TEST (2018) 27:716–748
https://doi.org/10.1007/s11749-018-0599-x

ORIGINAL PAPER

Comparing implementations of global and local indicators of spatial association

Roger S. Bivand[1] · David W. S. Wong[2]

Received: 4 April 2018 / Accepted: 14 July 2018 / Published online: 27 July 2018

Abstract
Functions to calculate measures of spatial association, especially measures of spatial autocorrelation, have been made available in many software applications. Measures may be global, applying to the whole data set under consideration, or local, applying to each observation in the data set. Methods of statistical inference may also be provided, but these will, like the measures themselves, depend on the support of the observations, chosen assumptions, and the way in which spatial association is represented; spatial weights are often used as a representational technique. In addition, assumptions may be made about the underlying mean model, and about error distributions. Different software implementations may choose to expose these choices to the analyst, but the sets of choices available may vary between these implementations, as may default settings. This comparison will consider the implementations of global Moran's I, Getis–Ord G and Geary's C, local I_i and G_i, available in a range of software including Crimestat, GeoDa, ArcGIS, PySAL and R contributed packages.

Keywords Software implementations · Global spatial autocorrelation · Local spatial autocorrelation · Lattice data

Mathematics Subject Classification 62P12 · 62P20 · 62P25

✉ Roger S. Bivand
roger.bivand@nhh.no

David W. S. Wong
dwong2@gmu.edu

[1] Department of Economics, Norwegian School of Economics, Helleveien 30, 5045 Bergen, Norway

[2] Department of Geography and GeoInformation Science, George Mason University, Exploratory Hall, 4400 University Dr., Fairfax, VA 22030, USA

1 Introduction

In application domains, problems involved in analyzing areal data have attracted attention for almost 60 years (Duncan et al. 1961). One set of problems has been associated with spatial heterogeneity, another with spatial autocorrelation. It has very often been the case that the polygonal areas available to analysts have not matched the footprint of spatial processes. This leads inevitably to problems, with relative spatial heterogeneity then used to attempt to regionalize the data, aggregating to more adequate, homogeneous, policy zones. Regionalization has developed further as a separate field with clear links to the study of spatial sorting and segregation. Spatial autocorrelation should arguably have stayed closer to spatial heterogeneity, and more recent work is moving in this direction (Ord and Getis 2012; Xu et al. 2014), to which we return in conclusion.

There have been implementations of global measures of spatial autocorrelation in open and closed[1] source software since the 1990's. These include the survey and Systat case in Bivand (1992)[2] and the then widely used SpaceStat implementation described in Anselin (1992). Provisions were also made within the ArcView and ArcInfo proprietary GIS (geographical information systems) through contributions written in Avenue and AML (advanced markup language), respectively. Following the introduction of ArcGIS superceding ArcView and ArcInfo, first Visual Basic then Python were used to provide implementations. This progression is described in detail by Wong and Lee (2005, first edition 2001) and is presented by Scott and Janikas (2010), also covering local measures of spatial autocorrelation introduced from the mid 1990s.

Table 1 shows one of the typical issues that differences in numerical results occur between implementations. The first four lines of the Table are copied from Table 6 in Bivand (1992, p. 957) and differ from those re-created using the current implementation in `spdep::moran.test()`, shown in the next four lines. The four lines differ among themselves in using binary or row-standardized spatial contiguity weights and using the normality or randomisation assumption for calculating the variance of Moran's I.

```
> eire <- rgdal::readOGR(system.file("shapes/eire.shp", package = "spData")[1])

> library(spdep)
> eire.nb <- poly2nb(eire)
> moran.test(eire$OWNCONS, nb2listw(eire.nb, style = "B"), randomisation = FALSE)

    Moran I test under normality

data:  eire$OWNCONS
weights: nb2listw(eire.nb, style = "B")

Moran I statistic standard deviate = 5.7608, p-value = 4.187e-09
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
       0.63262789       -0.04000000        0.01363295
```

[1] Some free software, like CrimeStat, is closed source.

[2] Source code now available from https://github.com/rsbivand/legacy_systat.

Table 1 Reproducing results for Moran's I under randomisation and normality assumptions for binary (B) and row-standardized (W) contiguity weights from Table 6 in Bivand (1992, p. 957) for 26 Irish counties and consumption of own produce as a percentage of gross agricultural output; `spdep::moran.test` are results without a neighbour link for a ferry between non-contiguous counties in the original data, and `spdep::moran.test (*)` following the insertion of the link

	Randomisation	Weights	I	$Z(I)$
Bivand (1992)	FALSE	B	0.626	5.748
Bivand (1992)	TRUE	B	0.626	5.706
Bivand (1992)	FALSE	W	0.722	6.073
Bivand (1992)	TRUE	W	0.722	6.027
spdep::moran.test	FALSE	B	0.633	5.761
spdep::moran.test	TRUE	B	0.633	5.719
spdep::moran.test	FALSE	W	0.713	5.909
spdep::moran.test	TRUE	W	0.713	5.864
spdep::moran.test (*)	FALSE	B	0.626	5.748
spdep::moran.test (*)	TRUE	B	0.626	5.706
spdep::moran.test (*)	FALSE	W	0.722	6.073
spdep::moran.test (*)	TRUE	W	0.722	6.027

The reason for the difference is that the contiguities used in Bivand (1992) follow Cliff and Ord (1969) and include a ferry link between the counties of Clare and Kerry (Bivand 2009, p. 377), a link that is not found when generating county contiguities based only on map boundaries. If we add in the symmetric ferry link, we see that the final four lines of Table 1 now agree with those from the original article.

```
> eire.nb[[3]] <- sort(c(eire.nb[[3]], 8L))
> eire.nb[[8]] <- sort(c(eire.nb[[8]], 3L))
> moran.test(eire$OWNCONS, nb2listw(eire.nb, style = "B"), randomisation = FALSE)

    Moran I test under normality

data:  eire$OWNCONS
weights: nb2listw(eire.nb, style = "B")

Moran I statistic standard deviate = 5.7483, p-value = 4.508e-09
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
       0.62601807       -0.04000000        0.01342444
```

Authors of implementations of global and local measures of spatial autocorrelation are often asked by users of the software why conducting the same calculation in different implementations appears to give different numerical results. While it is seldom the case that the inference would have differed, users express concern about the causes of the differences.[3] In this trivial case, the cause was a missing link in the graph of neighbours. A frequent cause of divergence in numerical results is that it may not be easy to exchange weights objects between implementations, so the difference

[3] For an example, see https://community.esri.com/thread/60740.

between weights is the cause of the difference in results. Another common cause of divergence is that the spatial weights and the variable of interest are not sorted in the same order or differ in some other way. Once we have established that the input data and the spatial weights being used are identical, we would expect all implementations to yield identical numerical output.

The purpose of this article is then to compare implementations of chosen global and local measures of spatial autocorrelation, and to establish reasons for any differences that are found, so that users can be surer that their choice of software is not prejudicing their work. In this comparison, we will not be considering spatial autocorrelation in categorical variables, and hope to return to join-count (Cliff and Ord 1981, pp. 18–20) and similar measures in the near future. The authors share an interest in benchmarking implementations of measures of spatial association, and Bivand (2009) and Bivand and Piras (2015) are similar in comparative approach.

2 Global and local indicators

Global measures express the strength of spatial autocorrelation present in the quantitative variable of interest across a whole areal data set, possibly after considering the influence of other variables. The underlying spatial process is expressed as a fixed spatial weights matrix chosen by the analyst, and the strength of spatial autocorrelation may vary if the spatial weights matrix is defined in a different way. For example, a chessboard might seem to display strong negative autocorrelation, but this only holds if the weights express contiguity between squares sharing edges, not edges and corners.

Local measures decompose the spatial autocorrelation present in the quantitative variable of interest across an areal data set to each of the component areas, also using a fixed spatial weights matrix chosen by the analyst. They will be affected by missing consideration of other variables, and/or of a global spatial process.

Both global and local measures may detect other forms of mis-specification, for example, a missing variable showing spatial pattern (see McMillen 2003; Schabenberger and Gotway 2005), or spatial heterogeneity. The use of local measures to detect hotspots is crucially impacted by their ability to pick up other forms of mis-specification. Further, because they may constitute multiple tests on the same data, inference needs to be able to handle multiple comparisons.

For convenience, we list standard representations of the measures as given in the now rather disperse literature. The development of the measures is covered in detail in the references given, together with further alternatives for join-count measures and ranked observations not covered here. We do not give the definitions of more specialized measures, such as those taking the incidence count and population at risk into account.

2.1 Global indicators

2.1.1 Moran's *I*

Moran's I, originally defined by Moran (1950), is without doubt the measure of choice for applied scientists, with over 2000 citations in *Web of Science*, concentrated in the environmental sciences, ecology and public health. Other authors have built on this work, notably Cliff and Ord (1969, 1973, 1981), and it is this development of a more general test statistic that is covered by Ripley (1981), Goodchild (1986) and Cressie (1993). The standard representation of the measure (Cliff and Ord 1981, p. 17, equation 1.15) is as follows:

$$I = \frac{n \sum_{(2)} w_{ij} z_i z_j}{S_0 \sum_{i=1}^{n} z_i^2}, \tag{1}$$

where x_i, $i = 1, \ldots, n$ are n observations on the numeric variable of interest, $z_i = x_i - \bar{x}$, $\bar{x} = \sum_{i=1}^{n} x_i / n$, $\sum_{(2)} = \underset{i \neq j}{\sum_{i=1}^{n} \sum_{j=1}^{n}}$, w_{ij} are the spatial weights, and $S_0 = \sum_{(2)} w_{ij}$. Note that by definition the principal diagonal of the weights matrix $w_{ii} = 0, i \in 1, \ldots, n$, so that in practice the condition $i \neq j$ on $\sum_{(2)}$ has no effect. Since many other weights are typically also 0, summation of products is often implemented over the nonzero values of w_{ij}. In early treatments, contiguity weights were by definition symmetric, $w_{ij} = w_{ji}$, as were weights based on a distance threshold, and weights could be seen as an undirected graph.

The expectation of Moran's I (Cliff and Ord 1981, p. 21, equation 1.37) for both the normality and randomisation assumptions used in the development may be taken as:

$$E(I) = -\frac{1}{(n-1)}, \tag{2}$$

if we do not question the size of n. Bivand and Portnov (2004) suggest that there are issues raised when x_i is observed for all $i = 1, \ldots, n$, but that there are no-neighbour observations, $\sum_{j=1}^{n} w_{ij} = 0$. Because neighbours are recorded as graph edges or as a sparse matrix, not as a dense matrix with many zero values, it is quite easy to generate no-neighbour observations. As Bivand and Portnov (2004, pp. 125–129) note, it is not obvious whether Cliff and Ord (1969, and their subsequent work) intended n to be the number of observations in total, or the number of observations with neighbours in the development of the inferential basis for Moran's I. In the **spdep** functions implementing global measures by default adjust n to the number of observations with neighbours once the user has also chosen to permit observations with no neighbours (leading to the curious lagged value of $\sum_{j=1}^{n} w_{ij} x_j = 0$). This path yields n' for use in the expectation and variance calculations:

$$n' = \sum_{i=1}^{n} \left[\left(\sum_{j=1}^{n} w_{ij} \right) > 0 \right], \tag{3}$$

where the logical variable $(\sum_{j=1}^{n} w_{ij}) > 0$ takes the value 1 and $(\sum_{j=1}^{n} w_{ij}) = 0$ the value 0 for summation.

