# Supplemental Appendix

## STADL Up! The Spatio-Temporal Autoregressive Distributed Lag Model for TSCS Data Analysis

Scott J. Cook, Jude C. Hays, and Robert J. Franzese, Jr.

# 1 TSCS Data in the *APSR*, *AJPS* and *JOP*

To calculate the number of articles that use time-series cross-sectional data, we analyzed 7336 research articles appearing in *The American Political Science Review*, *The American Journal of Political Science*, and *The Journal of Politics*, from 1980 to 2019. Using this set of articles, we counted the occurrence(s) of a list of keywords related to time-series cross-sectional data in each article. For the years 1980 to 2014, we use the data and metadata provided by JSTOR's API.[1] To collect the recent articles that are not covered by JSTOR's API, we directly scraped the text from each journal's website, the same keywords are used to count the number of occurrences in each article.[2]

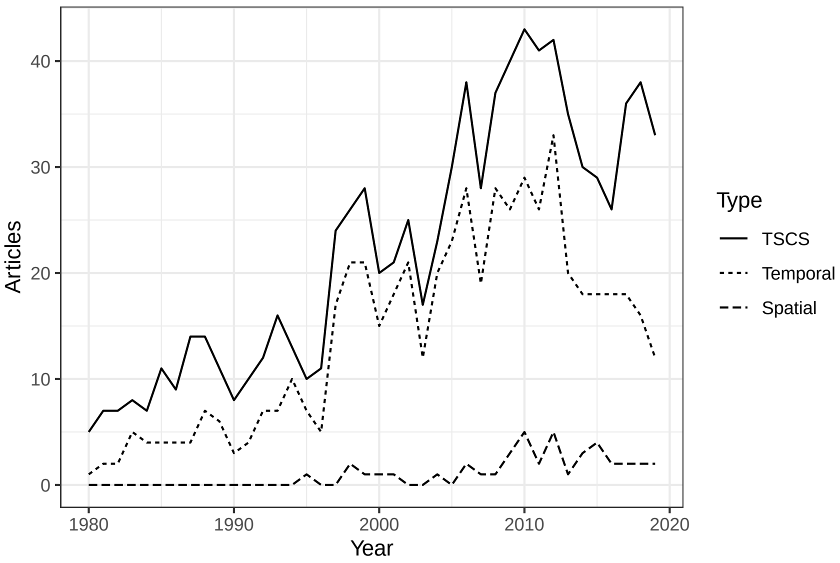
These results are summarized in Figure 1 which provides the yearly count of articles using keywords associated with TSCS data analysis. Specifically, TSCS (solid line) gives the number of articles using at least one of the following keywords: 'time series cross section(al)', 'tscs', 'panel data.'[3] Within the set of TSCS articles, we then count those that use keywords consistent with Temporal analysis (dotted line) – e.g., 'time series', 'time serial', 'temporal autocorrelation', 'temporal correlation','temporal dependence', 'temporal dynamics', 'time dependence', 'time lag(ged)', 'time lagged dependent', 'serial correlation', 'serially correlated', 'serial dependence.' – and Spatial analysis (dashed line) – e.g., 'spatial dependence', 'spatial autocorrelation', 'spatial correlation', 'spatially correlated', 'spatial lag', 'spatial-lag dependent', 'spatially lagged', 'spatially lagged dependent.'

[1] JSTOR's API has different time coverage for each journal. It provides upto 2014, 2015, 2018 for APSR, JOP, AJPS, respectively.

[2] For the replication data for this Appendix and the associated article see Cook, Hays, and Franzese (2022).

[3] Since JSTOR's API only provides the count of words up to trigrams, we use the keyword 'time series cross' instead of 'time series cross section' or 'time series cross sectional'.

**Figure 1:** Count of Articles Using TSCS Data in the Top-3, 1980-2019



The results demonstrate that TSCS data remain widely used, with 33 articles appearing in the top-3 (APSR, AJPS, and JOP) in 2019 alone. Few of these, however, seem to meaningfully consider both temporal and spatial dependence. In 2019, for example, of the 33 TSCS articles only 12 used keywords consistent with temporal analysis and only 2 with spatial analysis. As such, at most 2 could have jointly considered both temporal and spatial analysis, as our manuscript suggests in necessary.

# 2 Monte Carlo Analysis

## 2.1 Additional Design Details

The spatial locations for the units are generated by twice taking $N$ draws from a standard uniform to create xy-coordinates for each unit. The $w_{ij}$ relative connections between units are then generated using a $k$-Nearest Neighbor algorithm with $k = 5$, returning a binary $N$-by-$N$ matrix $\mathbf{W_N}$ with each element $w_{ij} = 1$ for the five closest $j$ to $i$ and 0 for all others (and all $w_{ii} = 0$ along the diagonal). The Kronecker product of this matrix and a $T$-dimensional identity matrix produces $\mathbf{W}$, an $NT$-by-$NT$ matrix with each $N$-by-$N$ block $\mathbf{W_N}$ along the prime diagonal giving the dyadic relations, which are assumed constant over time. We also assume that this same $\mathbf{W}$ operates with respect to spatial dynamics in $\mathbf{y}$, $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{e}$, and that it is known to the researcher.[4]

In order to focus on how varying the strength of $\phi_y$ and $\rho_y$ affect model performance, we fix the parameters for $\beta = 2$, $\phi_x = 0.6$, and $\rho_x = 0.3$, and confine attention to variation in the temporal dependence – $\phi_y = \{0,0.1,\ldots,0.4,0.5\}$ – and spatial dependence – $\rho_y = \{0,0.05,\ldots,0.25,0.3\}$ – in $\mathbf{y}$.[5] We are primarily interested in understanding how well the LDV and SAR models recover $\beta$, however, we also examine how the estimates of $\phi_y$ and $\rho_y$ are affected.

