The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–20

Spatial Unit Roots in Regressions: A Practitioner's Guide and a Stata Package

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

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

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

Abstract. Spatial unit roots can lead to spurious regression results. We present a brief overview of the methods developed in Müller and Watson (2024) to test for and correct for spatial unit roots. We also 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. We present a brief practitioner's guide for applied researchers.

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

1 Introduction: Spatial unit roots

Spatial data present challenges for statistical analysis because observations that are close to each other geographically tend to be correlated - violating the assumption of independent and identically distributed (i.i.d.) errors. Valid inference in such settings requires the use of heteroskedasticity and autocorrelation consistent (HAC) corrections or cluster standard errors at broader geographic levels (like states).

However, even these correction methods fail when spatial dependence is too strong ("spatial unit roots"): In such cases, these methods can produce spuriously significant regression coefficients even for completely independent variables. Müller and Watson (2024) develop new statistical tests to detect such strong dependence and 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.

It is well-known in the time series context that when the serial correlation in the regressors and regression errors is weak (i.e. I(0)), it is sufficient to apply HAC corrections to account for serial correlation. However, when the serial correlation is strong (i.e. I(1)), 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 finite-sample settings and corrects for size distortions when the regressor of interest (x) is nonstationary.[1] However, the methods presented in Müller and Watson (2022) and Müller and Watson (2023) are not dealing with the case of strong spatial auto-correlation in the outcome of interest (y). Müller and Watson (2024) introduce diagnostic tests for spatial unit roots and show how transformations of the dependent and independent variables eliminate spurious 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. We also present a practitioner's guide for applied researchers trying to detect potential spatial unit roots in their variables of interest: 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. We present the different spatial differencing methods proposed by Müller and Watson (2024) to correct spatial unit roots, and show that our routines replicate the results in Müller and Watson (2024) using data from Chetty et al. (2014).

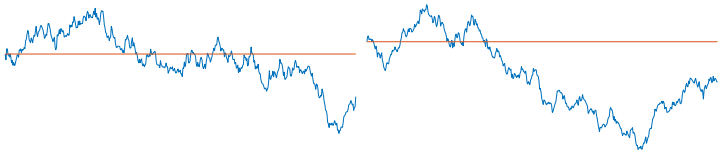
The rest of the article proceeds as follows: Section 2 summarizes and illustrates the tests developed by Müller and Watson (2024) to diagnose spatial unit roots, as well as their Stata implementation in the commands `spurtest` and `spurhalflife`. Section 3 explains the spatial differencing techniques they propose to eliminate unit roots, and presents how they can be applied using the command `spurtransform`. Section 4 presents a brief guide to applying these methods to common settings in applied research, and Section 5 demonstrates the functionality of our implementation by replicating results from Müller and Watson (2024). Section 6 concludes.

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 con-

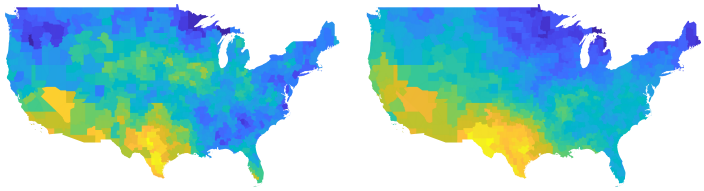
1. The methods developed in these two papers have been implemented in Stata by the authors in their -scpc- package.

(a): Independent time series unit root processes



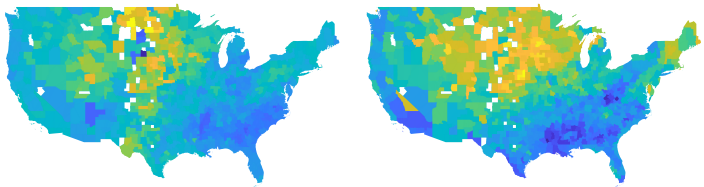
$R^2 = 0.33$
$t = 3.81$

(b): Independent spatial unit root processes



$R^2 = 0.33$
$t = 4.05$

(c): Data from Chetty et al. (2014)



$R^2 = 0.44$
$t = 7.89$

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.

text: 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) which show some visual resemblance with the unit-root processes in panel (b), thereby highlighting the potential relevance of strong spatial auto-correlation that needs to be detected and addressed in empirical work.

