Statistics/Data Analysis

```
      name:  <unnamed>
       log:  C:\Users\williamslaro\Documents\Research\Projects\Spatial Methods\Robustn
> ess\Spring 2018\R&R\Results\Spatial Issue Competition--Replication.smcl
  log type:  smcl
 opened on:   5 Jun 2018, 12:36:23

1 .
2 . ********************************************************************************
3 . *** Replication
4 . ********************************************************************************
5 .
6 . use `data', clear

7 .
8 . local W contiguity

9 . local stub con

10. local rs _rs

11.
12. preserve

13.         use "Data\W\W`W'`rs'.dta", clear

14.         mkmat `stub'*, matrix(W)

15.
16.         cap spmat drop w

17.         spmat dta w `stub'*, id(id)
  warning: spatial-weighting matrix contains 7 islands

18.         drop id

19.         tempfile w

20.         save `w', replace
  (note: file C:\Users\WILLIA~1\AppData\Local\Temp\ST_175c_000007.tmp not found)
  file C:\Users\WILLIA~1\AppData\Local\Temp\ST_175c_000007.tmp saved

21.
22.         spatwmat using `w', name(w)


  The following matrix has been created:

  1. Imported non-binary weights matrix w
     Dimension: 469x469

     Beware! 7 locations have no neighbors



23.
24.         keep in 1/4
  (465 observations deleted)

25.         keep `W'493*
```

```
26.         mkmat _all, matrix(Wcont)

27. restore

28.
29. sort issue ts party

30.
31. *** Ordinary least squares (OLS) regression
32. reg perc p*perc_tm1 mip_econ_tm1 conf_ch ep_election g_election2 G G_g_election2
```

Source	SS	df	MS		
Model	**3044.9553**	**10**	**304.49553**	Number of obs	= **469**
Residual	**15571.9457**	**458**	**33.9998815**	F(10, 458)	= **8.96**
				Prob > F	= **0.0000**
Total	**18616.901**	**468**	**39.779703**	R-squared	= **0.1636**
				Adj R-squared	= **0.1453**
				Root MSE	= **5.8309**

perc	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
p1_perc_tm1	**.1112422**	**.110217**	**1.01**	**0.313**	**-.1053514**	**.3278359**
p2_perc_tm1	**.243147**	**.081536**	**2.98**	**0.003**	**.082916**	**.4033779**
p3_perc_tm1	**.2148301**	**.0710858**	**3.02**	**0.003**	**.0751353**	**.3545249**
p4_perc_tm1	**.1556196**	**.1054409**	**1.48**	**0.141**	**-.0515884**	**.3628275**
mip_econ_tm1	**.050045**	**.0217026**	**2.31**	**0.022**	**.0073959**	**.092694**
conf_ch	**-2.577794**	**1.033803**	**-2.49**	**0.013**	**-4.60938**	**-.5462075**
ep_election	**3.103663**	**2.111213**	**1.47**	**0.142**	**-1.045202**	**7.252528**
g_election2	**1.400228**	**1.476332**	**0.95**	**0.343**	**-1.500996**	**4.301451**
G	**-1.269869**	**.578833**	**-2.19**	**0.029**	**-2.407366**	**-.1323708**
G_g_election2	**6.366956**	**2.090672**	**3.05**	**0.002**	**2.258458**	**10.47545**
_cons	**-2.864709**	**1.470903**	**-1.95**	**0.052**	**-5.755265**	**.0258476**

```
33.
34. *** Spatial Durbin model (best model)
35. spreg ml perc `W'`rs'_tlsl coalition_rs_tlsl p*_perc_tm1 mip_econ_tm1 conf_ch ep_ele
  > ction g_election2 G G_g_election2, id(id) dlmat(w, eig)
  warning: spatial-weighting matrix in w contains 7 islands

  Performing a grid search.... finished

  Iteration 0:   log likelihood = -1483.3275
  Iteration 1:   log likelihood = -1482.9931
  Iteration 2:   log likelihood = -1482.9931

  Optimizing unconcentrated log likelihood

  Iteration 0:   log likelihood = -1482.9931

  Spatial autoregressive model                    Number of obs   =        469
  (Maximum likelihood estimates)                  Wald chi2(12)   =      85.7269
                                                  Prob > chi2     =      0.0000
```