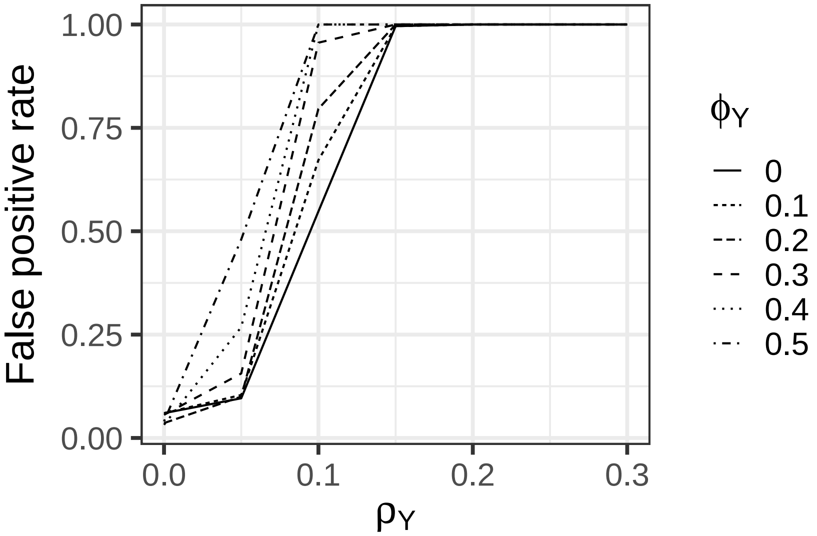
## 2.2 Lagrange Multiplier Tests

Current best practice (as advised by Beck and Katz 2011, e.g.) suggests post-estimation diagnostic tests for remaining serial dependence in estimated residuals, specifically Lagrange Multiplier tests of auxiliary regressions of estimated residuals on their lags. Unfortunately, these post-estimation tests for remaining *temporal* dependence will lead researchers astray when there is (unmodeled, i.e. remaining) residual *spatial* autocorrelation. Here, too, the unmodeled spatial dependence is "mistaken" for temporal dependence, causing researchers to over-reject the null, the frequency of which false-positive rate, intuitively, increases in $\rho_y$ as shown in Figure 2). The LM test for residual serial correlation has power against the incorrect alternative under these circumstances, erroneously registering the spatial dependence as temporal dependence, leading researchers to take inappropriate remedial actions – e.g., modeling higher-order time-lags of the outcome – rather than addressing the truly spatial cause of the dependence in the residuals.

[4] Noteworthy–in that these following are also commonly issues in TSCS data-analysis–among the simplifying assumptions in our simulations are (1) no parameter heterogeneity and (2) all regressors $\mathbf{X}$ are exogenous.

[5] The extent of the bias in these parameter estimates is a function of $\phi_x$ and $\rho_x$, but for tractability we fix and do not estimate these two parameters.

**Figure 2:** False-Positive Rate of Lagrange Multiplier test with Spatial Dependence



## 2.3 STADL Results

In the main text we demonstrate the bias of the SAR and the LDV models when there is unmodeled temporal or spatial dependence, respectively. We also imply that since the STADL model accounts for both temporal and spatial dependence, that it should be unbiased under these conditions. Here we demonstrate that explicitly, showing that under all conditions evaluated in our simulations the STADL model is an unbiased estimator of $\beta$ (Figure 3), $\rho$ (Figure 4), and $\phi$ (Figure 5).

**Figure 3:** STADL performance with Spatio-temporal dependence – Bias in $\beta$

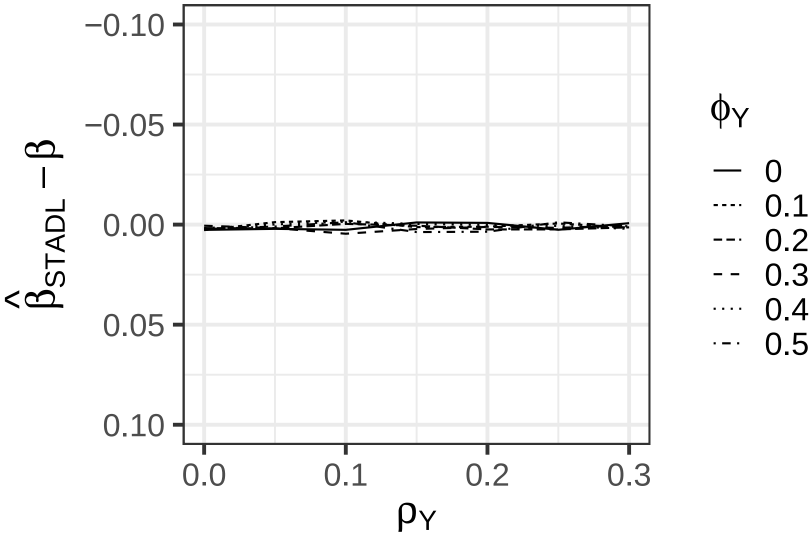


**Figure 4:** STADL performance with Spatio-temporal dependence – Bias in $\rho$

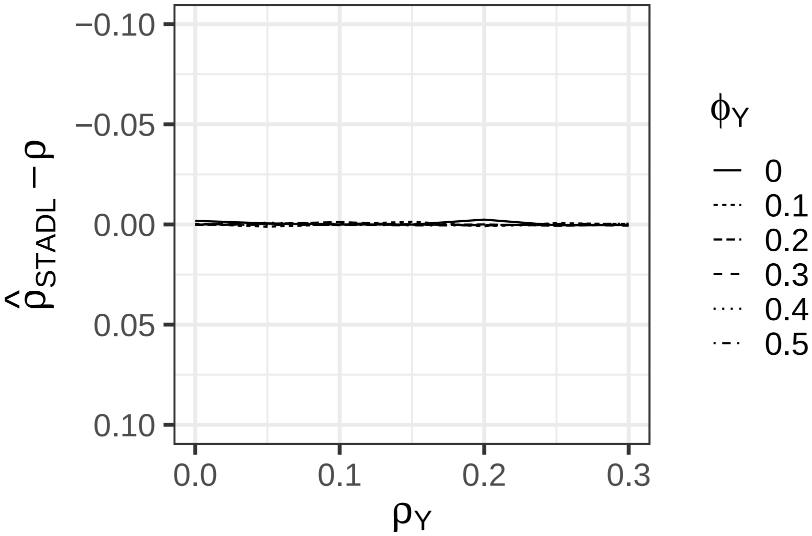
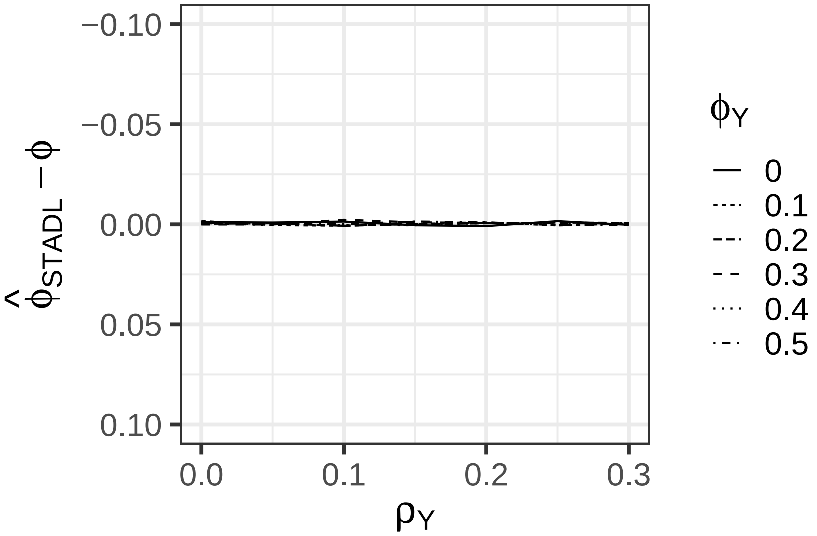