The analytical variance can be calculated under normality (N) or randomisation (R) assumptions. Under the normality assumption (Cliff and Ord 1981, p. 21, equation 1.38), it takes this form:

$$E_{\mathrm{N}}(I^2) = \frac{n^2 S_1 - nS_2 + 3S_0^2}{S_0^2(n^2-1)}, \tag{4}$$

where $S_1 = \frac{1}{2}\sum_{(2)}(w_{ij} + w_{ji})^2$ and $S_2 = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} w_{ij} + \sum_{j=1}^{n} w_{ji}\right)^2$. Under the randomisation assumption, which also accommodates divergences of the variable from normality by including a kurtosis term (Cliff and Ord 1981, p. 21, equation 1.39), it is:

$$E_{\mathrm{R}}(I^2) = \frac{n\left[(n^2-3n+3)S_1 - nS_2 + 3S_0^2\right] - b_2\left[(n^2-n)S_1 - 2nS_2 + 6S_0^2\right]}{(n-1)(n-2)(n-3)S_0^2}, \tag{5}$$

where $b_2 = \frac{m_4}{m_2^2}$, $m_4 = \sum_{i=1}^{n} z_i^4$ and $m_2 = \sum_{i=1}^{n} z_i^2$ (Cliff and Ord 1981, pp. 45–46). The variance is then calculated by subtracting the square of the expectation from the $E(I^2)$ term from the $E_*(I^2)$ term calculated under either the normality or the randomisation assumption (Cliff and Ord 1969, p. 28, equation 8):

$$\mathrm{Var}_*(I) = E_*(I^2) - [E(I)]^2. \tag{6}$$

Finally, we reach the standard normal deviate under one of the assumptions for evaluation (Cliff and Ord 1969, p. 28, equation 9):

$$Z_*(I) = \frac{I - E(I)}{\sqrt{\mathrm{Var}_*(I)}}. \tag{7}$$

Moran's I has also been developed for regression residuals, but for comparison is only available here for the **spdep** implementation, as neither GeoDa nor PySAL admit an intercept-only regression. In the intercept-only case, $Z(I)$ should agree exactly with the use of $\bar{x}$ as the mean model in standard Moran's I under the normality assumption.

None of the implementations considered here use the adjustment for small n considered in Cliff and Ord (1971) and discussed by Sokal and Oden (1978). There is as yet no implementation of the exact testing approach for regression residuals presented by Hepple (1998). Implementations of the Saddlepoint approximation for regression residuals proposed by Tiefelsdorf (2002) and the exact testing approach for regression residuals presented by Bivand et al. (2009) are available in **spdep** but not elsewhere. These approaches are based on Tiefelsdorf and Boots (1995), Tiefelsdorf and Boots (1997) and Tiefelsdorf (2000), and also apply to local Moran's I for regression residuals.

2.1.2 Geary's *C*

Geary's C (Geary 1954) was discussed by Duncan et al. (1961) and in Cliff and Ord (1969) and their subsequent work. It appears that this global measure has not been applied to the same extent as Moran's I, but it is implemented in a number of the software applications considered here. Geary's C is defined as (Cliff and Ord 1981, p. 17, equation 1.16):

$$C = \left(\frac{(n-1)}{2S_0}\right) \frac{\sum_{(2)} w_{ij}(x_i - x_j)^2}{\sum_{i=1}^{n} z_i^2}. \tag{8}$$

Its expectation is given as (Cliff and Ord 1981, p. 21, equation 1.40):

$$E(C) = 1. \tag{9}$$

Variance terms are defined again under assumptions of normality and randomisation. First the simpler randomisation definition is (Cliff and Ord 1981, p. 21, equation 1.41):

$$\mathrm{Var}_N(C) = \frac{(2S_1 + S_2)(n-1) - 4S_0^2}{2(n+1)S_0^2}. \tag{10}$$

The definition of the variance under randomisation is (Cliff and Ord 1981, p. 21, equation 1.42):

$$\begin{aligned} \mathrm{Var}_R(C) = \frac{1}{n(n-2)(n-3)S_0^2} \Big\{ &(n-1)S_1 \left[n^2 - 3n + 3 - (n-1)b_2\right] \\ &- \frac{1}{4}(n-1)S_2 \left[n^2 + 3n - 6 - (n^2 - n + 2)b_2\right] \\ &+ S_0^2 \left[n^2 - 3 - (n-1)^2 b_2\right]\Big\}. \end{aligned} \tag{11}$$

The standard normal deviate has a reversed numerator in the original development in Cliff and Ord (1969, p. 29, equation 13):

$$Z(C) = \frac{E(C) - C}{\sqrt{\mathrm{Var}_*(C)}}. \tag{12}$$

2.1.3 Getis–Ord *G*

The Getis–Ord global G measure arose in connection with exploration of local measures of spatial association in Getis and Ord (1992), intending to use G and its local variants to supplement Moran's I. The general G statistic is simplified by dropping the explicit $d()$ term in $w(d)_{ij}$ in their development (Getis and Ord 1992, p. 194, equation 5):

$$G = \frac{\sum_{(2)} w_{ij} x_i x_j}{\sum_{(2)} x_i x_j}, \tag{13}$$

Note that the summations as defined above strictly enforce $j \neq i$. The expectation, again adjusting n for no-neighbour observations at the choice of the implementation and analyst, is (Getis and Ord 1992, p. 195, equation 6):

$$E(G) = \frac{S_0}{n(n-1)} \tag{14}$$

The $E(G^2)$ term is relatively complicated, built up of many of the same building blocks as those used in the equivalent formulae for the analytical distributions of Moran's I and Geary's C (Getis and Ord 1992, p. 195):

$$E(G^2) = \frac{\left[B_0 m_2^2 + B_1 m_4 + B_2 m_1^2 m_2 + B_3 m_1 m_3 + B_4 m_1^4\right]}{(m_1^2 - m_2) n(n-1)(n-2)(n-3)} \tag{15}$$

where $m_j = n^{-1} \sum_{i=1}^{n} x_i^j$, $j = 1, 2, 3, 4$, and $B_0 = (n^2 - 3n + 3)S_1 - nS_2 + 3S_0^2$; $B_1 = -[(n^2 - n)S_1 - 2nS_2 + 6S_0^2]$ [see also correction in Getis and Ord (1993)]; $B_2 = -\left[2nS_1 - (n+3)S_2 + 6S_0^2\right]$; $B_3 = 4(n - 1S_1 - 2(n+1)S_2 + 8S_0^2$; and $B_4 = S_1 - S_2 + S_0^2$.

Finally we reach the variance term as (Getis and Ord 1992, p. 195, equation 7):

$$\mathrm{Var}(G) = E(G^2) - [E(G)]^2 \tag{16}$$

and the standard normal deviate:

$$Z(G) = \frac{G - E(G)}{\sqrt{\mathrm{Var}(G)}} \tag{17}$$

2.2 Local measures

At about the same time in the early and mid 1990s, local indicators of spatial association (LISA), spatially structured random effects, and spatial scan statistics emerged. The first two permitted the structure of spatial autocorrelation to be mapped to the units of observation in an inferential framework, while LISA and spatial scan statistics both claimed to make it possible to explore hotspots, although only spatial scan statistics have robust inferential underpinnings in this respect.

2.2.1 Getis–Ord G_i

In discussing G_i and G_i^*, Getis and Ord (1992) follow up incomplete work on spatial correlograms that had its origins in the 1970s by suggesting using distance to analyse spatial association. Since areal data may be represented by a point, perhaps a centroid, chosen to represent observations with polygonal support, or topological buffering may be used to find neighbours within distance bands. They followed up with a series of articles (Ord and Getis 1995; Getis and Ord 1996; Ord and Getis 2001) refining the measures, and removing some restrictions placed on the version presented in 1992.

The local G_i measure is in later work expressed as a standard deviate (Getis and Ord 1996, p. 263, equation 14.2):

$$Z(G_i) = \frac{\left[\sum_{j=1}^n w_{ij}x_j\right] - \left[\sum_{j=1}^n w_{ij}\bar{x}_i\right]}{s_i \left\{\left[\left((n-1)\sum_{j=1}^n w_{ij}^2 - \left(\sum_{j=1}^n w_{ij}\right)^2\right)\right]/(n-1)\right\}^{1/2}}, i \neq j, \quad (18)$$

where $s_i = \sqrt{((\sum_{j=1}^n x_j^2)/(n-1)) - [\bar{x}_i]^2}, i \neq j$, and $\bar{x}_i = (\sum_{j=1}^n x_j)/(n-1), i \neq j$. The left numerator component corresponds to G_i, the right to $E(G_i)$, and the denominator to $\mathrm{Var}(G_i)$.

In Eq. 18, the condition that $i \neq j$ is central. A further measure, local G_i^* relaxes this constraint, by including i as a neighbour of itself (thereby also removing the no-neighbour problem, because all observations have at least one neighbour). This local measure is expressed as (Getis and Ord 1996, p. 263, equation 14.3):

$$Z(G_i^*) = \frac{\left[\sum_{j=1}^n w_{ij}x_j\right] - \left[\left(\sum_{j=1}^n w_{ij}\right)\bar{x}^*\right]}{s^* \left\{\left[\left((n-1)\sum_{j=1}^n w_{ij}^2 - \left(\sum_{j=1}^n w_{ij}\right)^2\right)\right]/(n-1)\right\}^{1/2}}, \text{all } j, \quad (19)$$

where $s^* = \sqrt{((\sum_{j=1}^n x_j^2)/n) - \bar{x}^{*2}}$, and $\bar{x}^* = (\sum_{j=1}^n x_j)/n$, all j.

2.2.2 Moran's I_i

The local Moran's I_i measure of spatial association was introduced by Anselin (1995), and further elaborated in the context of the Moran scatterplot in Anselin (1996). The inferential development of the measure was considered by Getis and Ord (1996) and refined by Sokal et al. (1998). Work on Saddlepoint approximation and exact calculation of the standard normal deviate for regression residuals, including residuals from spatial regression models accounting for global autocorrelation, followed from similar developments for global Moran's I referred to in Sect. 2.1.1 (Tiefelsdorf 2002; Bivand et al. 2009).

Local Moran's I_i values are constructed as the n components used to reach global Moran's I (Anselin 1995, p. 99, equation 12):

$$I_i = \frac{z_i \sum_{j=1}^n w_{ij}z_j}{m_2}, \quad (20)$$

where[4] $m_2 = n^{-1}\sum_{i=1}^n z_i^2$. We once again assume that the global mean $\bar{x}$ is an adequate representation of the variable of interest x. The relationship between the

[4] In implementations we also find $m_2 = (n-1)^{-1}\sum_{i=1}^n z_i^2$, but this does not seem to have support in the original source.

sum of the local I_i and global I is (Anselin 1995, p. 99, equation 10):

$$I = \frac{\sum_{i=1}^{n} I_i}{S_0}, \tag{21}$$

Based on the development in Cliff and Ord (1981), the expectation and variance of I_i may be shown as follows; first the expectation (Anselin 1995, p. 99, equation 13):

$$E(I_i) = \frac{-w_i}{(n-1)} \tag{22}$$

where $w_i = \sum_{j=1}^{n} w_{ij}$. The variance under the randomisation assumption may be defined as (Anselin 1995, p. 99, equation 14, and p. 115):

$$\begin{aligned}\mathrm{Var}_{\mathrm{Anselin}}(I_i) = w_{i(2)}(n-b_2)/(n-1)\\ + 2w_{i(kh)}(2b_2-n)/\,[(n-1)(n-2)]\\ - \frac{w_i^2}{(n-1)^2}\end{aligned} \tag{23}$$

where[5] $b_2 = (n^{-1}\sum_{i=1}^{n} z_i^4)/m_2^2$, $w_{i(2)} = \sum_{j=1}^{n} w_{ij}^2$ and $w_{i(kh)} = \frac{1}{2}\sum_{k\neq i}\sum_{h\neq i} w_{ik}w_{ih}$. However, the $w_{i(kh)}$ term presents implementation difficulties, and Sokal et al (1998, p. 351) have argued that it should be further constrained by imposing $k \neq h$ in addition, leading to (Sokal et al. 1998, p. 334, equation 5, and p. 351, equation A4*):

$$\begin{aligned}\mathrm{Var}_{\mathrm{Sokal}}(I_i) = w_{i(2)}(n-b_2)/(n-1)\\ + (w_i^2 - w_{i(2)})(2b_2-n)/\,[(n-1)(n-2)]\\ - \left[\frac{-w_i}{(n-1)}\right]^2\end{aligned} \tag{24}$$

3 Software implementations

While it is probably the case that institutional setting and need determine the desirability of comparing and/or benchmarking implementations with each other, it is more likely that open source developers will wish to publish results. In earlier work, Bivand (1998, 2008) has attempted to show that implementations are equivalent in terms of results if not always in performance. Bivand and Piras (2015) survey a range of implementations of techniques for spatial econometrics. This article extends this work to cover implementations of some measures of spatial association and has taken into account chosen software applications.

