Problem of Joint Unit Root Test and Inconsistent Estimators in Over-Differenced Model

Delroy Low

6/26/2021

1.0 Introduction to the Problem

Stationarity is one of the core assumptions that is required in most time series research methodologies. To this end, researchers often have to ensure the presence of stationarity in the time series variables that they're dealing with and the go-to options to conduct such examination would be using the Augmented Dickey-Fuller (*ADF*) Test introduced in the paper prepared by Said and Dickey (1984) and the Kwiatkowski-Phillips-Schmidt-Shin (*KPSS*) Test, by Kwiatkowski, Phillips, Schmidt and Shin (1992).

However, past studies have shown that both tests have problematic finite sample property and this renders to the practice of using both tests simultaneously in determining if a particular time series is indeed stationary. In order to demonstrate, consider a stochastic process, $\{y_t\}_{t=1}^{T}$, time series researchers tend to subject this series to both *ADF* and *KPSS* Test and would make conclusion based on the decision matrix as below:

```
##                      KPSS: Stationary KPSS: Non-Stationary
## ADF: Stationary      "Stationary"     "Non-Stationary"
## ADF: Non-Stationary  "Non-Stationary" "Non-Stationary"
```

Thus, using such *Join-Test* approach, researchers would declare a series as a stationary process *if and only if* both tests indicates so. Otherwise, the series would be treated as a unit root process and to be taken the first difference before moving to the next stage of the analysis. Example of published papers that employed such *Join-Test* approach would be Ha and Shin (2021), Bernardi and Ricciuti (2021), among others.

Intuitively, such practice narrows the likelihood of one concluding stationarity. Thus, this implies that researchers would have to be more tolerant toward the practice of over-differencing. Nevertheless, it is not theoretically justified and in fact only worsens the already poor finite sample performance of these tests, and this project is to demonstrate the downside of using such procedure and also to point out the undesired consequence of over-differencing by conducting Monte Carlo (*MC*) Simulations.

2.0 The Downside of *Join-Test* Approach and Monte Carlo (*MC*) Setup

Given a simple first-order autoregressive model as followed:

$$(1 - \phi L)y_t = \mu_y + \epsilon_t, \;\; t = 1, 2, \cdots, T \tag{1}$$

Let $\mu_y = E[y_t]$, L be the lag operator and $\{\epsilon_t\}_{t=1}^{T} \sim_{WN} (0, \sigma_\epsilon^2)$ with *WN* be the short for "White Noise". Then, a particular time series is stationary *if and only if* $|\phi| < 1$ and is a unit root process for when $|\phi| = 1$. To this end, we'll conduct a *MC* Simulation to examine the performance of *ADF*, *KPSS* and the *Join-Test* Approach in distinguishing a stationary from unit root processes.

The *MC* setup is as followed:

- Generate a generic *AR(1)* Model:

$$y_t = \phi y_{t-1} + \epsilon_t, \;\; t = 1, 2, \cdots, T \tag{2}$$

- Subject the generic series to three different testing procedures, and create a bernoulli variable, r that:

$$r = \begin{cases} 1, if \;\; correct \;\; classification \\ 0, if \;\; otherwise \end{cases} \tag{3}$$

Then, this process will be repeated for m = 1,000 *MC* trials. Here, this process will be conducted for parameters, T $\in$ {100, 200, 300, 400, 500} and $\phi \in \{0.850, 0.900, 0.925, 0.950, 0.975, 1.000\}$. Specifically, the smaller T is to mimic the scenario in which we've scarce observations and $\phi \to 1$ to reflect the behavior of these tests for when the autoregressive parameter appraoches the unit root boundary. Also note that, we've $r \in \mathcal{R}^{mX3}$, where each column in matrix r records the prediction of each of the three tests, namely, *ADF*, *KPSS* and *Join-Test* Approach in each *MC* trial. Thus, let r_{adf} denotes the column in matrix r that records the performance of *ADF* test, then, the probability of correct classification is then estimated as:

$$\mathcal{P}(\widehat{Correct}_{adf}) = \frac{\sum_{i=1}^{m} r_{adf,i}}{m} \tag{4}$$

Similar formula applies to compute the estimated probability of correct classification for *KPSS* and the *Join-Test* Approach. In order to replicate this study, R Packages, `tseries` and `tidyverse` are required.