**Figure 5:** STADL performance with Spatio-temporal dependence – Bias in $\phi$



## 2.4 Additional Model Comparisons

In the main text we focus exclusively on bias when evaluating the models under different simulated conditions. Here we report additional quantities of interest, including bias (Bias), average standard error (Avg. SE), the standard deviation of the empirical distribution (SD), mean square error (MSE), and the coverage probabilities (CP).[6] Since reporting all of the simulated conditions in this manner would be unwieldy, we focus on 4 values of $\rho$ (0.0, 0.1, 0.2, 0.3), within each table, and 2 values for $\phi$ (0.0 and 0.5).[7] Both tables report the results for a set value of $\phi$ (Table 1 is $\phi = 0$, Table 2 is $\phi = 0.5$), and different values of $\rho$ (with values increasing as one moves down the Table). The results are consistent with what we would expect, in that failing to properly account for dependence between the observations induces bias, inaccurate standard errors (as demonstrated by the deviation of the Avg. SE and the SD), increased mean square error, and confidence intervals that rarely bound the true value (as indicated by CP).

[6] The coverage probabilities (CP) are calculated using the 95% confidence intervals (CIs) for each models sample coefficient and standard error estimate. The reported value for CP indicates the proportion of trials for which the 95% CI contains the true coefficient.

[7] We have also produced similar tables for other values of $\phi$ (0.1, 0.2, 0.3, 0.4), these results are available on Dataverse or by request.

**Table 1:** Simulation Results, $\phi = 0.0$

| | | $\beta$ | | | | $\phi$ | | $\rho$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Static | LDV | SAR | STADL | LDV | STADL | SAR | STADL |
| $\rho = 0.0$ | Bias | -0.000 | 0.002 | 0.001 | 0.003 | -0.001 | -0.001 | -0.002 | -0.002 |
| | Avg. SE | 0.012 | 0.026 | 0.013 | 0.027 | 0.013 | 0.013 | 0.011 | 0.011 |
| | SD | 0.011 | 0.027 | 0.013 | 0.027 | 0.014 | 0.014 | 0.010 | 0.010 |
| | MSE | 0.000 | 0.001 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 |
| | CP | 0.956 | 0.940 | 0.960 | 0.940 | 0.944 | 0.944 | 0.972 | 0.976 |
| $\rho = 0.1$ | Bias | 0.065 | 0.032 | 0.000 | 0.003 | 0.018 | -0.001 | -0.000 | -0.000 |
| | Avg. SE | 0.012 | 0.028 | 0.013 | 0.027 | 0.014 | 0.013 | 0.010 | 0.010 |
| | SD | 0.012 | 0.027 | 0.014 | 0.027 | 0.014 | 0.013 | 0.010 | 0.010 |
| | MSE | 0.004 | 0.002 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 |
| | CP | 0.000 | 0.808 | 0.956 | 0.952 | 0.732 | 0.948 | 0.952 | 0.940 |
| $\rho = 0.2$ | Bias | 0.145 | 0.009 | 0.001 | -0.001 | 0.071 | 0.001 | -0.002 | -0.002 |
| | Avg. SE | 0.014 | 0.032 | 0.013 | 0.026 | 0.015 | 0.013 | 0.009 | 0.010 |
| | SD | 0.012 | 0.029 | 0.014 | 0.026 | 0.014 | 0.012 | 0.010 | 0.010 |
| | MSE | 0.021 | 0.001 | 0.000 | 0.001 | 0.005 | 0.000 | 0.000 | 0.000 |
| | CP | 0.000 | 0.964 | 0.912 | 0.940 | 0.000 | 0.964 | 0.916 | 0.928 |
| $\rho = 0.3$ | Bias | 0.254 | -0.070 | -0.000 | -0.001 | 0.160 | 0.000 | 0.000 | 0.000 |
| | Avg. SE | 0.018 | 0.038 | 0.014 | 0.026 | 0.017 | 0.012 | 0.008 | 0.009 |
| | SD | 0.014 | 0.035 | 0.013 | 0.027 | 0.017 | 0.013 | 0.008 | 0.009 |
| | MSE | 0.065 | 0.006 | 0.000 | 0.001 | 0.026 | 0.000 | 0.000 | 0.000 |
| | CP | 0.000 | 0.528 | 0.968 | 0.944 | 0.000 | 0.940 | 0.944 | 0.964 |

**Table 2:** Simulation Results, $\phi = 0.5$

| | | $\beta$ | | | | $\phi$ | | $\rho$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | Static | LDV | SAR | STADL | LDV | STADL | SAR | STADL |
| $\rho = 0.0$ | Bias | 1.598 | 0.002 | 1.592 | 0.002 | -0.001 | -0.001 | 0.027 | -0.000 |
| | Avg. SE | 0.027 | 0.027 | 0.031 | 0.028 | 0.008 | 0.008 | 0.014 | 0.006 |
| | SD | 0.020 | 0.023 | 0.023 | 0.024 | 0.006 | 0.006 | 0.012 | 0.006 |
| | MSE | 2.553 | 0.001 | 2.534 | 0.001 | 0.000 | 0.000 | 0.001 | 0.000 |
| | CP | 0.000 | 0.988 | 0.000 | 0.984 | 0.996 | 0.988 | 0.484 | 0.916 |
| $\rho = 0.1$ | Bias | 1.836 | -0.028 | 1.581 | -0.001 | 0.045 | -0.000 | 0.113 | 0.000 |
| | Avg. SE | 0.032 | 0.032 | 0.032 | 0.027 | 0.008 | 0.008 | 0.012 | 0.005 |
| | SD | 0.022 | 0.028 | 0.023 | 0.026 | 0.008 | 0.008 | 0.009 | 0.005 |
| | MSE | 3.372 | 0.002 | 2.499 | 0.001 | 0.002 | 0.000 | 0.013 | 0.000 |
| | CP | 0.000 | 0.900 | 0.000 | 0.964 | 0.000 | 0.948 | 0.000 | 0.964 |
| $\rho = 0.2$ | Bias | 2.219 | -0.282 | 1.570 | 0.004 | 0.163 | -0.001 | 0.197 | 0.000 |
| | Avg. SE | 0.046 | 0.041 | 0.033 | 0.027 | 0.010 | 0.007 | 0.010 | 0.005 |
| | SD | 0.026 | 0.028 | 0.026 | 0.026 | 0.007 | 0.007 | 0.008 | 0.005 |
| | MSE | 4.926 | 0.081 | 2.465 | 0.001 | 0.027 | 0.000 | 0.039 | 0.000 |
| | CP | 0.000 | 0.000 | 0.000 | 0.952 | 0.000 | 0.956 | 0.000 | 0.948 |
| $\rho = 0.3$ | Bias | 2.897 | -0.724 | 1.553 | 0.001 | 0.324 | 0.000 | 0.280 | -0.000 |
| | Avg. SE | 0.075 | 0.048 | 0.035 | 0.025 | 0.009 | 0.007 | 0.008 | 0.005 |
| | SD | 0.030 | 0.027 | 0.025 | 0.024 | 0.006 | 0.007 | 0.006 | 0.005 |
| | MSE | 8.391 | 0.524 | 2.411 | 0.001 | 0.105 | 0.000 | 0.078 | 0.000 |
| | CP | 0.000 | 0.000 | 0.000 | 0.968 | 0.000 | 0.964 | 0.000 | 0.964 |