[5] Again, division by $(n-1)$ is encountered in implementations.

3.1 Crimestat

CrimeStat[6] is a closed-source Windows application that is free for download. It is described in Levine (2006, 2017), and the version used here is 4.02, running under Wine on Fedora Linux. CrimeStat is well-documented, but it appears that the multiple comparison issue is not highlighted in the online help page for hotspot analysis, although it is mentioned in the online manual. Crimestat does not output binary results, but permits export as rounded values in DBF files. It only permits fully connected inverse distance weights without row-standardization for I and I_i, and distance bands for G and G_i with user-choosable thresholds (for Euclidean and spherical distance). It does not permit import or export of weights; it can read point files in ESRI Shapefile format to import observations with point support. It provides global Moran's I, Geary's C and Getis–Ord G, and local Moran's I_i and Getis–Ord G_i. Its data and weights import and export facilities are the most limited, as are its range of choice of user-generated weights, and for this reason it provided the weights specifications used in most of this comparison.

3.2 ArcGIS

As Scott and Janikas (2010) recount, spatial statistics tools were added to ArcGIS 9 from 2004; as this release of ArcGIS supports Python as a tool and procedure development language, this is how the tools are written. It provides global Moran's I and Getis–Ord G, and local Moran's I_i and Getis–Ord G_i^*. The help pages explain clearly the multiple comparison problem for local measures and provide the possibility of reporting probability values adjusted for false discovery rate. Use of measures of spatial association in ArcView and other earlier ESRI products is described by Wong and Lee (2005). The version used here is ArcGIS 10.5 Desktop on Windows; rounded output in DBF files and crafted output through Python as numpy arrays has been used. Data and weights may be read in a large number of ways, using the spatial weights file (SWF) format also found in PySAL, so that we may be confident that the Python functions in ArcGIS are receiving the same input data and weights as the other implementations.

3.3 GeoDa

As Anselin et al. (2006) relate, GeoDa is a continuing reinvention of the original SpaceStat package (Anselin 1992) and has moved over time from a closed-source Windows implementation to an open source[7] multi-platform[8] application. The version used here is 1.12.1.59 for Windows running under Wine on Fedora Linux. The documentation explains clearly the multiple comparison problem for local measures, with reference to Caldas de Castro and Singer (2006). For global measures, GeoDa

[6] https://nij.gov/topics/technology/maps/pages/crimestat.aspx.

[7] https://github.com/GeoDaCenter/geoda/.

[8] https://spatial.uchicago.edu/geoda.

provides on-screen rounded output, and for local measures, rounded output in the DBF part exported in ESRI Shapefile format; it reads and writes many data and weights formats. Of the measures provided, we have used global Moran's I and local Moran's I_i and Getis–Ord G_i and G_i^*.

3.4 PySAL

The development of PySAL[9] is described by Rey and Anselin (2007) and Rey et al. (2015). It is an open source[10] package of Python modules for a growing range of tasks in spatial analysis. The version used here is 1.14.3 run from R using the **reticulate** package (Allaire et al. 2018). PySAL can read and write spatial weights files in a number of formats and can read and write data files. Using **reticulate**, binary input and output has been possible. We have used PySAL implementations of global Moran's I, Geary's C, Getis–Ord G, and local Moran's I_i and Getis–Ord G_i and G_i^*. The documentation of the local measures does not seem to discuss multiple comparisons.

3.5 R: spdep

There are a number of implementations of measures of spatial association in R packages, but because the **spdep**[11] contains most of those chosen for comparison, it will receive proportionate attention. The test functions have also been modified so as to permit the reproduction of matching results where other implementations have chosen other readings of the sources for the methods. The test functions were first described in Bivand and Gebhardt (2000) before being made available as a package (Bivand 2006). The version used here is 0.7–7, and like all published CRAN packages, **spdep**[12] is open source. The package provides a wide range of functions for creating, manipulating, reading and writing spatial weights, and implementations of global Moran's I, Geary's C, Getis–Ord G, and local Moran's I_i and Getis–Ord G_i and G_i^*. The local measures function documentation discusses the adjustment of probability values for multiple comparisons, using `p.adjust`, and the variant `spdep::p.adjustSP` which adjusts for the number of comparisons for nonzero neighbour weights only, provided as a less conservative speculation without proven theoretical bases.

4 Test data and locations

We have chosen to use data and locations utilized in a Consumer Data Research Centre (CDRC) tutorial[13] by Guy Lansley and James Cheshire, using UK 2011 census data for the London Borough of Camden and aggregation entity boundaries in planar coordinates. We are grateful to the authors of the tutorial for their permission to use

[9] http://pysal.readthedocs.io/en/latest/.

[10] https://github.com/pysal.

[11] https://cran.r-project.org/package=spdep.

[12] https://github.com/r-spatial/spdep/.

[13] https://data.cdrc.ac.uk/tutorial/an-introduction-to-spatial-data-analysis-and-visualisation-in-r.

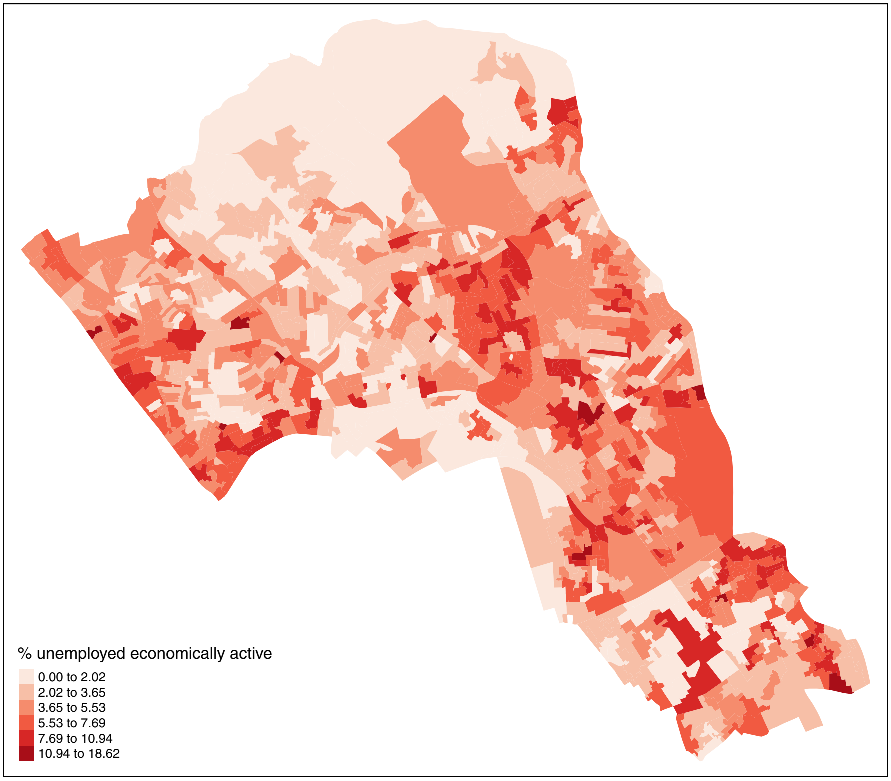


Fig. 1 London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

this data set for this comparison. Output Areas (OA) are the basic aggregation entities, grouped into LSOA and MSOA (Lower and Middle layer Super Output Areas); in the Borough of Camden there are 749 OA, 133 LSOA and 28 MSOA. In the tutorial, several rate variables are used; here we restrict ourselves to unemployment among economically active residents, calculated as a percentage from counts. Figure 1 shows the spatial distribution of the 2011 Census-based OA unemployment rates among economically active residents. The north of the borough contains Hampstead Heath, London Zoo is central, while the British Museum is toward the south of the borough.

```
> Employment0 <- read.csv("KS601EW_oa11.csv")
> Employment <- Employment0[, c(1:2, 6, 20)]
> names(Employment) <- c("OA11CD", "all_categories_economic_activity",
+      "economically_active_unemployed", "Unemployment")
> library(sf)
> output_areas <- st_read("Camden_oa11.shp")
> oa_census <- merge(output_areas, Employment, by = "OA11CD")
```

The aggregation entities have areal support (counts within polygons) which could be used in GeoDa, ArcGIS, PySAL and **spdep**; however as Crimestat requires point support, the positions of the observations are represented by polygon centroids. This

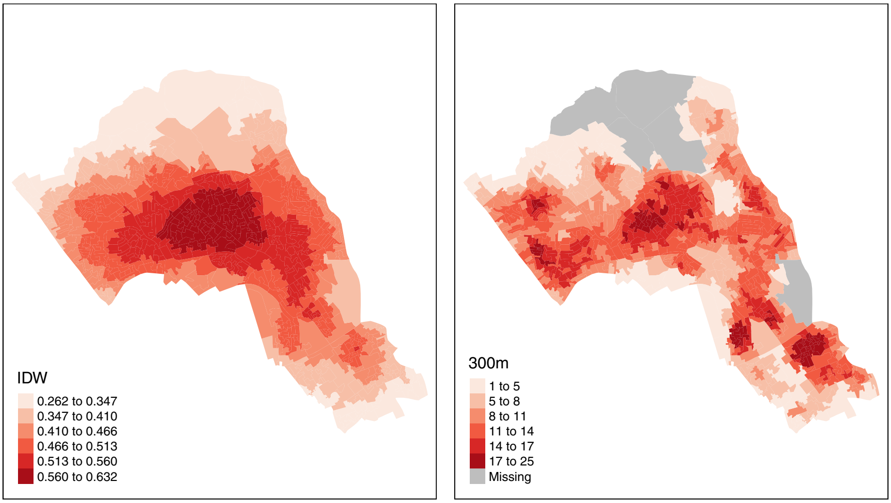


Fig. 2 Row sums of Output Area spatial weights; left panel: inverse distance weights between polygon centroids, right panel: 300 m binary centroid distance weights

departs from the use of polygon contiguities in the parts of the tutorial not dealing with Getis–Ord G and G_i measures, where contiguity neighbours were used.

