GeoDa
An Introduction to Spatial Data Analysis

Homepage Download View on GitHub Data Documentation Support 中文

Global Spatial Autocorrelation (1)

Moran Scatter Plot and Spatial Correlogram

Luc Anselin[1]

04/03/2018 (revised and updated)

Introduction

In this Chapter, we will explore the analysis of global spatial autocorrelation measures, focusing on the basics. We will use the Moran scatter plot and the non-parametric spatial correlogram to visualize the magnitude and the range of spatial autocorrelation. We will continue with the Cleveland house sales data set that we used in the analysis of distance-based spatial weights.

Objectives

- Visualize Moran's I by means of the Moran scatter plot
- Carry out inference using the permutation approach
- Make analyses reproducible using the random seed setting
- Nonlinear LOWESS smooth of the Moran scatter plot
- Brush Moran scatter plot to assess regional Moran's I
- Appreciate the difference between dynamic weights and static weights in Moran scatter plot regime regression
- Analyze the range of spatial autocorrelation by means of a spatial correlogram
- Address the sensitivity of the spatial correlogram to the choice of maximum distance and number of bins
- Address the computation of the spatial correlogram for large(r) data sets by relying on random sampling

GeoDa functions covered

- Space > Univariate Moran’s I
    - permutation inference
    - setting the random seed
    - LOWESS smoother of the Moran scatter plot
    - brushing the Moran scatter plot
    - save results (standardized value and spatial lag)
- Space > Spatial Correlogram
    - variable selection
    - selecting the number of bins
    - selecting the maximum distance
    - using a random sample of locations
    - changing the smoothing parameters

Getting started

With GeoDa launched and all previous projects closed, we load the project file for the Cleveland house price data, **clev_sls_154_core.gda**. This should bring up the by now familiar themeless point map for the house sales locations shown in Figure 1.

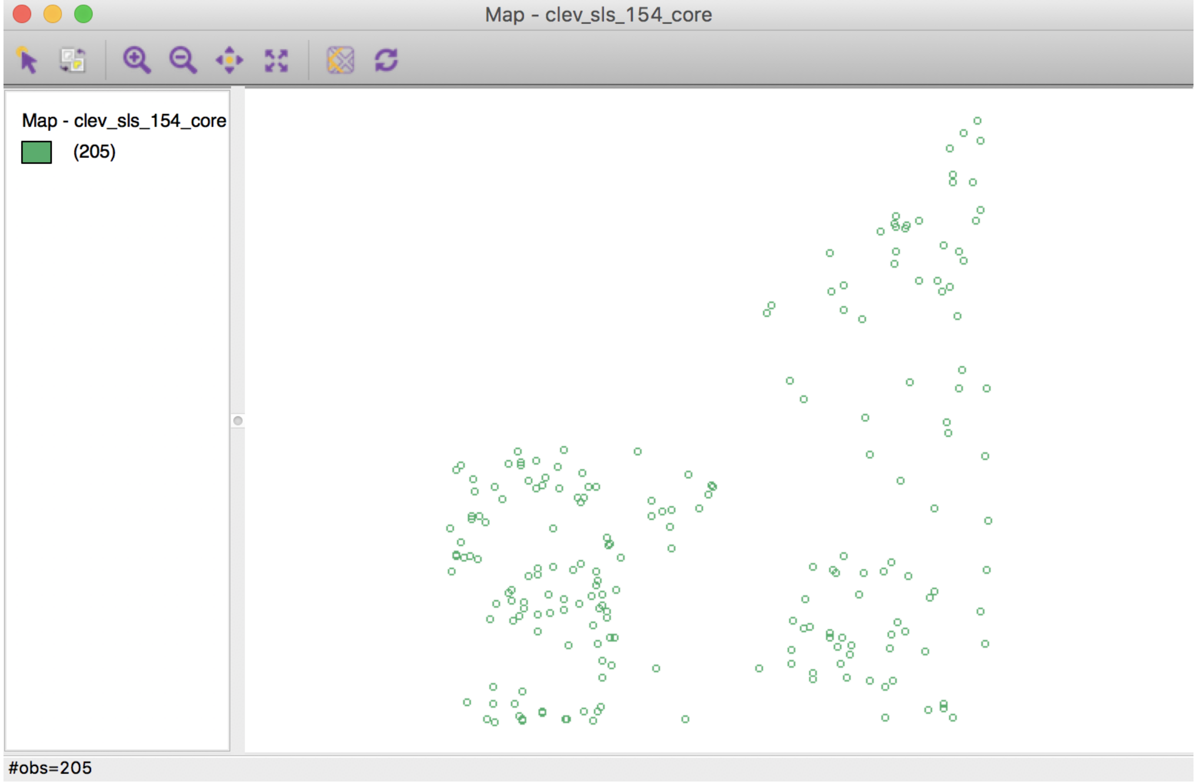
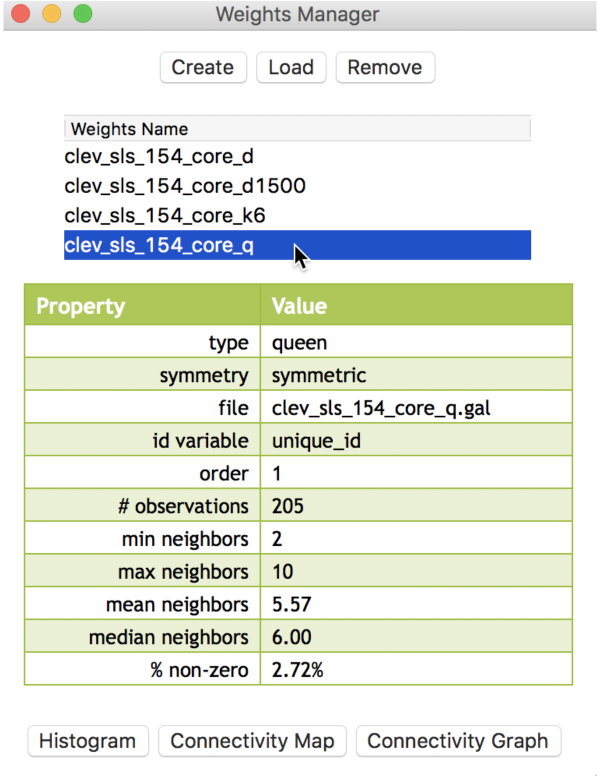


Figure 1: Cleveland sales themeless map

With the project file in place, the weights manager will contain all four weights that were created previously. The queen contiguity weights (from the Thiessen polygons) should be selected, as shown in Figure 2.[2]



Weights Name
clev_sls_154_core_d
clev_sls_154_core_d1500
clev_sls_154_core_k6
clev_sls_154_core_q

Property	Value
type	queen
symmetry	symmetric
file	clev_sls_154_core_q.gal
id variable	unique_id
order	1
# observations	205
min neighbors	2
max neighbors	10
mean neighbors	5.57
median neighbors	6.00
% non-zero	2.72%

Figure 2: Weights manager contents

We will use this weights file in the spatial autocorrelation analysis that follows.

The Moran Scatter PLot

Concept

Moran’s I

Moran’s I statistic is arguably the most commonly used indicator of global spatial autocorrelation. It was initially suggested by Moran (1948), and popularized through the classic work on spatial autocorrelation by Cliff and Ord (1973). In essence, it is a cross-product statistic between a variable and its spatial lag, with the variable expressed in deviations from its mean. For an observation at location i, this is expressed as $z_i = x_i - \bar{x}$, where $\bar{x}$ is the mean of variable x.

Moran’s I statistic is then:

$$I = \frac{\sum_i \sum_j w_{ij} z_i . z_j / S_0}{\sum_i z_i^2 / n}$$

with w_{ij} as the elements of the spatial weights matrix, $S_0 = \sum_i \sum_j w_{ij}$ as the sum of all the weights, and n as the number of observations.

Permutation inference

Inference for Moran’s I is based on a null hypothesis of spatial randomness. The distribution of the statistic under the null can be derived using either an assumption of normality (independent normal random variates), or so-called randomization (i.e., each value is equally likely to occur at any location).[3]

An alternative to an analytical derivation is a computational approach based on *permutation*. This calculates a reference distribution for the statistic under the null hypothesis of spatial randomness by randomly permuting the observed values over the locations. The statistic is computed for each of these randomly reshuffled data sets, which yields a *reference distribution*.

This distribution is then used to calculate a so-called pseudo p-value. This is found as

$$p = \frac{R + 1}{M + 1},$$

where R is the number of times the computed Moran's I from the spatial random data sets (the permuted data sets) is equal to or more extreme than the observed statistic. M equals the number of permutations. The latter is typically taken as $99, 999$, etc., to yield nicely rounded pseudo p-values.

The pseudo p-value is only a *summary* of the results from the reference distribution and should *not* be interpreted as an analytical p-value. Most importantly, it should be kept in mind that the extent of *significance* is determined in part by the number of random pemutations. More precisely, a result that has a p-value of 0.01 with 99 permutations is not necessarily more significant than a result with a p-value of 0.001 with 999 permutations.

Moran scatter plot

The Moran scatter plot, first outlined in Anselin (1996), consists of a plot with the spatially lagged variable on the y-axis and the original variable on the x-axis. The slope of the linear fit to the scatter plot equals Moran's I.

We consider a variable z, given in deviations from the mean. With row-standardized weights, the sum of all the weights (S_0) equals the number of obsevations (n). As a result, the expression for Moran's I simplifies to:

$$I = \frac{\sum_i \sum_j w_{ij} z_i . z_j}{\sum_i z_i^2} = \frac{\sum_i (z_i \times \sum_j w_{ij} z_j)}{\sum_i z_i^2}.$$

Upon closer examination, this turns out to be the slope of a regression of $\sum_j w_{ij} z_j$ on z_i.[4] This is the principle underlying the Moran scatter plot.

An important aspect of the visualization in the Moran scatter plot is the classification of the *nature* of spatial autocorrelation into four categories. Since the plot is centered on the mean (of zero), all points to the right of the mean have $z_i > 0$ and all points to the left have $z_i < 0$. We refer to these values respectively as *high* and *low*, in the limited sense of higher or lower than average. Similarly, we can classify the values for the spatial lag above and below the mean as *high* and *low*.

The scatter plot is then easily decomposed into four quadrants. The upper-right quadrant and the lower-left quadrant correspond with *positive* spatial autocorrelation (similar values at neighboring locations). We refer to them as respectively *high-high* and *low-low* spatial autocorrelation. In contrast, the lower-right and upper-left quadrant correspond to *negative* spatial autocorrelation (dissimilar values at neighboring locations). We refer to them as respectively *high-low* and *low-high* spatial autocorrelation.

The classification of the spatial autocorrelation into four types begins to make the connection between *global* and *local* spatial autocorrelation. However, it is important to keep in mind that the classification as such does not imply significance. This is further explored in our discussion of local indicators of spatial association (LISA).

Creating a Moran scatter plot

To start the Moran scatter plot, we select the left-most button in the spatial analysis group on the toolbar, as shown in Figure 3.

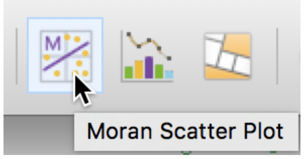


Figure 3: Moran scatter plot toolbar icon

After selecting the toolbar icon, the next prompt is for the type of analysis. For now, we will stick to the **Univariate Moran's I** option, first in the list shown in Figure 4.

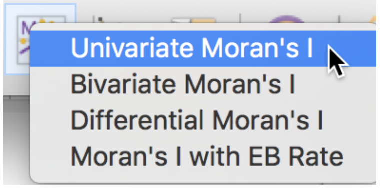


Figure 4: Univariate Moran's I

Alternatively, we can choose **Space > Univariate Moran's I** from the menu, listed in Figure 5.

