The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–24

Testing and Correcting for Spatial Unit Roots in Regression Analysis

Sascha O. Becker
University of Warwick
Coventry, United Kingdom
s.o.becker@warwick.ac.uk

P. David Boll
University of Warwick
Coventry, United Kingdom
david.boll@warwick.ac.uk

Hans-Joachim Voth
University of Zurich
Zurich, Switzerland
voth@econ.uzh.ch

Abstract. Spatial unit roots can lead to spurious regression results. We present an overview of the methods developed in Müller and Watson (2024) to test and correct for spatial unit roots, and introduce a suite of Stata commands (`spur`) implementing these techniques. Our commands exactly replicate results in Müller and Watson (2024) using the same Chetty et al. (2014) data. As a guide for applied researchers, we provide a practical algorithm for regression analysis using these methods, alongside a simulated illustration in Stata.

Keywords: st0001, spurtest, spurtransform, spurious spatial regression, spatial unit roots

1 Introduction: Spatial unit roots

Spatial data present challenges for statistical analysis: observations that are close to each other geographically tend to be correlated, violating the assumption of independent and identically distributed (i.i.d.) errors. In such settings, heteroskedasticity and autocorrelation consistent (HAC) corrections or clustered standard errors at broader geographic levels (like states) are often used.

However, these correction methods fail when spatial dependence is too strong ("spatial unit roots"). Even with clustering or HAC corrections, spuriously significant regression coefficients can arise. Müller and Watson (2024) develop new statistical tests to detect such strong dependence and procedures to correct for it, extending techniques from time series analysis. We present a Stata implementation of their original Matlab code, along with practical guidelines for applied researchers.

In the time series context, weak serial correlation in the regressors and regression errors (the I(0) case) can be dealt with by HAC corrections. However, when the serial correlation is strong (the I(1) case), inference fails and OLS produces "spurious regressions" (Granger and Newbold 1974). Furthermore, test statistics behave in non-standard ways (Phillips 1986).

The spatial context is similar (Fingleton 1999), but as Müller and Watson (2022) discuss, there are also important differences: First, time series operate in a one-dimensional space, whereas in the spatial context, we are dealing with two (or three) dimensions. Second, in the time series context, observations are usually equally spaced (... $t-1$, t, $t+1$, ...) whereas in the spatial context, the location of observations on a map can

be substantially different from a uniform distribution on a grid. Third, while there is a directionality in the time series context (... $t-1$, t, $t+1$, ...), in the spatial context, going east is as natural as going west or north or south. Müller and Watson (2022) propose a method for constructing confidence intervals that account for many forms of spatial correlation. It uses a projection-type variance estimator, where the projection weights are *spatial correlation principal components* (hence called SCPC) from a given "worst case" benchmark correlation matrix.

Müller and Watson (2022) require stationarity of both regressors and dependent variables for the large sample validity of their SCPC method. In Müller and Watson (2023), they present a robust version that can deal with some nonstationarities relevant to applied research. The methods developed in these two papers have been implemented in Stata by the authors in their `scpc` package,[1] which provides a postestimation command to correct regression inference for weak spatial dependence. However, as Müller and Watson (2024) show, these and other spatial HAC methods cannot deal with the case of *strong* spatial auto-correlation in the outcome of interest. Müller and Watson (2024) introduce diagnostic tests for such spatial unit roots and show how transformations of the dependent and independent variables eliminate spurious regression results in the presence of strong spatial dependence.

In this article, we provide a Stata version of the programs developed by Müller and Watson (2024) to test for and correct for spatial unit roots. These methods can be used in conjunction with the `scpc` package to correct regression inference for remaining weak spatial dependence after spatial unit roots have been corrected, but our package does not depend on `scpc` and can be used independently. We show that our routines replicate the results in Müller and Watson (2024) using data from Chetty et al. (2014).

We also provide practical guidelines for applied researchers dealing with potential spatial unit roots in regression analysis: how to test for non-stationarity or the presence of spatial unit roots, and what to do in case non-stationarity is detected, or when the presence of spatial unit roots cannot be rejected. To illustrate this algorithm and the use of our Stata commands, we present a simulated example and a Monte Carlo simulation.

The rest of the article proceeds as follows: Section 2 summarizes and illustrates the tests developed by Müller and Watson (2024) to diagnose spatial unit roots, as well as our Stata implementation of their Matlab code in the commands `spurtest` and `spurhalflife`. Section 3 explains the spatial differencing techniques they propose to eliminate unit roots, and presents how they can be applied using the command `spurtransform`. Appendix A demonstrates the functionality of our implementation by replicating results from Müller and Watson (2024). Section 4 presents a brief guide to using these methods in common settings in applied research, illustrated by an example application. Section 5 concludes.

1. This is available from https://github.com/ukmueller/SCPC.

2 Testing for spatial unit roots

This section discusses the approaches to inference about the degree of spatial dependence developed by Müller and Watson (2024). They motivate their analysis of spatial unit roots by starting from the time series analogue: in time series, the canonical $I(1)$ process is a Wiener process (also called Brownian motion). Its extension to the (two-dimensional) spatial case is via a so-called Lévy–Brownian motion. Figure 1 illustrates the similarity between spurious regressions in the time series context and spatial context: Panel (a) shows realizations of two independent Gaussian random walks, (b) shows independent simulated spatial unit root processes over $n = 722$ U.S. commuting zones. In each case, we report the R^2 and t-statistic from the linear regression (with HAC correction) of the first on the second process, which show spuriously significant correlation in both cases. Panel (c) shows two variables from Chetty et al. (2014): their outcome variable (mobility index) and one regressor (teen labor force participation). These resemble the unit-root processes in panel (b). This highlights the potential relevance of strong spatial auto-correlation, which needs to be detected and addressed in empirical work.

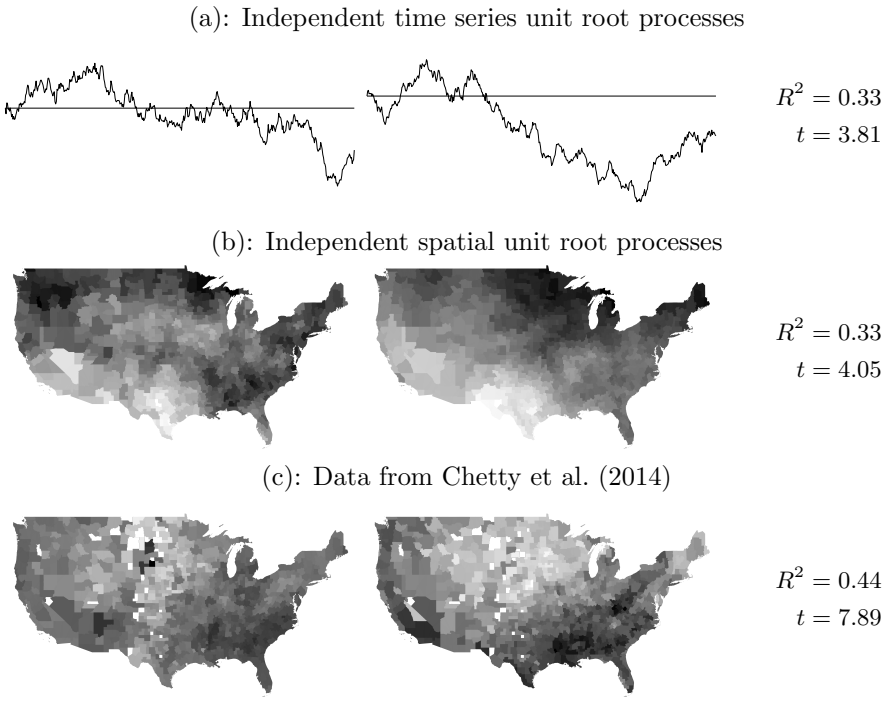

Figure 1: Spurious correlations with unit roots

Notes. – This figure is adapted from Figure 1 in Müller and Watson (2024); we thank Ulrich Müller and Mark Watson for kindly granting us permission for this.

Specifically, Müller and Watson (2024) develop four diagnostic tests, examining the following null hypotheses, respectively:

1. H_0: Scalar variable y is $I(1)$
2. H_0: Scalar variable y is $I(0)$
3. H_0: Linear regression residuals u are $I(1)$
4. H_0: Linear regression residuals u are $I(0)$

as well as a method to construct confidence intervals for the spatial half-life of a scalar variable. All of these tests exploit the different variance-covariance structures implied respectively by the canonical spatial $I(1)$ and local-to-unity (LTU) models, as defined by Müller and Watson (2024).

