Taking Time (and Space) Seriously:

How Scholars Falsely Infer Policy Diffusion from Model Misspecification

Online Appendix

Additional Experiments

In this *Online Appendix* we explore the results of other Monte Carlo experiments that we reference in the manuscript. We first detail how including irrelevant TLSLs potentially leads to bias in the other covariates and then present the inferential consequences to incorrectly specifying policy diffusion occurring through multiple channels.

Bias in Other Covariates

Recall that for each scenario we estimate two models, both of which include an irrelevant TLSL: the "underfitted" model omits a key variable from the model specification, and the "overfitted" model includes that relevant variable. Figure A.1 shows the TLSL false discovery rate for scenarios 2 and 3 in the left column and the coefficient bias for X_1 ($\hat{\beta}_1$) and X_2 ($\hat{\beta}_2$ in Scenario 2, $\hat{\theta}_{WZ}$ in Scenario 3) as measured by the root mean square error (RMSE)[1] for the two scenarios in the right column.

[1]We found the RMSE using $\sqrt{\frac{\sum_{i=1}^{N}(\hat{y}-y)^2}{n}}$, where i is an individual simulation, N is the total number of simulations, y is the actual parameter value from the DGP and $\hat{y}$ is the estimated value from the regression.

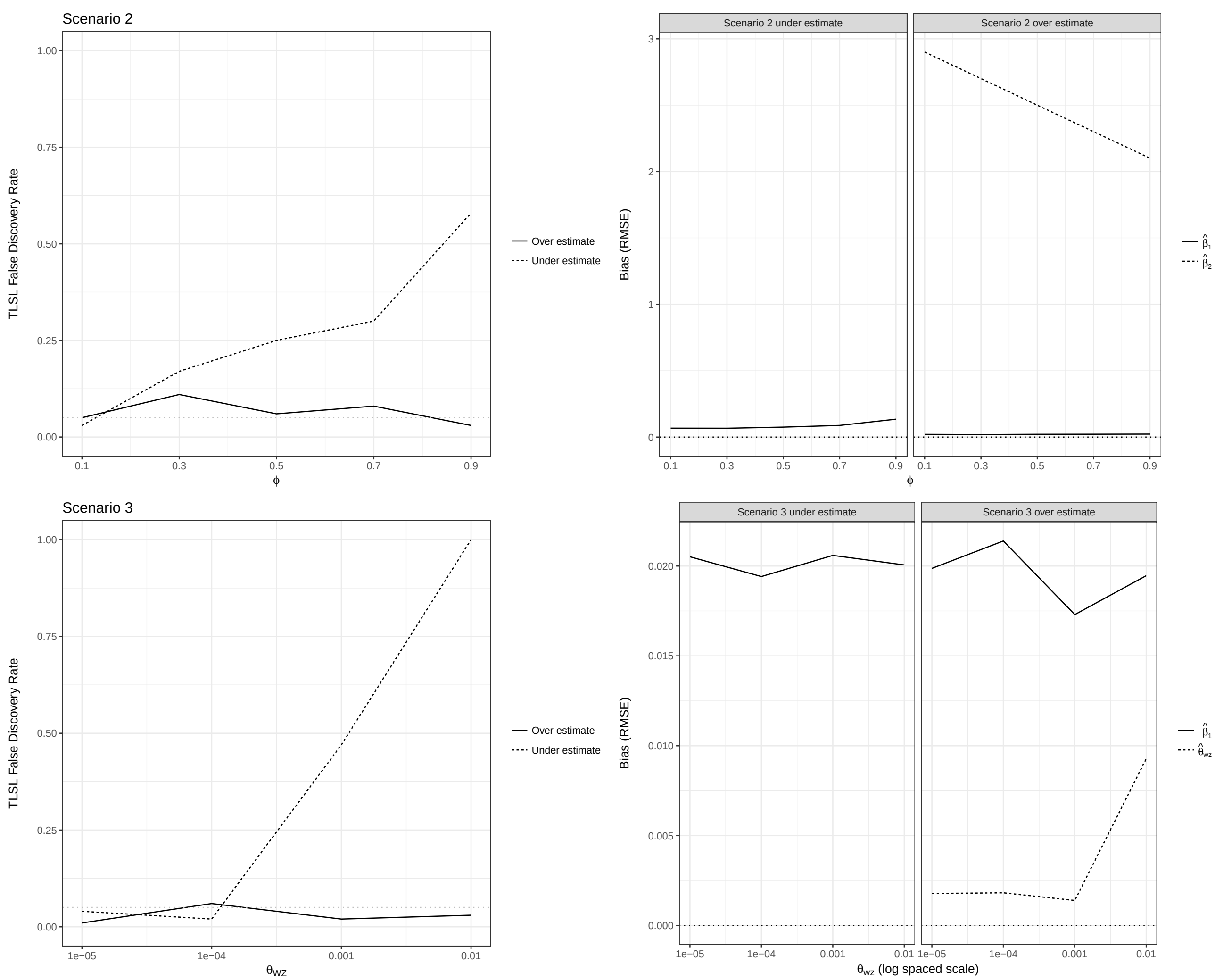
The under- and overfitted (or, "overestimated") models suffer from different problems. As we describe in the manuscript, the TLSL false discovery rate for the underestimated models is higher than indicated by its p-value. The problem is worse when autocorrelation (ϕ in scenario 2) or spatial clustering (θ_{WZ} in scenario 3) increases. Including the omitted autoregressive and/or spatially clustered variables in the overestimated models, on the other hand, returns the TLSL false discovery rate to a level closer to that suggested by the p-value.