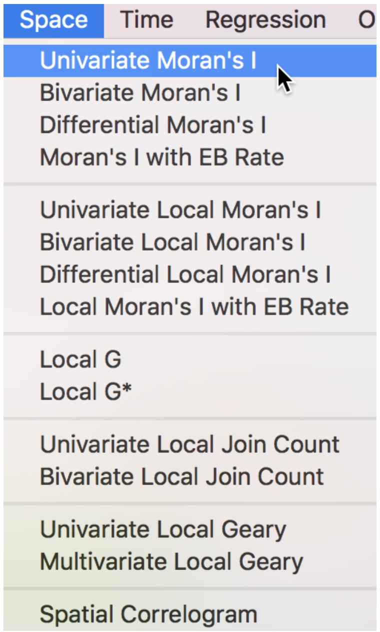
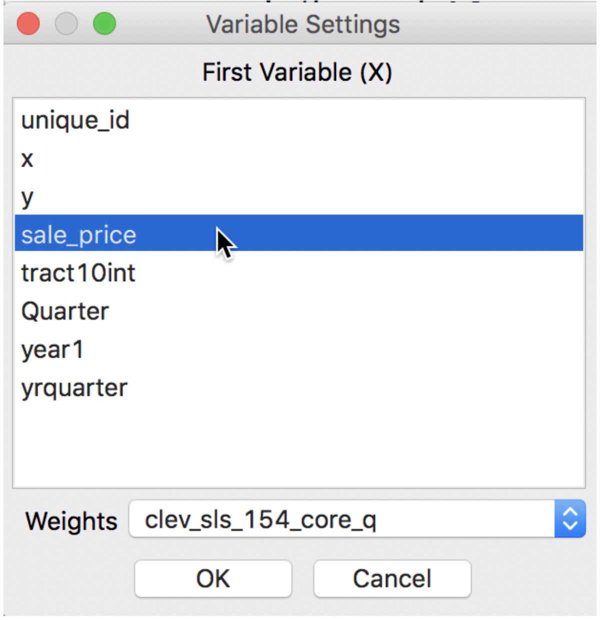


Figure 5: Moran scatter plot from menu

After the univariate Moran’s I analysis is initiated, the next prompt is for the variable name, to be selected from the **Variable Settings** dialog. In our example, we take **sale_price**. In the dialog, the **Weights** drop down list shows the currently active weights. As illustrated in Figure 6, in our example, this should be **clev_sls_154_core_q**.



Weights clev_sls_154_core_q

Figure 6: Moran’s I variable selection

Clicking **OK** brings up the Moran scatter plot, shown in Figure 7.

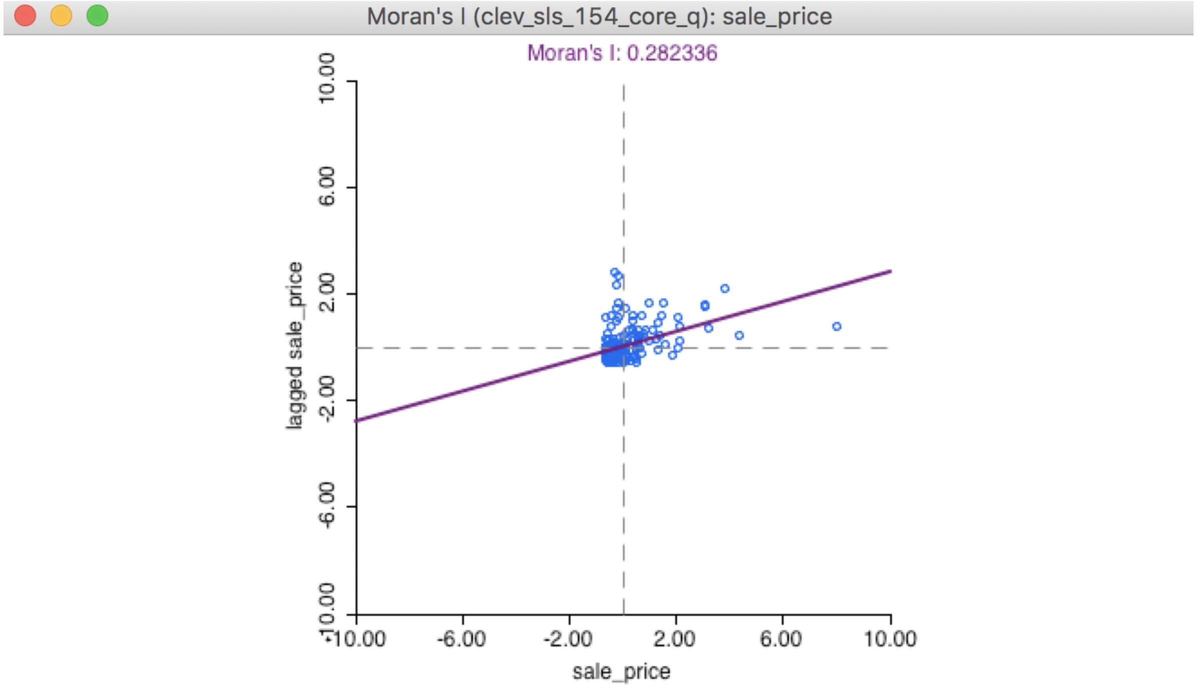


Figure 7: Moran scatter plot

Interpretation

In the Moran scatter plot in Figure 7, the points in the graph are a bit lopsided, because it is rendered as a square (the preferred approach when both axes are measured in the same units, to avoid distorting the data). The house prices are on the horizontal axis, and their spatially lagged counterparts on the vertical axis. Note that the house price values have been standardized, and are given in standard deviational units (the mean is zero and the standard deviation is one). Similarly, the spatial lag is computed for those standardized values.

In its default setting, the plot shows a linear fit through the point cloud. The slope of this line corresponds to Moran's I, and its value (0.282336) is listed at the top of the graph.

We can see that the shape of the point cloud is determined by the presence of several outliers on the high end (e.g., larger than three standard deviational units from the mean). One observation, with a sales price of $527,409 (compared to the median sales prices of $20,000), is as large as 8 standard deviational units above the mean. On the lower end of the spectrum (to the left of the dashed line in the middle that represents the mean), there is much less spread in the house prices, and those points end up bunched together. By eliminating some of the outliers, one may be able to see more detail for the remaining observations, but we will not pursue that here.

Finally, we can select points in each of the quadrants (highlighted by the dashed lines) and identify locations in the map (or, in any other open view) associated with each of the four types of spatial autocorrelation.

Assessing significance

As such, the slope of the linear fit only provides an estimate of Moran's I, but it does not reveal any information about the significance of the test statistic. This is obtained by means of the **Randomization** option. The full list of options is generated by right clicking (or, control clicking) on the plot, in the usual fashion. This yields the list shown in Figure 8. We select **Randomization** and the number of permutations in the side panel. This value corresponds to the M in the expression for the pseudo p-value given above.

In our example, as shown in Figure 8, we select **999 Permutations**, which is typically sufficient for reliable inference. The most *extreme* pseudo p-value possible under this scenario is 0.001, which means that none of the permuted data sets yielded a statistic larger than the one observed in the actual data.

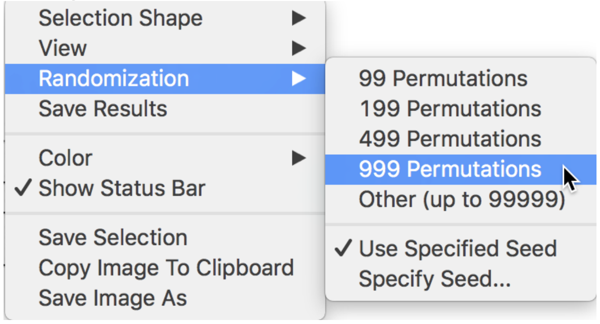
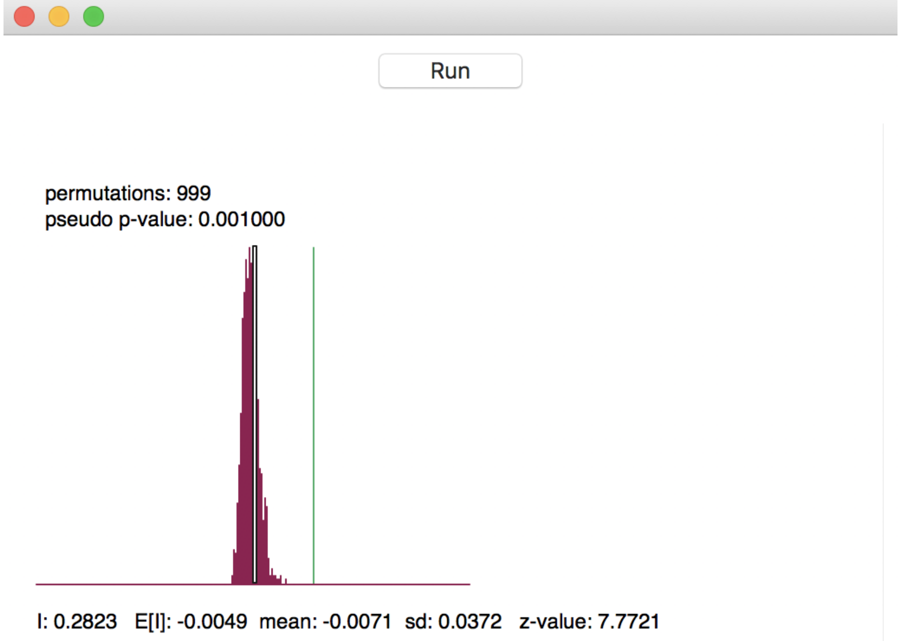


Figure 8: Randomization inference

Reference distribution

The result of the permutation operation is a reference distribution for the statistic, depicted as a histogram, as in Figure 9. The green line shows the value of the statistic for the actual data, placed at 0.2823 in our example, well to the right of the reference distribution. This suggests a strong rejection of the null hypothesis.



permutations: 999
pseudo p-value: 0.001000

I: 0.2823 E[I]: -0.0049 mean: -0.0071 sd: 0.0372 z-value: 7.7721

Figure 9: Reference distribution for Moran’s I

The graph contains several summary statistics. In the top left are shown the number of permutations used to construct the reference distribution (999) and the associated pseudo p-value. As mentioned, the latter is the ratio of the number of values for the statistic that are equal to or greater than the observed value (in our example, just 1 for the observed statistic itself) to the number of generated samples (999) + 1 (for the actual sample). Hence, the result is 1/(999+1) = 0.001.

As pointed out earlier, this is also the smallest (most extreme) p-value that can be obtained. It is important to keep this in mind when comparing results where a different number of permutations is used. For example, a pseudo p-value of 0.001 for 999 permutations is not necessarily *more significant* than a pseudo p-value of 0.01 for 99 permutations. In both cases, not a single statistic computed from the randomly generated samples exceeded the actual statistic. This contrasts with the usual interpretation of an analytically derived p-value.

In the status bar at the bottom of the graph appear several descriptive measures of the Moran’s I statistic. First is the actually observed value, I = 0.2823. Next follows the theoretical expected value, E[I], which equals -1/(n-1). The value of -0.0049 is indeed -1 / 204 (there are 205 observations in the data set). The mean is the average of the reference distribution. In our example, this result is -0.0071, slightly off from the theoretically expected value. The standard deviation of the reference distribution is 0.0372, compared to a theoretical value of 0.00158 under an analytical randomization approach (not computed in GeoDa).

These summary statistics illustrate a common feature in empirical work, namely that the theoretical indication of precision may be overly optimistic (the standard deviation is smaller in the analytical derivation). The z-value that corresponds to the computed Moran’s I, its empirical mean and standard deviation is 7.7721 (the last item to the right on the status bar). Even though a normal approximation would not be accurate, the z-value does suggest strong rejection of the null hypothesis.

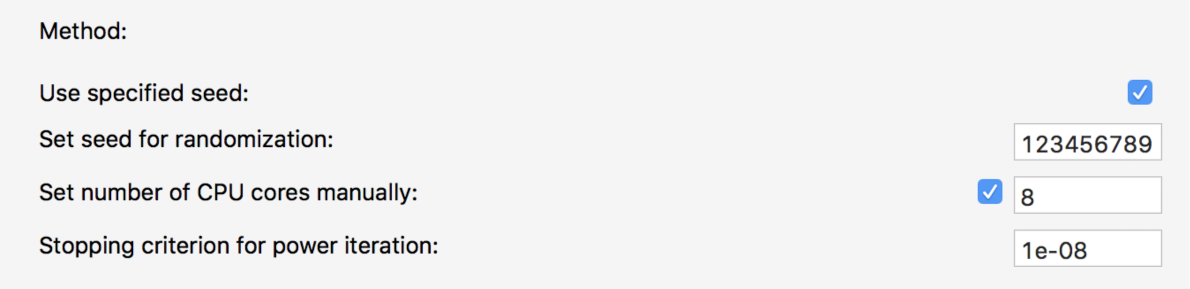
Clicking on the **Run** button in Figure 9 will generate a new empirical distribution. This allows for a sensitivity analysis of the results. Especially when only 99 permutations are used, the summary statistics may vary somewhat, but for 999 (and definitely for 99999, the largest possible value), they should be pretty stable.

Replicability - the random seed

In order to facilitate replication, the default setting in GeoDa is to use a specified seed for the random number generator. This is evidenced by the check mark next to **Use Specified Seed** in the options listed in Figure 8. The default seed value used is shown when selecting the **Specify Seed …** box, where it also can be changed.

The random seed can also be set globally in the GeoDa **Preferences**, under the **System** tab, as shown in Figure 10. At the bottom of the dialog, under the item **Method:**, the box for using the specified seed is checked by default, and the value 123456789 is used as the default seed. This can be changed by typing in a different value.

The same random seed is used in all operations in GeoDa that rely on some type of random permutations (i.e., all flavors of the Moran scatter plot, and all the local spatial autocorrelation statistics). This ensures that the results will be identical for each analysis that uses the same sequence of random numbers.