2.1 Discussion

Figure 1 presents the estimated probability of correct classification for both *ADF* and *KPSS* test respectively. Both tests have poor performance for when the ϕ is approaching the unit root boundary and such finite sample problem has been pointed out by various researchers, for instance, DeJong, Nankervis, Savin and Whiteman (1992), Caner and Kilian (2001) and others. However, for small sample size, $T < 300$, *KPSS* test does provide a relatively greater compared to that of *ADF* test. In fact, *KPSS* test does provide a comparatively stable result than *ADF* test irrespective of the sample size. However, for sample size, $T \geq 300$, *ADF* test would be a more preferable option.

```
## `summarise()` has grouped output by 'Phi'. You can override using the `.groups` argument.
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

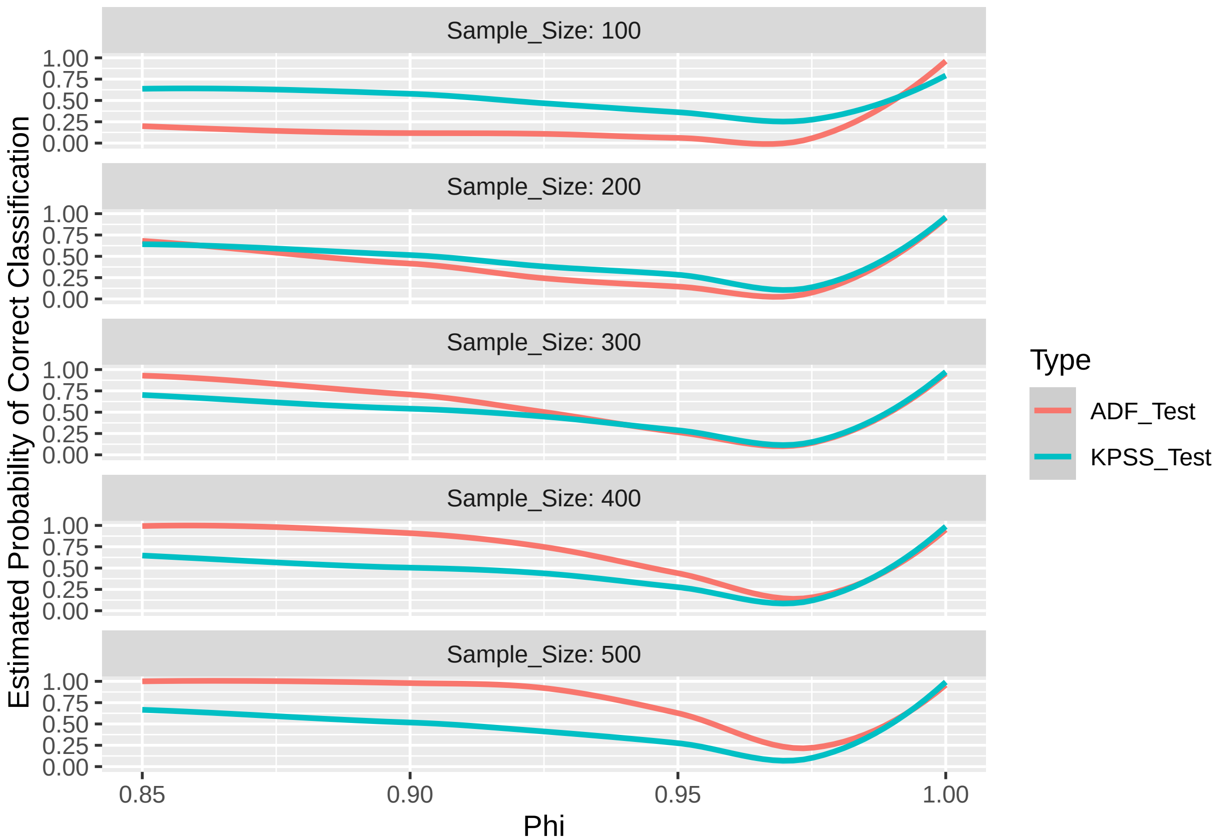

Figure 1: Performance of ADF and KPSS Test

Nevertheless, both *ADF* and *KPSS* test does have problem to indicate stationary series when $\phi \in [0.95, 1)$ To this end, researchers tend to rely on the result of both tests in identifying unit root series. Specifically, let p_i, for $i = adf, kpss$ as the *p-value* of the two respective tests; then, time series researchers tend to treat a series as unit root process for when $p_{adf} > \alpha \ \wedge \ p_{kpss} < \alpha$, $p_{adf} > \alpha \ \wedge \ p_{kpss} > \alpha$ or $p_{adf} < \alpha \ \wedge \ p_{kpss} < \alpha$, for a pre-specified significance level, α. Intuitively, such *Join-Test* approach increases the likelihood in which the researcher would detect unit root process; however, this is at the expense of reducing the classification accuracy for when the series is indeed stationary.