The canonical spatial $I(1)$ model is Lévy-Brownian motion, a spatial generalization of the Wiener process (Brownian motion) common in time series analysis. This can be thought of as a continuous-time analogue of a random walk. Conversely, LTU models describe stationary processes with weak mean reversion governed by a parameter $c > 0$. They are a generalization of the pure unit root model, in which the autoregressive root approaches unity as the sample size increases, at a rate determined by c. This allows them to behave very similarly to $I(1)$ processes for small c and very similarly to weakly dependent $I(0)$ processes for large c. Thus, they span a continuum of dependence between the dichotomous $I(0)$ and $I(1)$ cases. Their canonical form is the Ornstein-Uhlenbeck process, which can be thought of as a continuous-time analogue of an $AR(1)$ process in the time series context. The variance-covariance structure of these two canonical models in the spatial case is given by

$$\begin{aligned} \text{Canonical } I(1) \text{ model:} \quad & y_l = L(s_l), \quad E[L(s)L(r)] = \frac{1}{2}\left(|s| + |r| - |s-r|\right) \\ \text{Canonical LTU model:} \quad & y_l = J_c(s_l), \quad E[J_c(s)J_c(r)] = \exp\left[-c|s-r|\right]/(2c), \end{aligned}$$

where l indexes locations, s, r denote locations in space, $|x| = \sqrt{x'x}$, $L(\cdot)$ is Lévy-Brownian motion and $J_c(\cdot)$ is the spatial generalization of the Ornstein-Uhlenbeck process with mean-reversion parameter $c > 0$. These canonical processes provide asymptotic approximations for more general models[2] (see Theorem 2 in Müller and Watson 2024), and their properties can thus be used to discriminate between $I(1)$ and $I(0)$ processes.

2.1 Low-frequency weighted averages

The fundamental idea is to compare the performance of these two models in rationalizing the data. Rather than performing tests on the raw data, Müller and Watson (2024) build

2. "More general" in this case refers to models where the innovations are not necessarily white noise, but more general stationary processes. In the discrete-time time series context, this is comparable to the relationship between a random walk as the "canonical" unit root model with white noise increments to the "more general" ARIMA model with ARMA noise (Müller and Watson 2024).

on Müller and Watson (2008) and compute the test statistics from a fixed number q of *weighted averages* of the data. Specifically, given a data vector $\mathbf{y} = (y_1, \ldots, y_n)'$, define $\mathbf{\Sigma_L}$ as the $n \times n$ covariance matrix of $\mathbf{y}$ implied by the canonical $I(1)$ model (Lévy-Brownian motion $L(\cdot)$). In other words, $\mathbf{\Sigma_L}$ is the theoretical covariance matrix of the data under the $I(1)$ model. From this, derive $\mathbf{R}$ as the $n \times q$ matrix whose columns are the eigenvectors of $\mathbf{M\Sigma_L M}$ corresponding to the q largest eigenvalues, where $\mathbf{M} = \mathbf{I}_n - \mathbf{1}(\mathbf{1}'\mathbf{1})^{-1}\mathbf{1}'$ is the demeaning matrix, and scaled such that $n^{-1}\mathbf{R}'\mathbf{R} = \mathbf{I}_q$. Then, the weighted averages are computed as

$$\mathbf{z} = \mathbf{R}'\mathbf{M}\mathbf{y} = \mathbf{R}'\mathbf{y}$$

The j-th ($j = 1, \ldots, q$) weighted average is the linear combination of the data with the j-th largest population variance under the canonical $I(1)$ model; that is, the scalar z_j is the j-th largest principal component of $\mathbf{My}$ based on the assumed covariance matrix $\mathbf{M\Sigma_L M}$. As illustrated below, and discussed in detail in Müller and Watson (2019) for the time series case, this choice of weights extracts and summarizes *low-frequency* variation in the data.

Basing the tests on these weighted averages is useful in two broad ways: First, summarizing the data in a fixed number of averages yields an asymptotically multivariate (q-dimensional) *normal* distribution (following from a central limit theorem), which enables the use of standard inference methods. The covariance matrix of this limiting distribution is simply

$$\mathrm{Var}(\mathbf{z}) = \mathbf{R}'\mathbf{\Sigma}\mathbf{R} \equiv \mathbf{\Omega}$$

where $\mathbf{\Sigma}$ is the covariance matrix induced by the data generating process. For the purposes of this paper, $\mathbf{\Sigma}$ will be the covariance matrix implied by one of the two canonical models discussed above, which we denote by $\mathbf{\Sigma_L}$ for the $I(1)$ model and $\mathbf{\Sigma}(c)$ for the LTU model with decay parameter c. They imply different covariance structures $\mathrm{Var}(\mathbf{z})$, henceforth denoted as $\mathbf{\Omega_L}$ and $\mathbf{\Omega}(c)$. This is exploited to discriminate between broad models of persistence, which reduces to a standard problem of inference about the covariance matrix under normality.[3] Second, choosing the weights to extract only low-frequency variation makes the resulting tests robust to misspecification of the high-frequency variation: the accuracy of the approximations derived from the canonical models in finite samples now does not depend (much) on the ability of those models to match the high-frequency behavior of the data generating process. See Müller and Watson (2019) for a more extensive discussion.

Choice of q. An obvious practical question is how to choose the number of weighted averages q. This requires a trade-off: a large q increases the amount of data used in the tests, increasing power, but also makes the tests more sensitive to high-frequency noise in the data. Müller and Watson (2024) argue that a q between 10 and 20 captures most of the relevant low-frequency variation, and use $q = 15$ in their applications. Our

3. As discussed before, these canonical processes asymptotically approximate more general DGPs, and are thus a useful benchmark for inference (see Theorem 2 and Section 4.6 in Müller and Watson 2024).

numerical simulations show that the $q = 10$ ($q = 15$) [$q = 20$] largest eigenvectors capture ca. 85% (87%) [90%] of the variation in simulated LBM processes, respectively, while $q = 30$ ($q = 50$) only increases this share slightly to 92% (94%). In our Stata package, all test commands include the option `, q()`. We set `q(15)` as the default, and also recommend that users test the robustness of their results to different choices of q.

Illustration of weighted averages. We illustrate the construction of the weighted averages in a simple example. We randomly draw $n = 3000$ locations from a uniform distribution on the unit square, with coordinates s_l, $l = 1, \ldots, n$. The covariance matrix induced by Lévy-Brownian motion for these locations is then given by $\mathbf{\Sigma_L}$, where the (l, ℓ)-th element is $\frac{1}{2}(|s_l| + |s_\ell| - |s_l - s_\ell|)$. From there, it is straightforward to compute the eigenvectors of $\mathbf{M\Sigma_L M}$. The subplots of Figure 2 show the eigenvectors corresponding to the 1st, 2nd, 3rd, 4th, 10th, 15th, 20th and 50th highest eigenvalues, respectively, where the color of location l on the map indicates the value of the l-th element of the respective eigenvector. The "frequency" of the variation clearly increases

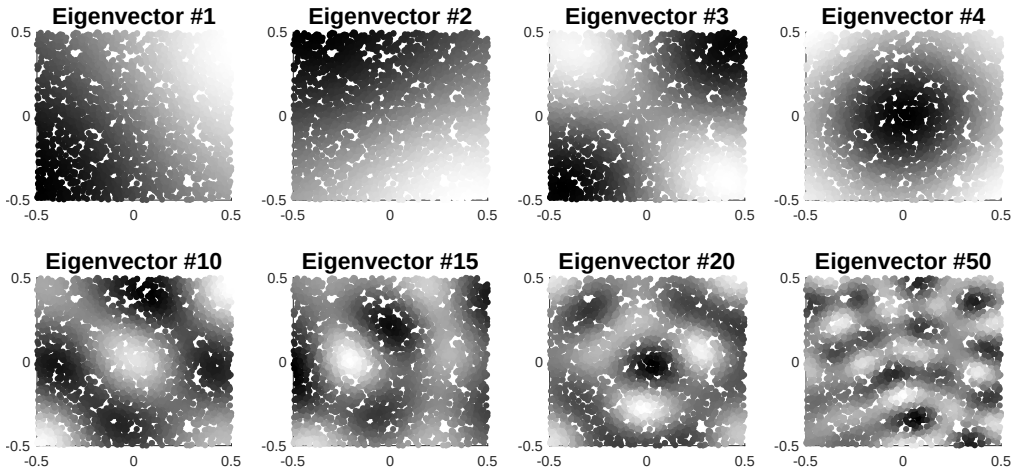

Figure 2: Illustration of the weights

with the order of the eigenvectors. To see how $\mathbf{z}$ extracts low-frequency variation from $\mathbf{y}$, notice that $\mathbf{z} = \mathbf{R}'\mathbf{y} = n(\mathbf{R}'\mathbf{R})^{-1}\mathbf{R}'\mathbf{y}$, since $n^{-1}\mathbf{R}'\mathbf{R} = \mathbf{I}_q$ by construction. $\mathbf{z}$ can therefore also be understood as coefficients (loadings) from projections of $\mathbf{y}$ on the q largest eigenvectors of $\mathbf{M\Sigma_L M}$. Inspecting the behavior of these eigenvectors in Figure 2 makes clear how this captures low-frequency variation.