Specifically, Müller and Watson (2024) develop four diagnostic tests, examing 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. LTU models are characterized by weak mean reversion as measured by a parameter $c > 0$. For small values of c, these processes behave nearly like $I(1)$ processes, and for large c, they share many characteristics of weakly dependent $I(0)$ series. The LTU model thus traces a continuous spectrum of dependence between the dichotomous $I(0)$ and $I(1)$ cases.

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

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 (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 basic idea underlying all of the tests is to compare the performance of these two models in rationalizing the data. Instead of performing tests on the raw data, Müller and Watson (2024) build 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 $L(\cdot)$). Further define $\mathbf{R}$ as the $n \times q$ matrix whose columns are the eigenvectors of $\mathbf{M\Sigma_L M}$ corresponding to the q largest eigenvalues, where $\mathbf{M} = \mathbf{I}_n - \mathbf{1}(\mathbf{1'1})^{-1}\mathbf{1'}$ is the demeaning matrix, and scaled such that $n^{-1}\mathbf{R'R} = \mathbf{I}_q$. Then, the weighted averages are computed as

$$\mathbf{Z} = \mathbf{R'MY} = \mathbf{R'Y}$$

The j-th ($j = 1, \ldots, q$) weighted average is the linear combination of the data with the j-th largest variance under the canonical $I(1)$ model. As 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:[2] 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

$$\mathbf{\Omega} = \mathbf{R'\Sigma R}$$

2. See Müller and Watson (2019) for a more extensive discussion.

where $\mathbf{\Sigma}$ is the covariance matrix induced by the canonical model that asymptotically approximates the data generating process (see again Theorem 2 in Müller and Watson 2024). 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.

Choice of q. An obvious practical question is how to choose the number of weighted averages q. The trade-off involved follows from the previous discussion: 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) suggest to keep q between 10 and 20, and use $q = 15$ in their applications. In our Stata package, all test commands include the option `, q()`. Following the previous discussion, we set `q(15)` as the default.

Illustration of weighted averages. To aid intuition, 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 $\frac{1}{2}\left(|s_l| + |s_\ell| - |s_l - s_\ell|\right)$. 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