Method:

Use specified seed: ☑

Set seed for randomization: 123456789

Set number of CPU cores manually: ☑ 8

Stopping criterion for power iteration: 1e-08

Figure 10: Random seed in GeoDa preferences

Using the same random seed ensures replicability on a particular machine. However, due to the way the random number generator is implemented, there may be differences depending on the hardware, specifically when a different number of CPU cores is used.[5] In order to control for any such discrepancies, it is possible to manually set the number of CPU cores. For example, in Figure 10, the box next to **Set number of CPU cores manually** is checked, and the number of cores is given as 8.[6]

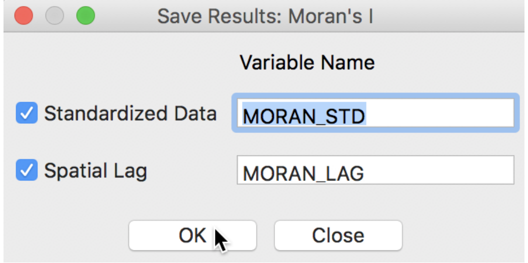
Moran scatter plot options

In the usual way, a right click in the plot brings up the available options, shown earlier in Figure 8.

Many of these are by now familiar. We have already discussed the **Randomization** option. We next briefly consider **Save Results** and **View**.

Saving scatter plot variables

When selecting the **Save Results** option, the dialog offers suggested variable names for the standardized variable (recall that the variable specified was **sale_price**, not its standardized version), with default name **MORAN_STD**, and for its spatial lag, with default name **MORAN_LAG**. This is illustrated in Figure 11. A click on **OK** will add these two variables to the data table.

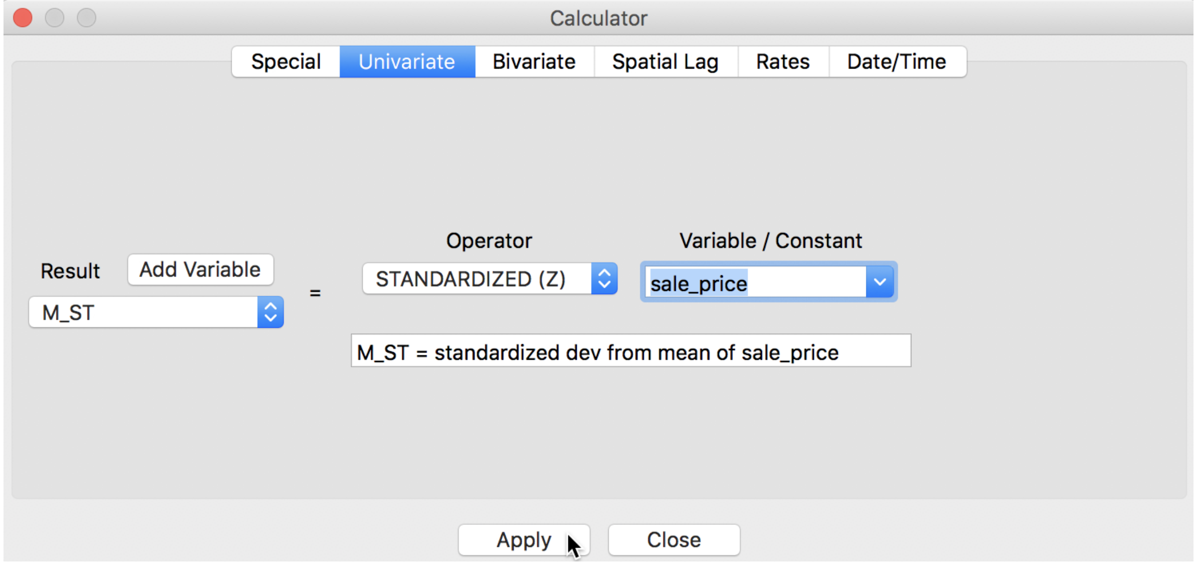


☑ Standardized Data: MORAN_STD

☑ Spatial Lag: MORAN_LAG

Figure 11: Saving variables from Moran scatter plot

We can quickly verify the results by using the **Table > Calculator** option to compute a standardized version of **sale_price** and its spatial lag. For the former, we use the **Univariate** tab and the **STANDARDIZED (Z)** operator (we first add **M_ST** as a new variable), as shown in Figure 12.[7]



Result: M_ST

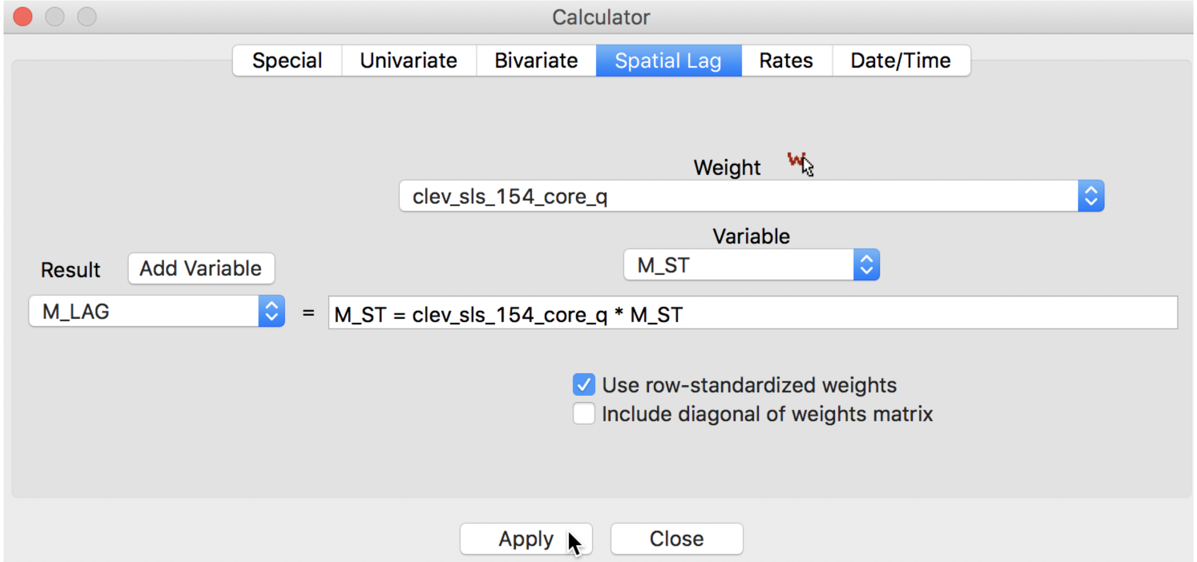
Operator: STANDARDIZED (Z)

Variable / Constant: sale_price

M_ST = standardized dev from mean of sale_price

Figure 12: Standardized variable in calculator

For the spatial lag, we use the **Spatial Lag** tab, with the weights file **clev_sls_154_core_q** specified and the previously standardized variable (again, after adding **M_LAG** as the new variable), as in Figure 13.



Weight: clev_sls_154_core_q

Variable: M_ST

Result: M_LAG

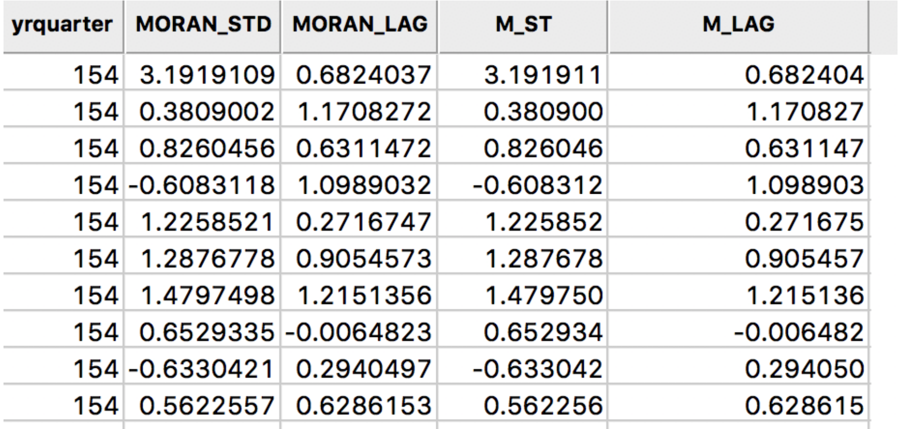
M_ST = clev_sls_154_core_q * M_ST

☑ Use row-standardized weights

☐ Include diagonal of weights matrix

Figure 13: Spatially lagged variable in calculator

The table now has four extra columns, two generated by the Moran scatter plot option, and two calculated explicitly to replicate these results, shown in Figure 14.



yrquarter	MORAN_STD	MORAN_LAG	M_ST	M_LAG
154	3.1919109	0.6824037	3.191911	0.682404
154	0.3809002	1.1708272	0.380900	1.170827
154	0.8260456	0.6311472	0.826046	0.631147
154	-0.6083118	1.0989032	-0.608312	1.098903
154	1.2258521	0.2716747	1.225852	0.271675
154	1.2876778	0.9054573	1.287678	0.905457
154	1.4797498	1.2151356	1.479750	1.215136
154	0.6529335	-0.0064823	0.652934	-0.006482
154	-0.6330421	0.2940497	-0.633042	0.294050
154	0.5622557	0.6286153	0.562256	0.628615

Figure 14: Variables added to table

Digression - creating a Moran scatter plot as a standard scatter plot

The preferred way to construct a Moran scatter plot is to use the designated functionality in the **Space** menu. However, it is perfectly possible to create a plot *the hard way*, as a special case of a standard bivariate scatter plot. To accomplish this, we select a variable (standardized) for the x-axis, and its spatial lag for the y-axis.

For example, in Figure [15](#), we select the just created **MORAN_STD** in the variable selection dialog for the x-axis, and its spatial lag **MORAN_LAG** as the y-axis (or, alternatively, **M_ST** and **M_LAG**).

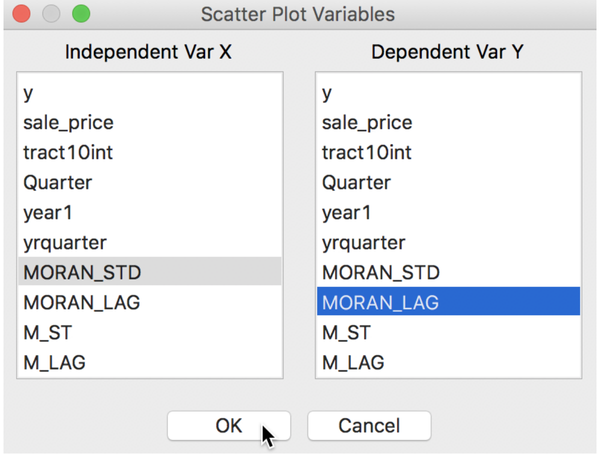
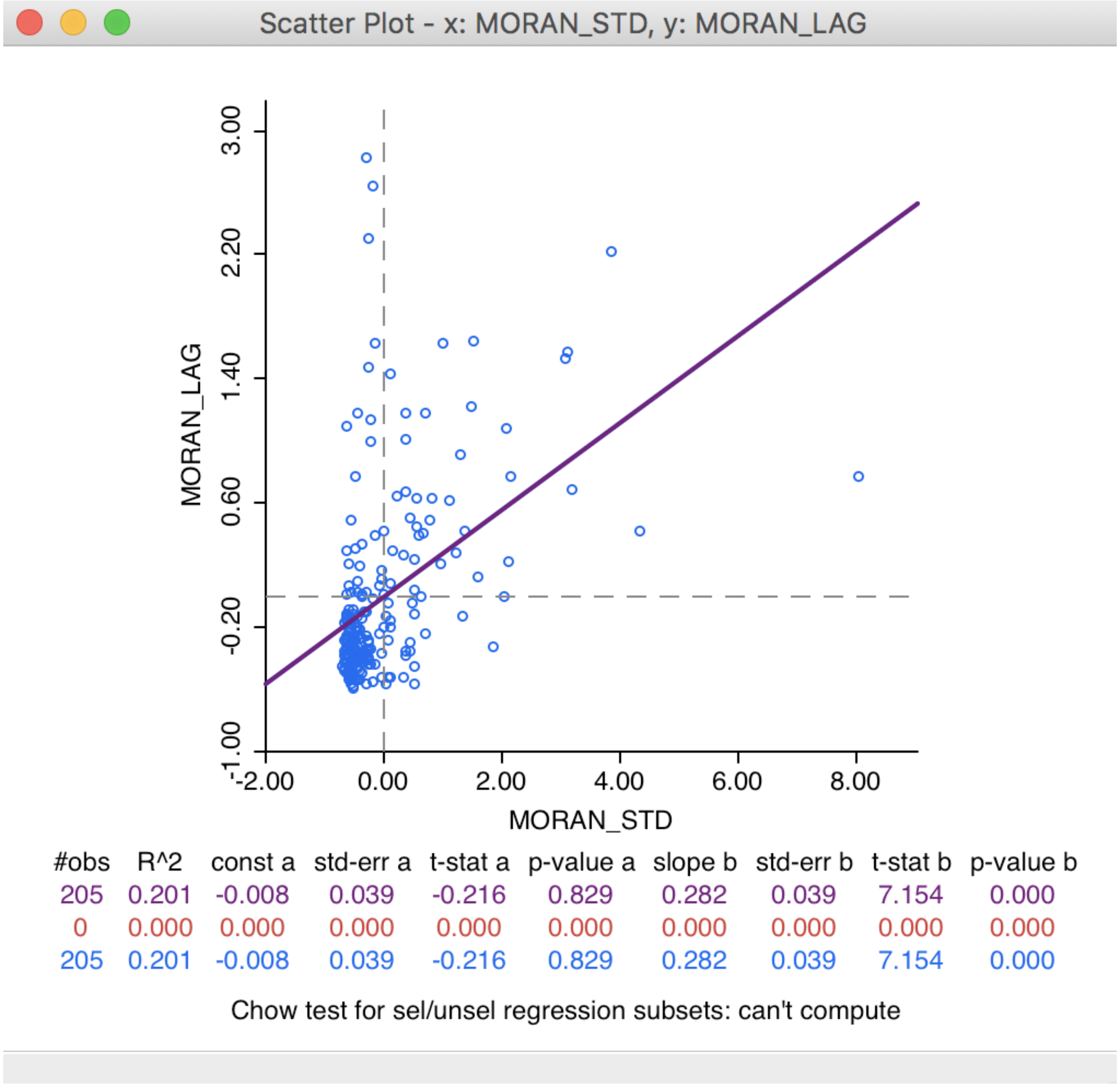


Figure 15: Scatter plot with spatial variables

This yields the standard scatter plot, shown in Figure [16](#). In contrast to the specific Moran scatter plot, here the statistics are displayed below the plot (the default setting for a standard scatter plot). This reveals the slope of the linear fit to be **0.282**, the same as Moran's I given by the Moran scatter plot.