perc	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
perc						
contiguity_rs~l	**-.0014915**	**.0487486**	**-0.03**	**0.976**	**-.097037**	**.094054**
coalition_rs_~l	**.1231481**	**.0708494**	**1.74**	**0.082**	**-.0157141**	**.2620103**
p1_perc_tm1	**.0362426**	**.1143788**	**0.32**	**0.751**	**-.1879357**	**.2604208**
p2_perc_tm1	**.201442**	**.0830205**	**2.43**	**0.015**	**.0387247**	**.3641592**
p3_perc_tm1	**.2089553**	**.0696053**	**3.00**	**0.003**	**.0725315**	**.345379**
p4_perc_tm1	**.1382091**	**.1039076**	**1.33**	**0.183**	**-.065446**	**.3418643**
mip_econ_tm1	**.0412172**	**.0215019**	**1.92**	**0.055**	**-.0009257**	**.0833601**
conf_ch	**-2.256142**	**1.02798**	**-2.19**	**0.028**	**-4.270946**	**-.241339**
ep_election	**2.757573**	**2.075387**	**1.33**	**0.184**	**-1.31011**	**6.825256**
g_election2	**1.209924**	**1.453047**	**0.83**	**0.405**	**-1.637996**	**4.057844**
G	**-1.239041**	**.5689266**	**-2.18**	**0.029**	**-2.354117**	**-.1239656**
G_g_election2	**5.606871**	**2.072886**	**2.70**	**0.007**	**1.544089**	**9.669653**
_cons	**-2.234203**	**1.459101**	**-1.53**	**0.126**	**-5.093988**	**.6255819**