As CrimeStat does not permit the import of spatial weights, its specifications have been replicated and used. For Moran's I, I_i and Geary's C, CrimeStat uses general inverse distance weighting (IDW) including all point observations, so here OA centroids are used, not polygon boundaries; all distances are measured in metres.

```
> oa_census_pt <- st_centroid(oa_census)
> crds <- st_coordinates(oa_census_pt)
> library(spdep)
> alldnb <- dnearneigh(crds, 0, 10000)
> dists <- nbdists(alldnb, crds)
> idw <- lapply(dists, function(x) 1/x)
> lw_idw <- nb2listw(alldnb, glist = idw, style = "B")
```

For Getis–Ord G and G_i, CrimeStat requires binary distance bands, here set to inter-centroid distances of 300 m or less. Figure 2 shows the sum of weights by OA for the two weighting schemes. The IDW scheme gives more weight to the central parts of the borough near Chalk Farm, while the binary 300 m weights accumulate in areas where the OAs are closer to each other.

```
> dnb_300 <- dnearneigh(crds, 0, 300)
> lwd_300 <- nb2listw(dnb_300, style = "B", zero.policy = TRUE)
> lwd_300s <- nb2listw(include.self(dnb_300), style = "B")
```

In a few cases where CrimeStat is not involved, polygon neighbour "queen" contiguities are used for polygons sharing at least one boundary point.

```
> nb_q <- poly2nb(as(oa_census, "Spatial"), queen = TRUE)
> lw <- nb2listw(nb_q, style = "B")
> lwW <- nb2listw(nb_q, style = "W")
```

Table 2 Global Moran's I, inverse distance weights, normality assumption; `spdep::moran.test (*)` gives results from `tstN_CS` with $E(I)^2$ omitted in Var(I)

	I	$E(I)$	Var(I)	$Z(I)$
CrimeStat	0.048459	– 0.001337	7.927432e–06	
spdep::moran.test (*)	0.048459	– 0.001337	7.927365e–06	17.685942
spdep::moran.test	0.048459	– 0.001337	6.140067e–06	20.095835
PySAL::Moran	0.048459	– 0.001337	6.140067e–06	20.095835
spdep::lm.morantest	0.048459	– 0.001337	6.140067e–06	20.095835
spdep::lm.morantest.sad	0.048459			8.256678
spdep::lm.morantest.exact	0.048459			4.991673

5 Global test results

The global test results are scalar, and so can be shown in tabular form. They are also not very exciting, as we wish to find output that is identical after rounding has been accounted for. This is similar to the kinds of results reported by Bivand and Piras (2015), and as experienced there, some differences have been removed during the preparation of this article (PySAL has been updated to address issues uncovered during work on this comparison). It is seldom the case that inferences would be changed by using other software on the same data, except where the standard deviate is close to a chosen confidence interval.

5.1 Moran's *I*

Starting with Moran's I with general IDW weights, we see that Table 2 with variance terms calculated under the normality assumption shows good agreement in estimates of Moran's I; CrimeStat 4.0.2 values are copied from rounded text file output but all others are binary, including PySAL 1.14.3 using **reticulate**. The `spdep::moran.test (*)` line shows that the standard deviance difference between CrimeStat and default `moran.test` are due to the omission of the $-E(I)^2$ term in Var(I) in CrimeStat (CrimeStat reports $E(I)$ and $\sqrt{\text{Var}(I)}$). Tests on regression residuals from a model only including the intercept give the same values of I, and the standard test `spdep::lm.morantest` is the same as `spdep::moran.test` under normality. However, Saddlepoint approximation (Tiefelsdorf 2002) and exact (Bivand et al. 2009) estimates of $Z(I)$ give very different values.

```
> tstN <- moran.test(oa_census$Unemployment, lw_idw, randomisation = FALSE)
> tstN_CS <- moran.test(oa_census$Unemployment, lw_idw, randomisation = FALSE,
+     drop.EI2 = TRUE)
> OLS <- lm(Unemployment ~ 1, oa_census)
> tstNlm <- lm.morantest(OLS, lw_idw)
> tstNsad <- lm.morantest.sad(OLS, lw_idw)
> tstNex <- lm.morantest.exact(OLS, lw_idw)
```

Since the ArcGIS `SpatialAutocorrelation_stats` function only seems to report Var(I) under randomisation, it is included in Table 3, and agrees with

Table 3 Global Moran's I, inverse distance weights, randomisation assumption; `spdep::moran.test (*)` gives results from `tstR_CS` with $E(I)^2$ omitted in $\text{Var}(I)$

	I	$E(I)$	$\text{Var}(I)$	$Z(I)$
CrimeStat	0.048459	– 0.001337	7.916183e–06	
spdep::moran.test (*)	0.048459	– 0.001337	7.916116e–06	17.698503
spdep::moran.test	0.048459	– 0.001337	6.128819e–06	20.114267
PySAL::Moran	0.048459	– 0.001337	6.128819e–06	20.114267
ArcGIS	0.048459	– 0.001337	6.128819e–06	20.114267

Table 4 Global Moran's I, binary 300 m distance weights, randomisation assumption

	adjust.n	I	$E(I)$	$\text{Var}(I)$	$Z(I)$
spdep::moran.test	TRUE	0.197848	– 0.001346	2.221678e–04	13.363973
spdep::moran.test	FALSE	0.199178	– 0.001337	2.221774e–04	13.452284
PySAL::Moran	FALSE	0.199178	– 0.001337	2.221774e–04	13.452284
ArcGIS	FALSE	0.199178	– 0.001337	2.221774e–04	13.452284

`spdep::moran.test` (default assumption randomisation) and `PySAL::Moran`. Once again, CrimeStat drops the $E(I)^2$ term in $\text{Var}(I)$. In `PySAL::Moran`, the $\text{Var}(I)$ term was affected by a bug for versions before 1.14.1.[14]

```
> tstR <- moran.test(oa_census$Unemployment, lw_idw)
> tstR_CS <- moran.test(oa_census$Unemployment, lw_idw, drop.EI2 = TRUE)
```

For the randomisation case, we also used the binary 300 m distance weights to check how the implementations handle no-neighbour observations. The results reported in Table 4 are for **spdep** adjusting n in the inferential basis (see Eq. 3), and for **spdep** not adjusting n to match `PySAL::Moran` and ArcGIS. For **spdep**, the `zero.policy=` argument needs to be set to accept 0 as the spatially lagged value of for observations with no neighbours.

```
> tst300 <- moran.test(oa_census$Unemployment, lwd_300, zero.policy = TRUE)
> tst300n <- moran.test(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     adjust.n = FALSE)
```

As it turned out, GeoDa silently row-standardizes imported general weights when reading the same GWT file that was used to read general weights in PySAL. Table 5 shows that we can replicate the value of I within rounding constraints. In addition, implementations in the R packages **ape** (Paradis et al. 2004) and **lctools** (Kalogirou 2017) using dense weights matrices also row-standardize weights internally; the **lctools** version provides $\text{Var}(I)$ under the normality (termed resampling) and randomisation assumptions.

```
> lwW_idw <- nb2listw(alldnb, glist = idw, style = "W")
> tstWN <- moran.test(oa_census$Unemployment, lwW_idw, randomisation = FALSE)
```

[14] https://github.com/pysal/pysal/issues/970.

Table 5 Global Moran's I, row-standardized inverse distance weights

	Randomisation	I	$E(I)$	$\text{Var}(I)$	$Z(I)$
GeoDa		0.048087			
spdep::moran.test	FALSE	0.048087	−0.001337	6.120535e–06	19.977699
lctools::moransI.w	FALSE	0.048087	−0.001337	6.120535e–06	19.977699
spdep::moran.test	TRUE	0.048087	−0.001337	6.109029e–06	19.996503
lctools::moransI.w	TRUE	0.048087	−0.001337	6.109029e–06	19.996503
ape::Moran.I	TRUE	0.048087	−0.001337	6.109029e–06	19.996503

Table 6 Global Moran's I, row-standardized contiguity weights, Monte Carlo (mc) and bootstrap (boot)

	nsim	I	$E(I)$	$\text{Var}(I)$	$Z(I)$
spdep::moran.test		0.268652	−0.001337	4.854350e–04	12.254073
GeoDa	999	0.268652	−0.001100	4.645282e–04	12.515800
spdep::moran.mc	999	0.268652	−0.001615	4.548740e–04	12.672066
PySAL::Moran	999	0.268652	−0.001003	4.923302e–04	12.152926
boot::boot (parametric)	999	0.268652	−0.000716	4.974985e–04	12.076735

```
> tstWR <- moran.test(oa_census$Unemployment, lwW_idw)
> B <- listw2mat(lw_idw)
> lctI <- lctools::moransI.w(x = oa_census$Unemployment, w = B)
> apeI <- ape::Moran.I(oa_census$Unemployment, B)
```

Most implementations offer bootstrap, Monte Carlo or Hope-type approaches to inference by permutation. The observed values are redistributed using sampling without replacement in the permutation cases. It is not possible to ensure the same stream of pseudorandom numbers across the implementations. The values reported in Table 6 of $E(I)$ and $\text{Var}(I)$ are the means and variances of the samples. For comparison, the output of Moran's I under randomisation for the same data and weights is provided. In addition, a parametric bootstrap is reported with input values drawn from the normal distribution using the mean and standard deviation of the input data. Inference on any of these would correspond to the standard result under randomisation, so the claim that these approaches provide robustness against distributional assumptions is probably not of practical importance.

```
> tstcWR <- moran.test(oa_census$Unemployment, lwW)
> set.seed(1)
> tstcWMC <- moran.mc(oa_census$Unemployment, lwW, nsim = 999, return_boot = TRUE)
```

```
> set.seed(1)
> boot_out <- EBImoran.mc(oa_census[[3]], oa_census[[2]], lwW, nsim = 999,
+     return_boot = TRUE)
```

There are several implementations of the Assunção and Reis (1999) Empirical Bayes Moran's I, taking the count of events and the base count rather than the rate. We again use row-standardized contiguity weights and permutation bootstrap for three

Table 7 Empirical Bayes Moran's I, row-standardized contiguity weights, Monte Carlo, bootstrap

	nsim	I	$E(I)$	$\mathrm{Var}(I)$	$Z(I)$
GeoDa	999	0.269000	– 0.002200	4.680018e–04	12.536200
spdep::EBImoran.mc	999	0.269000	– 0.001591	4.557487e–04	12.675073
PySAL::Moran_Rate	999	0.269000	– 0.000633	5.271059e–04	11.744200
DCluster::moranI.pboot	999	0.268652	– 0.000488	5.067401e–04	11.955984

Table 8 Summary of Moran's I capabilities, inverse distance weights

	spdep	PySAL	CrimeStat	ArcGIS
$Z(I)$ under normality	✓	✓	✓	✗
$Z(I)$ under randomisation	✓	✓	✓	✓
Equation 6 $\mathrm{Var}_*(I)$	✓	✓	✗	✓
Permutation $Z(I)$	✓	✓	✓	✓
Saddlepoint approximation $Z(I)$	✓	✗	✗	✗
Exact $Z(I)$	✓	✗	✗	✗

cases. The **DCluster** case is a Negative Binomial parametric bootstrap described by Gómez-Rubio et al. (2005). The results are shown in Table 7, and in this case show little difference from the global measure on the percentage rate for row-standardized contiguity weights.

Table 8 provides a summary of software capabilities for the base case of inverse distance weights without row-standardization. All of `spdep::moran.test()`, `PySAL::Moran()`, CrimeStat and `ArcGIS::GlobalI()` provide $Z(I)$ under randomisation and using permutation. `ArcGIS::GlobalI()` does not provide $Z(I)$ under normality, and CrimeStat does not subtract $[E(I)]^2$ in Eq. 6 when calculating $\mathrm{Var}_*(I)$. Only **spdep** provides exact and Saddlepoint approximation values of $Z(I)$.