#obs	R^2	const a	std-err a	t-stat a	p-value a	slope b	std-err b	t-stat b	p-value b
205	0.201	-0.008	0.039	-0.216	0.829	0.282	0.039	7.154	0.000
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
205	0.201	-0.008	0.039	-0.216	0.829	0.282	0.039	7.154	0.000

Chow test for sel/unsel regression subsets: can't compute

Figure 16: Moran as a standard scatter plot

We will revisit the plot in Figure [16](#) later when we consider the difference between **regime regression** in the Moran scatter plot compared to the standard scatter plot.

Apart from a purely pedagogic objective, there are some instances in practice when constructing a Moran scatter plot as a standard plot is the only way to obtain an estimate for Moran's I (as the slope of the linear fit). One example we already encountered is when we use row-standardized weights derived from an inverse distance measure. Such weights can be employed to create a spatially lagged variable in the usual way, which can then serve as the y-axis in the scatter plot.

The current implementation of the Moran scatter plot functionality in GeoDa ignores the specific values for the weights and only takes into account the presence of connectivity (i.e., non-zero weights). In the computation of the spatial lag, all the neighbors receive the same weight. However, for row-standardized inverse distance weights, the values will be different. Since the weights are row-standardized, the slope of a standard scatter plot of the spatial lag against the variable will still be Moran's I.

Note that this approach does not work for inverse distance weights that are not row-standardized, or for weights that include the diagonal element. In the former case, the slope estimate is off by a factor S_0/n. So, in order to recuperate the value for Moran's I, the slope estimate would have to be multiplied by n/S_0. However, the interpretation of Moran's I for spatial weights that are not row-standardized may be difficult.

In any case, the slope of the linear fit in a standard scatter plot is only an estimate for Moran's I, but does not provide any inference. The usual standard errors and t-statistics provided in the scatter plot are *not appropriate* in the spatial case. An explicity permutation procedure would need to be carried out separately.

LOWESS smoother

In addition to the traditional linear smoother, GeoDa also supports a Lowess smoother for the Moran scatter plot (similar to the functionality in the standard scatter plot). This is accomplished by selecting the **View > LOWESS Smoother** item in the options menu, illustrated in Figure 17.

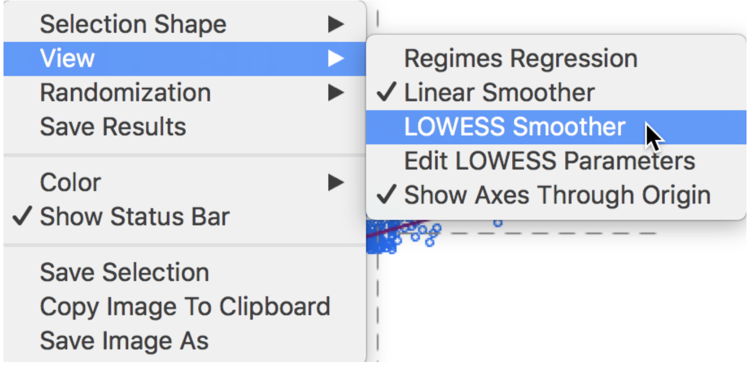


Figure 17: Moran scatter plot LOWESS option

With the **View > Linear Smoother** option turned off, only the nonlinear fit from the local regression is shown on the Moran scatter plot, as in Figure 18. The Lowess smoother can be explored to identify potential structural breaks in the pattern of spatial autocorrelation. For example, in some parts of the data set, the curve may be very steep and positive, suggesting strong positive spatial autocorrelation, whereas in other parts, it could be flat, indicating no autocorrelation.

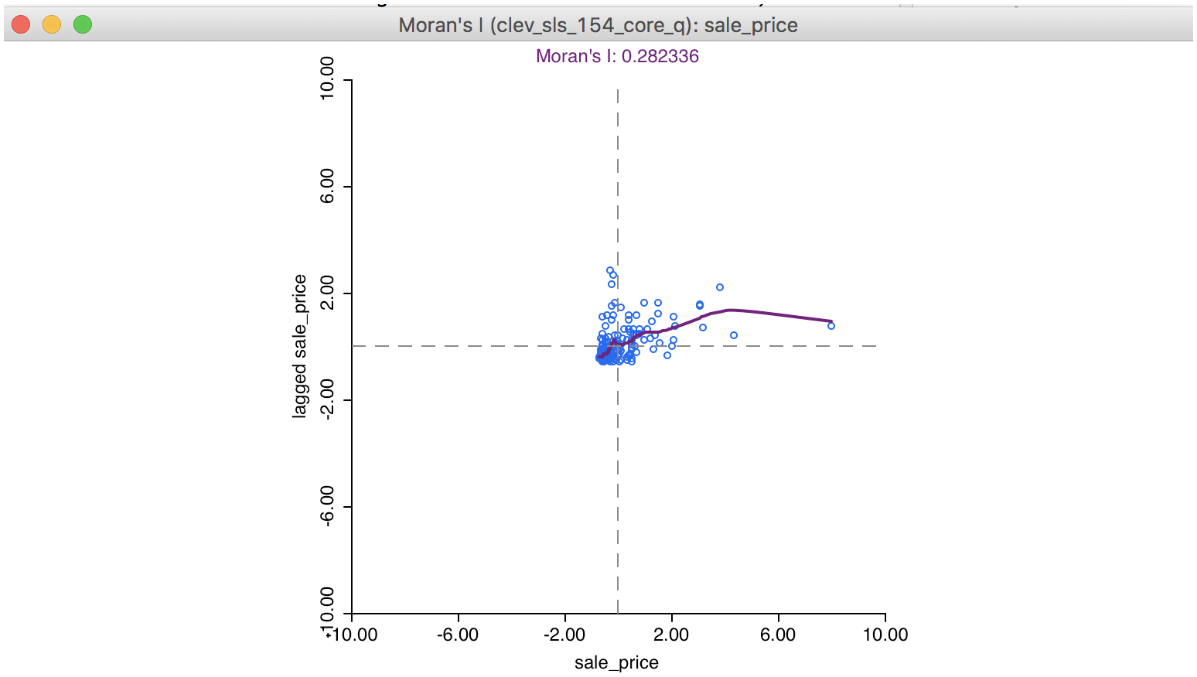


Figure 18: LOWESS smooth of Moran scatter plot

The local smoother is fit through the points in the Moran scatter plot by utilizing only those observations within a given bandwidth. As for the standard scatter plot, the default is to use 0.20 of the range of values. The bandwidth (as well as other, more technical parameters) can be specified by selecting the **View > Edit LOWESS Parameters** item in the options menu, as we have seen in the discussion of the standard scatter plot. All options work exactly the same as in the standard case.

Brushing the Moran scatter plot

A final option of the **View** setting that we consider initiates **Regimes Regression**. As in the standard scatter plot, with this option selected, statistics for the slope and intercept are recomputed as observations are selected and unselected. We proceed with the **LOWESS Smoother** turned off and the **Regimes Regression** selected, as in Figure 19.

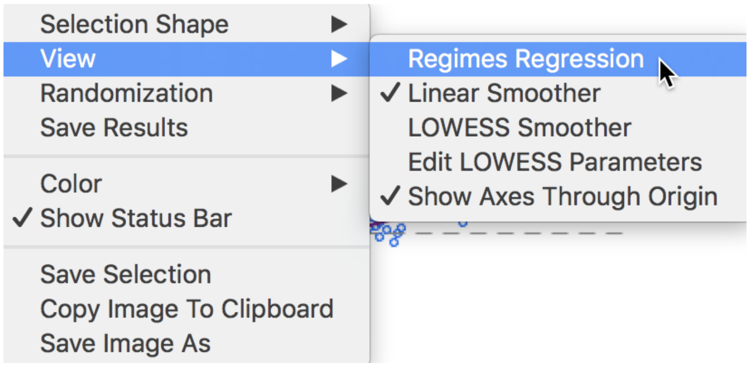


Figure 19: Moran scatter plot regimes

This brings up the initial setting for the brushing operation. As illustrated in Figure 20, there are now three Moran scatter plots. The one to the left (in red) is for the **selected** observations, currently empty, since no selection has been made. The one to the right is for the complement, referred to as **unselected** (in black), currently the same as the scatter plot in the center, which shows the slope for all observations.

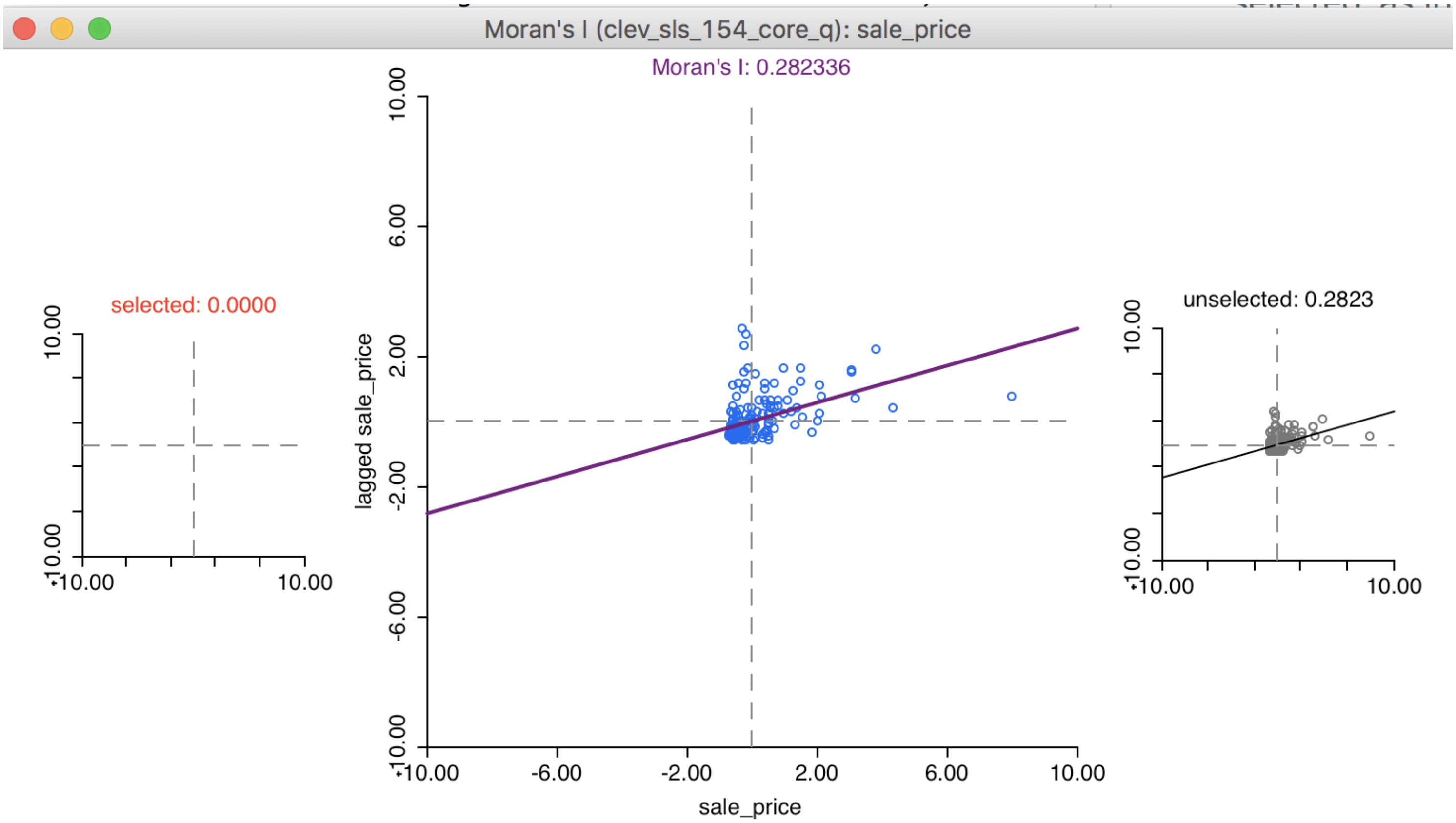


Figure 20: Regimes starting setup

We start the brushing operation by selecting a large rectangle with 51 observations (the number of selected observations is shown in the status bar) in the lower right of the point map, shown in the left panel of Figure 21. As soon as the selection is made, the red and black graphs are updated in the right panel of the Figure.