There is generally not much bias in the non-TLSL variables' coefficients in the underestimated models, regardless of the degree of autocorrelation or spatial clustering in the omitted variable. In some cases, however, the non-TLSL variables' coefficients become biased. The coefficient for the autoregressive X_2 variable in Scenario 2 is particularly affected by including a spurious TLSL. Keep in mind that $\beta_2 X_{AR}$ is part of the DGP, so this suggests that including an irrelevant TLSL produces inaccurate coefficient estimates for the variables that were indeed part of the data generating process. The coefficient for the non-autoregressive/non-spatially autocorrelated X_1 variable in Scenario 3 is also persistently biased in both the under and overestimated models, as was $\hat{\theta}_{\mathbf{W}Z}$ for higher values of $\theta_{\mathbf{W}Z}$, though to a smaller degree.

Policy Diffusion through Multiple Avenues

One of the strengths of the SLX model is the ability to easily and efficiently incorporate spatial diffusion through multiple connectivity matrices. In the case of temporally-delayed policy diffusion (described in the manuscript), this would occur through multiple TLSLs. For example, one could envision that governments would emulate tax policy of those that are more proximate and that have similar ideologies. In this case, the previous average of neighbors' tax rates would be weighted by both contiguity and ideological similarity, and one could include two TLSLs to reflect the alternative paths of policy diffusion.

Figure A.1: TLSL False Discovery Rate (Left) and Coefficient Bias (Right) for various ϕ (Scenario 2) and θ_{WZ} (Scenario 3)



The first row of plots shows results from Scenario 2. The second row is from Scenario 3.

It is clear that if tax rates are the end result of multiple avenues of policy diffusion, then it is best to measure those avenues correctly and include them in the model specification. In reality, however, it is often not clear whether there are multiple channels of diffusion or if the correct channels have been included. To assess the consequences of these types of model misspecification, we produce another series of Monte Carlo experiments. We begin with a data-generating process that includes an independent variable ($\mathbf{X}$) that influences the outcome ($\mathbf{y}$) directly (through β) and indirectly (through the two weights matrices, $\mathbf{W_1}$ and $\mathbf{W_2}$, and their effects, θ_1 and θ_2):

$$\mathbf{y} = \mathbf{X}\beta + \theta_1 \mathbf{W_1 X} + \theta_2 \mathbf{W_2 X} + \epsilon \tag{A.1}$$

You may notice that Equation A.1 is a more general version of the TLSL model presented in the manuscript. $\mathbf{X}$ is drawn from a normal distribution ($X \sim \mathcal{N}(0, 2)$), and the parameters are fixed at $\beta = 1$, $\theta_1 = 0.1$ and $\theta_2 = 0.25$. We produce 1000 simulations of 500 observations each. Since the primary driver in coefficient bias in the omitted variable case is correlation with the omitted variable, we create $\mathbf{W_1}$, $\mathbf{W_2}$ and $\mathbf{W_3}$ (described below) so that they vary in network correlation (from -0.6 to +0.6). We can use the varying degree of correlation to explore how misspecifying the avenues of diffusion influences the inferences one derives from the SLX model.