## 2.5 Spatiotemporal Error Autocorrelation

In the main text we focus primarily on model performance under different levels of spatial interdependence (i.e., $\rho$) and serial autodependence (i.e., $\phi$) in the outcome directly, reflecting the wide use of the SAR and LDV model in applied work. However, we might also be interested in how these models perform under varying levels of spatial and temporal error autocorrelation ($\lambda$ and $\delta$ respectively). To evaluate this, we generate data from a STADL($se^0, te^1$):

$$\mathbf{y}_t = \mathbf{x}_t\beta + (\mathbf{I} - \delta\mathbf{L} - \lambda\mathbf{W})^{-1}\boldsymbol{\varepsilon}_{\mathbf{y}},$$

with $\lambda$ and $\delta$ determining the level of spatial and temporal error autocorrelation.[8] Below we report various quantities of interest (bias, average standard error, standard deviation, mean square error, and the coverage probabilities) for a subset of the simulated conditions: 4 values of $\lambda$ (0.0, 0.1, 0.2, 0.3), within each table, and 2 values for $\delta$ (0.0 and 0.5).[9] Both Table 3 & 4 report the results for a set value of $\delta$ (Table 3 is $\delta = 0$ and Table 4 is $\delta = 0.5$), and different values of $\lambda$ (with values increasing as one moves down the Table). We focus on the estimator of $\beta$, since it is common across models and reflects the total effect of $\mathbf{x}_t$ on $\mathbf{y}_t$ under our simulated conditions.

The results are consistent with what we would expect, so we only briefly describe them globally here. First, under either (or both) forms of error dependence, the static model is unbiased by produces overly confident standard errors (as seen by the difference between Avg. SE and SD), resulting in poor coverage. Second, the respective error correlation models (SCE and SEM) are also always unbiased, however, they each produce overly confident standard errors when there is dependence in the unmodeled dimension. For example, when there is temporal error autocorrelation ($\delta \neq 0$), we see the standard errors of the SEM model is overconfident (i.e., Avg. SE $<$ SD) and, as a result, the coverage probabilities are lower than the targeted 95%. Third, the LDV and SAR models perform poorly under error dependence, as this is now partially captured by the (time or spatial) lag of the outcome, producing an inflationary bias in $\phi$ and $\rho$ and consequently an attenuating bias in $\beta$ (due to the covariate between the lags and $\mathbf{x}$).[10] This can be seen most acutely in the performance of the LDV model under higher values of $\delta$, as the bias increases and the coverage probabilities decrease. Fortunately, the STADL model performs well across all simulated conditions: unbiased, accurate SEs, low mean square error, and coverage probabilities consistently around 95%.

[8] All other model features are held fixed and identical to the DGP in the main text: $\beta = 2$, $\mathbf{x}$ is generated with spatial and temporal dependence, etc.

[9] We have also produced similar tables for other values of $\delta$ (0.1, 0.2, 0.3, 0.4), these results are available on Dataverse or by request.

[10] $\beta$ in the LDV and SAR models now reflects the short-run and pre-spatial effect respectively, however, under the simulated DGP the true long-run and post-spatial effects should be zero. As such, any bias given here does reflect an underestimation of the short-run, pre-spatial effects and a misattribution of the total effect of $\boldsymbol{x}_t$ on $\boldsymbol{y}_t$.

**Table 3:** Simulation Results, $\delta = 0.0$

| | | $\beta$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | Static | SCE | SEM | LDV | SAR | STADL |
| $\lambda = 0.0$ | Bias | -0.001 | -0.002 | -0.000 | -0.001 | 0.000 | -0.002 |
| | Avg. SE | 0.011 | 0.029 | 0.011 | 0.027 | 0.012 | 0.029 |
| | SD | 0.010 | 0.027 | 0.010 | 0.025 | 0.011 | 0.027 |
| | MSE | 0.000 | 0.001 | 0.000 | 0.001 | 0.000 | 0.001 |
| | CP | 0.956 | 0.980 | 0.956 | 0.976 | 0.956 | 0.976 |
| $\lambda = 0.1$ | Bias | -0.000 | -0.000 | -0.001 | 0.000 | -0.002 | 0.000 |
| | Avg. SE | 0.011 | 0.029 | 0.011 | 0.027 | 0.012 | 0.029 |
| | SD | 0.010 | 0.030 | 0.010 | 0.028 | 0.012 | 0.030 |
| | MSE | 0.000 | 0.001 | 0.000 | 0.001 | 0.000 | 0.001 |
| | CP | 0.964 | 0.936 | 0.956 | 0.936 | 0.952 | 0.928 |
| $\lambda = 0.2$ | Bias | 0.001 | 0.000 | 0.001 | 0.001 | -0.003 | -0.000 |
| | Avg. SE | 0.011 | 0.029 | 0.011 | 0.027 | 0.012 | 0.029 |
| | SD | 0.012 | 0.030 | 0.012 | 0.028 | 0.013 | 0.029 |
| | MSE | 0.000 | 0.001 | 0.000 | 0.001 | 0.000 | 0.001 |
| | CP | 0.936 | 0.964 | 0.940 | 0.940 | 0.948 | 0.968 |
| $\lambda = 0.3$ | Bias | -0.002 | -0.002 | -0.002 | -0.002 | -0.009 | -0.001 |
| | Avg. SE | 0.011 | 0.029 | 0.012 | 0.027 | 0.013 | 0.030 |
| | SD | 0.012 | 0.031 | 0.011 | 0.029 | 0.012 | 0.029 |
| | MSE | 0.000 | 0.001 | 0.000 | 0.001 | 0.000 | 0.001 |
| | CP | 0.924 | 0.944 | 0.972 | 0.936 | 0.896 | 0.948 |