lambda							
	_cons	**.0712139**	**.0364749**	**1.95**	**0.051**	**-.0002756**	**.1427035**
sigma2							
	_cons	**32.55781**	**2.128738**	**15.29**	**0.000**	**28.38556**	**36.73006**

```
36.
37. ********************************************************************************
38. *** Long-Term Effects for OLS
39. ********************************************************************************
40.
41. qui reg perc p*perc_tm1 mip_econ_tm1 conf_ch ep_election g_election2 G G_g_election2

42. mat b = e(b)

43. qui sum mip_econ_tm1 if e(sample)

44. local sd = r(sd)

45.
46. foreach p of numlist 1(1)4 {
    2.          di _newline(2) "Party = `p'"
    3.          di "LTE of Economic Salience = " b[1,5]/(1-b[1,`p'])
    4.                  di "1-standard deviation ("round(`sd', .01) ") LTE of Economic Sa
  > lience = " `sd' * (b[1,5]/(1-b[1,`p']))
    5.          di "LTE of Consumer Confidence = " b[1,6]/(1-b[1,`p'])
    6.
47.          di "LTE of EP Election = " b[1,7]/(1-b[1,`p'])
    7.
48.          di "LTE of General Election (Opposition) = " b[1,8]/(1-b[1,`p'])
    8.
49.          di "LTE of General Election (Government) = " (b[1,8]+b[1,10])/(1-b[1,`p'])
  >
    9.
50.          di "LTE of Government (No GE) = " b[1,9]/(1-b[1,`p'])
   10.
51.          di "LTE of Government (GE) = " (b[1,9]+b[1,10])/(1-b[1,`p'])
   11. }


  Party = 1
  LTE of Economic Salience = .0563089
  1-standard deviation (13.26) LTE of Economic Salience = .74638075
  LTE of Consumer Confidence = -2.9004457
  LTE of EP Election = 3.4921361
  LTE of General Election (Opposition) = 1.5754886
  LTE of General Election (Government) = 8.739371
  LTE of Government (No GE) = -1.4288129
  LTE of Government (GE) = 5.7350695


  Party = 2
  LTE of Economic Salience = .06612244
  1-standard deviation (13.26) LTE of Economic Salience = .87646035
  LTE of Consumer Confidence = -3.4059368
  LTE of EP Election = 4.1007473
  LTE of General Election (Opposition) = 1.8500655
  LTE of General Election (Government) = 10.262473
  LTE of Government (No GE) = -1.6778271
  LTE of Government (GE) = 6.7345802
```

```
  Party = 3
  LTE of Economic Salience = .06373776
  1-standard deviation (13.26) LTE of Economic Salience = .84485113
  LTE of Consumer Confidence = -3.2831029
  LTE of EP Election = 3.9528553
  LTE of General Election (Opposition) = 1.7833436
  LTE of General Election (Government) = 9.8923605
  LTE of Government (No GE) = -1.6173169
  LTE of Government (GE) = 6.4917


  Party = 4
  LTE of Economic Salience = .05926827
  1-standard deviation (13.26) LTE of Economic Salience = .78560759
  LTE of Consumer Confidence = -3.0528817
  LTE of EP Election = 3.6756691
  LTE of General Election (Opposition) = 1.6582901
  LTE of General Election (Government) = 9.198678
  LTE of Government (No GE) = -1.5039057
  LTE of Government (GE) = 6.0364822

52.
53.
54.
55. ******************************************************************************
56. *** Multicollinearity
57. ******************************************************************************
58.
59. qui reg contiguity_rs_tlsl p*_perc_tm1 coalition_rs_tlsl mip_econ_tm1 conf_ch ep_ele
  > ction g_election2 G G_g_election2

60. di "Variance inflation factor = " 1/(1-e(r2))
  Variance inflation factor = 1.2735718

61.
62. qui reg coalition_rs_tlsl p*_perc_tm1 contiguity_rs_tlsl mip_econ_tm1 conf_ch ep_ele
  > ction g_election2 G G_g_election2

63. di "Variance inflation factor = " 1/(1-e(r2))
  Variance inflation factor = 1.5265513

64.
65. ******************************************************************************
66. *** Secondary tests
67. ******************************************************************************
68.
69. *** Summary statistics for emphasis by party
70. bys party: sum perc, det

  ------------------------------------------------------------------------------
  -> party = 1 Greens

       Percentage of press releases on that issue at time
                                 t
  -------------------------------------------------------------
        Percentiles      Smallest
   1%    -9.178772      -14.17877
   5%    -8.005932      -9.178772
  10%    -5.204413      -9.094026       Obs                 119
  25%     -2.27401      -8.518395       Sum of Wgt.         119

  50%     .9155679                      Mean           .7097804
                         Largest        Std. Dev.      4.959207
  75%     3.339476       12.22123
  90%     5.821229       13.45281       Variance       24.59373
  95%     8.969376       16.59046       Skewness       .3864921
  99%     16.59046       17.82123       Kurtosis       4.638054

  ------------------------------------------------------------------------------
  -> party = 2 SPD
```

Percentage of press releases on that issue at time t

Percentiles		Smallest		
1%	-13.95913	-13.95913		
5%	-10.06303	-13.95913		
10%	-7.50752	-13.95913	Obs	116
25%	-4.435057	-10.6258	Sum of Wgt.	116
50%	-1.008823		Mean	.3201313
		Largest	Std. Dev.	6.867896
75%	5.128332	13.31359		
90%	9.95391	17.29087	Variance	47.16799
95%	11.68189	19.3742	Skewness	.4713823
99%	19.3742	19.3742	Kurtosis	2.997571

-> party = 3 FDP

Percentage of press releases on that issue at time t

Percentiles		Smallest		
1%	-12.62048	-12.62048		
5%	-12.62048	-12.62048		
10%	-9.763339	-12.62048	Obs	115
25%	-5.878908	-12.62048	Sum of Wgt.	115
50%	-.1204815		Mean	-.0870257
		Largest	Std. Dev.	7.739002
75%	3.817875	14.04619		
90%	7.692019	14.40655	Variance	59.89216
95%	13.69531	20.71285	Skewness	1.040942
99%	20.71285	37.37952	Kurtosis	6.668366