When the outcome is the result of multiple avenues of diffusion, what are the consequences of only modeling one avenue of diffusion? How does that change our inferences about the pattern of policy diffusion? In other words, if Equation A.1 depicts the data-generating process, then what happens when we estimate the following and omit a relevant TLSL ($\theta_2 \mathbf{W_2 X}$)?

$$\mathbf{y} = \mathbf{X}\beta + \theta_1 \mathbf{W_1 X} + \epsilon \tag{A.2}$$

This would be a scenario where tax rates depend on the weighted average of tax rates among neighbors—both proximate and ideological—but only the proximate neighbors are included in the model. This is a classic case of underfitting the model by omitting a relevant variable.

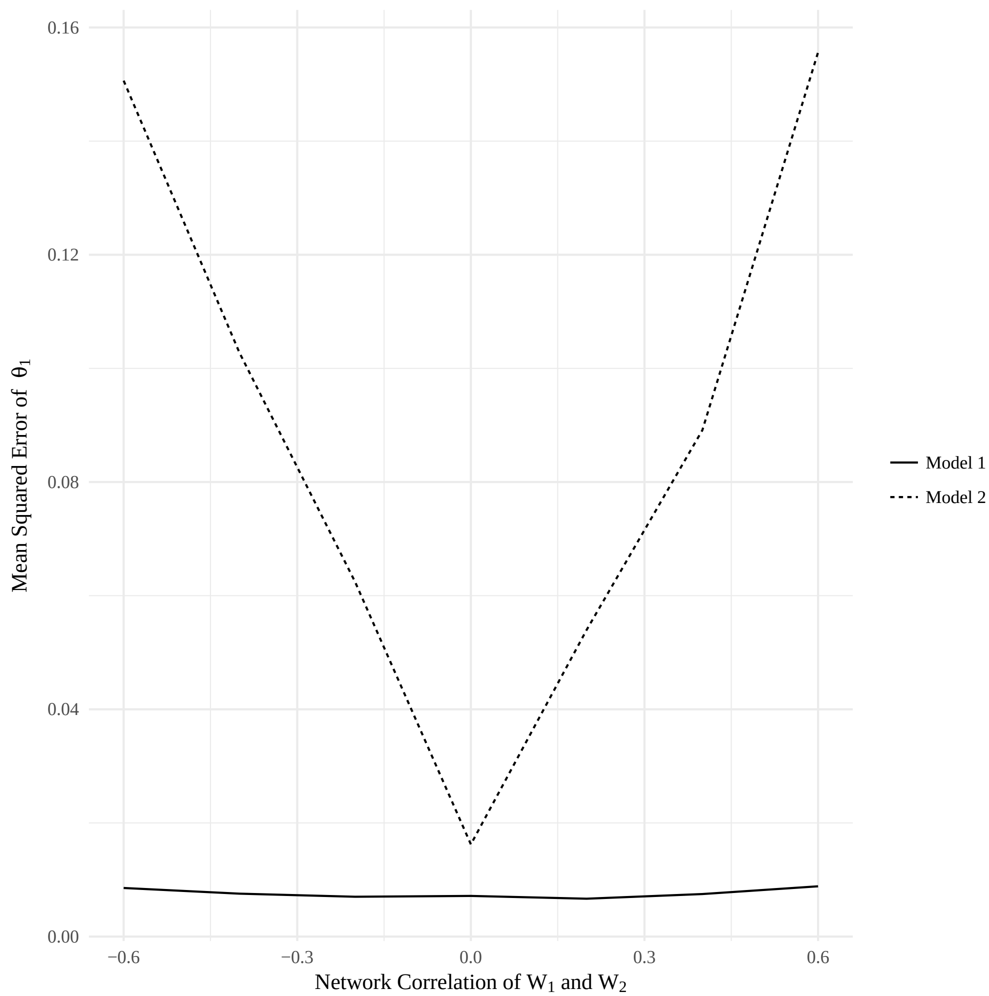
As Gujarati (2003: 510-513) demonstrates, the severity of the consequences depend on the amount of correlation between $\mathbf{W_1 X}$ and $\mathbf{W_2 X}$. If the omitted variable is not correlated with the regressor (or has no effect on $\mathbf{y}$), then there is only a small impact on model efficiency. In studies of policy diffusion, it is unlikely that policy diffusion would occur through multiple avenues that are completely distinct and uncorrelated with each other. A more probable consequence of omitting the relevant TLSL is that the effect of θ_1 will be biased (because some of the effect of $\mathbf{W_2X}$ will be falsely attributed to $\mathbf{W_1X}$). The direction of the bias is a function of the omitted coefficient, $\theta_2 = 0.25$, and the degree of correlation between $\mathbf{W_1X}$ and $\mathbf{W_2X}$.[2] Since $\theta_2 = 0.25$ across all simulations, the bias will be negative for those scenarios with negative network correlation, and positive for those scenarios with positive network correlation. Another troubling consequence is that the standard errors are biased estimates (though the direction of the bias depends on the relative importance of the regressors, see Gujarati 2003: 512), which means that confidence intervals will be misleading.