```
## `summarise()` has grouped output by 'Phi'. You can override using the `.groups` argument.
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Figure 2 compares the performance if a researcher were to rely on either *ADF* or *KPSS* test alone rather than adopting the *Join-Test* approach. For all sample sizes, the improvement of prediction at $\phi = 1$ is insignificant

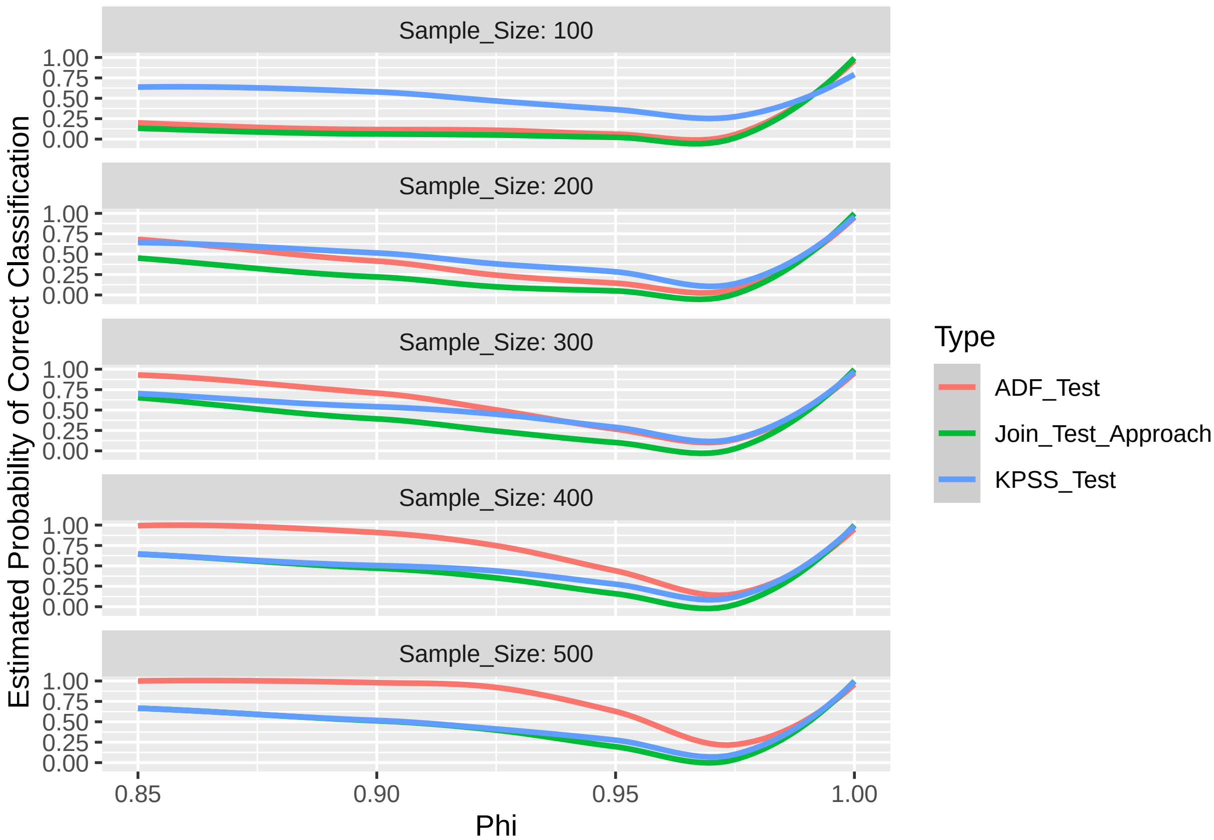

Figure 2: Comparing Join-Test Approach

compared to the fall in the prediction accuracy over the range of $\phi \in [0.85, 1)$. Such cost is more severe for when the sample size is small. In fact, with large sample size, the performance of *KPSS* as well as the *Join-Test* approach is not compatible to the one of *ADF* test. Thus, researchers should rely on either one of the two tests in detecting non-stationarity. Moreover, such *literal Join-Test* approach lacks theoretical justification; and there are researchers such as Charemza and Syczewska (1998) and Keblowski and Welfe (2004) who propose a more systematic approach that is theoretically justified to implement these tests jointly.

3.0 Consequence of Over-Differencing Problem and Monte Carlo (*MC*) Setup

It is also true that it is inevitable that one would commit to over-differencing a stationary series given the poor performance of these conventional unit root test in identifying the presence of non-stationary problem. In fact, it is also conventionally belief that over-differencing would not be as troublesome as under-differencing, that is to proceed to conduct time series analyses with non-stationary series. However, there are undesired consequences of such higher tolerant to the problem of over-differencing. Specifically, the *least square* estimators obtained will be an inconsistent. Suppose a simple *AR(1)* model as:

$$\begin{aligned} y_t &= \mu_y + \phi y_{t-1} + \epsilon_t, \quad t = 1, 2, \cdots, T \\ \Delta y_t &= \phi y_{t-1} - (\phi y_{t-2} + \epsilon_{t-1}) + \epsilon_t \\ \Delta y_t &= \phi \Delta y_{t-1} + \eta_t, \quad \eta_t = \Delta \epsilon_t \end{aligned}$$

Suppose the ordinary condition to ensure stationarity holds, that is $\phi < 1$ and $\epsilon_t \sim_{WN} (0, \sigma^2_\epsilon)$; then, one can prove that:

$$\hat{\phi} \rightarrow_p \frac{\phi - 1}{2} \tag{5}$$

Where, $\rightarrow_p$ symbolizes convergence in probability. In fact, such problem does not alleviate as sample size grows and and the downward finite sample bias becomes more severe as $\phi \rightarrow 1$. Such theoretical derived result does provide an explanation to the observation by Cochrane (2012).

A *MC* Simulation will be conducted in order to verify such asymptotic behavior derived. The *MC* setup is as followed:

- Generate a stationary *AR(1)* series as:

$$y_t = \phi y_{t-1} + \epsilon_t, \ t = 1, 2, \cdots, T \tag{6}$$

- Take the first difference of the model and obtain the the estimator, $\hat{\phi}$ in the following equation:

$$\Delta y_t = \hat{\phi} \Delta y_{t-1} + \Delta \epsilon_t \tag{7}$$

This process will be repeated for m = 2,500 *MC* trials. Also, we'll be considering parameters, $T \in \{100, 500, 1000\}$ and $\phi \in \{0.85, 0.90, 0.95, 0.99\}$. R Packages, `tidyverse`, `tseries` and `data.table` are required to replicate the following *MC* result.