5.2 Other global indicators

5.2.1 Geary's *C*

We return to the IDW general weights to accommodate CrimeStat for a comparison of Geary's C (Table 9). There are many fewer implementations of Geary's C, probably because it is more computationally demanding, especially when the spatial weights are dense, as in this case where there are many more pair differences to compute. PySAL and CrimeStat output uses the standard z-value, so reversing the sign (Eq. 12, and Cliff and Ord 1969, p. 29, equation 13).

```
> tstN <- geary.test(oa_census$Unemployment, lw_idw, randomisation = FALSE)
> tstR <- geary.test(oa_census$Unemployment, lw_idw)
```

Table 9 Global Geary's C, inverse distance weights

	Randomisation	C	$E(C)$	Var(C)	$Z(C)$
CrimeStat	FALSE	0.954406	1.000000		– 5.299659
PySAL::Geary	FALSE	0.954406	1.000000	7.401481e–05	– 5.299659
spdep::geary.test	FALSE	0.954406	1.000000	7.401481e–05	5.299659
PySAL::Geary	TRUE	0.954406	1.000000	1.218734e–04	– 4.130026
spdep::geary.test	TRUE	0.954406	1.000000	1.218734e–04	4.130026

Table 10 Global Getis–Ord G, binary 300 m distance weights

	adjust.n	G	$E(G)$	Var(G)	$Z(G)$
CrimeStat	FALSE	0.017971	0.015714		8.462774
PySAL::G	FALSE	0.017971	0.015714	7.107530e–08	8.462774
spdep::globalG	FALSE	0.017971	0.015714	7.107530e–08	8.462774
spdep::globalG	TRUE	0.017971	0.015926	6.786192e–08	7.846753

5.2.2 Getis–Ord *G*

The comparisons shown in Table 10 use the binary distance definition used by CrimeStat; the cut off threshold is set to 300 m. The three implementations (CrimeStat, `PySAL::G` and `spdep::globalG`) are identical apart from rounding. Earlier, some implementations differed by not correcting the variance using Getis and Ord (1993), but this has been dealt with now. CrimeStat and PySAL do not adjust n for no-neighbour observations.

```
> tst <- globalG.test(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     adjust.n = TRUE)
> tstn <- globalG.test(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     adjust.n = FALSE)
```

Getting an exact match for the ArcGIS global Getis–Ord G with binary 300 m distance weights turned out to be quite demanding. In ArcGIS, some internal products are accumulated only for observations with neighbours, but others use the full vector of the variable of interest. If `adjust.x=TRUE`, the x vector is shortened by dropping the non-neighbour observations. However, the denominator in Eq. 13, $\sum_{(2)} x_i x_j,\ j \neq i$, is implemented as sum of the product of x'_i dropping no-neighbour observations with x_j, the complete x vector, and then subtracting the cross-product of x'_i. ArcGIS does not adjust n for no-neighbour observations.

```
> tstx <- globalG.test(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     adjust.n = FALSE, adjust.x = TRUE, Arc_all_x = FALSE)
> tstxAG <- globalG.test(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     adjust.n = FALSE, adjust.x = TRUE, Arc_all_x = TRUE)
```

The `adjust.x = TRUE` argument drops no-neighbour observation x values, and the `Arc_all_x = TRUE` uses the complete x vector in one product sum. Table 11 shows

Table 11 Reproducing ArcGIS output for global Getis–Ord G, binary 300 m distance weights

	adjust.x	Arc_all_x	G	$E(G)$	Var(G)	$Z(G)$
spdep::globalG	TRUE	FALSE	0.018131	0.015714	7.276249e–08	8.958640
spdep::globalG	TRUE	TRUE	0.018050	0.015714	7.276249e–08	8.660501
ArcGIS			0.018050	0.015714	7.276249e–08	8.660501

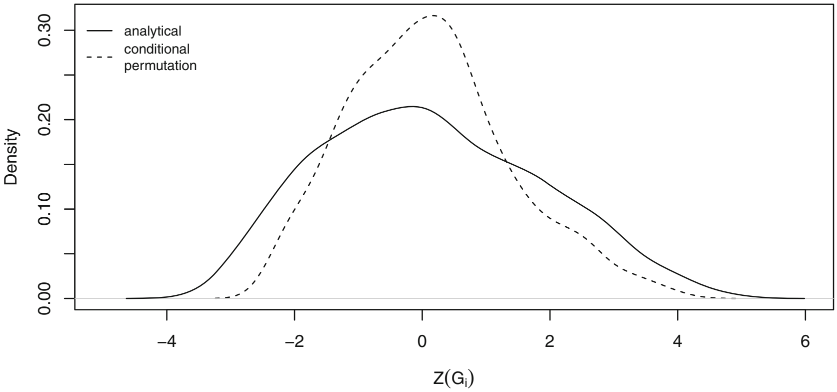


Fig. 3 Density plots of analytical and conditional permutation-based $Z(G_i)$ values, PySAL, 999 samples; London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

that when `Arc_all_x = TRUE`, the value of G is slightly smaller as the denominator is slightly larger. $E(G)$ and Var(G) are the same because the moments of x are calculated leaving out no-neighbour x values consistently, so the difference in $Z(G)$ is caused by the difference in G.

6 Local test results

The comparison of local results is less easy to convey, because each scalar output in the global case is replaced by a vector of n values. This means that we will need to compare vector values between implementations within given precision, while taking into account the precision output to, for example, DBF files.

6.1 Getis–Ord G_i

The CrimeStat, PySAL and **spdep** implementations return values of G_i, $E(G_i)$, Var(G_i) and $Z(G_i)$, while GeoDa returns only G_i. The implementations differ in the values assigned to no-neighbour observations; here these are set to missing (`NA`) if not already so reported for purposes of comparison.

```
> Gi <- localG(oa_census$Unemployment, lwd_300, return_internals = TRUE)
```

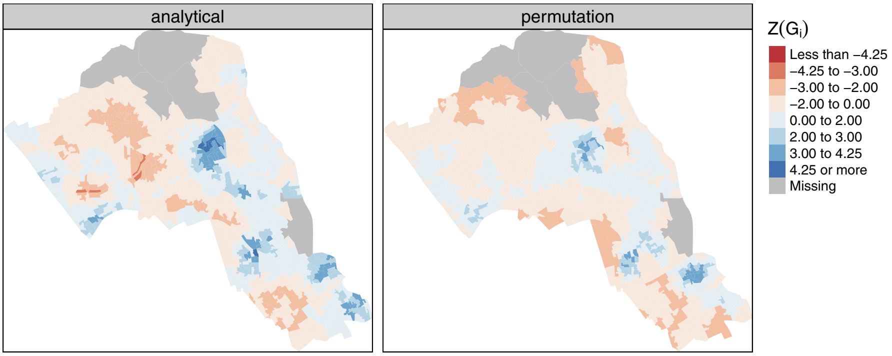


Fig. 4 Analytical and conditional permutation-based $Z(G_i)$ values, PySAL, 999 samples; London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

The results for the PySAL and **spdep** implementations using the binary 300 m distance threshold weights are identical within machine precision, and these agree with those for CrimeStat after rounding to six digits after the decimal sign. Several of the implementations provide conditional permutation-based inference, where all observations except x_i are randomly re-assigned without replacement for the test for observation i, here 999 times. Figure 3 shows density plots of $Z(G_i)$ computed analytically (Eq. 18) and by conditional permutation from the PySAL implementation; it is clear that the conditional permutation-based are more concentrated in the centre of the distribution than the analytical values. Figure 4 contrasts the same values; recall that positive values (blue) of $Z(G_i)$ here correspond to spatial autocorrelation with respect to high unemployment, and negative values (red) to spatial autocorrelation with respect to low unemployment. The correlation between the analytical and permutation-based $Z(G_i)$ values is only 0.737; this result is consistent and is not affected by the number of draws as explored in more detail for the local Moran's I_i case below.

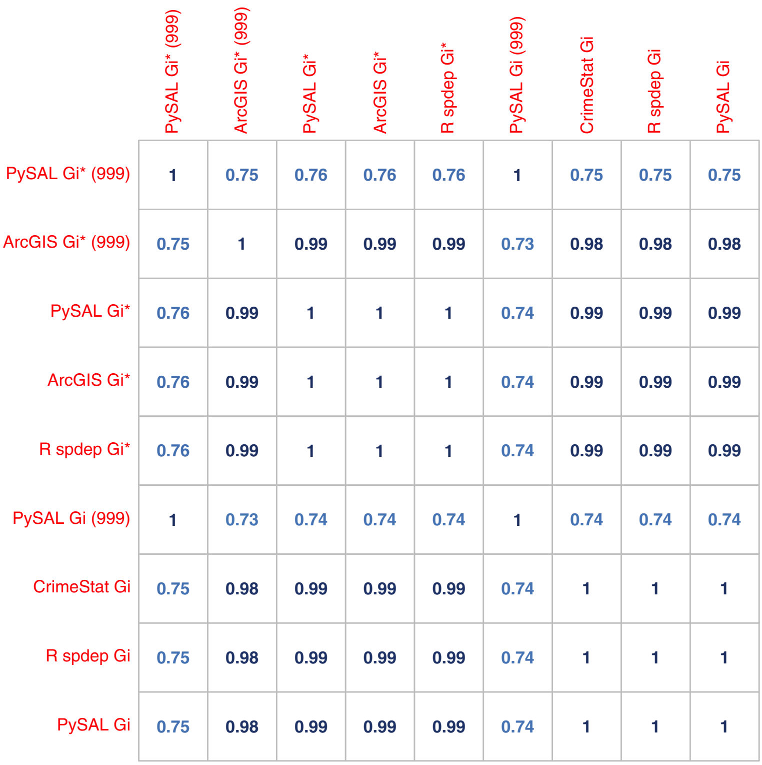
The G_i values returned by GeoDa agree with **spdep** when they are rounded to seven digits after the decimal sign, and when **spdep** uses the `GeoDa=TRUE` argument to accommodate the fact that GeoDa drops x_i values for observations with no neighbours from summations.[15]

```
> Gi_gd <- localG(oa_census$Unemployment, lwd_300, return_internals = TRUE,
+     GeoDa = TRUE)
```

ArcGIS only provides the G_i^* measure, and the ArcGIS values of $Z(G_i^*)$ agree within machine precision with the PySAL and **spdep** implementations for the binary 300 m distance threshold weights. Once again, the GeoDa G_i^* values agree with those from **spdep** when they are rounded to seven digits after the decimal sign, and **spdep** uses the `GeoDa=TRUE` argument. In the G_i^* case, GeoDa only seems to include x_i values in summations when observations have more than one neighbour (not counting itself as a valid neighbour).

```
> lwd_300s <- nb2listw(include.self(dnb_300), style = "B")
> Gi_s <- localG(oa_census$Unemployment, lwd_300s, return_internals = TRUE)
```

[15] https://github.com/GeoDaCenter/geoda/blob/master/Explore/GStatCoordinator.cpp, lines 338–342, 526–527.



	PySAL Gi* (999)	ArcGIS Gi* (999)	PySAL Gi*	ArcGIS Gi*	R spdep Gi*	PySAL Gi (999)	CrimeStat Gi	R spdep Gi	PySAL Gi
PySAL Gi* (999)	1	0.75	0.76	0.76	0.76	1	0.75	0.75	0.75
ArcGIS Gi* (999)	0.75	1	0.99	0.99	0.99	0.73	0.98	0.98	0.98
PySAL Gi*	0.76	0.99	1	1	1	0.74	0.99	0.99	0.99
ArcGIS Gi*	0.76	0.99	1	1	1	0.74	0.99	0.99	0.99
R spdep Gi*	0.76	0.99	1	1	1	0.74	0.99	0.99	0.99
PySAL Gi (999)	1	0.73	0.74	0.74	0.74	1	0.74	0.74	0.74
CrimeStat Gi	0.75	0.98	0.99	0.99	0.99	0.74	1	1	1
R spdep Gi	0.75	0.98	0.99	0.99	0.99	0.74	1	1	1
PySAL Gi	0.75	0.98	0.99	0.99	0.99	0.74	1	1	1

Fig. 5 Correlations between values of $Z(G_i)$ (Eq. 18) and $Z(G_i^*)$ (Eq. 19)—conditional permutation (with numbers of permutations) and analytical; London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

```
> Gi_s_gd <- localG(oa_census$Unemployment, lwd_300s, return_internals = TRUE,
+      GeoDa = TRUE)
```

Figure 5 summarizes the inferential bases for local G_i and G_i^* for analytical and conditional permutation approaches. All the analytical $Z(G_i)$ and $Z(G_i^*)$ are effectively identical within and between groups, suggesting that only providing G_i (CrimeStat) or G_i^* (ArcGIS) is not a problem. The ArcGIS conditional permutation $Z(G_i^*)$ values are very close to the analytical values, but have been reconstructed here from their p values. It is unknown why the PySAL and ArcGIS conditional permutation $Z(G_i^*)$ values differ as much as they do, but this may relate to the reconstruction of the ArcGIS values. As GeoDa reported conditional permutation p values are folded to combine tails, it is not possible to include them in this comparison.