We first compare the performance of Model 1 ("correct" model shown in Equation A.1) to Model 2 (omits a relevant TLSL, shown in Equation A.2) in terms of mean squared error (MSE). As MSE is a measure of both bias and efficiency, lower values indicate better model specification. Figure A.2 shows that Model 1 provides lower levels of MSE than Model 2 across all degrees of network correlation between $\mathbf{W_1X}$ and $\mathbf{W_2X}$. More intuitively, it is useful to see how omitting a relevant TLSL will influence the inferences that one makes about policy diffusion. In Figure A.3 we show the median value of θ_1 (represented by the dot) and lower and upper 90% confidence intervals across values of network correlation between $\mathbf{W_1X}$ and $\mathbf{W_2X}$. The horizontal dashed line provides the true value of θ_1. The only scenario where the confidence intervals would overlap the true value of θ_1 is when the network correlation is 0. In all other cases, the effect of θ_1 is statistically different from the true effect. Particularly problematic—though potentially rare—is the scenario with a correlation of -0.6. If the two avenues of diffusion are negatively correlated with each other, then omitting $\mathbf{W_2X}$ will force the effect of θ_1 to be negative (recall that it is actually 0.1). These experiments show that scholars should specify multiple diffusion channels if

[2] More specifically, it is the slope coefficient in the auxiliary regression of $\mathbf{W_2X}$ on $\mathbf{W_1X}$.

necessary, and that the degree of bias will increase as the correlations between diffusion channels increases.

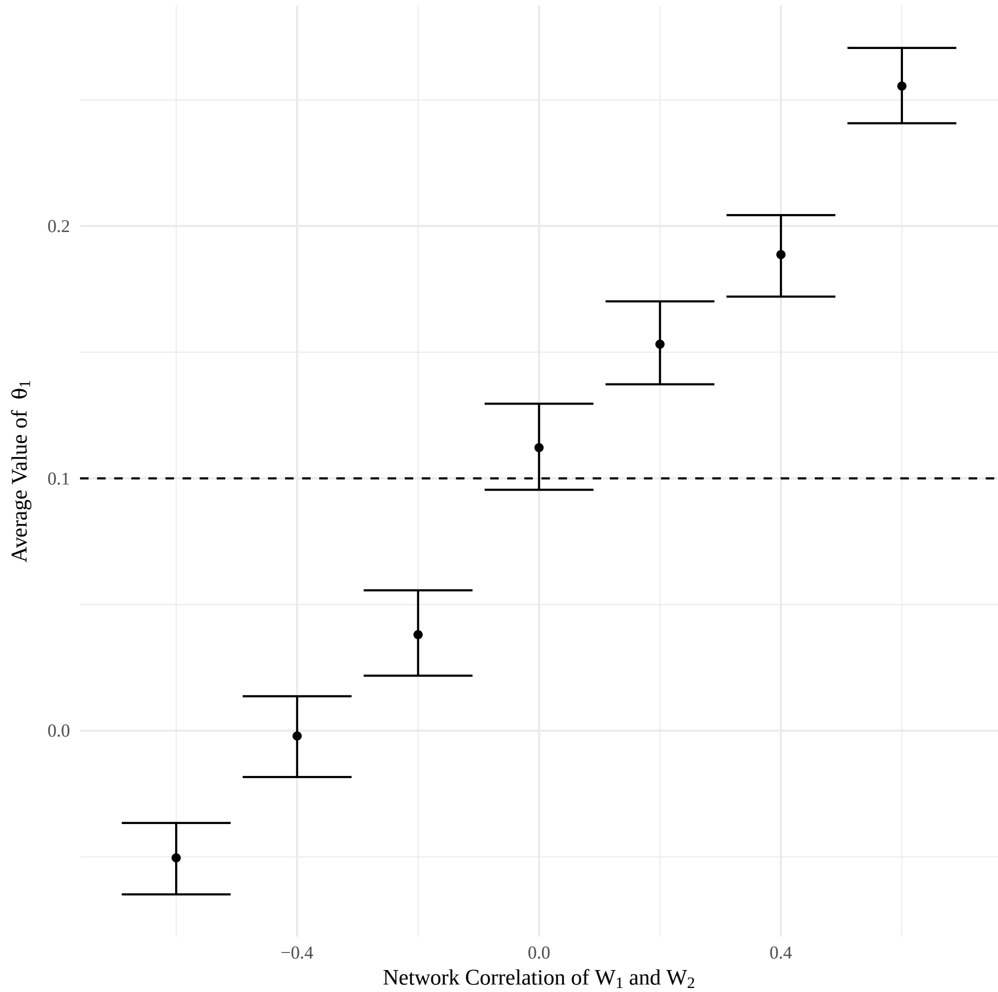
Figure A.2: Mean Squared Error of θ_1 for Models that Include (Model 1) and Omit (Model 2) a Relevant Avenue of Diffusion ($\mathbf{W_2X}$)