##3.1 Discussion

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

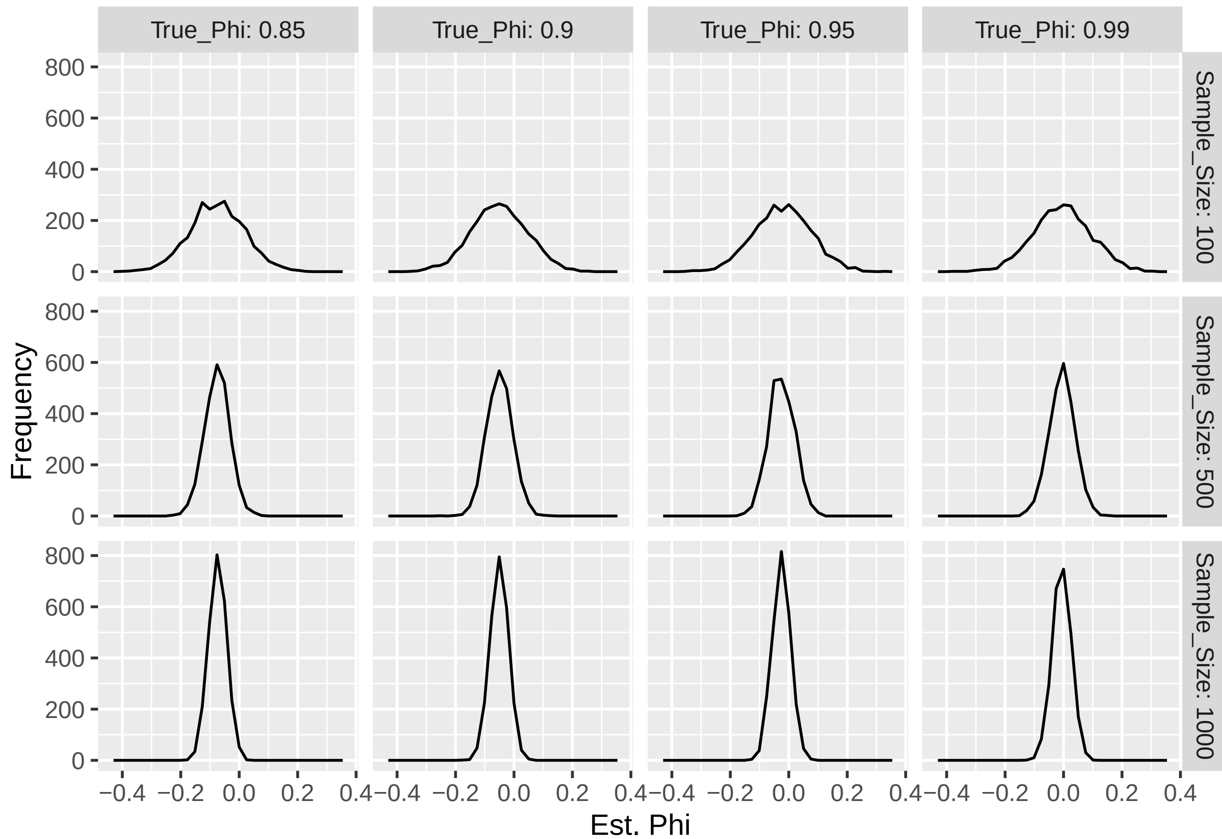

Figure 3: Fig. 3: Sampling Distribution

Figure 3 shows the sampling distribution of estimator, $\hat{\phi}$ under different combination of true coefficient, ϕ and sample size, T. It is obvious that the estimator, $\hat{\phi}$ does not converge to the true parameter, ϕ as T grows and therefore, they are inconsistent. In fact, comparing the actual mean of distribution to the calculated convergent as derived above:

```
## `summarise()` has grouped output by 'Sample_Size'. You can override using the `.groups` argument.
```

```
## # A tibble: 12 x 5
## # Groups:   Sample_Size [3]
##    Sample_Size True_Phi Mean_Est_Phi Calculated_Convergent Discrepancy
##          <dbl>    <dbl>        <dbl>                 <dbl>       <dbl>
##  1         100     0.85       -0.076                -0.075    -0.00100
##  2         100     0.9        -0.05                 -0.05      0
##  3         100     0.95       -0.021                -0.025     0.004
##  4         100     0.99       -0.005                -0.005     0
##  5         500     0.85       -0.076                -0.075    -0.00100
##  6         500     0.9        -0.049                -0.05      0.00100
##  7         500     0.95       -0.024                -0.025     0.00100
##  8         500     0.99       -0.005                -0.005     0
##  9        1000     0.85       -0.074                -0.075     0.00100
## 10        1000     0.9        -0.051                -0.05     -0.00100
## 11        1000     0.95       -0.025                -0.025     0
```

```
## 12         1000      0.99        -0.006                     -0.005    -0.001
```

Where $Calculated\_Convergent = \frac{\phi - 1}{2}$. Note that the discrepancy between the two statistics is very trifle even for small sample size, $T = 100$. Thus, this provides a sanitary check to the validity of the theoretical result that was driven above.

4.0 Conclusion and Recommendation

This project has shown that:

1. Joint Unit Root Test in the setting that is described as aforementioned is not reliable and it worsens the prediction accuracy if one would have relied on either *ADF* or *KPSS* test alone,