Table 12 summarizes the capabilities of five software implementations of local G_i and G_i^*: `spdep::localG()`, `PySAL::G_Local()`, CrimeStat, `ArcGIS::LocalG()`, and GeoDa. GeoDa, `spdep::localG()` and `PySAL::G_Local()` provide both local G_i and G_i^*, while CrimeStat provides only local G_i and `ArcGIS::LocalG()` only G_i^*. GeoDa does not provide analytical $Z(G_i)$ values, and **spdep** does not provide conditional permutation $Z(G_i)$ values. Taking Fig. 5 into account, it is not obvious

Table 12 Summary of local G_i and G_i^* capabilities, binary 300 m distance weights

	spdep	PySAL	CrimeStat	ArcGIS	GeoDa
Analytical $Z(G_i)$	✓	✓	✓	✓	✗
Conditional permutation $Z(G_i)$	✗	✓	✓	✓	✓
G_i	✓	✓	✓	✗	✓
G_i^*	✓	✓	✗	✓	✓

that the provision of both local G_i and G_i^* is essential; it is further not obvious that conditional permutation offers a stronger inferential basis than analytical values of $Z(G_i)$.

6.2 Moran's I_i

As in the global case for Moran's I, CrimeStat uses general inverse distance weights between the centroids of all output area polygons for local Moran's I_i. Starting with this case, we again note that CrimeStat and GeoDa export results in DBF format subject to rounding. In both **spdep** and CrimeStat, local Moran's I_i is calculated such that the denominator of m_2 in Eq. 20 is n, but may be set to $n - 1$ in **spdep**, and equivalently in the b_2 term, if the argument `mlvar=FALSE`. The values of I_i returned by **spdep** and CrimeStat agree to six digits after the decimal sign with default `mlvar=TRUE`; however, the values of $Z(I_i)$ differ somewhat (mean absolute difference: 0.0004928), although they are perfectly correlated. This suggests that CrimeStat perhaps uses Eq. 23, since **spdep** uses Eq. 24 to define $\text{Var}(I_i)$.

```
> base_Ii_ml <- localmoran(oa_census$Unemployment, lw_idw)
```

Setting `mlvar=FALSE` in **spdep** gives output that agrees within machine precision for I_i and $Z(I_i)$ to that of ArcGIS, implying that both use Eq. 24 to define $\text{Var}(I_i)$. Comparing **spdep** with `mlvar=FALSE` and PySAL gives agreement within machine precision for I_i, but does not compute or return any analytical inferential results.

```
> base_Ii <- localmoran(oa_census$Unemployment, lw_idw, mlvar = FALSE)
```

Again, GeoDa appears to row-standardize on reading the general inverse distance weights. The values of I_i reported by GeoDa agree with **spdep** with `mlvar=FALSE` when they are rounded to seven digits after the decimal sign, and PySAL and **spdep** I_i values with `mlvar=FALSE` for the row-standardized case agree within machine precision with PySAL.

```
> base_IiW <- localmoran(oa_census$Unemployment, lwW_idw, mlvar = FALSE)
```

The R **lctools** package provides the `l.moransI` function, which presupposes k-nearest neighbour weights and permits row-standardized or Bisquare kernel weights; for $k = 6$, the values of I_i agree with **spdep** with `mlvar=TRUE`.

```
> crds <- st_coordinates(pt_out)
> k6 <- knn2nb(knearneigh(crds, k = 6))
> klw <- nb2listw(k6, style = "W", zero.policy = TRUE)
```

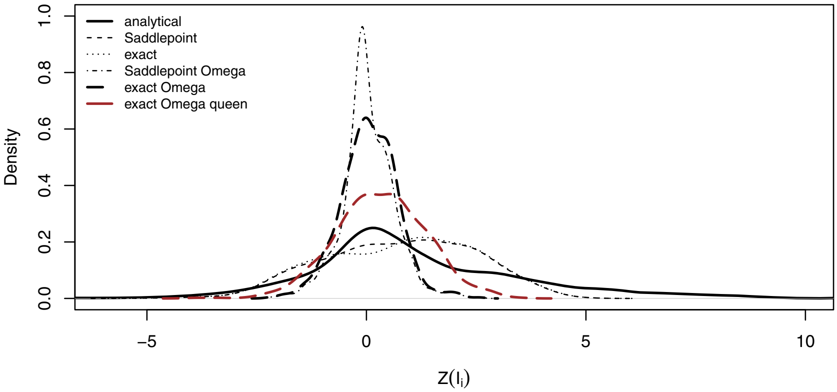


Fig. 6 Density plots of analytical, Saddlepoint approximation and exact $Z(I_i)$ values using the same inverse distance weights and for exact Ω queen binary contiguities for the exact measures only with inverse distance weights for the SAR model, **spdep**; horizontal axis truncated

```
> base_Ii_k6 <- localmoran(oa_census$Unemployment, klw, zero.policy = TRUE)
> lct_Ii <- lctools::l.moransI(crds, 6, x = oa_census$Unemployment,
+     WType = "Binary", scatter.plot = FALSE)
```

The R package **ncf** (Bjornstad 2018) function `lisa` uses row-standardized distance-based weights, and the I_i values agree with **spdep** for a threshold of 300 m and `mlvar=TRUE`.

```
> lwd_300W <- nb2listw(dnb_300, style = "W", zero.policy = TRUE)
> base_Ii_300 <- localmoran(oa_census$Unemployment, lwd_300W, zero.policy = FALSE)
> ncf_Ii <- ncf::lisa(crds[, 1], crds[, 2], oa_census$Unemployment,
+     300, resamp = 0L, quiet = TRUE)
```

Local Moran's I_i as calculated using Saddlepoint approximation (Tiefelsdorf 2002) and exact (Bivand et al. 2009) methods provide inferential alternatives to the analytical methods presented above, and to conditional permutation to be considered later. The I_i values returned are equal to the values with `mlvar=TRUE` when multiplied by $n/2$. Both approaches permit the inclusion of explanatory variables, and the use of a global spatial process to account for global autocorrelation before local autocorrelation is explored. Here we use an intercept-only linear model, and an intercept-only simultaneous autoregressive model to remove a global process defined by the same inverse distance weights matrix. We extend this approach to explore local autocorrelation with different spatial weights to those used to remove the global process, because the actual global data generation process may not be fully captured by the chosen weights.

```
> sad_Ii <- as.data.frame(localmoran.sad(OLS, nb = alldnb, glist = idw,
+     style = "B"))
> exact_Ii <- as.data.frame(localmoran.exact(OLS, nb = alldnb, glist = idw,
+     style = "B"))
> SEM <- errorsarlm(Unemployment ~ 1, data = oa_census, listw = lw_idw)
> lm.target <- lm(SEM$tary ~ SEM$tarX - 1)
> Omega <- invIrW(lw_idw, rho = SEM$lambda)
> sad_Ii_Omega <- as.data.frame(localmoran.sad(lm.target, nb = alldnb,
+     glist = idw, style = "B", Omega = Omega))
```

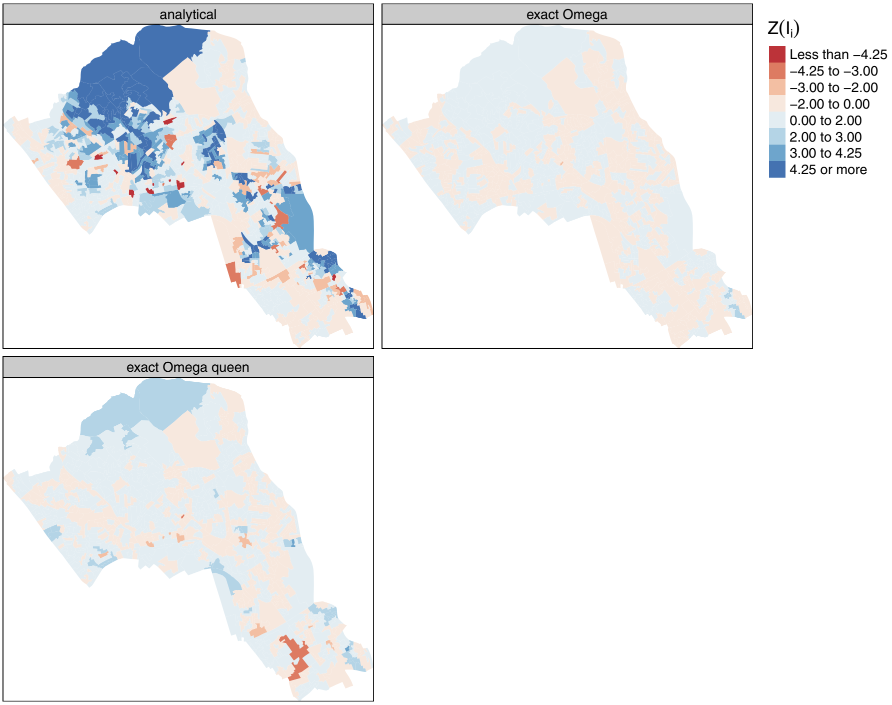


Fig. 7 Analytical and exact $Z(I_i)$ values; the exact values have been calculated after global autocorrelation has been removed by fitting a SAR model; London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

Table 13 Tabulation of OA $Z(I_i)$ values by conventional normal confidence levels, analytical, Saddlepoint approximation and exact $Z(G_i)$ values using the same inverse distance weights and for exact Ω queen binary contiguities for the exact measures only with inverse distance weights for the SAR model

	Analytical	Saddle	Exact	Saddle Ω	Exact Ω	Exact Ω queen
< -4.25	10	1	1	0	0	0
-4.25 to -3	12	6	6	0	0	2
-3 to -2	28	32	33	1	1	9
-2 to 0	193	213	214	380	348	270
0 to 2	289	294	289	364	395	434
2 to 3	67	135	138	4	5	32
3 to 4.25	70	65	66	0	0	2
> 4.25	80	3	2	0	0	0

```
> exact_Ii_Omega <- as.data.frame(localmoran.exact.alt(lm.target,
+     nb = alldnb, glist = idw, style = "B", Omega = Omega))
> exact_Ii_Omega_q <- as.data.frame(localmoran.exact.alt(lm.target,
+     nb = nb_q, style = "B", Omega = Omega))
```

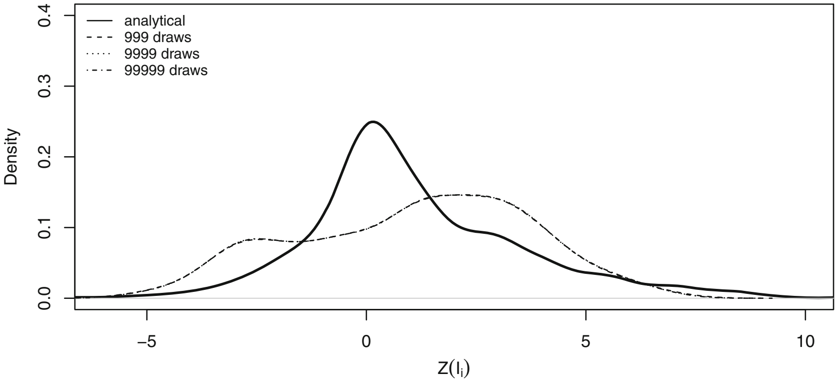


Fig. 8 Density plots of analytical and conditional permutation $Z(I_i)$ for increasing numbers of draws—the lines for conditional permutation $Z(I_i)$ overplot and show how little they differ; horizontal axis truncated

Figures 6, 7 and Table 13 show the dramatic effect on inferential output of using Saddlepoint approximation or exact methods, especially when global autocorrelation has been removed by modelling and only searching for residual local spatial autocorrelation. In this case, and for the choice of inverse distance weights, there is effectively no residual local spatial autocorrelation. When we remove global inverse distance weight-based autocorrelation, and test using binary contiguity weights (exact Ω queen), some residual local spatial autocorrelation is found, but still less than when global spatial autocorrelation is not removed. Even if we had not modelled global spatial autocorrelation, we could have introduced covariates into the mean model with a potentially similar effect, or added a Lower layer Super Output Area random effect in a multilevel approach (as a speculation—the block diagonal group effect might replace the Ω term instead of a global spatial process).