This is further illustrated by the subplots of Figure 3: The first two subplots show simulated data for an LBM process, $\mathbf{y}_{\text{LBM}} \sim N(\mathbf{0}, \mathbf{\Sigma_L})$, and an LTU process with much lower persistence, $\mathbf{y}_{\text{LTU}} \sim N(\mathbf{0}, \mathbf{\Sigma}(10))$, respectively. The difference in low-frequency variation is clearly visible. The third subplot illustrates how the weighted averages discussed above can be used for inference about spatial persistence. The black lines show the absolute values of the elements of $\mathbf{z}_{\text{LBM}} = \mathbf{R}'\mathbf{y}_{\text{LBM}}$ and $\mathbf{z}_{\text{LTU}} = \mathbf{R}'\mathbf{y}_{\text{LTU}}$,

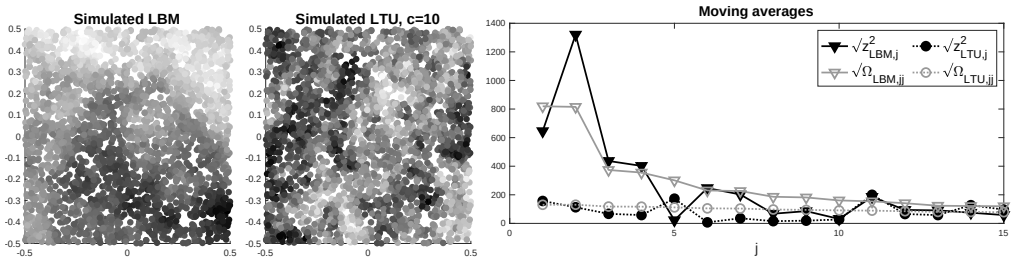

Figure 3: Simulated data and weighted averages

respectively, where $\mathbf{R}$ collects the eigenvectors of $\mathbf{M\Sigma_L M}$ corresponding to the $q = 15$ largest eigenvalues (so that $j = 1$ is the largest eigenvector, $j = 2$ the second largest, etc.). The difference in behavior is stark: the LBM process loads heavily on the first few eigenvectors (low frequencies) and then quickly decays, while the LTU process loads evenly across the spectrum. This empirical behavior can be compared to the expected behavior of $\mathbf{z}$ under the two models, given by $\mathbf{z}_{\text{LBM}} \sim N(\mathbf{0}, \mathbf{\Omega}_{\text{LBM}})$ and $\mathbf{z}_{\text{LTU}} \sim N(\mathbf{0}, \mathbf{\Omega}_{\text{LTU}})$, respectively, where $\mathbf{\Omega}_{\text{LBM}} = \mathbf{R}'\mathbf{\Sigma_L R}$ and $\mathbf{\Omega}_{\text{LTU}} = \mathbf{R}'\mathbf{\Sigma}(10)\mathbf{R}$. This implies that $E[\mathbf{z}_j^2] = \mathbf{\Omega}_{j,j}$ for the respective model, shown by the gray lines. By construction, $\mathbf{\Omega}_{\text{LBM}}$ describes the behavior of $\mathbf{z}_{\text{LBM}}$ much better than that of $\mathbf{z}_{\text{LTU}}$, and vice versa. The next sections formalize such comparisons to distinguish between $I(1)$ and $I(0)$ processes.

2.2 Generic testing procedure

Given the weighted averages $\mathbf{z}$ whose limiting distribution is multivariate normal, inference boils down to testing hypotheses about its covariance matrix $\mathbf{\Omega}$. In all tests, the hypotheses are of the form

$$H_0 : \mathbf{\Omega} = \mathbf{\Omega}_0 \quad \text{vs.} \quad H_a : \mathbf{\Omega} = \mathbf{\Omega}_a$$

Müller and Watson (2024) suggest to use the likelihood ratio test statistic of $\mathbf{z}/\sqrt{\mathbf{z}'\mathbf{z}}$

$$\frac{\mathcal{L}(\mathbf{\Omega}_a \mid \mathbf{z})}{\mathcal{L}(\mathbf{\Omega}_0 \mid \mathbf{z})} \propto \frac{\mathbf{z}'\mathbf{\Omega}_0^{-1}\mathbf{z}}{\mathbf{z}'\mathbf{\Omega}_a^{-1}\mathbf{z}} \equiv \Lambda$$

with critical value CV that solves

$$\Pr(\Lambda > CV \mid H_0) = \alpha$$

By the Neyman-Pearson lemma, this is the most powerful level α scale invariant test. In practice, the critical value is computed by

1. drawing N_{rep} random $q \times 1$ vectors $\hat{\mathbf{z}}$ from the distribution $N(\mathbf{0}, \mathbf{\Omega}_0)$,
2. computing the test statistic $\hat{\Lambda} = \hat{\mathbf{z}}'\mathbf{\Omega}_0^{-1}\hat{\mathbf{z}}/\hat{\mathbf{z}}'\mathbf{\Omega}_a^{-1}\hat{\mathbf{z}}$ for each draw,

3. setting CV as the empirical $1-\alpha$ quantile of the resulting distribution of $\hat{\Lambda}$.

The test then rejects H_0 if $\Lambda > CV$.[4] All test commands in our package include the option `, nrep()`, which sets the sample size N_{rep} for the Monte Carlo simulation. The default is `nrep(100000)`.

2.3 I(1) test

The $I(1)$ test examines the presence of a unit root in a scalar variables y, i.e. the $I(1)$ model against the LTU model. The hypotheses are therefore

$$H_0 : \mathbf{\Omega} = \mathbf{\Omega}_L = \mathbf{R}'\mathbf{\Sigma}_L\mathbf{R} \quad \text{vs.} \quad H_a : \mathbf{\Omega} = \mathbf{\Omega}(c_a) = \mathbf{R}'\mathbf{\Sigma}(c_a)\mathbf{R}$$

where $\mathbf{\Sigma}_L$ is the covariance matrix implied by the canonical $I(1)$ model and $\mathbf{\Sigma}(c_a)$ is the covariance matrix implied by the LTU model with mean-reversion parameter c_a. The choice of c_a determines the power of the test across the alternative hypothesis space $c > 0$. No uniformly most powerful test exists, so Müller and Watson (2024) propose setting c_a such that a level 5% test has 50% power, following King (1987).[5] The test statistic,[6] following the discussion in Section 2.2, is

$$\text{LFUR} = \frac{\mathbf{z}'\mathbf{\Omega}_L^{-1}\mathbf{z}}{\mathbf{z}'\mathbf{\Omega}^{-1}(c_a)\mathbf{z}}$$

and the test rejects H_0 if LFUR is larger than the critical value (computed as described in Section 2.2).