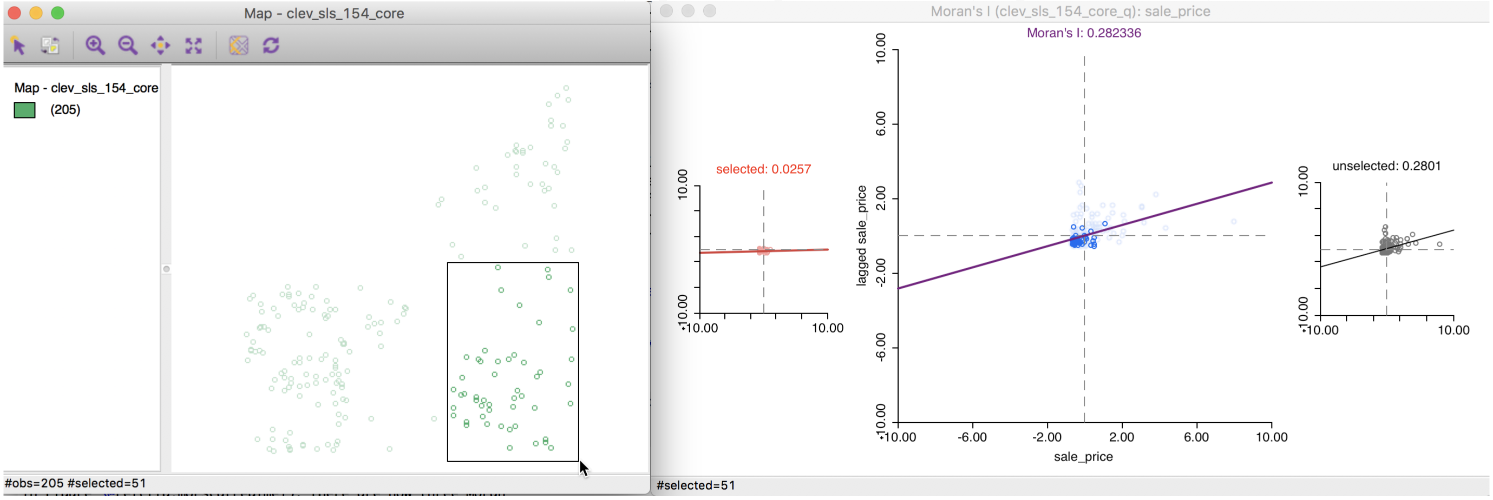


Figure 21: Brushing the Moran scatter plot - 1

The distinguishing characteristics of this brushing operation (or, any selection) is that the spatial weights are dynamically updated for both the selection and the unselected observations. In other words, the edge effects are corrected for and no links are included to locations that are not part of the selection. The same holds for the unselected observations.

As a result, the slope of the Moran scatter plots on the left and the right are as if the data set consisted only of the selected/unselected points. This is a visual implementation of a *regionalized* Moran's I, where the indicator of spatial autocorrelation is calculated for a subset of the observations (Munasinghe and Morris 1996).

In our initial selection, the selected observations show no spatial autocorrelation at all (a value of 0.0257), whereas the complement (unselected) obtain a Moran's I of 0.2801, basically the same as the overall statistic of 0.282. This would suggest the presence of spatial heterogeneity in the strength of the spatial autocorrelation, in the sense that the subset selected shows a very different degree of dependence than its complement (or the data set as a whole).

Note that this is purely exploratory, and there is no actual *test* for the difference between the two statistics (this contrasts with the availability of the Chow test in the standard scatter plot).

As we move the *brush* to the left in the point map, the selected and unselected scatter plots are updated in real time, showing the corresponding regional Moran's I statistics in the panels of Figure 22.

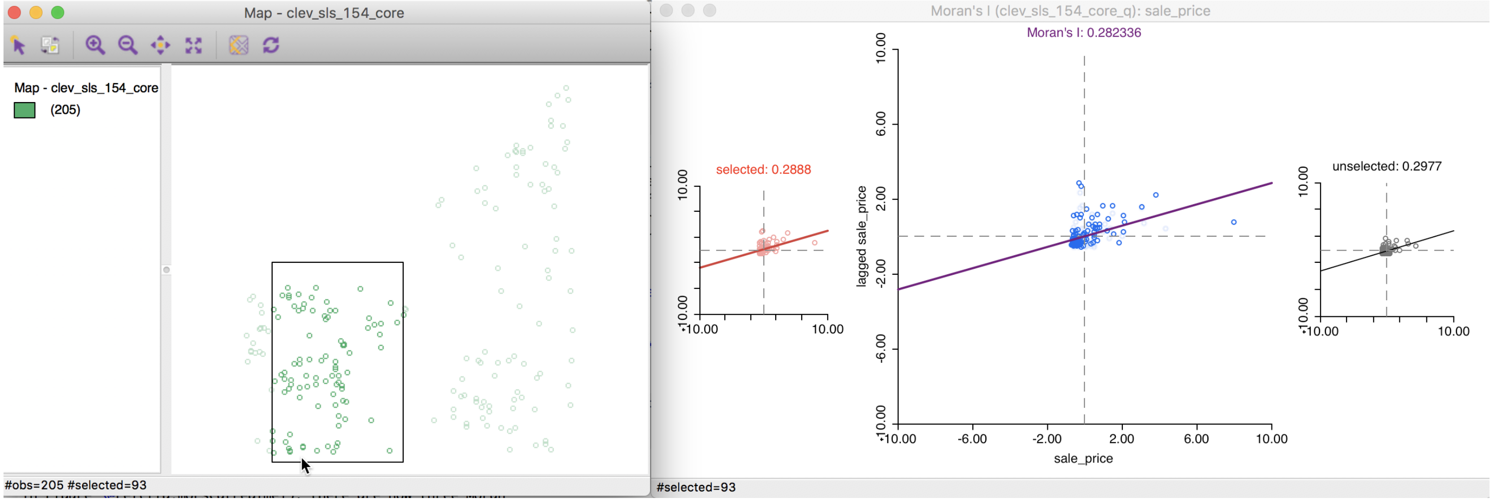


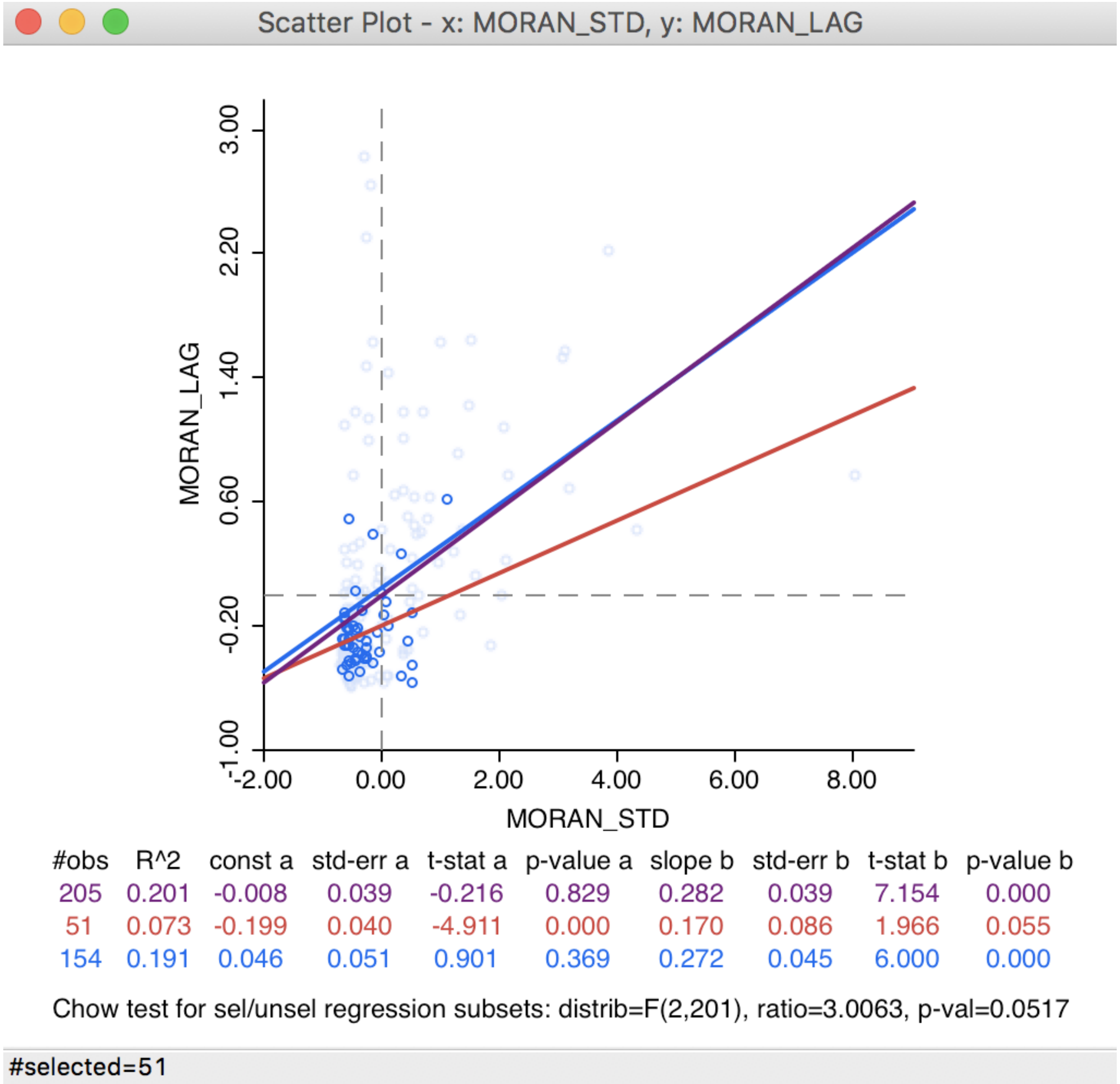
Figure 22: Brushing the Moran scatter plot - 2

In our example, the second selection shows much less evidence of spatial heterogeneity, with a Moran's I of 0.2888 for the selected observations and 0.2977 for the complement, as shown in the Figure.

Of course, because of the linking of all graphs in GeoDa, the selection is also reflected in the *standard* scatter plot we constructed for the standardized price variable and its spatial lag.

The difference between this and the Moran scatter plot brushing pertains to the dynamically updated spatial weights. In the standard scatter plot, the spatially lagged variable is included in the selection as is, which also includes neighbors that are *not* part of the selection. This contrasts to the selection in the Moran scatter plot, which dynamically updates the spatial weights for both the selected and unselected.

The difference is highlighted in Figure 23.



#obs	R^2	const a	std-err a	t-stat a	p-value a	slope b	std-err b	t-stat b	p-value b
205	0.201	-0.008	0.039	-0.216	0.829	0.282	0.039	7.154	0.000
51	0.073	-0.199	0.040	-4.911	0.000	0.170	0.086	1.966	0.055
154	0.191	0.046	0.051	0.901	0.369	0.272	0.045	6.000	0.000

Chow test for sel/unsel regression subsets: distrib=F(2,201), ratio=3.0063, p-val=0.0517

Figure 23: Brushing the standard scatter plot

The scatter plot with the same 51 selected observations as in the first example above yields a slope of 0.170 for the selected (in red, contrasted with 0.026 above) and 0.272 for the unselected (in blue, compared with 0.280 above). The usual Chow test does not provide strong evidence for spatial heterogeneity.

This result needs to be interpreted with caution, since the weights structure has not been adjusted to reflect the actual *breaks* within the data set. On the other hand, the standard scatter plot *ignores* boundary effects (in the sense that it includes neighbors *outside* the regimes), so in some contexts, this may be the interpretation sought. GeoDa provides both options.