-> party = 4 CDU/CSU

Percentage of press releases on that issue at time t

Percentiles		Smallest		
1%	-12.8534	-12.8534		
5%	-8.628045	-12.8534		
10%	-7.477053	-10.52782	Obs	119
25%	-3.329587	-10.22182	Sum of Wgt.	119
50%	.0498282		Mean	.0245342
		Largest	Std. Dev.	5.382011
75%	3.337079	10.22353		
90%	6.843573	11.38903	Variance	28.96604
95%	8.575175	12.37183	Skewness	.1819244
99%	12.37183	18.72555	Kurtosis	3.53185

```
71.
72. *** Test the lag structure
73. reg perc mip_econ mip_econ_tm1 conf_ch ep_election g_election2 G G_g_election2
```

Source	SS	df	MS		
Model	2395.11562	7	342.159374	Number of obs	= 469
Residual	16221.7854	461	35.1882547	F(7, 461)	= 9.72
				Prob > F	= 0.0000
				R-squared	= 0.1287
				Adj R-squared	= 0.1154
Total	18616.901	468	39.779703	Root MSE	= 5.932

perc	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
mip_econ	-.0519111	.0450973	-1.15	0.250	-.1405329	.0367107
mip_econ_tm1	.1056541	.0423591	2.49	0.013	.0224132	.188895
conf_ch	-3.572111	1.059735	-3.37	0.001	-5.654621	-1.489601
ep_election	3.567366	2.14202	1.67	0.097	-.6419659	7.776699
g_election2	1.700062	1.509587	1.13	0.261	-1.266462	4.666585
G	-1.507633	.5701571	-2.64	0.008	-2.628062	-.3872045
G_g_election2	7.746471	2.057884	3.76	0.000	3.702476	11.79047
_cons	-3.032806	1.558521	-1.95	0.052	-6.095492	.0298794

```
74.
75. *** Are the phis statistically different?
76. spreg ml perc p*_perc_tm1 `W'`rs'_tlsl coalition_rs_tlsl mip_econ_tm1 conf_ch ep_ele
  > ction g_election2 G G_g_election2, id(id) dlmat(w, eig)
  warning: spatial-weighting matrix in w contains 7 islands

  Performing a grid search.... finished

  Iteration 0:   log likelihood = -1483.3275
  Iteration 1:   log likelihood = -1482.9931
  Iteration 2:   log likelihood = -1482.9931

  Optimizing unconcentrated log likelihood

  Iteration 0:   log likelihood = -1482.9931

  Spatial autoregressive model                    Number of obs   =        469
  (Maximum likelihood estimates)                  Wald chi2(12)   =    85.7269
                                                  Prob > chi2     =     0.0000
```