2.4 I(0) test

Testing the $I(0)$ null hypothesis, i.e. spatial stationarity, is not as straightforward: the LTU model, as discussed in Section 1, is similar to an $I(1)$ process for small c, and similar to an $I(0)$ process for large c. Therefore, to specify an $I(0)$ null hypothesis, one must take a stance on the value of c that separates the two. Müller and Watson (2024) propose to set this value to $c_{0.03}$, defined as the value of c such that the average pairwise correlation induced by $\mathbf{\Sigma}(c)$ is 0.03.[7] They then propose the hypotheses

$$H_0 : \mathbf{\Omega} = \mathbf{\Omega}(c),\ c \geq c_{0.03} \quad \text{vs.} \quad H_a : \mathbf{\Omega} = \mathbf{\Omega}(c) + g_a^2\mathbf{\Omega}_L,\ g_a > 0$$

4. P-values are computed as $\sum_i^{N_{rep}} \mathbf{1}[\hat{\Lambda}_i > \Lambda]/N_{rep}$
5. In practice, this is achieved through Monte Carlo simulation: For a given set of locations and some value c_a, $\mathbf{\Sigma_L}$ and $\mathbf{\Sigma}(c_a)$ are known theoretical objects. The critical value for a given level (here 5%) is computed as described in Section 2.2, and power can be computed in an analogue fashion, drawing data from the alternative distribution instead of the null distribution. This is then repeated for different c_a values until an approximate solution to $\text{Power}(c_a) = 0.5$ is found.
6. Müller and Watson (2024) label the statistic LFUR in reference to the Low Frequency Unit Root statistic in Müller and Watson (2008). Similarly, the I(0) test statistic is labelled LFST in reference to their Low Frequency Stationarity test.
7. See Müller and Watson (2024) for details.

where the alternative hypothesis is a mixture of the $I(0)$ and $I(1)$ models, which gets closer to the $I(1)$ model as g_a increases. g_a thus plays the same role as c_a for the $I(1)$ test in controlling the distance between the null and alternative hypotheses, and its choice determines the power profile of the test for different levels of persistence. Müller and Watson (2024) propose to set this value analogously to c_a by targeting 50% power. To construct a test statistic in the form of Section 2.2, we need simple hypotheses, which in turn requires a choice for c: Müller and Watson (2024) suggest that setting $c = c_{0.001}$ under both H_0 and H_a and thus computing the test statistic

$$\text{LFST} = \frac{\mathbf{z}'\mathbf{\Omega}(c_{0.001})^{-1}\mathbf{z}}{\mathbf{z}'[\mathbf{\Omega}(c_{0.001}) + g_a^2\mathbf{\Omega}_L]^{-1}\mathbf{z}}$$

yields a test that works well for a wide range of $c \geq c_{0.03}$. The test rejects H_0 if LFST is larger than the critical value (computed as described in Section 2.2, with the modification that first the critical value is computed for a range of values $c \geq c_{0.03}$, and then the highest of those values is used to compare to the test statistic).

2.5 I(1) and I(0) tests for regression residuals

In many practical applications, the econometrician wants to test the persistence of the errors of a regression model $y_l = \mathbf{x}_l'\beta + u_l$. With β unknown and its estimates biased in the presence of unit roots, u_l is unobserved and thus the previous tests cannot be directly applied. Müller and Watson (2024) propose a simple solution for the case where $\mathbf{u}$ is independent of $\mathbf{X}$, which is to condition on $\mathbf{X}$ in the construction of the weighted averages:

$$\mathbf{z}_X = \mathbf{R}_X\mathbf{y}$$

where $\mathbf{R}_X$ collects the eigenvectors of $\mathbf{M}_X\mathbf{\Sigma}_L\mathbf{M}_X$ corresponding to the largest q eigenvalues, and $\mathbf{M}_X = \mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}$. Then, the LFUR and LFST statistics can be computed as before, with $\mathbf{z}_X$ instead of $\mathbf{z}$.

2.6 The spurtest command

All four tests described in the previous sections are implemented in the Stata command `spurtest`, which has four versions for the four different tests.

Syntax

`spurtest i1` *varname* [*if*] [*in*] [`, q(#) nrep(#) latlong`]
`spurtest i0` *varname* [*if*] [*in*] [`, q(#) nrep(#) latlong`]

In each case, *varname* is the numerical variable to be tested for stationarity.

`spurtest i1resid` *depvar* [*indepvars*] [*if*] [*in*] [`, q(#) nrep(#) latlong`]

`spurtest i0resid` *depvar* [*indepvars*] [*if*] [*in*] [`, q(#) nrep(#) latlong`]

In each case, *depvar* is the numerical dependent variable, and *indepvars* are the numerical independent variables of the regression model (a constant is always included).

All our commands require that the variables containing the spatial coordinates are named `s_1`, `s_2`, ..., `s_p`. This is for consistency with the `scpc` command developed by Müller and Watson (2022, 2023), which we use below. If the option `latlong` is specified, `s_1` is interpreted as latitude and `s_2` as longitude, and no other `s_*` variables may be present. If the option is not specified, the p `s_*` variables present are interpreted as coordinates in $p-$dimensional Euclidean space.

Most of the code underlying this and the other commands in our package is written in Mata. We provide an `.mlib` library of compiled Mata functions, which is required to run the commands.[8] This is installed automatically when following the installation instructions in Section 7. Further, this and all other commands in this package rely on the `moremata` package (Jann 2005).

Options

`q(#)` specifies the number of weighted averages to be used in the test. The default is `q(15)`.

`nrep(#)` specifies the number of Monte Carlo draws to be used to simulate the distribution of the test statistic. The default is `nrep(100000)`.

`latlong` specifies that the spatial coordinates are given in latitude (stored in `s_1`) and longitude (stored in `s_2`) (see above).

Stored results

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

Scalars	
`r(teststat)`	Test statistic (LFUR or LFST)
`r(p)`	P-value of the test
`r(ha_param)`	Parameter for alternative hypothesis (c_a or g_a)
Matrices	
`r(cv)`	Critical values at 1%, 5%, and 10% levels

2.7 Confidence sets for spatial half-life and the spurhalflife command

For completeness, we also implement a method proposed in Müller and Watson (2024) to construct confidence sets for the spatial half-life of a process, that is, the spatial distance at which the correlation in the process is equal to 1/2. In the local-to-unity framework, this is directly connected to the parameter c, specifically the half-life h is equal to $\ln 2/c$. Confidence intervals can then be constructed as the sets of values of

8. The source code for the Mata functions is available at https://github.com/pdavidboll/SPUR.

h for which the null hypothesis $H_0 : h_0 = h$ cannot be rejected. The test statistic suggested by Müller and Watson (2024)

$$\frac{\int_0^{\Delta_{\max}} \det\big(\boldsymbol{\Omega}(\ln 2/h)\big)^{-1/2}\big(\mathbf{z}'\boldsymbol{\Omega}(\ln 2/h)^{-1}\mathbf{z}\big)^{-q/2}\,dh}{\big(\mathbf{z}'\boldsymbol{\Omega}(\ln 2/h_0)^{-1}\mathbf{z}\big)^{-q/2}}.$$

compares how well the data fit under H_0 to their average fit across a range $[0, \Delta_{\max}]$ of alternative values of h, where $\Delta_{\max}$ is the maximum pairwise distance of the sample locations, with the weighting chosen to be uniform in h. For a given h_0, a critical value $CV(h_0)$ for this test statistic can be computed using Monte Carlo simulation as described in Section 2.2 by drawing from the null distribution $\mathbf{z} \sim N(\mathbf{0}, \boldsymbol{\Omega}(\ln 2/h_0))$ and computing the $1-\alpha$ quantile of the resulting distribution. Comparing the test statistic based on data to this critical value yields a test of $H_0 : h_0 = h$. Repeating this for a grid of values h_0 and collecting all values that are not rejected then yields a $100(1-\alpha)\%$ confidence set for h. For further details we refer the interested reader to Section 4.4 of Müller and Watson (2024).

Syntax

`spurhalflife` *varname* [*if*] [*in*] [`, q(#) nrep(#) level(#) latlong normdist`]

varname is the numerical variable whose spatial half-life is of interest.

The variables containing the spatial coordinates must be named `s_1`, `s_2`, ..., `s_p`. (See explanation in Section 2.6.)

Options

`q(#)` specifies the number of weighted averages to be used in the test. The default is `q(15)`.

`nrep(#)` specifies the number of Monte Carlo draws to be used to simulate the distribution of the test statistic. The default is `nrep(100000)`.

`level(#)` specifies the desired confidence level in percent. The default is `level(95)`.

`latlong` specifies that the spatial coordinates are given in latitude (stored in `s_1`) and longitude (stored in `s_2`) (see above).

`normdist` specifies that the results are to be returned as fractions of the maximum pairwise distance in the sample. Otherwise, they are returned in meters (if `latlong`) or the units of the original Euclidean coordinates (if not `latlong`).