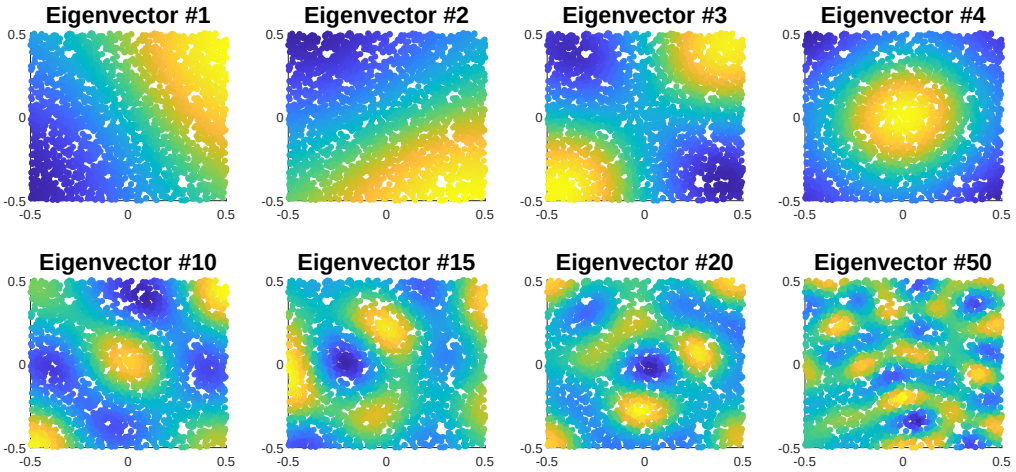


Figure 2: Illustration of the weights

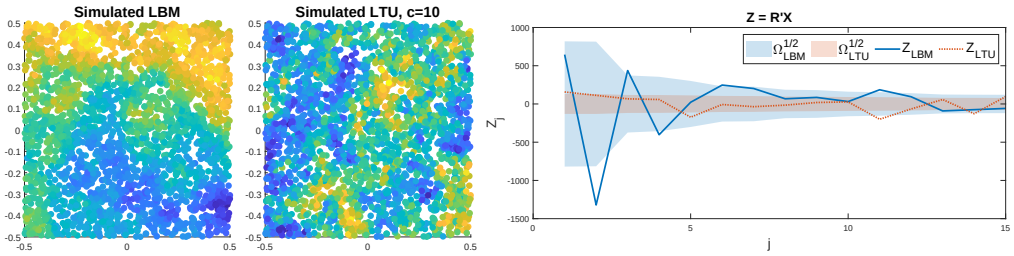


Figure 3: Simulated data and weighted averages

increases with the order of the eigenvectors, and thus projecting[3] the data $\mathbf{Y}$ on the first q eigenvectors extracts the low-frequency variation. This is further illustrated by the subplots of Figure 3: The first two subplots show simulated data for an LBM (unit root) and an LTU process with very low persistence ($c = 10$), respectively. The difference in low-frequency variation is clearly visible. The third subplot shows the weighted averages $\mathbf{Z}_{LBM}, \mathbf{Z}_{LTU}$ resulting from pre-multiplying the two data vectors with the eigenvectors of $\mathbf{M}\mathbf{\Sigma}_\mathbf{L}\mathbf{M}$ corresponding to the $q = 15$ largest eigenvalues. The difference in behavior is very 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. The two shaded areas show the range $[-\sqrt{\mathbf{\Omega}_{j,j}}, \sqrt{\mathbf{\Omega}_{j,j}}]$ of the covariance matrices $\mathbf{\Omega}_{LBM}, \mathbf{\Omega}_{LTU}$ implied by the two processes: by construction, $\mathbf{\Omega}_{LBM}$ describes the behavior of $\mathbf{Z}_{LBM}$ much better than that of $\mathbf{Z}_{LTU}$, and vice versa. The next sections formalize such comparisons to discriminate 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

3. Notice that $\mathbf{R}'\mathbf{Y} = n(\mathbf{R}'\mathbf{R})^{-1}\mathbf{R}'\mathbf{Y}$ by construction. The j-th element of $\mathbf{Z}$ is thus the (scaled) coefficient of a regression of $\mathbf{Y}$ on the j-th column of $\mathbf{R}$.

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 tests for 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). The test statistic,[5] 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 quite as straightforward: the LTU model, as discussed in Section 1, is very similar to an $I(1)$ process for small c, and very 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.[6] They then propose the hypothesis

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

4. P-values are computed as $\sum_i^{N_{rep}} \mathbf{1}[\hat{\Lambda}_i > \Lambda]/N_{rep}$
5. Müller and Watson (2024) label the statistic LFUR in reference to the Low Frequency Unit Root statistic in Müller and Watson (2008).
6. 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. To construct a test statistic in the form of Section 2.2, we require simple hypotheses. Müller and Watson (2024) suggest that setting $c = c_{0.001}$ under both H_0 and H_a and thus computing the test statistic

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

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 $y_l = x_l'\beta + u_l$. 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 $\mathbf{M}_X\boldsymbol{\Sigma}_L\mathbf{M}_X$ 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).

For this command (and all other commands in this package) to work, the spatial coordinates must be stored in the variables `s_*`, where `*` is a positive integer. 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.

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 $\ln 2/c$. Therefore, confidence intervals can be constructed as the sets of values of h for which the null hypothesis $H_0 : h_0 = h$ cannot be rejected. 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.

For this command to work, the spatial coordinates must be stored in the variables `s_*`, where `*` is a positive integer. (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 transformations, the last of which 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.

Nearest Neighbor (NN) Differences

Perhaps the most 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

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

where $\hat{\mathbf{H}}_{\text{NN},lj} = 1$ 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

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

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

Clustered demeaning

A third option is to partition the data into K clusters and subtracting from each observation the mean within its cluster (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

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

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

LBM-GLS transformation

The previous three transformations are somewhat 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

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

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 ($\mathbf{H}\mathbf{Y}$).[7]