2. Greater tolerant to the problem of over-differencing is harmful in that it causes our *least square* estimator to be inconsistent.

Thus, some modification has to be done on these conventional unit root test for them to be more reliable in their practical application. To this end, bootstrap unit root test, check Palm, Smeekes, and Urbain (2008) or model-averaging unit root test such as the one proposed by Hansen and Racine (2018) might be some resources to be explored further. In fact, to encourage more usage of these bootstrapped unit root test in applied research, R Packages such as `bootUR` and `hr` have been created.

Reference

- Charemza, Wojciech W. and Syczewska, Ewa M. (1998). Joint application of the Dickey-Fuller and KPSS test. *Economics Letters*, Vol. 61, pp. 17-21.
- Phillips, P., Schmidt, P. and Shin, Y. (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*, Vol. 54, pp. 159-178.
- Said, Said E. and Dickey, David A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, Vol. 71, pp. 599-607.
- Ha, J. and Shin, Y. (2021). Lead-lag relationship between the shipping freight rate and agricultural commodity import price in Korea. *J Navig Port Res.*, Vol. 45, pp. 69-74.
- Bernardi, D. and Ricciuti, R. (2021). An Economic Analysis of 'Quota 90'. Working Paper Number 9, Department of Economics University of Verona.
- Dejong, David N., Nankervis, John C., Savin, N. and Whiteman, Charles H. (2002). The power problems of unit root test in time series with autoregressive errors. *Journal of Econometrics*, Vol. 53, pp. 323-343.
- Caner, M. and Kilian, L. (2001). Size distortions of tests of the null hypothesis of stationarity: evidence and implications for the PPP debate. *Journal of International Money and Finance*, Vol. 20, pp. 639-657.
- Kębłowski, P. and Welfe, A. (2004). The ADF–KPSS test of the joint confirmation hypothesis of unit autoregressive root. *Economics Letters*, Vol. 85, pp. 257-263.
- Cochrane, John H. (2018). A Brief Parable of Over-Di§erencing.
- Palm, Franz C., Smeekes, S. and Urbain, J. (2008). Bootstrap unit-root tests: Comparison and Extensions. *Journal of Time Series Analysis*, Vol. 29, pp. 371-401.
- Hansen, B. and Racine, J. (2018). Bootstrap model averaging unit root inference. Working Paper Number 2018-09, Department of Economics McMaster University.