In the case of $Z(G_i)$, we saw (Figs. 3, 4) that the values returned by analytical and conditional permutation were not very similar, both in terms of distribution as might be expected but also in terms of the spatial patterning of tail values in the distribution.

```
> pslm <- pysal$Moran_Local(y, w, transformation = "O", permutations = 999L)
> pslm_9999 <- pysal$Moran_Local(y, w, transformation = "O", permutations = 9999L)
> pslm_99999 <- pysal$Moran_Local(y, w, transformation = "O", permutations = 99999L)
```

We will use the PySAL implementation here, but can note that a non-optimized implementation in R, and the PySAL and ArcGIS implementations of conditional permutation yield $Z(I_i)$ correlated with each other by more than 0.999 (see also Fig. 10); GeoDa does not return $Z(I_i)$ values. This suggests that the implementations are using the same understanding of conditional permutation, and that remaining trivial differences are related to different streams of random numbers.

Figures 8, 9 and Table 14 show not only that increasing the number of draws beyond 999 has no effect (the $Z(I_i)$ values are correlated by more than 0.999), but that the procedure generates more values outside the −24 to 2 range compared to the analytical approach for this data set and weights. Even adjusting probability values by false discovery rate will leave more “unusual” values of local autocorrelation than

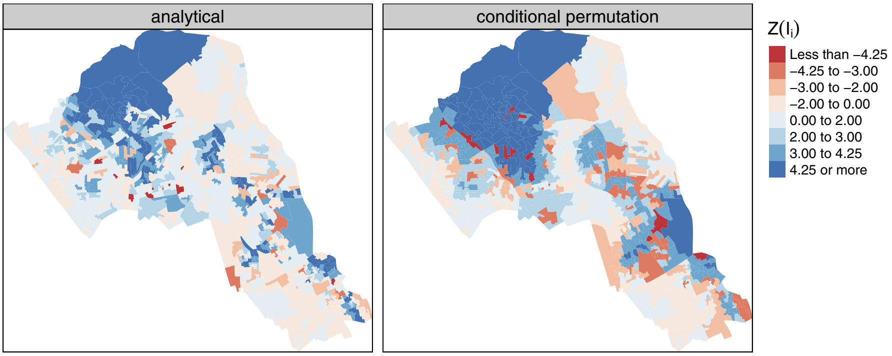


Fig. 9 Analytical and conditional permutation-based $Z(I_i)$ values (99999 draws); London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

Table 14 Tabulation of OA $Z(I_i)$ values by conventional normal confidence levels, analytical and conditional permutation $Z(I_i)$ for increasing numbers of draws, PySAL

	Analytical	999 draws	9999 draws	99999 draws
< -4.25	10	12	11	12
-4.25 to -3	12	48	51	48
-3 to -2	28	67	65	67
-2 to 0	193	127	128	127
0 to 2	289	193	193	193
2 to 3	67	108	104	107
3 to 4.25	70	120	127	123
> 4.25	80	74	70	72

when the analytical approach is used, and the contrast with Saddlepoint approximation and exact methods does not need stressing.

Finally, since some issues were observed in the handling of no-neighbour observations, we reproduce parts of the comparison for the binary 300 m distance threshold weights.

```
> sp_liF <- localmoran(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     mlvar = FALSE)
> sp_liFa <- localmoran(oa_census$Unemployment, lwd_300, zero.policy = TRUE,
+     mlvar = FALSE, adjust.x = TRUE)
```

The I_i values returned by PySAL and **spdep** agree when `mlvar=FALSE`, and the GeoDa values agree with **spdep** when `mlvar=FALSE` and the weights are row-standardized. The ArcGIS I_i values agree with **spdep** when `mlvar=FALSE` and `adjust.x=TRUE`, indicating that summations in ArcGIS omit values of x for no-neighbour observations.

Figure 10 summarizes the results of the different ways of calculating $Z(I_i)$, the standard deviate of local Moran's I_i. The first block of values with a Pearson correlation of 1 is returned by conditional permutation methods. It seems that no advantage is

	PySAL (999)	PySAL (99999)	R (999)	ArcGIS (999)	R spdep (N)	R spdep (R)	CrimeStat (R)	ArcGIS (R)	R spdep (Saddlepoint)	R spdep (Exact)	R spdep (Saddlepoint, Omega)	R spdep (Exact, Omega)
PySAL (999)	1	1	1	1	0.79	0.79	0.79	0.79	0.88	0.88	0.72	0.75
PySAL (99999)	1	1	1	1	0.79	0.79	0.79	0.79	0.88	0.88	0.72	0.75
R (999)	1	1	1	1	0.79	0.79	0.79	0.79	0.88	0.88	0.72	0.75
ArcGIS (999)	1	1	1	1	0.79	0.79	0.79	0.79	0.88	0.88	0.72	0.75
R spdep (N)	0.79	0.79	0.79	0.79	1	1	1	1	0.95	0.95	0.87	0.87
R spdep (R)	0.79	0.79	0.79	0.79	1	1	1	1	0.95	0.95	0.87	0.87
CrimeStat (R)	0.79	0.79	0.79	0.79	1	1	1	1	0.95	0.95	0.87	0.87
ArcGIS (R)	0.79	0.79	0.79	0.79	1	1	1	1	0.95	0.95	0.87	0.87
R spdep (Saddlepoint)	0.88	0.88	0.88	0.88	0.95	0.95	0.95	0.95	1	1	0.87	0.89
R spdep (Exact)	0.88	0.88	0.88	0.88	0.95	0.95	0.95	0.95	1	1	0.87	0.89
R spdep (Saddlepoint, Omega)	0.72	0.72	0.72	0.72	0.87	0.87	0.87	0.87	0.87	0.87	1	0.99
R spdep (Exact, Omega)	0.75	0.75	0.75	0.75	0.87	0.87	0.87	0.87	0.89	0.89	0.99	1

Fig. 10 Correlations between values of $Z(I_i)$—conditional permutation (with numbers of permutations), analytical (Normal and Randomized), Saddlepoint approximation and exact methods (without and with Omega used to remove global autocorrelation); London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

obtained by increasing the number of iterations. The next clear block is generated by the use of analytical methods (one normal assumption returned by the Saddlepoint function in **spdep**, the others under the randomisation assumption) to calculate the expectation and variance of local Moran's I_i. The final two blocks bring together exact and Saddlepoint approximation vales, first without the prior modelling of global autocorrelation, the second based on providing the Ω matrix calculated from earlier fitting of a global model. These latter methods are only available in **spdep**.

Only `PySAL::Moran_Local_Rate` and GeoDa provide Empirical Bayes local I_i rates; these agree within rounding error for row-standardized contiguity weights.

Table 15 gives a summary of local Moran's I_i capabilities for five software implementations: **spdep** (`localmoran()`, `localmoran.sad()` and `localmoran.exact()`), `PySAL::Moran_Local()`, CrimeStat, `ArcGIS::LocalI()` and GeoDa. The correlations shown in Fig. 10 indicate that exact or Saddlepoint approximation $Z(I_i)$ values are a useful contrast to analytical or conditional permutation $Z(I_i)$ values. Uses of Eq. 20 with default $(n-1)$ will seldom change inferences but do confuse users, as do the analytical definitions of $\text{Var}(I_i)$.

Table 15 Summary of local I_i capabilities, inverse distance weights

	spdep	PySAL	CrimeStat	ArcGIS	GeoDa
Analytical $Z(I_i)$	✓	✗	✓	✓	✗
Conditional permutation $Z(I_i)$	✗	✓	✓	✓	✓
Equation 20 default $(n-1)$	✗	✓	✗	✓	✓
Equation 24 $\mathrm{Var}_{\mathrm{Sokal}}(I_i)$	✓	✗	✗	✓	✗
Saddlepoint approximation $Z(I_i)$	✓	✗	✗	✗	✗
Exact $Z(I_i)$	✓	✗	✗	✗	✗

7 Conclusions

In this comparative review of implementations of global and local measures of spatial autocorrelation, we have been able to establish the conditions under which we can account for observed numerical differences in output. These differences are unlikely to affect inferential outcomes for global measures, but user choices for local measures both of software and of inferential method over and above the handling of multiple comparisons will have consequences for conclusions drawn. Only users of **spdep** have access to Saddlepoint approximation of exact methods for local Moran's I_i, and thus to the possibility of the removal of global mis-specification from the data before exploring local measures. In any case, applied users are unlikely to choose to do this, despite these methods often not needing more computing time than conditional permutations.

In particular, it is a matter of concern that the spatial patterns of Z values generated by conditional permutation for local measures differ considerably from those calculated using analytical methods. This means that the further use of local measures to "detect" "hotspots" which is prevalent in applied fields, needs to take account not only of the pressing need to handle false discovery rates, but also of the differences between "hotspots" that might be "detected" using analytical or conditional permutation. Table 14 is particularly worrying, as conditional permutation for this data set generates far more output area $Z(I_i)$ values that exceed $|2|$ than the analytical method, and many of them are in different output areas (182 more conditional permutation values of $Z(I_i) \geq |2|$ compared with analytical; 20 more analytical values of $Z(I_i) \geq |2|$ compared with conditional permutation).

```
> LOSH_idw <- LOSH.cs(oa_census$Unemployment, lw_idw)
> nn <- card(lwd_300$neighbours) > 0
> LOSH_d300 <- LOSH.cs(oa_census$Unemployment[nn], subset(lwd_300,
+     nn))
```

Since conditional permutation assumes that the local and (conditional, without x_i) global distributions of x are equivalent, perhaps the divergence between analytical and conditional permutations is driven by local spatial heterogeneity. Ord and Getis (2012) propose a measure of local spatial heterogeneity (LOSH), and very recently an implementation has been added to **spdep** thanks to Rene Westerholt in connection with Westerholt et al. (2015, 2018). The implementation also includes inferential mechanisms proposed by Xu et al. (2014). Figure 11 shows the values of the measure for two different sets of spatial weights. Values of the measure greater than unity

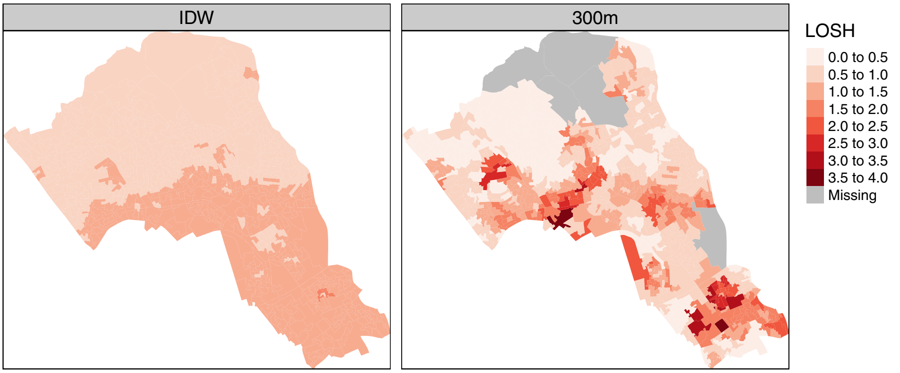


Fig. 11 Local spatial heterogeneity measure for general inverse distance and binary 300 m distance threshold weights; London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

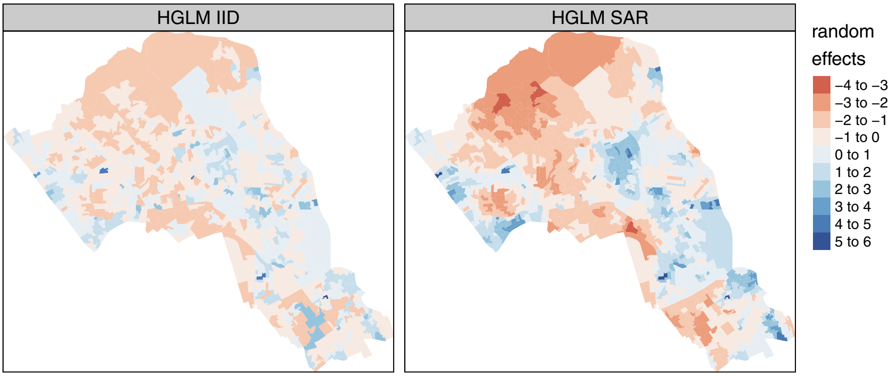


Fig. 12 HGLM IID and SAR random effects—SAR random effects for contiguous queen neighbours; London Borough of Camden: 2011 Census unemployment rates of resident working age population by Output Area

indicate heightened local spatial heterogenity. Not only can we see that local spatial heterogenity is present, but also that general inverse distance weights induce strong smoothing compared to binary 300 m distance threshold weights. This measure is fairly new, and its implementation has only been made available recently, so we can expect more studies of the impact of local spatial heterogeneity, for example, on spatial discrepancies in the inferential bases of local measures of spatial autocorrelation.