**Table 4:** Simulation Results, $\delta = 0.5$

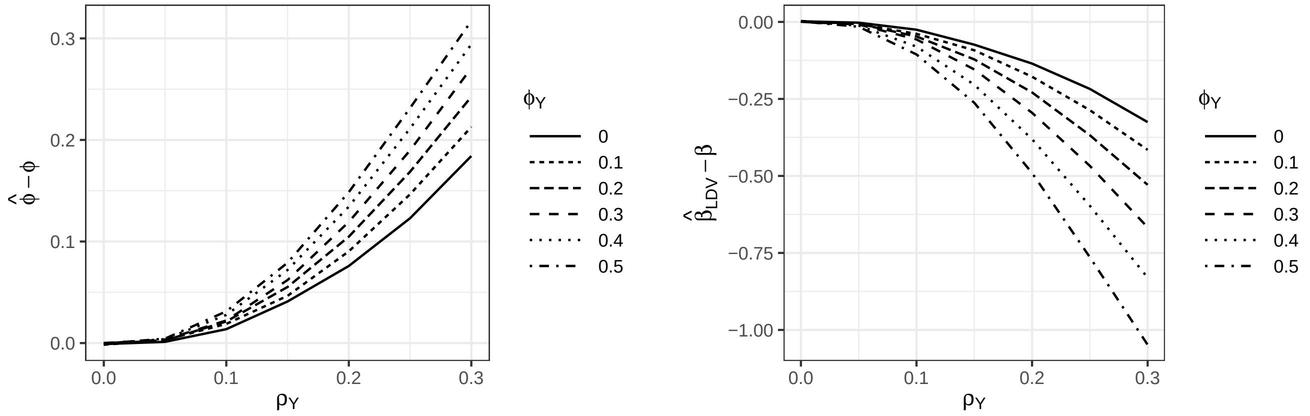
|  |  | $\beta$ |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
|  |  | Static | SCE | SEM | LDV | SAR | STADL |
| $\lambda = 0.0$ | Bias | -0.001 | 0.002 | -0.001 | -0.186 | -0.001 | 0.002 |
|  | Avg. SE | 0.012 | 0.029 | 0.012 | 0.029 | 0.014 | 0.028 |
|  | SD | 0.020 | 0.029 | 0.020 | 0.033 | 0.023 | 0.029 |
|  | MSE | 0.000 | 0.001 | 0.000 | 0.036 | 0.001 | 0.001 |
|  | CP | 0.784 | 0.928 | 0.784 | 0.000 | 0.776 | 0.928 |
| $\lambda = 0.1$ | Bias | -0.000 | -0.003 | -0.000 | -0.192 | -0.004 | -0.004 |
|  | Avg. SE | 0.012 | 0.029 | 0.013 | 0.029 | 0.014 | 0.029 |
|  | SD | 0.020 | 0.031 | 0.020 | 0.033 | 0.022 | 0.031 |
|  | MSE | 0.000 | 0.001 | 0.000 | 0.038 | 0.000 | 0.001 |
|  | CP | 0.756 | 0.928 | 0.788 | 0.000 | 0.800 | 0.928 |
| $\lambda = 0.2$ | Bias | -0.002 | 0.002 | -0.002 | -0.197 | -0.010 | 0.002 |
|  | Avg. SE | 0.012 | 0.029 | 0.013 | 0.030 | 0.014 | 0.029 |
|  | SD | 0.022 | 0.030 | 0.021 | 0.031 | 0.022 | 0.029 |
|  | MSE | 0.000 | 0.001 | 0.000 | 0.040 | 0.001 | 0.001 |
|  | CP | 0.760 | 0.940 | 0.800 | 0.000 | 0.768 | 0.960 |
| $\lambda = 0.3$ | Bias | 0.001 | 0.000 | 0.001 | -0.223 | -0.014 | -0.000 |
|  | Avg. SE | 0.013 | 0.030 | 0.014 | 0.030 | 0.015 | 0.030 |
|  | SD | 0.027 | 0.034 | 0.024 | 0.039 | 0.024 | 0.033 |
|  | MSE | 0.001 | 0.001 | 0.001 | 0.051 | 0.001 | 0.001 |
|  | CP | 0.644 | 0.920 | 0.748 | 0.000 | 0.708 | 0.928 |

## 2.6 Alternative covariate dependence patterns

In footnotes 32 and 36 we discuss how the patterns of bias presented in the main text (for the LDV and SAR models) are a function of the spatio-temporal dependence in the covariate, $x$. To demonstrate this, we re-create the coefficient bias plots presented in the main text (Figures 6 and 7 for the LDV model, and Figures 10 and 11 for the SAR model) under for different conditions for $x$. In the main text we held the dependence in $x$ fixed at $\phi_x = 0.6$ and $\rho_x = 0.3$, here we show the effect of increasing $\phi_x$ or $\rho_x$ respectively.

### 2.6.1 Increased temporal dependence in $x$

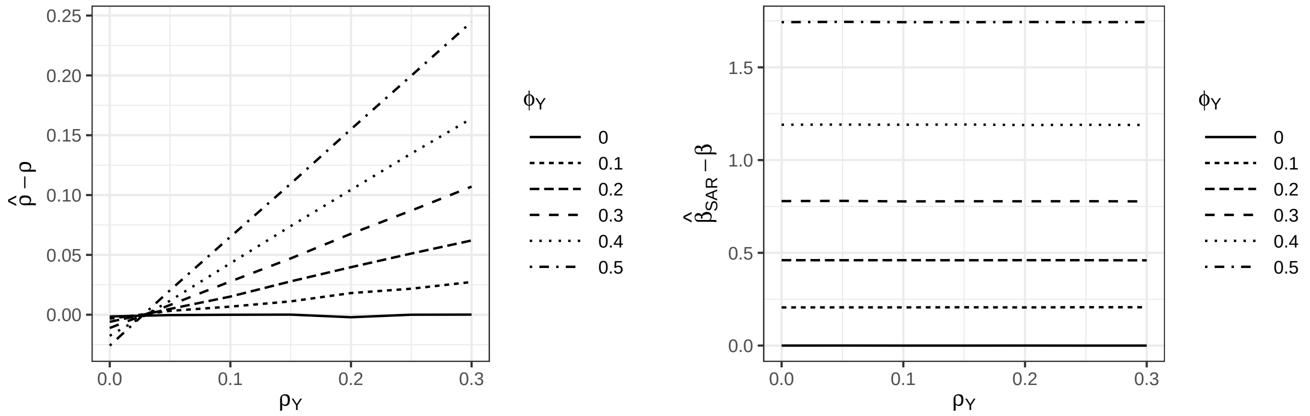
In the first set of results, we increase the temporal dependence in $x$ relative to the spatial dependence, i.e., $\phi_x = 0.8$ and $\rho_x = 0.1$. The results are presented in Figures 6 and 7, and indicate similar, if more pronounced, biases to those reported in the main text. Specifically, we see that the increase in $\phi_x$ (and decrease in $\rho_x$) actually increases the attenuation bias in $\hat{\beta}_{LDV}$ (as seen in Figure 6b), with $\hat{\beta}_{LDV}$ now also clearly decreasing with increases to $\rho_y$.



