1.719)	2.760 (2.922)	1.186 (1.466)	1.596 (2.104)	2.514 (3.989)
	200	1.202 (1.216)	1.659 (1.687)	2.774 (2.853)	1.187 (1.311)	1.617 (1.840)	2.621 (3.212)
Level stationarity	50	0.351 (0.352)	0.462 (0.462)	0.724 (0.724)	0.353 (0.411)	0.450 (0.499)	0.655 (0.798)
	75	0.350 (0.350)	0.462 (0.462)	0.731 (0.730)	0.350 (0.374)	0.451 (0.464)	0.671 (0.651)
	100	0.349 (0.349)	0.462 (0.462)	0.734 (0.734)	0.349 (0.363)	0.452 (0.458)	0.682 (0.660)
	200	0.348 (0.348)	0.461 (0.462)	0.739 (0.738)	0.348 (0.353)	0.455 (0.459)	0.706 (0.697)
Trend stationarity	50	0.121 (0.121)	0.149 (0.149)	0.213 (0.213)	0.124 (0.224)	0.147 (0.437)	0.196 (3.361)
	75	0.121 (0.121)	0.148 (0.148)	0.214 (0.215)	0.122 (0.148)	0.146 (0.177)	0.199 (0.393)
	100	0.120 (0.120)	0.148 (0.148)	0.215 (0.215)	0.121 (0.133)	0.146 (0.157)	0.201 (0.215)
	200	0.120 (0.120)	0.148 (0.148)	0.216 (0.217)	0.120 (0.122)	0.146 (0.148)	0.207 (0.206)

NOTE: Sephton's (2017) critical values are given in parentheses, while those from *kpsstest* are not.