Spatial Correlogram

Concept

A non-parametric spatial correlogram is an alternative measure of global spatial autocorrelation that does not rely on the specification of a spatial weights matrix. Instead, a local regression is fit to the covariances or correlations computed for all pairs of observations as a function of the distance between them (for example, as outlined in Bjornstad and Falck 2001).[8]

With standardized variables z, this boils down to a local regression:

$$z_i . z_j = f(d_{ij}) + u,$$

where d_{ij} is the distance between a pair of locations $i - j$, u is an error term, and f is the non-parametric function to be determined from the data. Typically, the latter is a LOWESS or kernel regression.

GeoDa implements a spatial correlogram, i.e., the computation is based on standardized variables such that the cross-products correspond to correlations. The nonlinear curve is a LOWESS regression fit to the average correlation for all pairs of observations in a distance bin. This uses a similar logic as underlying the empirical variogram of geostatistics, but with a non-linear smoother applied to the bin estimates.

Creating a spatial correlogram

The spatial correlogram functionality is invoked by clicking on the corresponding icon in the toolbar (the middle icon in the spatial analysis group, as in Figure 24), or by selecting **Space > Spatial Correlogram** from the menu (the item at the very bottom of the list of options).

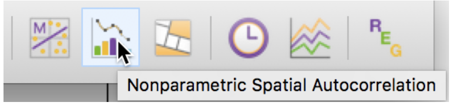
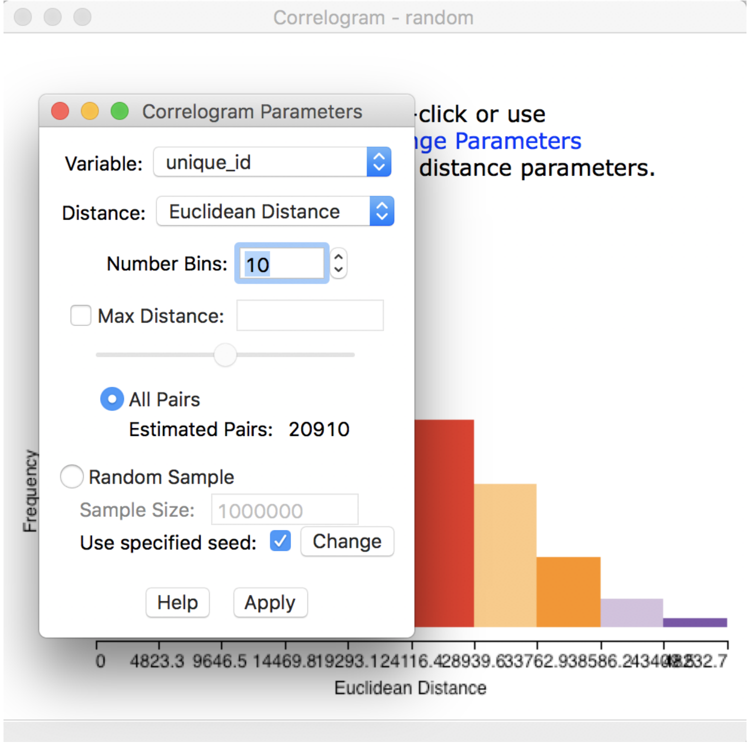


Figure 24: Non-parametric correlogram icon

This brings up a dialog with the default parameter settings and a graph in the background, as shown in Figure 25. This graph is not informative at this point, since it shows a correlogram for the first variable, which is **unique_id**. First, we need to choose the proper settings in the dialog.



Variable: unique_id

Distance: Euclidean Distance

Number Bins: 10

☐ Max Distance:

● All Pairs

Estimated Pairs: 20910

○ Random Sample

Sample Size: 1000000

Use specified seed: ☑

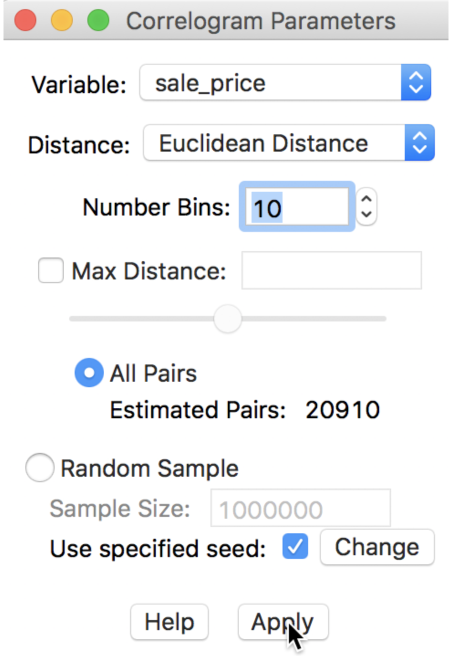
Figure 25: Initial correlogram dialog

The items at the top of the dialog are the **Variable** (available from a drop down list), the **Distance** metric (default is Euclidean, but Arc distance is also available), and the **Number of Bins**.

The non-parametric correlogram is computed by means of a local regression on the pairwise correlations that fall within a distance bin. The number of bins determines the distance range of each bin. This range is the maximum distance divided by the number of bins. The more bins are chosen, the more fine-grained the correlogram will be. However, this also potentially can lead to too few pairs in some bins (the rule of thumb is to have at least 30).

The number of elements in each bin is a result of the interplay between the number of bins and the maximum distance. As the default, all pairs are used (in our example, with 205 observations, this yields: $[205^2 - 205]/2 = 20910$ pairs), and thus all the pairwise distances are taken into account. However, in many instances in practice, this may not be a good choice. For example, when there are many observations, the number of pairs quickly becomes unwieldy and GeoDa will run out of memory. Also, the correlations computed for pairs that are far apart are not that meaningful (they should be zero due to Tobler's law). The bottom half of the dialog provides options for fine-tuning these choices.

First, we consider the default, which uses all pairs and does not impose any constraints in terms of maximum distance. The number of bins is set to **10**. As before, the **Variable** we analyze is **sale_price**. These selections are shown in the dialog in Figure 26.



Variable: sale_price

Distance: Euclidean Distance

Number Bins: 10

☐ Max Distance:

● All Pairs

Estimated Pairs: 20910

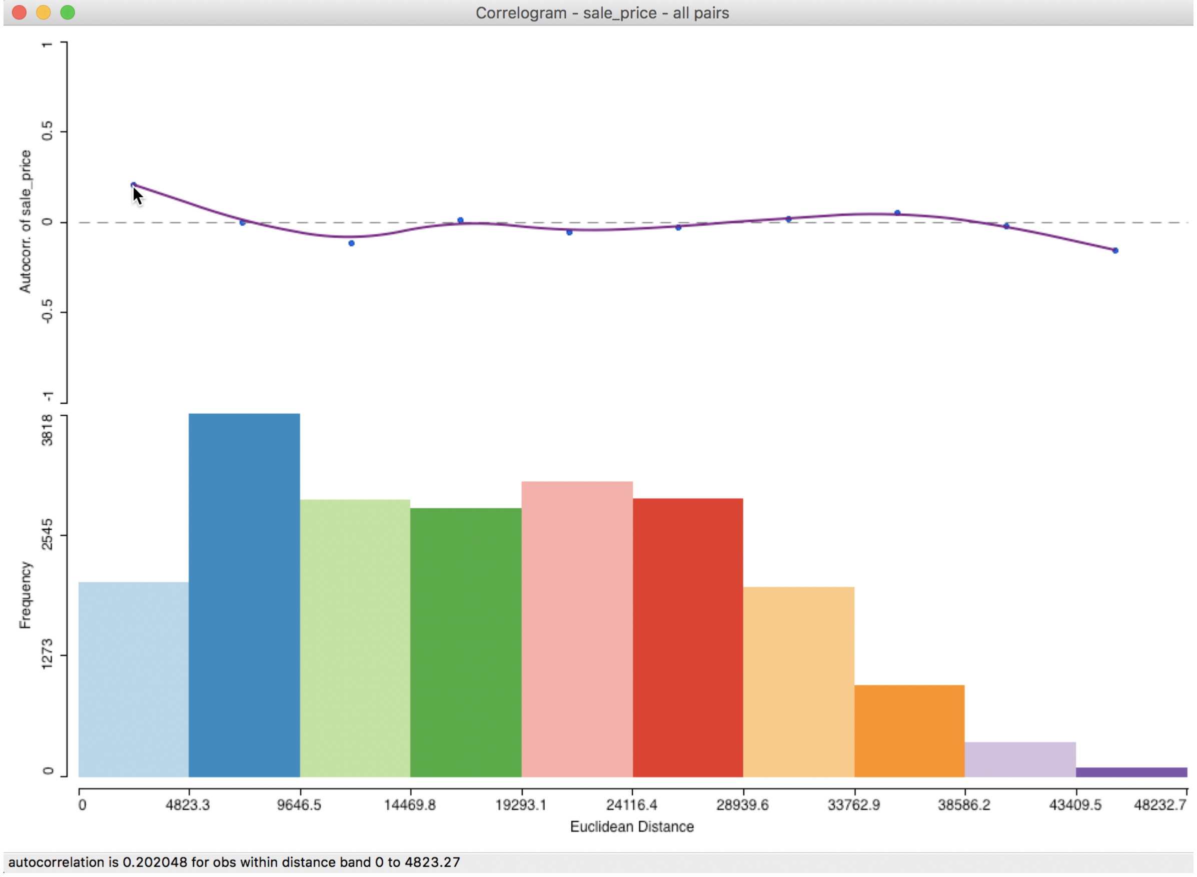
○ Random Sample

Sample Size: 1000000

Use specified seed: ☑

Figure 26: Variable setting for correlogram

Clicking the **Apply** button yields the spatial correlogram shown in Figure 27. Slight adjustments to the size of the window may be necessary in order to see all the detail on the horizontal axis, especially since the distances are expressed in feet (so they are large numbers).



autocorrelation is 0.202048 for obs within distance band 0 to 4823.27

Figure 27: Default spatial correlogram

Interpretation

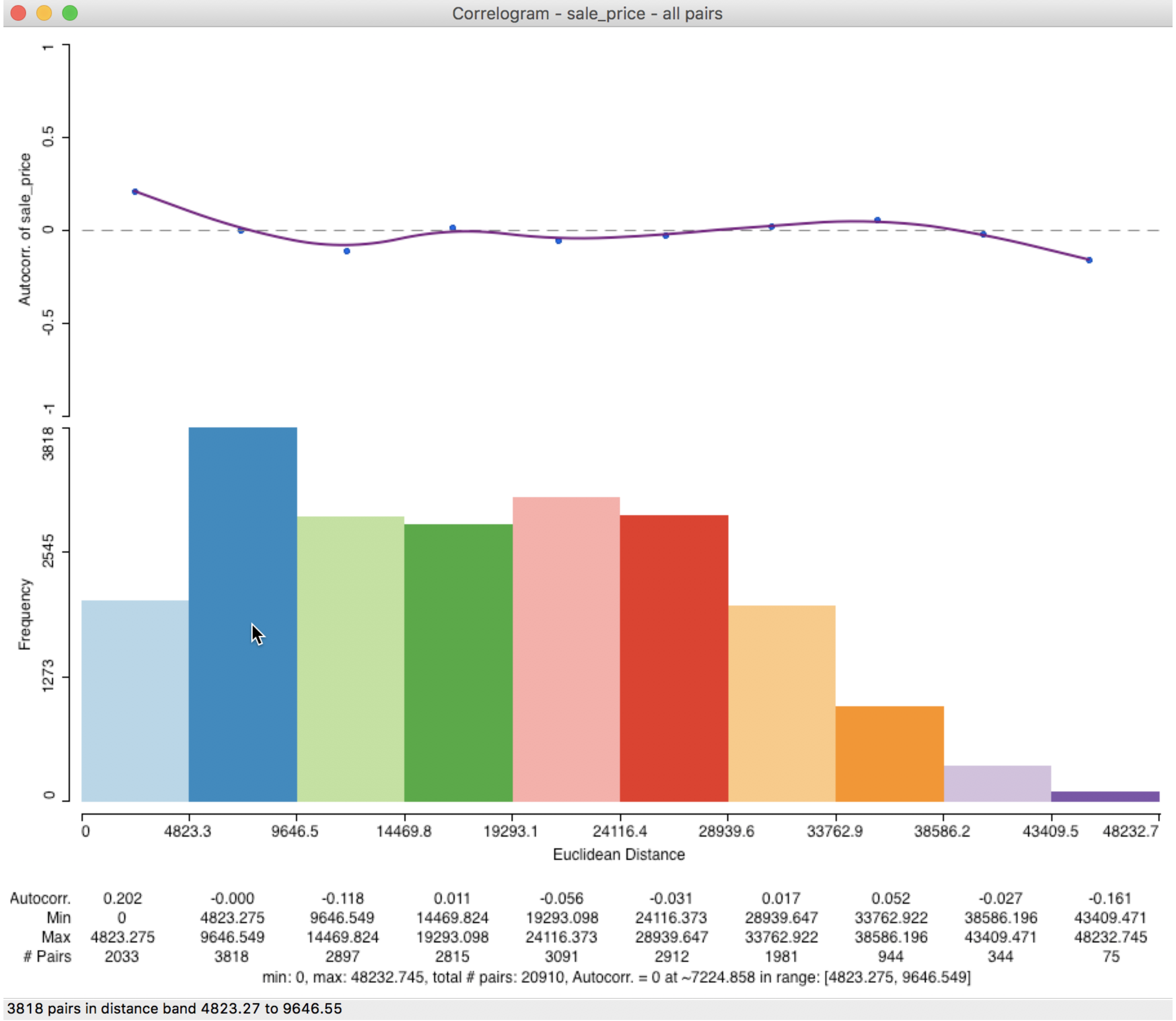
At the top of the graph in Figure 27 is the actual correlogram, depicting how the spatial autocorrelation changes with distance. Hovering the pointer over each blue dot, gives the spatial autocorrelation associated with that distance band in the status bar. In our example, the first dot corresponds to 0.202048 for distances between 0 and 4823 feet (or about 0.9 miles).