perc	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
perc						
p1_perc_tm1	.0362426	.1143788	0.32	0.751	-.1879357	.2604208
p2_perc_tm1	.201442	.0830205	2.43	0.015	.0387247	.3641592
p3_perc_tm1	.2089553	.0696053	3.00	0.003	.0725315	.345379
p4_perc_tm1	.1382091	.1039076	1.33	0.183	-.065446	.3418643
contiguity_rs~l	-.0014915	.0487486	-0.03	0.976	-.097037	.094054
coalition_rs_~l	.1231481	.0708494	1.74	0.082	-.0157141	.2620103
mip_econ_tm1	.0412172	.0215019	1.92	0.055	-.0009257	.0833601
conf_ch	-2.256142	1.02798	-2.19	0.028	-4.270946	-.241339
ep_election	2.757573	2.075387	1.33	0.184	-1.31011	6.825256
g_election2	1.209924	1.453047	0.83	0.405	-1.637996	4.057844
G	-1.239041	.5689266	-2.18	0.029	-2.354117	-.1239656
G_g_election2	5.606871	2.072886	2.70	0.007	1.544089	9.669653
_cons	-2.234203	1.459101	-1.53	0.126	-5.093988	.6255819
lambda						
_cons	.0712139	.0364749	1.95	0.051	-.0002756	.1427035
sigma2						
_cons	32.55781	2.128738	15.29	0.000	28.38556	36.73006

```
77.
78. test p2_perc_tm1 = p1_perc_tm1

   ( 1)  - [perc]p1_perc_tm1 + [perc]p2_perc_tm1 = 0

             chi2(  1) =    1.60
           Prob > chi2 =    0.2060
```

```
79. test p3_perc_tm1 = p1_perc_tm1

   ( 1)  - [perc]p1_perc_tm1 + [perc]p3_perc_tm1 = 0

             chi2(  1) =    1.67
           Prob > chi2 =    0.1959

80. test p2_perc_tm1 = p4_perc_tm1

   ( 1)  [perc]p2_perc_tm1 - [perc]p4_perc_tm1 = 0

             chi2(  1) =    0.24
           Prob > chi2 =    0.6256

81. test p3_perc_tm1 = p4_perc_tm1

   ( 1)  [perc]p3_perc_tm1 - [perc]p4_perc_tm1 = 0

             chi2(  1) =    0.32
           Prob > chi2 =    0.5700

82.
83. ******************************************************************************
84. ******************************************************************************
85. *** Substantive effects for spatial econometric models
86. ******************************************************************************
87. ******************************************************************************
88.
89.
90. ******************************************************************************
91. *** Illustration of all spatial-temporal effects
92. ***
93. *** German coalition: SPD + Greens
94. ******************************************************************************
95.
96. *use `data', clear
97.
98. local W contiguity

99. local stub con

100 local rs _rs

101 *local rs
102
103 preserve

104         use "Data\W\W`W'`rs'.dta", clear

105         mkmat `stub'*, matrix(W)

106
107         cap spmat drop w

108         spmat dta w `stub'*, id(id)
  warning: spatial-weighting matrix contains 7 islands

109         drop id

110         tempfile w
```

```
111         save `w', replace
  (note: file C:\Users\WILLIA~1\AppData\Local\Temp\ST_175c_000009.tmp not found)
  file C:\Users\WILLIA~1\AppData\Local\Temp\ST_175c_000009.tmp saved

112
113         spatwmat using `w', name(w)


  The following matrix has been created:

  1. Imported non-binary weights matrix w
     Dimension: 469x469

     Beware! 7 locations have no neighbors



114
115         keep in 1/4
  (465 observations deleted)

116         keep `W'493*

117         mkmat _all, matrix(Wcont)

118 restore

119
120 sort issue ts party

121
122 *** Ordinary least squares (OLS) regression
123 reg perc mip_econ_tm1 conf_ch ep_election g_election2 G G_g_election2
```

Source	SS	df	MS		
Model	2348.49089	6	391.415149	Number of obs	= 469
Residual	16268.4101	462	35.213009	F(6, 462)	= 11.12
				Prob > F	= 0.0000
Total	18616.901	468	39.779703	R-squared	= 0.1261
				Adj R-squared	= 0.1148
				Root MSE	= 5.9341

perc	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
mip_econ_tm1	.0635454	.0213636	2.97	0.003	.0215635	.1055273
conf_ch	-3.310807	1.0355	-3.20	0.001	-5.345681	-1.275933
ep_election	3.576773	2.142757	1.67	0.096	-.633985	7.787531
g_election2	1.414036	1.489519	0.95	0.343	-1.513036	4.341108
G	-1.510892	.5703506	-2.65	0.008	-2.631695	-.3900896
G_g_election2	7.74973	2.058605	3.76	0.000	3.704339	11.79512
_cons	-3.675256	1.455671	-2.52	0.012	-6.535811	-.8147001

```
124 qui sum mip_econ_tm1 if e(sample)

125 local sd = r(sd)

126
127 *** Spatial Durbin model (best model)
128 spreg ml perc p*_perc_tm1 `W'`rs'_tlsl coalition_rs_tlsl mip_econ_tm1 conf_ch ep_ele
  > ction g_election2 G G_g_election2, id(id) dlmat(w, eig)
  warning: spatial-weighting matrix in w contains 7 islands

  Performing a grid search.... finished

  Iteration 0:   log likelihood = -1483.3275
  Iteration 1:   log likelihood = -1482.9931
  Iteration 2:   log likelihood = -1482.9931

  Optimizing unconcentrated log likelihood
```

Iteration 0: log likelihood = **-1482.9931**

Spatial autoregressive model
(Maximum likelihood estimates)