Stored results

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

Scalars

`r(ci_l)`	Lower bound of confidence interval
`r(ci_u)`	Upper bound of confidence interval
`r(max_dist)`	Maximum pairwise distance in the sample

3 Correction through spatial differencing and the spurtransform command

Having tested for and found evidence of the presence of spatial unit roots, the econometrician needs a way to correct for them in order to be able to estimate regression coefficients consistently. The standard approach in the time series literature is to take first differences of the data:

$$y_t = y_{t-1} + \epsilon_t$$
$$\Delta y_t = y_t - y_{t-1} = \epsilon_t$$

which yields a stationary process that can be used in regressions. The equivalent transformation in the spatial context is not obvious: observations in space cannot be ordered in the way that a time series can, and they are unevenly spaced, so which value to subtract from each observation is not clear. Müller and Watson (2024) propose four possible linear transformations. The last of them they find to be the most powerful in their simulations. The following presents all four and illustrates their effects using the simulated LBM from Section 2.1. Throughout, the vectors $\mathbf{y} = (y_1, \ldots, y_n)$ and $\mathbf{y}^* = (y_1^*, \ldots, y_n^*)$ refer to the raw and transformed data vectors, respectively. Further, $\mathbf{H} = \mathbf{I} - \tilde{\mathbf{H}}$ refers to the respective transformation matrix, such that $\mathbf{y}^* = \mathbf{H}\mathbf{y} = \mathbf{y} - \tilde{\mathbf{H}}\mathbf{y}$.

Nearest Neighbor (NN) Differences

One obvious differencing procedure would be

$$y_l^* = y_l - y_{\ell(l)}$$

where $s_{\ell(l)}$ is the location nearest to s_l. This is equivalent to

$$\mathbf{y}^* = \mathbf{H}_{\text{NN}}\mathbf{y} = (\mathbf{I}_n - \tilde{\mathbf{H}}_{\text{NN}})\mathbf{y}$$

where $\tilde{\mathbf{H}}_{\text{NN},lj} = 1$ if $j = \ell(l)$ and 0 otherwise.

Isotropic Differences

Instead of taking differences only with respect to the nearest neighbor, another option would be to subtract the mean of all observations in a neighborhood of radius b:

$$y_l^* = y_l - \bar{y}_l(b)$$

where

$$\bar{y}_l(b) = \frac{1}{m_l(b)} \sum_{j \neq l} \mathbf{1}[|s_l - s_j| < b] y_j$$
$$m_l(b) = \sum_{j \neq l} \mathbf{1}[|s_l - s_j| < b]$$

This is equivalent to

$$\mathbf{y}^* = \mathbf{H}_{\text{ISO}}\mathbf{y} = (\mathbf{I}_n - \tilde{\mathbf{H}}_{\text{ISO}})\mathbf{y}$$

where $\tilde{\mathbf{H}}_{\text{ISO},lj} = m_l(b)^{-1}\mathbf{1}[|s_l - s_j| < b]y_j$ for $j \neq l$ and 0 for $j = l$.

Clustered demeaning

A third option is to partition the data into K clusters and subtract the mean within its cluster from each observation (or, equivalently, including cluster fixed effects in the regressions). These clusters could be based on knowledge of the structure of the data (e.g., states), or constructed through techniques like k-means clustering. The transformed data is then

$$y_l^* = y_l - \bar{y}_{k(l)}$$

where

$$\bar{y}_{k(l)} = \frac{1}{m_{k(l)}} \sum_{j} \mathbf{1}[k(j) = k(l)] y_j$$
$$m_{k(l)} = \sum_{j} \mathbf{1}[k(j) = k(l)]$$

and $k(l)$ is the cluster that l belongs to. This is equivalent to

$$\mathbf{y}^* = \mathbf{H}_{\text{CL}}\mathbf{y} = (\mathbf{I}_n - \tilde{\mathbf{H}}_{\text{CL}})\mathbf{y}$$

where $\tilde{\mathbf{H}}_{\text{CL},lj} = m_{k(l)}^{-1}\mathbf{1}[k(j) = k(l)]y_j$.

LBM-GLS transformation

The previous three transformations are ad hoc ways of correcting strong spatial dependence. Following their characterization of spatial unit root processes as approximated by Lévy-Brownian motion, Müller and Watson (2024) propose a GLS transformation based on the covariance matrix induced by LBM. Recall that, under LBM, the demeaned data are distributed as $\mathbf{y} \sim N(0, \mathbf{M}\boldsymbol{\Sigma}_L\mathbf{M})$. The standard GLS transform is then

$$\begin{aligned}\mathbf{y}^* &= (\mathbf{M}\boldsymbol{\Sigma}_L\mathbf{M})^{-1/2}\mathbf{y} \\ &\equiv \mathbf{H}_{\text{LBMGLS}}\mathbf{y} \equiv (\mathbf{I}_n - \tilde{\mathbf{H}}_{\text{LBMGLS}})\mathbf{y}\end{aligned}$$

where $(\mathbf{M}\boldsymbol{\Sigma}_L\mathbf{M})^{-1/2}$ is the Moore-Penrose inverse of $(\mathbf{M}\boldsymbol{\Sigma}_L\mathbf{M})^{1/2}$. To see how this transformation can be described as "spatial differencing", it is useful to relate this back to the time series case: It is easy to show that taking first differences of any evenly spaced time series is exactly equivalent to a (particular) GLS transformation based on the covariance matrix of a standard random walk. The LBM-GLS transformation translates this logic to the multidimensional spatial case, using the LBM covariance matrix. Figure 4 further illustrates the effects of the transformation.

Figure 4 illustrates all four transformations. The single plot at the top is the "raw" data used for this illustration, which is the simulated LBM process from Figure 3. The four columns below show the four described transformations, respectively. Within each column, the top panel illustrates the transformation for one single data point (in red): the blue dots are the data points whose weighted values are subtracted from the red point, with a stronger blue indicating a larger weight. In the NN transformation, only the closest neighbour is subtracted. In the isotropic and cluster transformations, an unweighted mean of surrounding observations is subtracted. The LBMGLS transformation subtracts a weighted mean of all surrounding observations, with weights quickly decaying with distance. The middle panel shows the values which are subtracted from the raw data ($\hat{\mathbf{H}}\mathbf{y}$), and the bottom panel shows the transformed data ($\mathbf{H}\mathbf{y}$).[9]

Syntax

`spurtransform` *varlist* [*if*] [*in*] `,` `prefix(`*string*`)` [`transformation(`*string*`)` `radius(#)` `clustvar(`*varname*`)` `latlong` `replace` `separately`]

varlist is the list of variables to be transformed. The transformed variables will be stored under the original variables names prefixed with `prefix`. If *varlist* contains several variables, they are all transformed using the same matrix $\mathbf{H}$, meaning that only observations where *all* specified variables are non-missing will be included. To override this behavior, specify the option `separately`.

The variables containing the spatial coordinates must be named `s_1`, `s_2`, ..., `s_p`. (See explanation in Section 2.6.)

Options

`prefix(`*string*`)` specifies the prefix for the variable names under which the transformed data will be stored.

`transformation(`*string*`)` specifies the type of transformation. Must be one of `nn, iso, cluster, lbmgls`. Defaults to `lbmgls`.

`radius(#)` specifies the radius in metres (if `latlong`), or in the units of the original coordinates (if not `latlong`), which is to be used for isotropic differencing (b in the