What if we are wrong about the particular channel of policy diffusion? If there are multiple channels of policy diffusion, but we mistakenly include a channel with no direct effect on the outcome, what are the consequences?

$$\mathbf{y} = \mathbf{X}\beta + \theta_3\mathbf{W_3X} + \epsilon \tag{A.3}$$

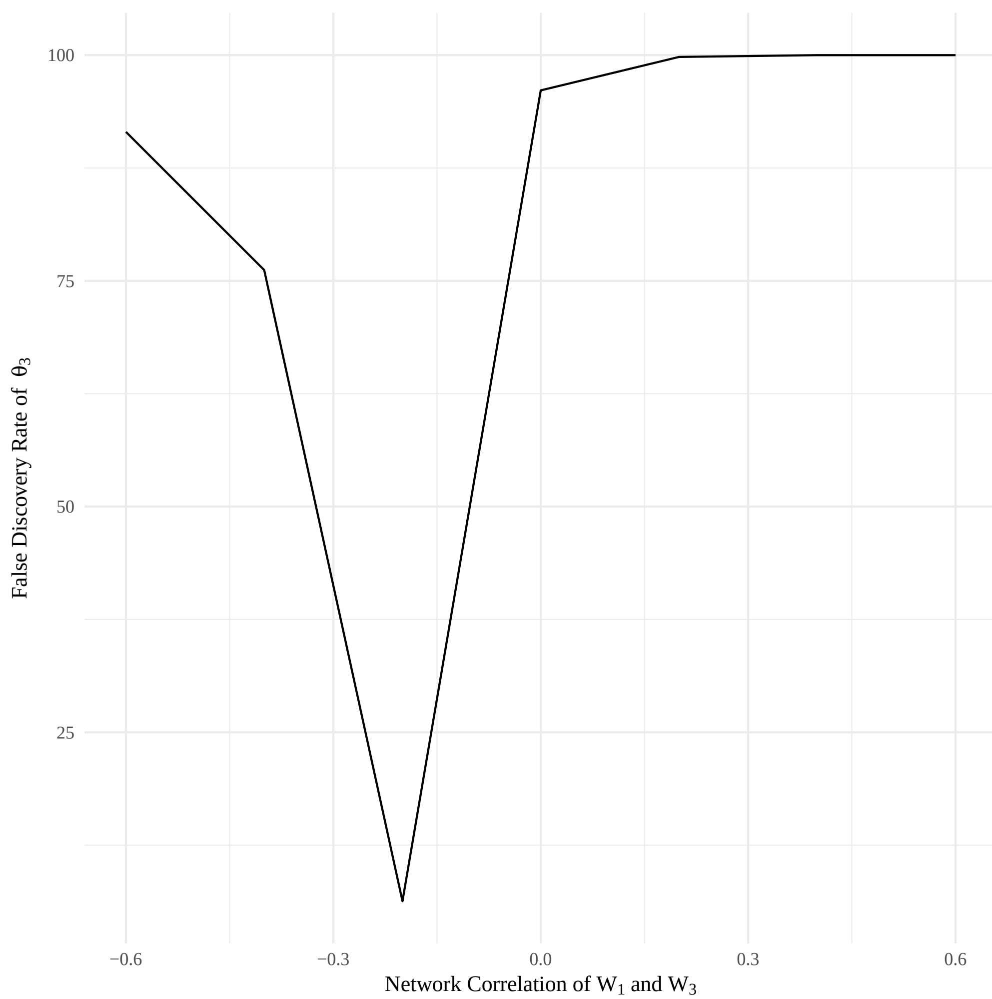
Figure A.3: Median and 90% Confidence Intervals of the Effects of θ_1 Compared to the True Value ($\theta_1 = 0.1$) in Model 2