The intersection between the correlogram and the dashed zero axis (which determines the range of spatial autocorrelation) happens in the midpoint of the second range (4823 to 9647 feet), i.e., 7235, or roughly around 1.4 miles. Beyond that range, the autocorrelation is first negative and then fluctuates around the zero line.

At the bottom of the graph is a histogram that shows the number of pairs of observations in each bin. Hovering the pointer over a given bin shows how many pairs are contained in the bin in the status bar. In our example, each bin clearly has more than sufficient observation pairs. Even the last bin, which seems small (a function of the vertical scale), uses 75 pairs for the computation.

Further detail is provided when selecting the **Display Statistics** option (right click on the graph). As shown in Figure 28, for each bin, the computed autocorrelation is provided, as well as the distance range for the bin (lower and upper bound), and the number of pairs used to compute the statistic.

In addition, there is a summary with the minimum and maximum distance, the total number of pairs, and an estimate for the range (i.e., the distance at which the estimated autocorrelation first becomes zero). In our example, the latter is estimated to be about 7,225 feet (about 1.4 miles). Right after the listing of the value for the range, the lower and upper bound of the bin where the correlogram crosses the axis is given (this provides an alternative, but less precise approximation of the range).



```
Autocorr.   0.202     -0.000    -0.118     0.011     -0.056     -0.031     0.017     0.052     -0.027    -0.161
      Min   0         4823.275  9646.549   14469.824 19293.098  24116.373  28939.647 33762.922 38586.196 43409.471
      Max   4823.275  9646.549  14469.824  19293.098 24116.373  28939.647  33762.922 38586.196 43409.471 48232.745
  # Pairs   2033      3818      2897       2815      3091       2912       1981      944       344       75
          min: 0, max: 48232.745, total # pairs: 20910, Autocorr. = 0 at ~7224.858 in range: [4823.275, 9646.549]
3818 pairs in distance band 4823.27 to 9646.55
```

Figure 28: Spatial correlogram with statistics

Spatial correlogram options

In GeoDa, there are two sets of spatial correlogram options. One consists of the **parameters** of the model, as specified in the dialog shown in Figure 26. The other set of options are invoked in the usual way, by right clicking on the graph. We already referred to the **Display Statistics** option above. The full list of options is shown in Figure 29.

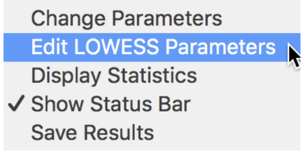


Figure 29: Spatial correlogram options

One interesting option is to **Save Results**. This provides a record of the descriptive statistics of the correlogram in a text file (with file extension csv). The file contains the information listed at the bottom of the graph when **Display Statistics** has been selected. This includes the estimates of the spatial autocorrelation, bin ranges, number of observations in the bins and the summary items.

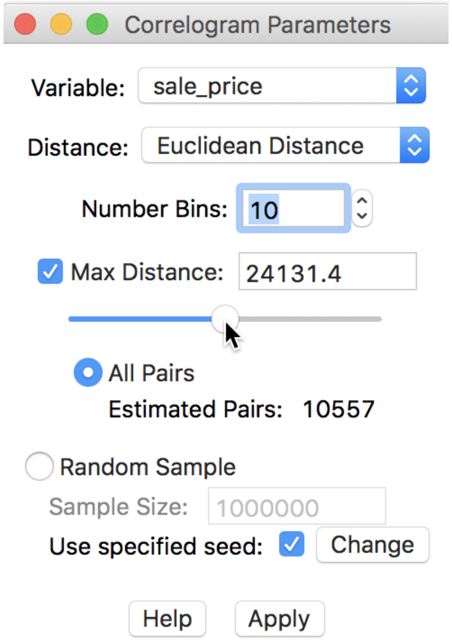
LOWESS parameters

The selection highlighted in Figure 29 pertains to the **LOWESS Parameters**, i.e., the bandwidth and other technical options that can be specified for any LOWESS smoother. The default is to use a bandwidth of 0.20, which works well in most situations. As usual, a larger bandwidth will yield a (slightly) smoother curve, and a smaller bandwidth the opposite. This works in exactly the same way as for the standard LOWESS nonlinear smoother.

Distance settings

In most situations, the default use of all the distance pairs is too overwhelming. Also, there is a good reason to limit the maximum distance, since correlations at large(r) distances are both sparser (fewer pairs in a bin, which leads to less precise estimates) and supposed to be near zero (Tobler's law).

There are several rules of thumb to set the maximum distance. GeoDa uses half of the maximum distance as a point of departure. By checking the box next to **Max Distance** and clicking on the button in the middle of the slider bar, half the maximum distance (**24131.4**) is listed in the box, shown in Figure 30. This reduces the number of pairs used to compute the correlations almost by half, from 20910 to roughly 10557.[9]



Correlogram Parameters

Variable: sale_price

Distance: Euclidean Distance

Number Bins: 10

☑ Max Distance: 24131.4

● All Pairs

Estimated Pairs: 10557

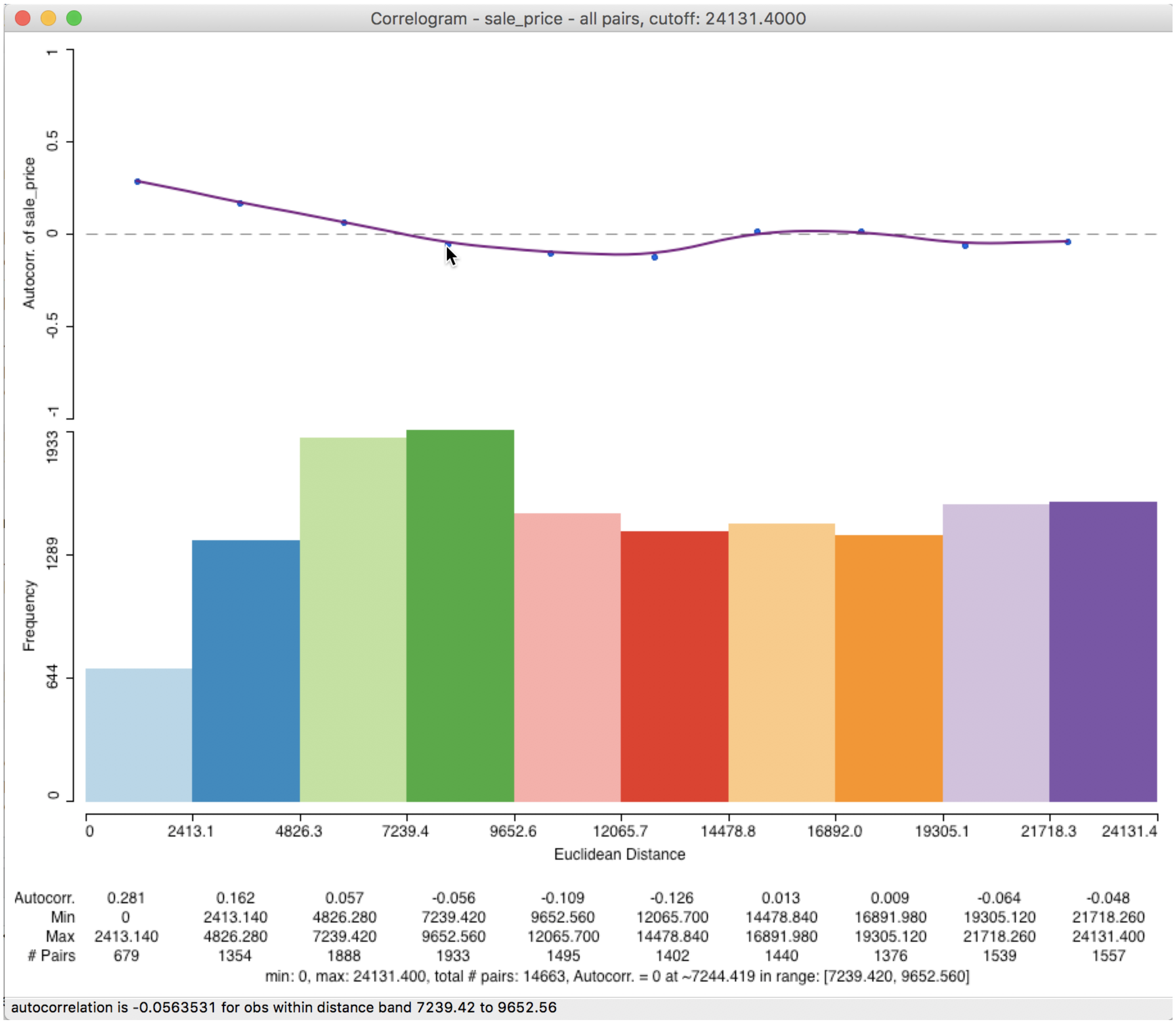
○ Random Sample

Sample Size: 1000000

Use specified seed: ☑

Figure 30: Adjusting maximum distance

The corresponding correlogram in Figure 31 emphasizes the autocorrelation pattern over the shorter distance ranges. It suggests a range for the correlation that is basically the same as the initial result, about 7,244 feet (compared to 7,225). The status bar lists the autocorrelation given by the point over which the pointer hovers (in our example, about -0.056 for the fourth bin).



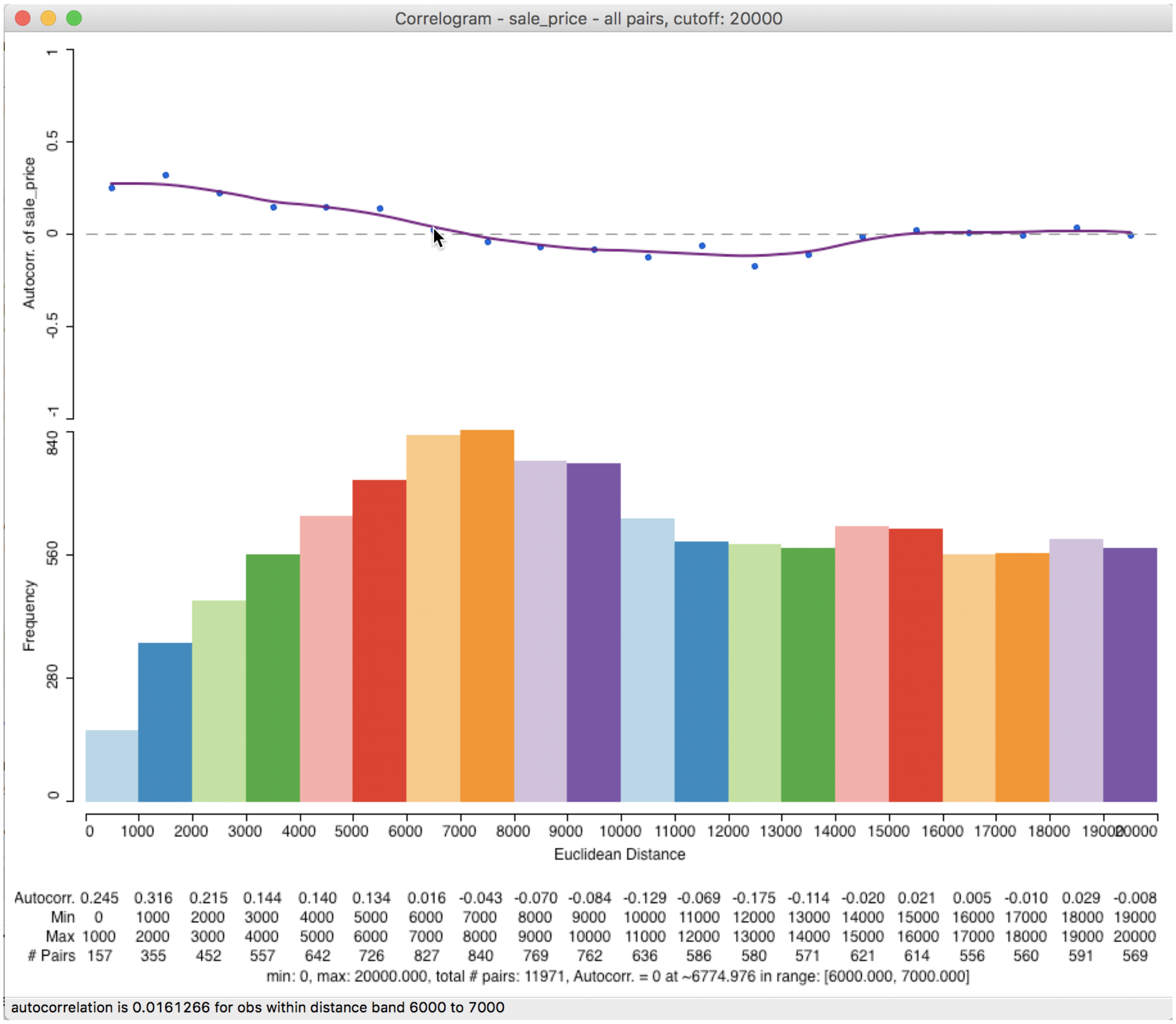
```
Autocorr.    0.281     0.162     0.057    -0.056    -0.109    -0.126     0.013     0.009    -0.064    -0.048
      Min        0  2413.140  4826.280  7239.420  9652.560 12065.700 14478.840 16891.980 19305.120 21718.260
      Max 2413.140  4826.280  7239.420  9652.560 12065.700 14478.840 16891.980 19305.120 21718.260 24131.400
  # Pairs      679      1354      1888      1933      1495      1402      1440      1376      1539      1557
          min: 0, max: 24131.400, total # pairs: 14663, Autocorr. = 0 at ~7244.419 in range: [7239.420, 9652.560]
autocorrelation is -0.0563531 for obs within distance band 7239.42 to 9652.56
```