**(a)** Bias in $\hat{\phi}_y$

**(b)** Bias in $\hat{\beta}$

**Figure 6:** LDV with unmodeled spatial dependence ($\phi_x = 0.8$, $\rho_x = 0.1$)



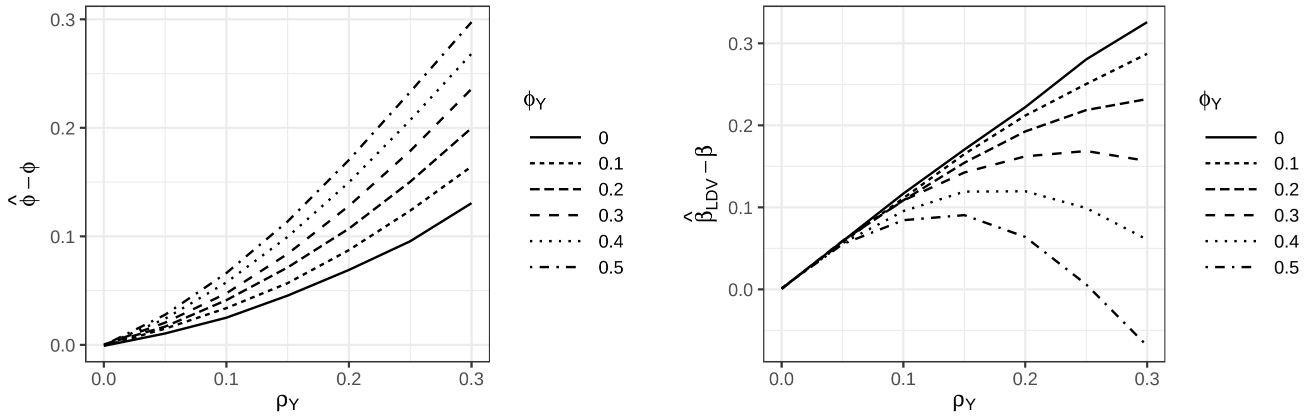
**(a)** Bias in $\hat{\rho}_y$

**(b)** Bias in $\hat{\beta}$

**Figure 7:** SAR with unmodeled temporal dependence ($\phi_x = 0.8$, $\rho_x = 0.1$)

### 2.6.2 Increased spatial dependence in *x*

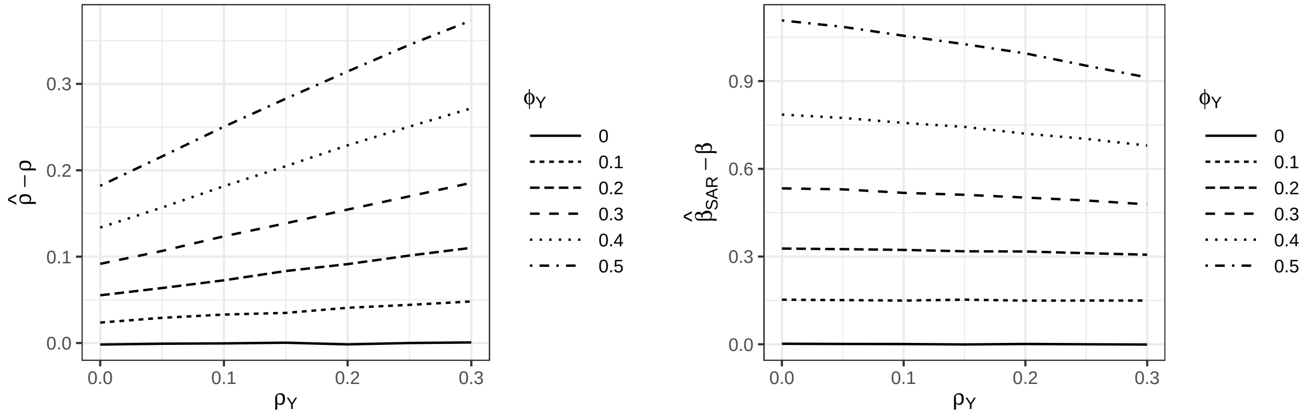
In the next set of results, we increase the spatial dependence in $x$ relative to the temporal dependence, i.e., $\rho_x = 0.6$ and $\phi_x = 0.3$. While these conditions are less likely in real-world data—as we more often observe $\phi > \rho$—it is still illustrative to see how the biases are affected under these conditions. The results are presented in Figures 8 and 9. The estimators of the dependence parameters $\phi_y$ (Figure 8a) and $\rho_y$ (Figure 9a) are similar to what we report in the main text, though slightly lesser for $\hat{\phi}_{y,LDV}$ (due to the decreased temporal dependence in $x$) and slightly greater for $\hat{\rho}_{y,SAR}$ (due to the increased spatial dependence in $x$). The results for $\beta$ reveal more interesting differences, as the bias in the estimator $\beta_{L}DV$ is now *inflationary* under many conditions (those with lower values of $\phi_y$ and $\rho_y$), again reflecting the increased spatial dependence in $x$. The SAR estimator for $\beta$ (9b) is more similar to the results given in the main text, though here we can more clearly see that the bias is a decreasing function of $\rho_y$ under some conditions (high $\phi_y$) as discussed in footnote 36.