Syntax

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

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

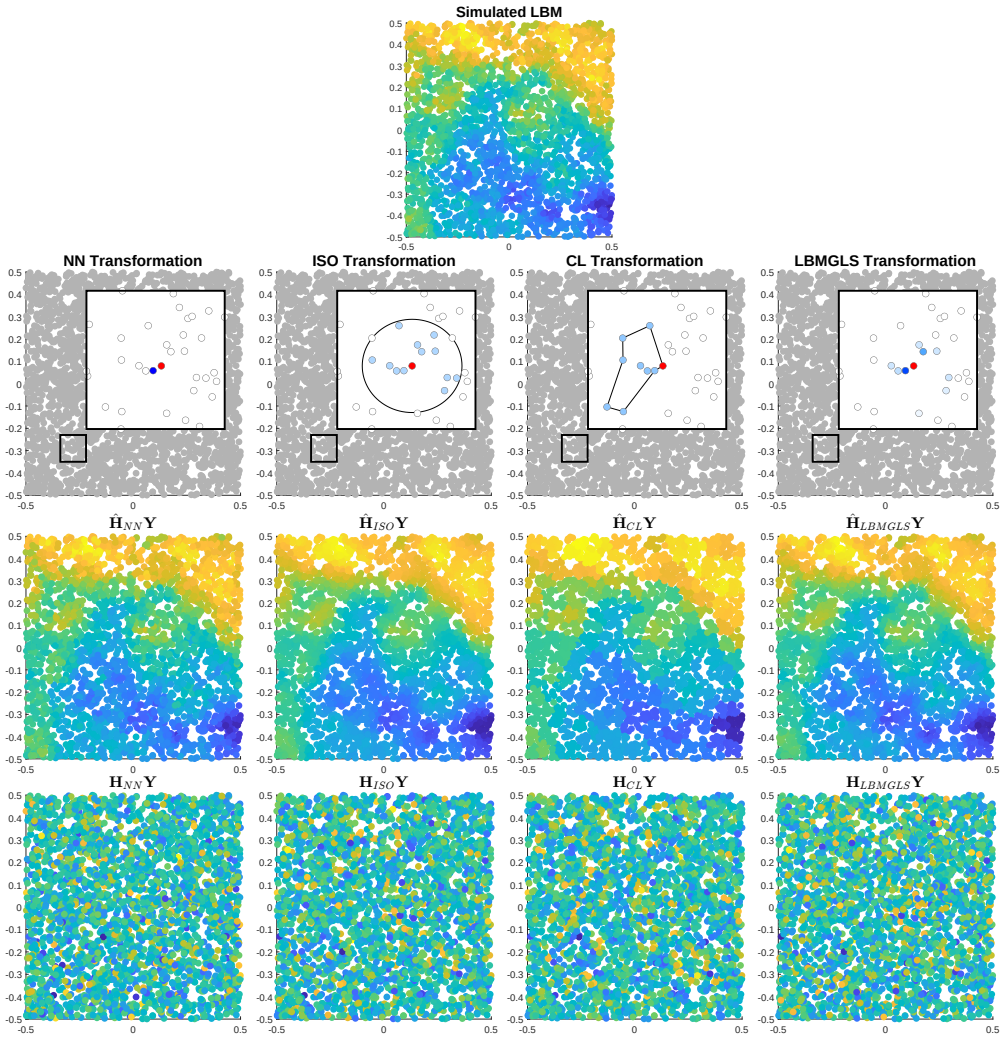


Figure 4: Differencing transformations

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`, or, equivalently, execute the command separately for all variables.

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 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 Practitioner's guide

How should the Müller-Watson approach be used in practice? Figure 5 summarizes the key steps in applying the spatial unit root 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 mostly likely present; we need to use the methods in Müller and Watson (2024), combined with the SCPC approach in Müller and Watson (2022, 2023). In this case, variables on both sides of the equation need to be differenced, independent of whether x is I(0) or I(1). If we rejected I(0) but also I(1), the case is indeterminate; it is arguably wise to difference and report results from using transformed variables. If we cannot reject the dependent variable being I(0), but we can reject that it is I(1), we can confidently use standard methods.