Number of obs = **469**
Wald chi2(**12**) = **85.7269**
Prob > chi2 = **0.0000**

perc	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
perc						
p1_perc_tm1	.0362426	.1143788	0.32	0.751	-.1879357	.2604208
p2_perc_tm1	.201442	.0830205	2.43	0.015	.0387247	.3641592
p3_perc_tm1	.2089553	.0696053	3.00	0.003	.0725315	.345379
p4_perc_tm1	.1382091	.1039076	1.33	0.183	-.065446	.3418643
contiguity_rs~l	-.0014915	.0487486	-0.03	0.976	-.097037	.094054
coalition_rs_~l	.1231481	.0708494	1.74	0.082	-.0157141	.2620103
mip_econ_tm1	.0412172	.0215019	1.92	0.055	-.0009257	.0833601
conf_ch	-2.256142	1.02798	-2.19	0.028	-4.270946	-.241339
ep_election	2.757573	2.075387	1.33	0.184	-1.31011	6.825256
g_election2	1.209924	1.453047	0.83	0.405	-1.637996	4.057844
G	-1.239041	.5689266	-2.18	0.029	-2.354117	-.1239656
G_g_election2	5.606871	2.072886	2.70	0.007	1.544089	9.669653
_cons	-2.234203	1.459101	-1.53	0.126	-5.093988	.6255819
lambda						
_cons	.0712139	.0364749	1.95	0.051	-.0002756	.1427035
sigma2						
_cons	32.55781	2.128738	15.29	0.000	28.38556	36.73006

```
129 mat b = e(b)

130 mat V = e(V)

131
132 local W coalition

133 local stub coal

134 local rs _rs

135 *local rs
136
137 preserve

138         use "Data\W\W`W'`rs'.dta", clear

139         keep in 1/4
  (465 observations deleted)

140         keep `W'493*

141         mkmat _all, matrix(Wcoal)

142 restore

143
144 * What is the change in X?
145 scalar X = `sd'
```

```
146
147 * How many simulations?
148 local draws = 1000

149
150 * Now use -corr2data- to simulate N draws from the multivariate normal distribution
  > based on the maximum likelihood estimates.
151 set seed 8675309

152
153 clear

154 corr2data b1 - b12 alpha rho sigma, n(`draws') means(b) cov(V)
  (obs 1,000)

155
156 * Make this into an Nx2 matrix
157 mkmat _all, matrix(draws)

158
159 * Which column is rho?
160 local rho = 14

161
162 * Which columns are phi?
163 local phi1 = 1

164 local phi4 = 4

165
166 * Which column is theta?
167 *local theta = 5                                /* Contiguity TLSL */
168 local theta = 6                                 /* Coalition TLSL */

169
170 * Which parameter (beta)?
171 *local beta = 8                                 /* Change in consumer confidence */
172 local beta = 7                                  /* MIP economy t-1 */

173
174 * Local macro representing the number of observations
175 local N = rowsof(Wcont)

176
177 * Identity matrix
178 mat I = I(`N')

179
180 * A draws x 1 matrix of missing values; we will fill in these values with the direct
  > , spatial and total effects in the loop.
181 foreach t of numlist 0(1)10 {
    2.         mat tp`t'_total_draws = J(`draws',4,.)
    3.         mat tp`t'_ctotal_draws = J(`draws',4,.)
    4.         mat tp`t'_direct_draws = J(`draws',4,.)
    5.         mat tp`t'_spatial_draws = J(`draws',4,.)
    6. }

182
183 mat total_draws = J(`draws',4,.)
```

```
184
185 *** Make all the matrices accessible in mata
186 mata: I = st_matrix("I")

187 mata: Wcont = st_matrix("Wcont")

188 mata: Wcoal = st_matrix("Wcoal")

189 mata: X = st_numscalar("X")

190
191 local d = 1

192 while `d' <= `draws' {
    2.          scalar beta = draws[`d',`beta']
    3.          scalar theta = draws[`d',`theta']
    4.          scalar rho = draws[`d',`rho']
    5.          matrix phi = draws[`d',`phi1'..`phi4']