**(a)** Bias in $\phi_y$

**(b)** Bias in $\beta$

**Figure 8:** LDV with unmodeled spatial dependence ($\rho_x = 0.6$, $\phi_x = 0.3$)



**(a)** Bias in $\rho_y$

**(b)** Bias in $\beta$

**Figure 9:** SAR with unmodeled temporal dependence ($\rho_x = 0.6$, $\phi_x = 0.3$)

# 3 tscsdep: A New R Package

Among the most significant obstacles to modeling geographical spatial dependence in the analysis of time-series-cross-sectional data, particularly with unbalanced panels, is creating the spatial weights matrix, $\mathbf{W}$. For balanced panels with fixed $N$ and $T$, this task is relatively easy. One simply pre-Kroenecker-multiplies the spatial weights matrix for a single cross-section by a $T \times T$ identity matrix:

$$\mathbf{W_{NT}} = \mathbf{I_T} \otimes \mathbf{W_N}.$$

Unfortunately, in applied work, it is much more common to have unbalanced panels where $T$ varies across the sample's units. This can arise either due to the entry and exit of units from one's sample (e.g., the recognition of South Sudan), or data missingness. Consequently, the cross-sectional weights matrices are time-period specific—in other words, there are multiple cross-sectional weights matrices associated with a sample. Moreover, because patterns of missingness vary across variables, sample dimensions vary across different regression models. As such, each regression will have a unique TSCS weights matrix. This makes accounting for spatial dependence prohibitively costly.

The package `tscsdep` was created to make it easy to account for (geographical) spatial dependence when working with TSCS data. It draws heavily from the `Cshapes` (Weidmann, Schvitz, and Girardin 2021) and `spatialreg` packages (Bivand and Piras 2015). At the moment, there are two main functions in `tscsdep`. The first, `make_ntspmat`, generates a nearest neighbor spatial weights matrix for an unbalanced TSCS sample of countries observed annually, which was used to estimate a non-spatial linear regression model. The call to execute the function is

```
wm <- make_ntspmat(lmobj,ci,yi,k)
```

In this call, `wm` stores the output, the weights matrix; `lmobj` is an object created by the `lm` function. This object contains information about the data and regression specification; `ci` and `yi` are names of variables that identify the country name and year for each observation in the sample;

and `k` is the number of nearest neighbors used for the spatial weights. The second function, `ntspreg`, re-estimates the original linear regression with a spatial lag using the TSCS weights matrix created by `make_ntspmat`. The call is

```
sar <- ntspreg(lmobj,wm)
```

In this line, `sar` stores the spatial regression output; `lmobj` is the same `lm` object used to create the TSCS spatial weights matrix, `wm`, which is also included as the second and final argument in the function. The package is available at https://github.com/judechays/STADL.

## 3.1 Political Accountability / Infant Mortality Reanalysis

In this section, we show how to use `tscsdep` to estimate Model 2 in our reanalysis of Lührmann, Marquardt, and Mechkova (2020). Our package uses `cshapes` and a nearest neighbor algorithm to create spatial weights matrices and provides a convenience wrapper for the spatial regression functions in `spatialreg`. We estimate the model in first differences using the `slide` function from the package `DataCombine` (Gandrud 2016). To illustrate an important feature of the `make_ntspmat` function, we start by (incorrectly) assuming the country names in the Lührmann, Marquardt, and Mechkova (2020) dataset match the country names in `cshapes`.

```
library (DataCombine)

data<-read.csv("accountability_data_regressions_old.csv")

# Lag the variable one time period by country
data <- slide(data = data, Var = ’infant’, GroupVar = ’country_id’,
NewVar = ’lag_inf’, slideBy = -1)
data$diff_inf <- data$infant - data$lag_inf

reg<-lm(formula = diff_inf ~ lag_inf + Accountability + aid + loggdp + gdp_grow +
    resourcesdep_hm + gini2 + lnpop + urban_cow + violence_domestic + rx_infant +
    communist + v2x_corr + as.factor(year) + as.factor(country_name), data = data)
wm <- make_ntspmat_ch(reg,country_name,year,5)
```

We read the data, estimate the non-spatial linear regression (in first differences), and then run the `make_ntspmat` function, hoping to create a nearest neighbor spatial weights matrix, using the five nearest neighbors. Unfortunately, the initial output tells us that the country names are not perfectly matched. There are 24 country names in the dataset that do not match the country names in `cshapes`.

```
================================================================================
         Data Country Name       Data Start Year COW Country Name COW Start Year
--------------------------------------------------------------------------------
1              Belarus                2005
2     Bosnia and Herzegovina          1996
3           Burkina Faso              1974
4           Burma/Myanmar             1967
5             Cambodia                1995
6  Congo, Democratic Republic of      2006
7     Congo, Republic of the          1968
8              Germany                1992
9               Iran                  1965
10             Italy                  1960
11           Ivory Coast              1966
12          Korea, South              1960
13           Kyrgyzstan               1992
14            Macedonia               1993
15           Madagascar               1965
16             Romania                1963
17             Russia                 1993
18            Sri Lanka               1960
19            Swaziland               1969
20            Tanzania                1988
21             Turkey                 1967
```

```
22         United States              1960
23             Yemen                  1992
24           Zimbabwe                 1971
--------------------------------------------------------------------------------
Error in make_ntspmat(reg, country_name, year, 5) : Some of your Country-Years are not Matched.
```