Figure 31: Correlogram with half max distance

In order to get an even more precise measure of the range, we can specify the maximum distance as 20000 feet (3.8 miles) by typing this value into the bin, and select 20 bins (there are plenty of observations, so there is no danger of having fewer than 30 pairs in any bin).

The resulting correlogram, shown in Figure 32, has a distance range for each bin of 1000 feet (about 0.2 miles). This provides a much finer grained measure of the range of spatial autocorrelation (this may require some expansion of the window to clearly see all the values on the horizontal axis and the displayed statistics).

The intersection between the correlogram and the zero axis is almost exactly in the middle of the 6000-7000 foot range, about 6775 feet (or 1.3 miles). In sum, by changing the correlogram parameters from the default to a much smaller value and increasing the number of bins, the estimate of our range of spatial interaction goes from about 7200 feet to about 6800 feet. This highlights the importance of sensitivity analysis and going beyond the default values.



Autocorr.	0.245	0.316	0.215	0.144	0.140	0.134	0.016	-0.043	-0.070	-0.084	-0.129	-0.069	-0.175	-0.114	-0.020	0.021	0.005	-0.010	0.029	-0.008
Min	0	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000	12000	13000	14000	15000	16000	17000	18000	19000
Max	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000	12000	13000	14000	15000	16000	17000	18000	19000	20000
# Pairs	157	355	452	557	642	726	827	840	769	762	636	586	580	571	621	614	556	560	591	569

min: 0, max: 20000.000, total # pairs: 11971, Autocorr. = 0 at ~6774.976 in range: [6000.000, 7000.000]

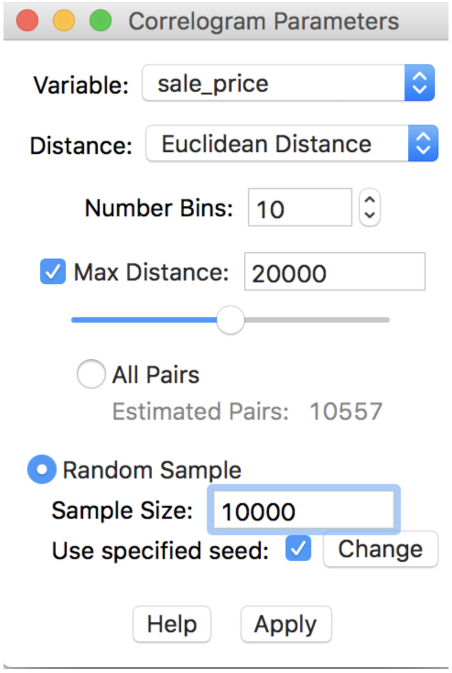
autocorrelation is 0.0161266 for obs within distance band 6000 to 7000

Figure 32: Custom distance and bins

Large(r) data issues – random sampling

A final feature of the correlogram in GeoDa is the ability to compute the spatial correlations for a sample of locations, which reduces the number of pairs used in the calculations. This is especially useful when the data set is large, in which case the number of pairs could quickly become prohibitive.

In our example, this is not really necessary, since the default sample size of 1,000,000 that is used to generate the random sample exceeds the current total number of pairs in the actual data. Nevertheless, to illustrate the process, we go back and use **20000** for the maximum distance, with 10 bins, and check the **Random Sample** radio button, with **10000** for the sample size, as in Figure 33. To allow exact replication, we make sure that the **Use specified seed** option is checked.



Correlogram Parameters

Variable: sale_price

Distance: Euclidean Distance

Number Bins: 10

☑ Max Distance: 20000

○ All Pairs

Estimated Pairs: 10557

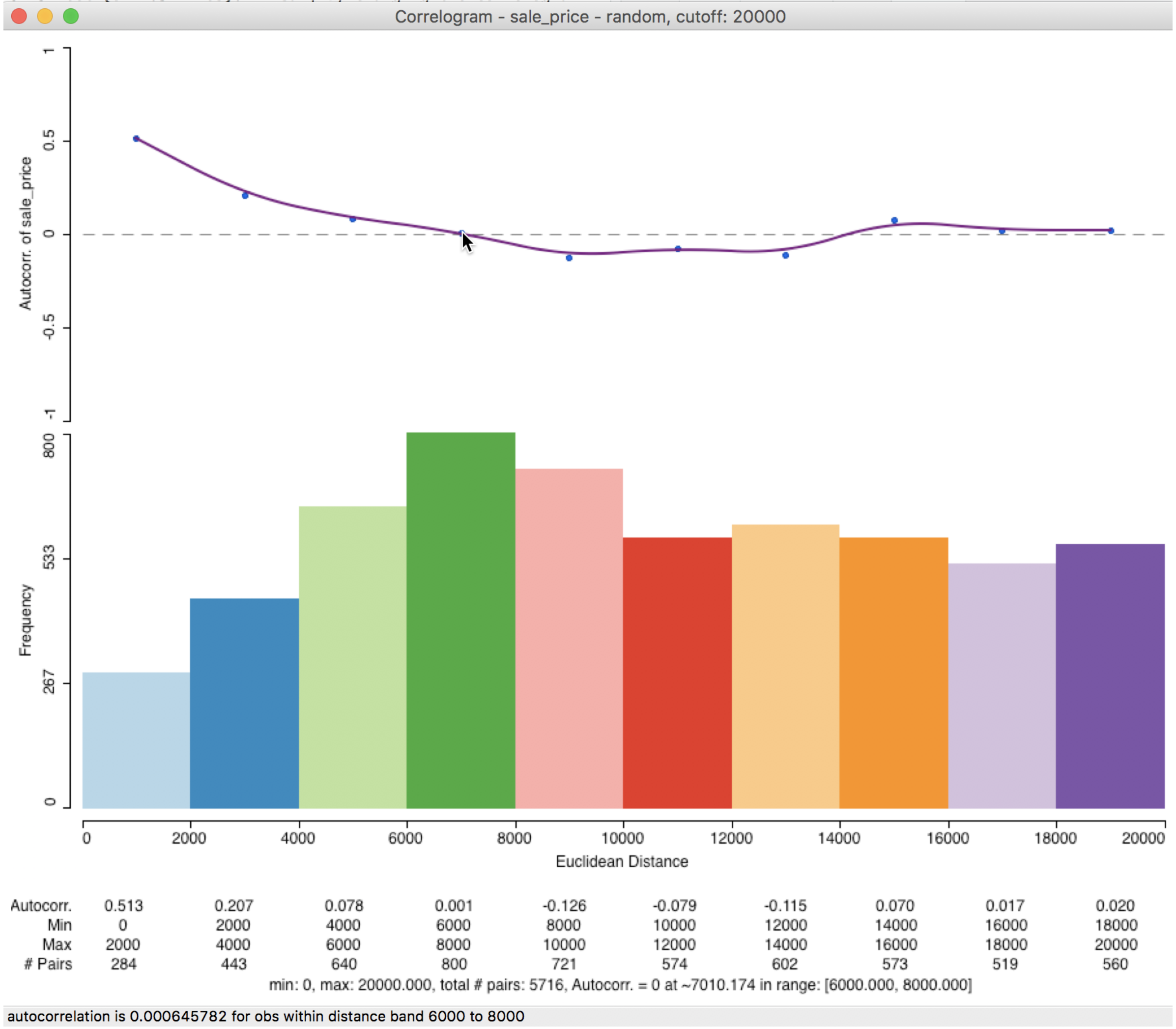
● Random Sample

Sample Size: 10000

Use specified seed: ☑

Figure 33: Randomly sampled observation pairs

In our example, the correlogram that results from the sampled observation pairs is not that different from the default case. As demonstrated in Figure 34, the first correlation is higher, at 0.513, but the range is again roughly 7000 feet. In general, the sampling approximation is quite good, as long as the selected sample size is not too small relative to the original data size.



Correlogram - sale_price - random, cutoff: 20000

Autocorr.	0.513	0.207	0.078	0.001	-0.126	-0.079	-0.115	0.070	0.017	0.020
Min	0	2000	4000	6000	8000	10000	12000	14000	16000	18000
Max	2000	4000	6000	8000	10000	12000	14000	16000	18000	20000
# Pairs	284	443	640	800	721	574	602	573	519	560

min: 0, max: 20000.000, total # pairs: 5716, Autocorr. = 0 at ~7010.174 in range: [6000.000, 8000.000]

autocorrelation is 0.000645782 for obs within distance band 6000 to 8000

Figure 34: Correlogram for random observation pairs

References

Anselin, Luc. 1996. “The Moran Scatterplot as an ESDA Tool to Assess Local Instability in Spatial Association.” In *Spatial Analytical Perspectives on Gis in Environmental and Socio-Economic Sciences*, edited by Manfred Fischer, Henk Scholten, and David Unwin, 111–25. London: Taylor; Francis.

Bjornstad, Ottar N., and Wilhelm Falck. 2001. “Nonparametric Spatial Covariance Functions: Estimation and Testing.” *Environmental and Ecological Statistics* 8:53–70.

Cliff, Andrew, and J. Keith Ord. 1973. *Spatial Autocorrelation*. London: Pion.

———. 1981. *Spatial Processes: Models and Applications*. London: Pion.

Hall, P., and P. Patil. 1994. “Properties of Nonparametric Estimators of Autocovariance for Stationary Random Fields.” *Probability Theory and Related Fields* 99:399–424.

Moran, Patrick A.P. 1948. “The Interpretation of Statistical Maps.” *Biometrika* 35:255–60.

Munasinghe, Rajika L., and Robert D. Morris. 1996. “Localization of Disease Clusters Using Regional Measures of Spatial Autocorrelation.” *Statistics in Medicine* 15:893–905.

University of Chicago, Center for Spatial Data Science – anselin@uchicago.edu↩

If the project was started without the right project file, then at the very least, the queen contiguity weights need to be created that are based on the Thiessen polygons (previously **clev_sls_154_core_q**).↩

See Cliff and Ord (1973) or Cliff and Ord (1981) for an extensive technical discussion.↩

In a bivariate linear regression $y = \alpha + \beta x$, the least squares estimate for β is $\sum_i (x_i \times y_i) / \sum_i x_i^2$. In the Moran scatter plot, the role of y is taken by the spatial lag $\sum_j w_{ij} z_j$.↩

The implementation of the random numbers in GeoDa takes advantage of multi-threading, a built-in form of parallel computing.↩

This is the number of software multithreading cores, typically twice the number of physical cores (the example shown is on a machine with 4 CPU hardware cores, which yield 8 multi-threading cores, the number listed).↩

The **STANDARDIZED (Z)** option is the usual standardization, based on subtracting the mean and dividing by the standard deviation. The Calculator also contains an alternative, based on the standardized mean absolute deviation, **STANDARDIZED (MAD)**. This operation is not used in the Moran scatter plot.↩

See also Hall and Patil (1994) for a technical discussion of the general principle.↩

Since the estimated number of pairs is computed on the fly, as the maximum distance is adjusted by means of the slider, it is only an approximation. The actual number of pairs used in the calcuation of the pairwise correlations is given in the status bar of the correlogram.↩