9. The cluster transformation uses $K = 200$ clusters constructed through k-means clustering.

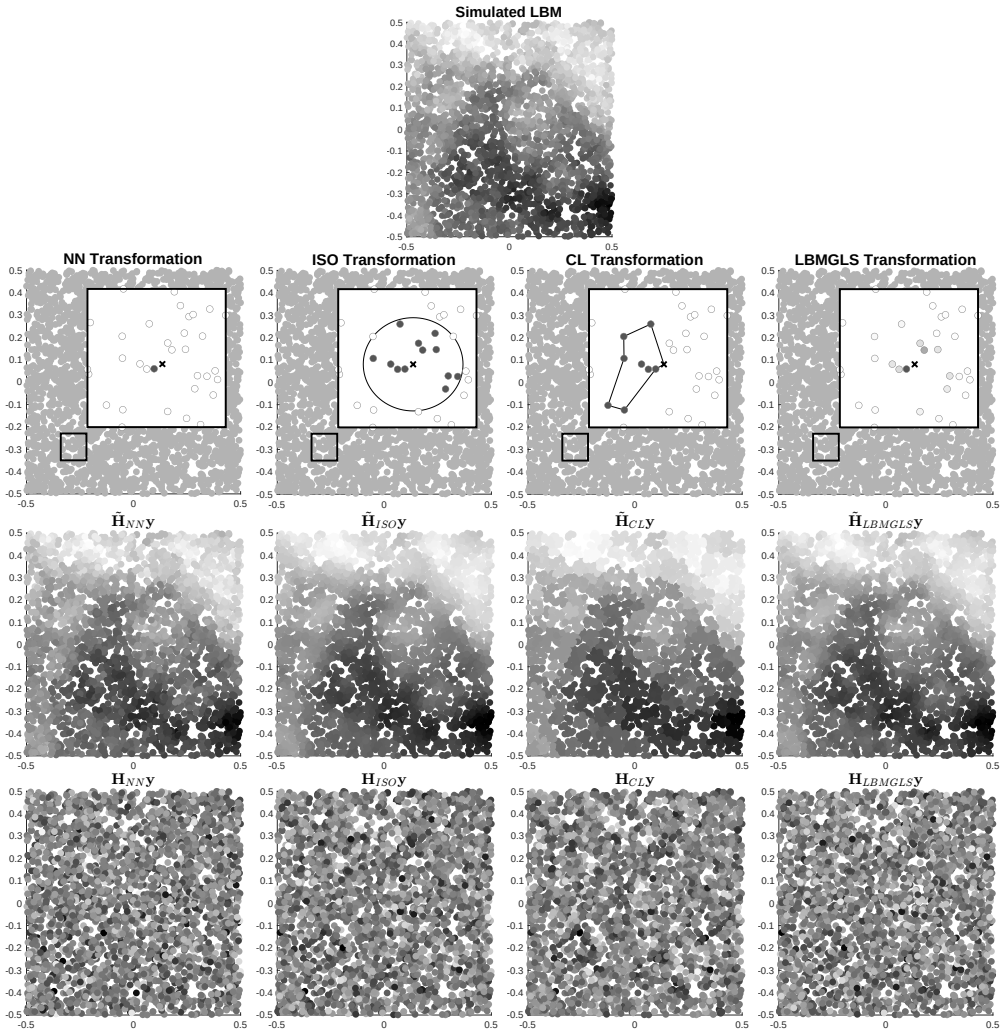

Figure 4: Differencing transformations

notation above). Only allowed with `transformation(iso)`.

`clustvar(`*varname*`)` specifies the variable that is to be used for clustering. Only allowed with `transformation(cluster)`.

`latlong` specifies that the spatial coordinates are given in latitude (stored in `s_1`) and longitude (stored in `s_2`) (see above).

`replace` allows the command to overwrite variables when storing the transformed data.

`separately` executes the transformation separately for all variables in *varlist*. This

leads to different results if there are missing observations in some variables, because the default behavior is to construct the H matrix based only on those observations for which all variables are non-missing.

4 Regression analysis using the Müller-Watson approach

4.1 Proposed procedure

Having outlined the methods presented in Müller and Watson (2024) as well as our implementation thereof, we now turn to their practical application in regression analyses of spatial data. We propose a simple algorithm, summarized in Figure 5:

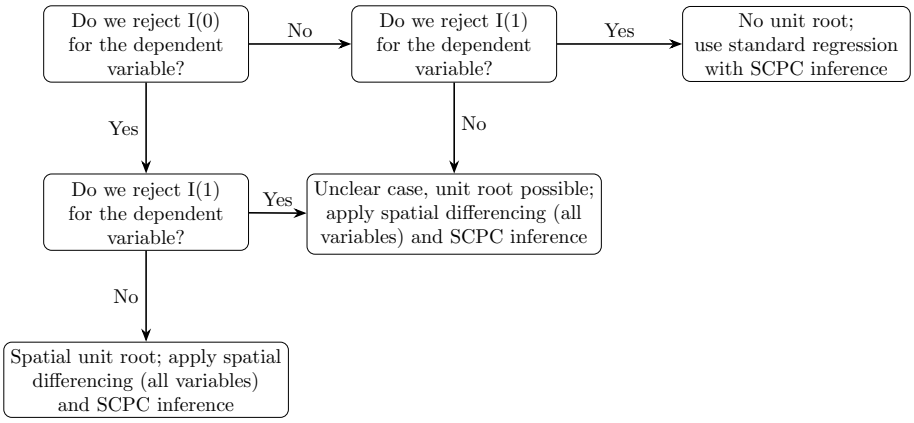

Figure 5: Flow diagram showing how to apply the Müller-Watson approach

We first test whether the dependent variable contains a unit root. To this end, we examine whether we can reject that it is I(0). If so, we test whether we can reject that it is I(1). If we cannot reject, a unit root is most likely present, and we need to apply one of the transformation methods discussed above to remove it. In this case, we propose to difference *both* the dependent *and* the independent variable(s), for ease of interpretation of the regression coefficients. If we reject I(0) but also I(1), or neither, the case is indeterminate; it is arguably wise to difference and report results using transformed variables. If we do not reject the dependent variable being I(0), but we can reject that it is I(1), we can confidently proceed without differencing. In all cases, regression inference still needs to take any remaining (weak) spatial correlation into account; we suggest using the SCPC approach in Müller and Watson (2022, 2023).

Multivariate cases as well as well as instrumental variables can be handled analogously. Since the hypothesized relationship involves x and y, we should proceed with differencing *all* independent variables. Also, because IV estimation represents a rescaling of the relationship between y and z via x, we can proceed analogously in this case.[10]

10. We thank Ulrich Müller and Mark Watson for clarifying this point.

4.2 Illustration using Stata

To illustrate this procedure and the use of our Stata commands, we simulate two *independent* LTU processes x and y with very high persistence ($c = 0.01$), using 722 US commuting zone centroids as locations. We take the location data from the replication package of Müller and Watson (2024), who in turn obtained it from Chetty et al. (2014), and provide it in a supplementary file `example.dta`, along with the simulated data.[11] This data file also includes further variables from Chetty et al. (2014), which we use in Appendix A; however, here we only require the latitude and longitude variables s_1 and s_2 alongside y and x:

```
. use s_1 s_2 y x using "example.dta", clear
```

Figure 6 plots the simulated variables using the `geoplot` command (Jann 2023), the code for which we omit here for brevity, but is included in the `example_reg.do` file in the supplementary materials. The shade of each dot indicates the value of the respective variable at that location. Strong spatial dependence is clearly visible in both variables.

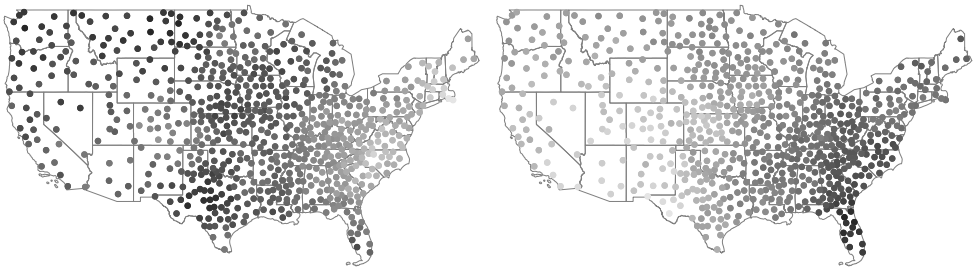

Figure 6: Simulated dependent variable y (left) and independent variable x (right)

Running a simple regression of y on x and applying SCPC inference again illustrates the issue of spurious regression results in the presence of (near) unit roots: In this case, there is a strongly significant negative correlation between y and x, even though they are independent in population:

```
. qui reg y x, r

. scpc, latlong
found 722 observations / clusters and 2-dimensional locations in s_*
Computing distances on surface of sphere treating s_1 as latitude and
s_2 as longitude
SCPC optimal q = 8 for maximal average pairwise correlation = 0.030

SCPC Inference for first 2 coefficients
```

11. The Stata code to generate the simulated data is also available in the supplementary materials as `make_example_data.do`.

```
             |      Coef     Std_Err      t      P>|t|    95% Conf   Interval
-------------+------------------------------------------------------------------
           x | -.4916417    .1246614   -3.94    0.020   -.8813726  -.1019108
       _cons |  1.817547      .08275   21.96    0.000    1.558602   2.076492
```

We now follow the procedure outlined above, by applying the $I(0)$ and $I(1)$ tests to y using the `spurtest` command. We can reject that y is $I(0)$ with very high confidence, and we cannot reject that it is $I(1)$, indicating the presence of a unit root:

```
. spurtest i0 y, latlong
found 722 observations and 2-dimensional locations in s_*
Computing distances on surface of sphere treating s_1 as latitude and
s_2 as longitude
Spatial I(0) Test Results
----------------------------------------
Test Statistic (LFST) :    3.4638
P-value               :    0.0004
----------------------------------------

. spurtest i1 y, latlong
found 722 observations and 2-dimensional locations in s_*
Computing distances on surface of sphere treating s_1 as latitude and
s_2 as longitude
Spatial I(1) Test Results
----------------------------------------
Test Statistic (LFUR) :    4.8015
P-value               :    0.5736
----------------------------------------
```