In the next chunk of code, we correct the country names to match `cshapes`, using the `recode_factor` function dplyr.

```
data$country_name<-recode_factor(data$country_name,"Burma/Myanmar"="Myanmar (Burma)")
data$country_name<-recode_factor(data$country_name,"Korea, South"="Korea, Republic of")
data$country_name<-recode_factor(data$country_name,"Russia"="Russia (Soviet Union)")
data$country_name<-recode_factor(data$country_name,"Yemen"="Yemen (Arab Republic of Yemen)")
data$country_name<-recode_factor(data$country_name,"United States"="United States of America")
data$country_name<-recode_factor(data$country_name,"Kosovo"="Yugoslavia/Serbia")
data$country_name<-recode_factor(data$country_name,"Tanzania"="Tanzania (Tanganyika)")
data$country_name<-recode_factor(data$country_name,"Burkina Faso"="Burkina Faso (Upper Volta)")
data$country_name<-recode_factor(data$country_name,"Cambodia"="Cambodia (Kampuchea)")
data$country_name<-recode_factor(data$country_name,"Zimbabwe"="Zimbabwe (Rhodesia)")
data$country_name<-recode_factor(data$country_name,"Ivory Coast"="Cote D’Ivoire")
data$country_name<-recode_factor(data$country_name,"Germany"="German Federal Republic")
data$country_name<-recode_factor(data$country_name,"Iran"="Iran (Persia)")
data$country_name<-recode_factor(data$country_name,"Italy"="Italy/Sardinia")
data$country_name<-recode_factor(data$country_name,"Turkey"="Turkey (Ottoman Empire)")
data$country_name<-recode_factor(data$country_name,"Belarus"="Belarus (Byelorussia)")
data$country_name<-recode_factor(data$country_name,"Congo, Democratic Republic of"="Congo, Democratic Republic of (Zaire)")
data$country_name<-recode_factor(data$country_name,"Congo, Republic of the"="Congo")
data$country_name<-recode_factor(data$country_name,"Kyrgyzstan"="Kyrgyz Republic")
data$country_name<-recode_factor(data$country_name,"Madagascar"="Madagascar (Malagasy)")
data$country_name<-recode_factor(data$country_name,"Sri Lanka"="Sri Lanka (Ceylon)")
data$country_name<-recode_factor(data$country_name,"Swaziland"="Swaziland (Eswatini)")
data$country_name<-recode_factor(data$country_name,"Bosnia and Herzegovina"="Bosnia-Herzegovina")
data$country_name<-recode_factor(data$country_name,"Macedonia"="Macedonia (FYROM/North Macedonia)")
data$country_name<-recode_factor(data$country_name,"Romania"="Rumania")

reg <-lm(formula = diff_inf ~ lag_inf + Accountability + aid + loggdp + gdp_grow +
    resourcesdep_hm + gini2 + lnpop + urban_cow + violence_domestic + rx_infant +
    communist + v2x_corr + as.factor(year) + as.factor(country_name), data = data)

wm <- make_ntspmat_ch(reg,country_name,year,5)
```

After the country names are matched, we re-estimate the regression and call `make_ntspmat` to create the nearest-neighbor weights matrix (using distance between capital cities to calculate distance). Now that the country names are matched, the console output changes. Instead of an error message, the year and COW country codes for the corresponding cross-section will be printed to the console. When finished, the function `make_ntspmat` returns a list. The first element of the list is the original dataset. The second element is the desired $NT \times NT$ weights matrix. The last element is a vector of unique country-year identifiers.

```
[1] 1960
 [1]  70 380 740 290 140   2 750 732  20 220  90 645 325 600 210  95 200 780 390 375 350 820 920 385
[25] 310
[1] All of your Countries are Matched.
[1] 1961
 [1] 70  380 740 290 140 2   92  135 750 732 20  220 90  645 325 600 210 95  616 200 165 780 390 375
[25] 350 820 920 385 310
[1] All of your Countries are Matched.
.
.
.
[1] 2006
 [1] 70  452 560 365 339 651 100 290 140 92  771 145 91  432 770 135 625 700 160 475 840 510 800 500
[25] 101 760 439 811 541 790 436 551 438 437 435 516 482 155 94  130 90  645 663 367 450 712 600 95
[49] 640 369 165 371 373 370 471 490 42  372 705 703 812 580 359 517 780 572 702 461 346 355 344 950
[73] 368 343 820 150 360 317 310
[1] All of your Countries are Matched.
```

In the chunk of code below, we extract the second element of this list, the nearest-neighbor weights matrix, and run the function `ntspreg`, to estimate a STADL in first differences. The function returns a list of output from the function `lagsarlm`, which is a part of the `spatialreg` package. The nearest-neighbors weights matrix is automatically row-standardized by `ntspreg`.

```
w <- as.matrix(wm[[2]])
sar <- ntspreg(reg,w)
summary(sar)
```

The results are summarized below (fixed-effects estimates omitted to save space).

```
Call:spatialreg::lagsarlm(formula = formula, data = df, listw = listw, method = "eigen", zero.policy = TRUE, tol.solve = 1e-10)

Residuals:
      Min        1Q    Median        3Q       Max
-8.708167 -0.347469  0.024157  0.350663 27.982009

Type: lag
Coefficients: (asymptotic standard errors)
                                                                Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)                                                  -14.7507422   2.0933916  -7.0463 1.837e-12
lag_inf                                                       -0.0199248   0.0016723 -11.9143 < 2.2e-16
Accountability                                                -0.1902587   0.0383825  -4.9569 7.162e-07
aid                                                            0.0155231   0.0033215   4.6735 2.961e-06
loggdp                                                         0.7972011   0.0859525   9.2749 < 2.2e-16
gdp_grow                                                      -0.0193535   0.0025472  -7.5979 3.020e-14
resourcesdep_hm                                                0.0132692   0.0023254   5.7063 1.155e-08
gini2                                                          0.0064230   0.0033563   1.9137 0.0556545
lnpop                                                          0.7061523   0.1775994   3.9761 7.006e-05
urban_cow                                                      0.0225790   0.0030610   7.3762 1.628e-13
violence_domestic                                             -0.0159651   0.0139242  -1.1466 0.2515585
rx_infant                                                      0.0088278   0.0024662   3.5795 0.0003443
communist                                                     -0.7459046   0.1751434  -4.2588 2.055e-05
v2x_corr                                                      -0.2535541   0.2057153  -1.2325 0.2177442


Rho: 0.032504, LR test value: 2.4805, p-value: 0.11527
Asymptotic standard error: 0.01982
    z-value: 1.64, p-value: 0.10101
Wald statistic: 2.6895, p-value: 0.10101
```

# References

Beck, Nathaniel, and Jonathan N Katz. 2011. “Modeling Dynamics in Time-Series-Cross-Section Political Economy Data.” *Annual Review of Political Science* 14:331–352.

Bivand, Roger, and Gianfranco Piras. 2015. “Comparing Implementations of Estimation Methods for Spatial Econometrics.” *Journal of Statistical Software* 63 (18): 1–36. https://www.jstatsoft.org/v63/i18/.

Cook, Scott, Jude Hays, and Robert Franzese. 2022. *Replication Data for: STADL Up! The Spatio-Temporal Autoregressive Distributed Lag Model for TSCS Data Analysis.* V. DRAFT VERSION. https://doi.org/10.7910/DVN/DBMJQZ. https://doi.org/10.7910/DVN/DBMJQZ.

Gandrud, Christopher. 2016. *DataCombine: Tools for Easily Combining and Cleaning Data Sets.* R package version 0.2.21. https://CRAN.R-project.org/package=DataCombine.

Lührmann, Anna, Kyle L Marquardt, and Valeriya Mechkova. 2020. “Constraining Governments: New Indices of Vertical, Horizontal, and Diagonal Accountability.” *American Political Science Review* 114 (3): 811–820.

Weidmann, Nils B., Guy Schvitz, and Luc Girardin. 2021. *cshapes: The CShapes 2.0 Dataset and Utilities.* R package version 2.0. https://CRAN.R-project.org/package=cshapes.