Multivariate cases as well as well as instrumental variables (IV) 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.[8]

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

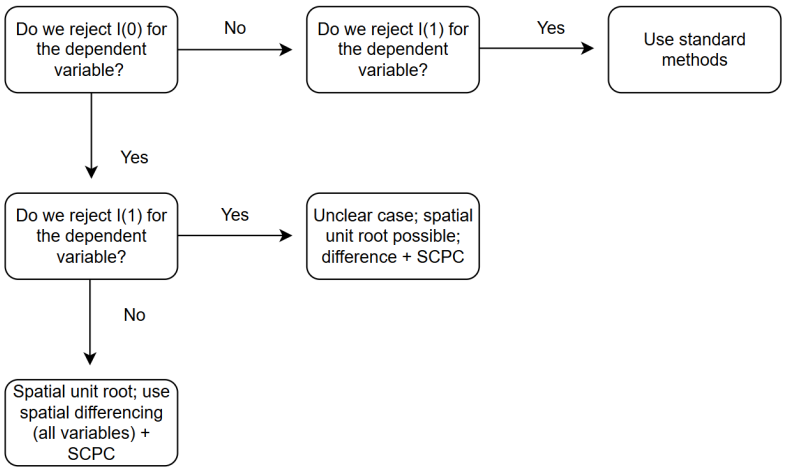


Figure 5: Flow diagram: Steps to apply the spatial unit root approach

5 Application: 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). The respective data comes in xlsx format and was obtained from the replication package accompanying Müller and Watson (2024). 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 1.

Our code starts with some preparatory work: reading in the Chetty et al. (2014) data; defining labels; defining lists of variables:

```
clear all
mata mata clear

// import variable labels
import excel "../example_data/Chetty_Data_Labels.xlsx", sheet("Sheet1") firstrow case(lower) clear
local i 0
foreach v of varlist * {
    local lab`++i´ = `v´[1]
}

// import data
import excel "../example_data/Chetty_Data_1.xlsx", sheet("Sheet1") firstrow case(lower) clear

// assign variable labels
```

```
local i 0
foreach v of varlist * {
    label variable `v´ `"`lab`++i´´"´
}

// drop non-contiguous states
drop if state == "HI"
drop if state == "AK"

// rename lat and lon
rename lat s_1
rename lon s_2

// 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"
```

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.

```
// 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 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.

6 Conclusion

The need to adjust for spatial dependence in regression inference using spatial data is well-known. It is routinely addressed through well-developed HAC methods such as Conley (1999) or regional clustering. However, recent econometric advances by Müller and Watson (2024) demonstrate that this may be insufficient when spatial dependence is strong: analogously to well-established results in time series econometrics, strong spatial autocorrelation can lead to spuriously significant coefficients in regressions of independent processes. Diagnosing and correcting for this is therefore important to applied research that exploits on spatial variation.

Here, we present a new Stata package `spur` that diagnoses the presence of spatial unit

	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	0.79	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]	0.42	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]	0.30	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	[0.15, ∞]	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]	0.11	0.52	−0.60[−0.94, −0.26]
Divorce Rate	0.05	0.17	[0.02, ∞]	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, 0.49]	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	[0.02, ∞]	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]	0.37	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]	0.29	0.04	0.26[−0.06, 0.58]
Migration Inflow	0.30	0.08	[0.00, ∞]	0.07	−0.27[−0.42, −0.12]	0.32	0.02	−0.12[−0.27, 0.04]
Migration Outlflow	0.34	0.01	[0.08, ∞]	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	[0.16, ∞]	0.00	−0.03[−0.16, 0.10]	0.39	0.02	−0.12[−0.29, 0.06]

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

roots and creates transformed variables that are cleansed of strong spatial dependence, using the methods developed in Müller and Watson (2024). We demonstrate that our package can exactly replicate the empirical results in Müller and Watson (2024), and we provide a guide to applying this new package in applied settings. In follow-up work, we plan to apply these methods to several influential studies using spatial data to gauge the magnitude of these issues in practice.

7 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 for excellent research assistance.

8 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)

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

9 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.

Conley, T. G. 1999. GMM Estimation with Cross Sectional Dependence. *Journal of Econometrics* 92(1): 1–45.

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.

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*.

———. 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, CESifo, CEH@ANU, CReAM, CEPR, Ifo, IZA, ROA, RF Berlin, and SoDa Labs.

P. David Boll is a Ph.D. 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.

Please address any correspondence to Sascha O. Becker (s.o.becker@warwick.ac.uk), David Boll (David.Boll@warwick.ac.uk) or Hans-Joachim Voth (voth@econ.uzh.ch).