We therefore use `spurtransform` to difference both y and x using the LBM-GLS transformation, which adds the transformed variables h_y and h_x to the dataset. Figure 7 plots the transformed variables, now showing much less spatial dependence.

```
. qui spurtransform y x, latlong prefix(h_)
```

Finally, we run the regression of h_y on h_x and apply SCPC inference. The coefficient is now close to zero and not statistically significant, showing that our procedure correctly diagnosed and corrected the spurious regression problem:

```
. qui reg h_y h_x, r

. scpc, latlong
found 722 observations / clusters and 2-dimensional locations in s_*
Computing distances on surface of sphere treating s_1 as latitude and
s_2 as longitude
SCPC optimal q = 8 for maximal average pairwise correlation = 0.030

SCPC Inference for first 2 coefficients

             |      Coef     Std_Err      t      P>|t|    95% Conf   Interval
-------------+------------------------------------------------------------------
         h_x |  .0019491    .0369693    0.05    0.968   -.0955051   .0994032
       _cons | -4.09e-11    .0017588   -0.00    0.990   -.0046002   .0046002
```

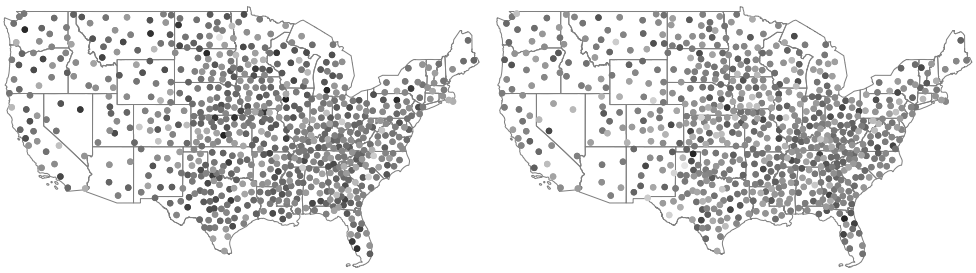

Figure 7: Transformed dependent variable h_y (left) and independent variable h_x (right)

4.3 Monte Carlo simulations

We repeat the above exercise 200 times, each time simulating independent x and y processes as above. We omit the simulation code here for brevity, but provide it in the supplementary materials as `example_montecarlo.do`. For each repetition, we draw the dependence parameters c_x and c_y from a log-normal distribution such that $\log(c_x) \sim N(3, 2)$ and $\log(c_y) \sim N(3, 2)$, which yields a range of realistic persistence levels in this setting. We first estimate the uncorrected regression of y on x with SCPC inference, then apply our procedure to test for unit roots (with a 5% threshold for significance) and, if necessary, difference both variables before re-estimating the regression with SCPC inference. Figure 8 summarizes the results. The left panel shows the estimated coefficients from the uncorrected and corrected regressions, respectively. The right panel shows the share of repetitions in which the null hypothesis of no effect is rejected at the 5% level. This shows that the correction substantially reduces the variance of the estimated coefficients around the true value of zero, and that it reduces the (false) rejection rate from over 10% to under 5%.

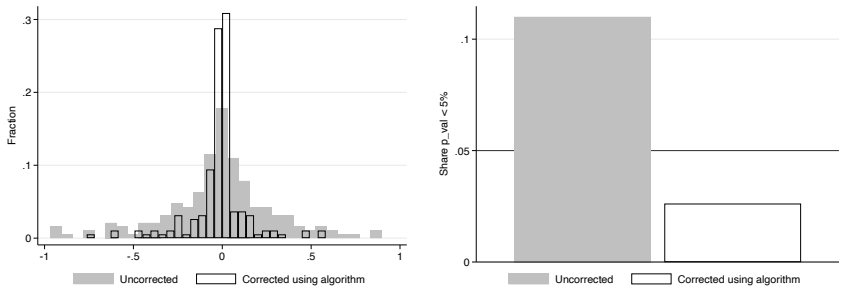

Figure 8: Simulation results: Estimated coefficients (left) and rejection shares at 5% level (right)

5 Conclusions

We present `spur`, a Stata implementation of newly-developed econometric methods that help to diagnose and correct spatial unit roots (Müller and Watson 2024) and discuss their use in regression analysis. How these new methods perform compared to alternative methods to correct for strong spatial dependence is an open question. In follow-up work, we plan to apply this approach as well as several alternatives to both simulated and observational data, examining their power and size properties. This will clarify when each method is best applied, given a particular setting.

6 Acknowledgments

The Stata code is based on the Matlab code provided by Ulrich Müller and Mark Watson https://doi.org/10.5281/zenodo.11199509. Our Stata code replicates the results in Müller and Watson (2024) based on their Matlab code 1:1. Any errors in the Stata code remain our own. We are obliged to Ulrich Müller and Mark Watson for useful conversations and suggestions. We thank Daniel Göttlich and Elie Malhaire for excellent research assistance.

7 Programs and supplemental material

To install the software files as they existed at the time of publication of this article, type (NB: SJ template) . net sj 24-3
. net install st0751
. net get st0751 (to install program files, if available) (to install ancillary files, if available)

The latest version of the programs can be installed using

```
. net install spur, replace from(https://raw.githubusercontent.com/pdavidboll/spur/main/)
```

and (optional) example data and do files can be downloaded using

```
. net get spur, replace from(https://raw.githubusercontent.com/pdavidboll/spur/main/)
```

Revised and improved versions of the programs may become available in the future at https://github.com/pdavidboll/SPUR or on our web pages (https://www.sobecker.de and https://pauldavidboll.com and https://www.jvoth.com).

We provide example do-files and data to replicate the results in Section 4 and Appendix A. Please refer to the provided `readme.txt` file for further information.

8 References

Chetty, R., N. Hendren, P. Kline, and E. Saez. 2014. Where Is the Land of Opportunity? The Geography of Intergenerational Mobility in the United States. *Quarterly Journal of Economics* 129(4): 1553–1623.

Fingleton, B. 1999. Spurious Spatial Regression: Some Monte Carlo Results with a Spatial Unit Root and Spatial Cointegration. *Journal of Regional Science* 39(1): 1–19.

Granger, C., and P. Newbold. 1974. Spurious regressions in econometrics. *Journal of Econometrics* 2(2): 111–120.

Jann, B. 2005. MOREMATA: Stata module (Mata) to provide various functions. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s455001.html.

———. 2023. GEOPLOT: Stata module to draw maps. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s459211.html.

King, M. L. 1987. Towards a Theory of Point Optimal Testing. *Econometric Reviews* 6(2): 169–218.

Müller, U. K., and M. W. Watson. 2008. Testing Models of Low-Frequency Variability. *Econometrica* 76(5): 979–1016.

———. 2019. Low-Frequency Analysis of Economic Time Series. Working paper, in preparation for *Handbook of Econometrics*. https://www.princeton.edu/~umueller/HOE.pdf.

———. 2022. Spatial Correlation Robust Inference. *Econometrica* 90(6): 2901–2935.

———. 2023. Spatial Correlation Robust Inference in Linear Regression and Panel Models. *Journal of Business & Economic Statistics* 41(4): 1050–1064.

———. 2024. Spatial Unit Roots and Spurious Regression. *Econometrica* 92(5): 1661–1695.

Phillips, P. 1986. Understanding spurious regressions in econometrics. *Journal of Econometrics* 33(3): 311–340.

About the authors

Sascha O. Becker is Professor of Economics at the University of Warwick, UK, and Xiaokai Yang Chair of Business and Economics at Monash University, Australia. He is also affiliated with CAGE, CEH@ANU, CEPH, CESifo, CReAM, CEPR, Ifo, IZA, ROA, RF Berlin, and SoDa Labs.

P. David Boll is a PhD candidate at the University of Warwick, UK.

Hans-Joachim Voth is UBS Foundation Professor of Economics, University of Zurich, Switzerland and Scientific Director of the UBS Center for Economics in Society. He is also affiliated with CEPR and CAGE.

A Appendix: Reproducing the Chetty et al. (2014) results in Müller and Watson (2024)

To demonstrate that our Stata code works as expected, we reproduce Table 1 in Müller and Watson (2024) which uses data from Chetty et al. (2014). These data are originally in xlsx format and were obtained from the replication package accompanying Müller and Watson (2024). We read and clean these data in the script `make_example_data.do` and save the resulting dataset as `example.dta`. We keep their variable names 1:1. The key outcome variable is called "am" (absolute mobility) whereas all other variables are predictors of the potential for absolute mobility, such as "tlfpr", the teenage labor force participation rate. "am" and "tlfpr" are the two variables depicted in Figure 1, panel (c). In what follows, we list the sequence of Stata commands that produces our Table A.1.