Appendix 1: Proof of Inconsistency

Suppose a stationary *AR(1)* model, with *i.i.d.* white noise error series, $\epsilon_t \sim_{iid} (0, \sigma_\epsilon^2)$, then:

$$\begin{aligned} y_t &= \phi y_{t-1} + \epsilon_t, \quad t = 1, 2, \cdots, T \\ \Delta y_t &= \phi \Delta y_{t-1} + \eta_t, \quad \eta_t = \Delta \epsilon_t \end{aligned}$$

Then, for $|\phi| < 1$, we can express series, $\{y_t\}_{t=1}^T$ as a infinite sum of the white noise error, ϵ_t as $y_t = \sum_{i=0}^\infty \phi^i \epsilon_{t-i}$, thus:

$$\Delta y_t = \sum_{i=0}^\infty \phi^i \epsilon_{t-i} - \sum_{j=0}^\infty \phi^j \epsilon_{t-j-1} \tag{8}$$

In this case, we'll have $E[\Delta y_t] = 0, \forall t$, for $\epsilon_t \sim_{WN} (0, \sigma_\epsilon^2)$. Then, its variance, $\sigma_{\Delta y_t}^2$ can be computed as:

$$\begin{aligned} \sigma_{\Delta y_t}^2 &= \sum_{i=0}^\infty \phi^{2i} Var(\epsilon_{t-i}) + \sum_{j=0}^\infty \phi^{2j} Var(\epsilon_{t-j-1}) - 2 \sum_{i=0}^\infty \sum_{j=0}^\infty \phi^i \phi^j Cov(\epsilon_{t-i}, \epsilon_{t-j-1}) \\ &= \sigma_\epsilon^2 \{2(1-\phi^2)^{-1}) - 2\phi(1-\phi^2)^{-1}\} \\ &= 2\sigma_\epsilon^2 (1+\phi)^{-1}, \forall t \end{aligned}$$

Thus, in this case, we'll have:

$$\hat{\phi} \to_p \phi - \frac{\phi+1}{2} = \frac{\phi-1}{2} \tag{9}$$

Appendix 2: Summary Table for Figure 1 and Figure 2

```
## `summarise()` has grouped output by 'Phi'. You can override using the `.groups` argument.
```

```
##      Phi Sample_Size Correct_ADF Correct_KPSS Correct_Join
## 1  0.850         100         199          637          133
## 2  0.900         100         117          578           63
## 3  0.925         100         108          467           50
## 4  0.950         100          61          362           22
## 5  0.975         100          56          274           14
## 6  1.000         100         961          792          992
## 7  0.850         200         681          642          452
## 8  0.900         200         415          515          221
## 9  0.925         200         243          381          100
## 10 0.950         200         145          284           50
## 11 0.975         200          74          137           11
## 12 1.000         200         949          959          998
## 13 0.850         300         929          702          652
## 14 0.900         300         707          541          392
## 15 0.925         300         502          449          243
## 16 0.950         300         266          288          102
## 17 0.975         300         138          149           26
## 18 1.000         300         953          973          997
## 19 0.850         400         993          647          645
## 20 0.900         400         909          505          472
## 21 0.925         400         748          438          353
## 22 0.950         400         440          276          159
## 23 0.975         400         156          119           24
## 24 1.000         400         951          988          999
## 25 0.850         500        1000          666          666
## 26 0.900         500         979          517          512
## 27 0.925         500         921          412          398
## 28 0.950         500         627          275          195
## 29 0.975         500         220          101           38
## 30 1.000         500         958          992          998
```