In a survey of ways of calculating independent and identically distributed (IID) and spatially structured (here simultaneous autoregressive, SAR) random effects in multilevel models, Bivand et al. (2017) draw attention to the possibility of using spatially structured random effects to explore local spatial autocorrelation. Since random effects estimates come with standard errors, as, for example, in the use of hierarchical generalized linear models by Alam et al. (2015), they may provide an additional way of modelling spatial dependence. Here we have not added covariates or grouping at more aggregated levels, not used the possibility of handling the underlying discrete response by fitting a Poisson regression with an offset, but such flexibility is easily available. Figure 12 shows the fitted IID and queen contiguity SAR random effects for the data set under investigation. Further fitting techniques are reviewed by Bivand et al. (2017).

```
> library(hglm)
> X_hglm <- model.matrix(~1, data = oa_census)
> Z_hglm <- model.matrix(~-1 + factor(OA11CD), data = oa_census)
> hglm_iid <- hglm(y = oa_census$Unemployment, X = X_hglm, Z = Z_hglm)
> M_hglm <- listw2mat(lwW)
> hglm_sar <- hglm(y = oa_census$Unemployment, X = X_hglm, Z = Z_hglm,
+     rand.family = SAR(D = M_hglm))
```

In the course of our comparison, we have established the reasons for observed differences in numerical results between implementations of global and local measures of spatial autocorrelation. We have pointed to the need to draw users' attention to the issue of multiple comparisons in making inferential judgements based on local measures. We have further examined the way in which implementations handle no-neighbour observations. We have raised questions about the appropriateness of relying on conditional permutation as an inferential basis for local measures, and suggested a link to a newer measure of local spatial heterogeneity. We indicate that local measures of spatial autocorrelation are also likely to mislead users in the presence of global autocorrelation, and where the mean model is mis-specified in other ways. These doubts have already been highlighted in the literature, often in the articles introducing local measures, but have unfortunately often been put aside by users. We continue to hope that implementations and this comparison will offer the guidance required to assist users in their application of these measures.

Acknowledgements We would like to thank the editors and reviewers for constructive suggestions that we hope have clarified the conclusions of this comparative study. We would also like to thank Shiyang Ruan for assistance with ArcGIS Python programming.

References

Alam M, Rönnegård L, Shen X (2015) Fitting conditional and simultaneous autoregressive spatial models in hglm. R J 7(2):5–18. http://journal.r-project.org/archive/2015-2/alam-ronnegard-shen.pdf

Allaire JJ, Ushey K, Tang Y (2018) reticulate: interface to 'Python'. https://CRAN.R-project.org/package=reticulate, R package version 1.8

Anselin L (1992) SpaceStat, a software program for analysis of spatial data. National Center for Geographic Information and Analysis (NCGIA), University of California, Santa Barbara

Anselin L (1995) Local indicators of spatial association—LISA. Geogr Anal 27(2):93–115

Anselin L (1996) The Moran scatterplot as an ESDA tool to assess local instability in spatial association. In: Fischer MM, Scholten HJ, Unwin D (eds) Spatial analytical perspectives on GIS. Taylor & Francis, London, pp 111–125

Anselin L, Syabri I, Kho Y (2006) GeoDa: an introduction to spatial data analysis. Geogr Anal 38:5–22

Assunção R, Reis EA (1999) A new proposal to adjust Moran's I for population density. Stat Med 18:2147–2162

Bivand RS (1992) SYSTAT-compatible software for modeling spatial dependence among observations. Comput Geosci 18(8):951–963. https://doi.org/10.1016/0098-3004(92)90013-H

Bivand RS (1998) Software and software design issues in the exploration of local dependence. The Statistician 47:499–508

Bivand RS (2006) Implementing spatial data analysis software tools in R. Geogr Anal 38:23–40

Bivand RS (2008) Implementing representations of space in economic geography. J Reg Sci 48:1–27

Bivand RS (2009) Applying measures of spatial autocorrelation: computation and simulation. Geogr Anal 41(375–384):10

Bivand RS, Gebhardt A (2000) Implementing functions for spatial statistical analysis using the R language. J Geogr Syst 2:307–317

Bivand RS, Piras G (2015) Comparing implementations of estimation methods for spatial econometrics. J Stat Softw 63(1):1–36. https://doi.org/10.18637/jss.v063.i18

Bivand RS, Portnov BA (2004) Exploring spatial data analysis techniques using R: the case of observations with no neighbours. In: Anselin L, Florax RJGM, Rey SJ (eds) Advances in spatial econometrics: methodology, tools, applications. Springer, Berlin, pp 121–142

Bivand RS, Müller W, Reder M (2009) Power calculations for global and local Moran's *I*. Comput Stat Data Anal 53:2859–2872

Bivand RS, Sha Z, Osland L, Thorsen IS (2017) A comparison of estimation methods for multilevel models of spatially structured data. Spat Stat. https://doi.org/10.1016/j.spasta.2017.01.002

Bjornstad ON (2018) ncf: spatial covariance functions. https://CRAN.R-project.org/package=ncf, R package version 1.2-5

Caldas de Castro M, Singer BH (2006) Controlling the false discovery rate: a new application to account for multiple and dependent tests in local statistics of spatial association. Geogr Anal 38(2):180–208. https://doi.org/10.1111/j.0016-7363.2006.00682.x

Cliff AD, Ord JK (1969) The problem of spatial autocorrelation. In: Scott AJ (ed) London Papers in Regional Science 1, Studies in Regional Science. Pion, London, pp 25–55

Cliff AD, Ord JK (1971) Evaluating the percentage points of a spatial autocorrelation coefficient. Geogr Anal 3(1):51–62. https://doi.org/10.1111/j.1538-4632.1971.tb00347.x

Cliff AD, Ord JK (1973) Spatial autocorrelation. Pion, London

Cliff AD, Ord JK (1981) Spatial processes. Pion, London

Cressie NAC (1993) Statistics for spatial data. Wiley, New York

Duncan OD, Cuzzort RP, Duncan B (1961) Statistical geography: problems in analyzing areal data. Free Press, Glencoe

Geary RC (1954) The contiguity ratio and statistical mapping. Inc Stat 5:115–145

Getis A, Ord JK (1992) The analysis of spatial association by the use of distance statistics. Geogr Anal 24(2):189–206

Getis A, Ord JK (1993) Erratum: The analysis of spatial association by the use of distance statistics. Geogr Anal 25(3):276

Getis A, Ord JK (1996) Local spatial statistics: an overview. In: Longley P, Batty M (eds) Spatial analysis: modelling in a GIS environment. GeoInformation International, Cambridge, pp 261–277

Gómez-Rubio V, Ferrándiz-Ferragud J, López-Quílez A (2005) Detecting clusters of disease with R. J Geogr Syst 7(2):189–206

Goodchild MF (1986) Spatial autocorrelation. Geobooks, Norwich. https://alexsingleton.files.wordpress.com/2014/09/47-spatial-aurocorrelation.pdf

Hepple LW (1998) Exact testing for spatial autocorrelation among regression residuals. Environ Plan A 30:85–108

Kalogirou S (2017) lctools: local correlation, spatial inequalities, geographically weighted regression and other tools. https://CRAN.R-project.org/package=lctools, R package version 0.2-6

Levine N (2006) Crime mapping and the CrimeStat program. Geogr Anal 38(1):41–56

Levine N (2017) Crimestat: a spatial statistical program for the analysis of crime incidents. In: Shekhar S, Xiong H, Zhou X (eds) Encyclopedia of GIS. Springer, Cham, pp 381–388. https://doi.org/10.1007/978-3-319-17885-1_229

McMillen DP (2003) Spatial autocorrelation or model misspecification? Int Reg Sci Rev 26:208–217

Moran PAP (1950) Notes on continuous stochastic phenomena. Biometrika 37:17–23

Ord JK, Getis A (1995) Local spatial autocorrelation statistics: distributional issues and an application. Geogr Anal 27(3):286–306

Ord JK, Getis A (2001) Testing for local spatial autocorrelation in the presence of global autocorrelation. J Reg Sci 41(3):411–432

Ord JK, Getis A (2012) Local spatial heteroscedasticity (LOSH). Ann Reg Sci 48(2):529–539

Paradis E, Claude J, Strimmer K (2004) APE: analyses of phylogenetics and evolution in R language. Bioinformatics 20:289–290

Rey SJ, Anselin L (2007) Pysal: a python library of spatial analytical methods. Rev Reg Stud 37(1):5–27

Rey SJ, Anselin L, Li X, Pahle R, Laura J, Li W, Koschinsky J (2015) Open geospatial analytics with pysal. ISPRS Int J Geoinf 4(2):815–836. https://doi.org/10.3390/ijgi4020815

Ripley BD (1981) Spatial statistics. Wiley, New York

Schabenberger O, Gotway CA (2005) Statistical methods for spatial data analysis. Chapman & Hall, Boca Raton

Scott LM, Janikas MV (2010) Spatial statistics in ArcGIS. In: Fischer MM, Getis A (eds) Handbook of applied spatial analysis: software tools, methods and applications. Springer, Berlin, pp 27–41. https://doi.org/10.1007/978-3-642-03647-7_2

Sokal RR, Oden NL (1978) Spatial autocorrelation in biology: 1. methodology. Biol J Linn Soc 10(2):199–228. https://doi.org/10.1111/j.1095-8312.1978.tb00013.x

Sokal RR, Oden NL, Thomson BA (1998) Local spatial autocorrelation in a biological model. Geogr Anal 30:331–354

Tiefelsdorf M (2000) Modelling spatial processes: the identification and analysis of spatial relationships in regression residuals by means of Moran's *I*. Springer, Berlin

Tiefelsdorf M (2002) The saddlepoint approximation of Moran's I and local Moran's I_i reference distributions and their numerical evaluation. Geogr Anal 34:187–206

Tiefelsdorf M, Boots BN (1995) The exact distribution of Moran's I. Environ Plan A 27:985–999

Tiefelsdorf M, Boots BN (1997) A note on the extremities of local Moran's I and their impact on global Moran's I. Geogr Anal 29:248–257

Westerholt R, Resch B, Zipf A (2015) A local scale-sensitive indicator of spatial autocorrelation for assessing high- and low-value clusters in multiscale datasets. Int J Geogr Inf Sci 29(5):868–887. https://doi.org/10.1080/13658816.2014.1002499

Westerholt R, Resch B, Mocnik FB, Hoffmeister D (2018) A statistical test on the local effects of spatially structured variance. Int J Geogr Inf Sci 32(3):571–600. https://doi.org/10.1080/13658816.2017.1402914

Wong DWS, Lee J (2005) Statistical analysis of geographic information with ArcView GIS and ArcGIS. Wiley, New York

Xu M, Mei CL, Yan N (2014) A note on the null distribution of the local spatial heteroscedasticity (LOSH) statistic. Ann Reg Sci 52(3):697–710