After this, we call the different commands in the Stata SPUR suite: `spurtest`, `spurhalflife` and `spurtransform`, before finally applying the `scpc` command made available by Müller and Watson (2023) on their website. The latter apply the proper standard errors appropriate in the context of spatial auto-correlation on the (transformed) data.

```
use "example.dta", clear

// make list of covariates
local myvars "fracblack racseg segpov25 fraccom15 hipc gini incsh1 tsr tsperc hsdrop scind
fracrel crimer fracsm fracdiv fracmar loctr colpc coltui colgrad manshare chimp tlfpr
migirate migorate fracfor"

// loop over variables
foreach var of varlist am `myvars´ {
    local label_`var´: variable label `var´

    // i1 test
    spurtest i1 `var´, latlong
        local tab_1_`var´ = `r(p)´

    // i0 test
    spurtest i0 `var´, latlong
        local tab_2_`var´ = `r(p)´

    // half-life
    spurhalflife `var´, latlong normdist nrep(10000)
        local tab_3_`var´ = `r(ci_l)´
        local tab_4_`var´ = `r(ci_u)´

    // note that "am" (=absolute mobility) is the dependent variable
    if "`var´"!="am" {

        preserve
        // Standardize variables
        qui sum am if !missing(am) & !missing(`var´)
        qui replace am = (am - `r(mean)´)/`r(sd)´ if !missing(am) & !missing(`var´)
        qui sum `var´ if !missing(am) & !missing(`var´)
        qui replace `var´ = (`var´ - `r(mean)´)/`r(sd)´ if !missing(am) & !missing(`var´)
```

```
            // Naive OLS
            reg am `var´, noconstant vce(cluster state)
                local tab_5_`var´ = `e(r2)´
                matrix res = r(table)
                local tab_6_`var´ = res[1,1]
                local tab_7_`var´ = res[5,1]
                local tab_8_`var´ = res[6,1]

            // Residual I(1) test
            spurtest i1resid am `var´, latlong
                local tab_9_`var´ = `r(p)´

            // Residual I()) test (not in table)
            spurtest i0resid am `var´, latlong

            // LBMGLS transformation
            qui spurtransform am `var´, prefix("h_") latlong replace

            // OLS on transformed
            qui reg h_am h_`var´, noconstant robust
                local tab_10_`var´ = `e(r2)´
            scpc, latlong
                matrix res = e(scpcstats)
                local tab_11_`var´ = res[1,1]
                local tab_12_`var´ = res[1,5]
                local tab_13_`var´ = res[1,6]

            restore

        } end of "am" if-condition

    } // end loop
```

We follow the exact same ordering of columns as Müller and Watson (2024) to allow for comparison of results of their original Matlab code and our Stata code. Our results are shown in Table A.1. Apart from minor differences in the second decimal place, which are explained by the fact that the methods use simulations based on random numbers, our code reproduces the results in Müller and Watson (2024) exactly.

Note that in the vast majority of cases, applying the LBM-GLS transformation does not turn significant results in levels into insignificant ones. While there are occasional cases like the effect of the manufacturing share or Chinese import growth (significant in levels, but not after the transformation), where the new 95% confidence interval includes zero, these are rare. This is true despite the fact that the overwhelming majority of dependent variables appear to be I(1), exhibiting a strong form of spatial dependence.

	Spatial Persistence Statistics			Regression of AMI onto Variable				
	p-Value of Test		Half-life	Levels			LBM-GLS	
Variable	$I(1)$	$I(0)$	95% CI	R^2	$\hat{\beta}$[95% CI] Cluster	p-Value Resid. $I(1)$	R^2	$\hat{\beta}$[95% CI] C-SCPC
Absolute Mobility Index	0.38	0.00	[0.09, ∞]	NA	NA	NA	NA	NA
Frac. Black Residents	0.11	0.01	[0.04, ∞]	0.36	−0.60[−0.74, −0.47]	0.21	0.10	−0.42[−0.70, −0.15]
Racial Segregation	0.01	0.13	[0.00, 0.28]	0.14	−0.38[−0.47, −0.29]	0.29	0.18	−0.24[−0.35, −0.12]
Segregation of Poverty	0.28	0.03	[0.05, ∞]	0.18	−0.43[−0.56, −0.29]	0.27	0.15	−0.21[−0.36, −0.05]
Frac. ¡ 15 Mins to Work	0.57	0.00	[0.14, ∞]	0.48	0.69[0.54, 0.85]	0.14	0.15	0.37[0.08, 0.65]
Mean Household Income	0.13	0.14	[0.02, ∞]	0.00	0.05[−0.10, 0.20]	0.38	0.00	−0.01[−0.26, 0.24]
Gini	0.78	0.00	[0.26, ∞]	0.37	−0.60[−0.79, −0.42]	0.24	0.10	−0.22[−0.38, −0.05]
Top 1 Perc. Inc. Share	0.31	0.02	[0.07, ∞]	0.04	−0.21[−0.36, −0.06]	0.36	0.02	−0.07[−0.13, −0.00]
Student-Teacher Ratio	0.23	0.13	[0.05, ∞]	0.12	−0.35[−0.55, −0.14]	0.45	0.03	−0.17[−0.44, 0.11]
Test Scores (Inc. adjusted)	0.30	0.06	[0.07, ∞]	0.34	0.58[0.39, 0.76]	0.41	0.30	0.42[0.15, 0.69]
High School Dropout	0.09	0.02	[0.03, ∞]	0.34	−0.58[−0.75, −0.41]	0.49	0.21	−0.29[−0.56, −0.02]
Social Capital Index	0.72	0.00	[0.22, ∞]	0.41	0.64[0.46, 0.82]	0.29	0.08	0.28[−0.02, 0.59]
Frac. Religious	0.27	0.04	[0.07, ∞]	0.28	0.53[0.35, 0.70]	0.26	0.14	0.32[0.14, 0.50]
Violent Crime Rate	0.54	0.02	[0.14, ∞]	0.21	−0.45[−0.68, −0.23]	0.34	0.04	−0.14[−0.26, −0.03]
Frac. Single Mothers	0.18	0.00	[0.05, ∞]	0.59	−0.77[−0.92, −0.62]	0.12	0.52	−0.60[−0.94, −0.26]
Divorce Rate	0.05	0.17	[0.02, 3.00]	0.27	−0.52[−0.71, −0.33]	0.50	0.26	−0.37[−0.63, −0.11]
Frac. Married	0.05	0.08	[0.01, ∞]	0.31	0.56[0.43, 0.68]	0.22	0.31	0.35[0.11, 0.59]
Local Tax Rate	0.02	0.24	[0.01, 0.42]	0.12	0.35[0.21, 0.48]	0.39	0.01	0.07[−0.10, 0.23]
Colleges per Capita	0.23	0.07	[0.06, ∞]	0.06	0.24[−0.02, 0.49]	0.27	0.00	0.01[−0.24, 0.26]
College Tuition	0.38	0.00	[0.09, ∞]	0.00	−0.02[−0.16, 0.12]	0.28	0.00	0.01[−0.05, 0.08]
Coll. Grad. Rate (Inc. Adjusted)	0.04	0.03	[0.00, 3.00]	0.02	0.15[0.03, 0.28]	0.35	0.03	0.08[0.01, 0.15]
Manufacturing Share	0.20	0.00	[0.06, ∞]	0.09	−0.30[−0.47, −0.12]	0.37	0.01	0.07[−0.09, 0.23]
Chinese Import Growth	0.02	0.07	[0.02, 0.46]	0.03	−0.17[−0.33, −0.02]	0.38	0.00	0.03[−0.01, 0.06]
Teenage LFP Rate	0.51	0.00	[0.12, ∞]	0.44	0.66[0.49, 0.83]	0.28	0.04	0.26[−0.06, 0.58]
Migration Inflow	0.29	0.08	[0.00, ∞]	0.07	−0.27[−0.42, −0.12]	0.33	0.02	−0.12[−0.27, 0.04]
Migration Outlflow	0.34	0.01	[0.07, ∞]	0.03	−0.16[−0.31, −0.02]	0.37	0.01	−0.08[−0.16, 0.01]
Frac. Foreign Born	0.56	0.04	[0.17, ∞]	0.00	−0.03[−0.16, 0.10]	0.39	0.02	−0.12[−0.29, 0.06]

Table A.1: Reproducing the Chetty et al. (2014) results in Müller and Watson (2024) using our Stata commands.