This would be a scenario where tax rates depend on the weighted average of tax rates among proximate and ideological neighbors but we mistakenly identify neighbors as those states that have similarly-composed economies. This is a classic case of both underfitting the model (by omitting the two relevant TLSLs) and overfitting the model (by including the irrelevant TLSL). The consequences of this are more severe than the previous example of model misspecification because it involves both types of model misspecification.

To explore the consequences of incorrect specification of the avenue of policy diffusion, we create another weights matrix, $\mathbf{W_3}$, and we vary the degree of network correlation between it and $\mathbf{W_1}$ (the correlation between $\mathbf{W_3}$ and $\mathbf{W_2}$ is held constant at +0.1). We expect that the SLX will falsely reject the true null hypothesis of no effect of $\mathbf{W_3X}$ (or that $\theta_3 = 0$) at unacceptably high levels, depending on the degree of network correlation between both omitted TLSLs.

Figure A.4: False Discovery Rate for θ_3 in a Model that Omits Two True Avenues of Policy Diffusion



In the vast majority of cases, including an irrelevant TLSL in the place of the relevant avenues of policy diffusion will cause scholars to falsely conclude that policy diffuses through that avenue. The one exception is when the network correlation between $\mathbf{W_1X}$ and $\mathbf{W_3X}$ is -0.2, and this is due to chance; this negative correlation is being cancelled out by the positive correlation (+0.1) between $\mathbf{W_2X}$ and $\mathbf{W_3X}$, thus lowering the false discovery rate to approximately 10% (acceptable given the 90% confidence level). Overall, incorrectly specifying the avenue of policy diffusion—especially with one that is correlated with the true avenues—will result in incorrect inferences about the spatial diffusion process.

spatialWeights

This project has shown the importance of testing for spatial autocorrelation before including a temporally-lagged spatial-lag (TLSL) in a regression model. While there is abundant statistical software to test for spatial autocorrelation, it is not easy to do so with time-series cross sectional data on a per-time interval basis.

We introduce the `spatialWeights` package for the R programming language to make it easy for researchers to calculate spatial weights for TSCS data and report spatial clustering test statistics.

The following is a simple demonstration of the package's syntax for creating a TLSL and reporting per-time-period Moran's I tests using simulated data that exhibits strong spatial autocorrelation:

```
# Create simulated TSCS data
sims <- expand.grid(ID = letters, year = 2012:2017)
sims$located_continuous <- nrow(sims):1
sims$y <- nrow(sims):1 - 200

# Find TLSL weights and Moran's I for continuous simulated data
df_weights_cont_tlsl <- monadic_spatial_weights(
```

```
df = sims, id_var = 'ID',
time_var = 'year',
location_var = 'located_continuous',
y_var = 'y', mc_cores = 1, tlsl = TRUE)
```

This call returns both a data frame with the calculated spatial weights (note: only showing the first six rows):

```
ID year sp_wght_located_continuous_y lag_sp_wght_located_continuous_y
1  a 2010                       -19825                              NA
2  a 2011                       -28275                          -19825
3  a 2012                       -36725                          -28275
4  a 2013                       -45175                          -36725
5  a 2014                       -53625                          -45175
6  a 2015                       -62075                          -53625
```

and prints to the console the Moran's I for each time-period:

```
Continuous location variable detected. Proximity found using
method = euclidean.

2010: Moran's I p-value: <2e-16
2011: Moran's I p-value: <2e-16
2012: Moran's I p-value: <2e-16
2013: Moran's I p-value: <2e-16
2014: Moran's I p-value: <2e-16
2015: Moran's I p-value: <2e-16
```