    6.
193         mata: beta = st_numscalar("beta")
    7.          mata: theta = st_numscalar("theta")
    8.          mata: rho = st_numscalar("rho")
    9.          mata: phi = st_matrix("phi")
   10.
194         mata: pd = luinv(I-rho*Wcont)*X*beta
   11.
195         mata: tp0_direct = diagonal(pd)'
   12.          mata: tp0_spatial = colsum(pd) :- tp0_direct
   13.          mata: tp0_total = tp0_direct + tp0_spatial
   14.          mata: tp0_ctotal = tp0_direct + tp0_spatial
   15.
196         mata: st_matrix("tp0_direct", tp0_direct)
   16.          mata: st_matrix("tp0_spatial", tp0_spatial)
   17.          mata: st_matrix("tp0_total", tp0_total)
   18.          mata: st_matrix("tp0_ctotal", tp0_ctotal)
   19.
197         mata: tp1_direct = phi :* tp0_total
   20.          mata: tp1_spatial = (theta*Wcoal*tp0_total')'
   21.          mata: tp1_total = tp1_direct + tp1_spatial
   22.          mata: tp1_ctotal = tp1_direct + tp1_spatial + tp0_total
   23.
198         mata: st_matrix("tp1_direct", tp1_direct)
   24.          mata: st_matrix("tp1_spatial", tp1_spatial)
   25.          mata: st_matrix("tp1_total", tp1_total)
   26.          mata: st_matrix("tp1_ctotal", tp1_ctotal)
   27.
199         mata: tp2_direct = phi :* tp1_total
   28.          mata: tp2_spatial = (theta*Wcoal*tp1_total')'
   29.          mata: tp2_total = tp2_direct + tp2_spatial
   30.          mata: tp2_ctotal = tp2_direct + tp2_spatial + tp1_ctotal
   31.
200         mata: st_matrix("tp2_direct", tp2_direct)
   32.          mata: st_matrix("tp2_spatial", tp2_spatial)
   33.          mata: st_matrix("tp2_total", tp2_total)
   34.          mata: st_matrix("tp2_ctotal", tp2_ctotal)
   35.
201         mata: tp3_direct = phi :* tp2_total
   36.          mata: tp3_spatial = (theta*Wcoal*tp2_total')'
   37.          mata: tp3_total = tp3_direct + tp3_spatial
   38.          mata: tp3_ctotal = tp3_direct + tp3_spatial + tp2_ctotal
   39.
```

```
202         mata: st_matrix("tp3_direct", tp3_direct)
  40.          mata: st_matrix("tp3_spatial", tp3_spatial)
  41.          mata: st_matrix("tp3_total", tp3_total)
  42.          mata: st_matrix("tp3_ctotal", tp3_ctotal)
  43.
203         mata: tp4_direct = phi :* tp3_total
  44.          mata: tp4_spatial = (theta*Wcoal*tp3_total')'
  45.          mata: tp4_total = tp4_direct + tp4_spatial
  46.          mata: tp4_ctotal = tp4_direct + tp4_spatial + tp3_ctotal
  47.
204         mata: st_matrix("tp4_direct", tp4_direct)
  48.          mata: st_matrix("tp4_spatial", tp4_spatial)
  49.          mata: st_matrix("tp4_total", tp4_total)
  50.          mata: st_matrix("tp4_ctotal", tp4_ctotal)
  51.
205         mata: tp5_direct = phi :* tp4_total
  52.          mata: tp5_spatial = (theta*Wcoal*tp4_total')'
  53.          mata: tp5_total = tp5_direct + tp5_spatial
  54.          mata: tp5_ctotal = tp5_direct + tp5_spatial + tp4_ctotal
  55.
206         mata: st_matrix("tp5_direct", tp5_direct)
  56.          mata: st_matrix("tp5_spatial", tp5_spatial)
  57.          mata: st_matrix("tp5_total", tp5_total)
  58.          mata: st_matrix("tp5_ctotal", tp5_ctotal)
  59.
207         mata: tp6_direct = phi :* tp5_total
  60.          mata: tp6_spatial = (theta*Wcoal*tp5_total')'
  61.          mata: tp6_total = tp6_direct + tp6_spatial
  62.          mata: tp6_ctotal = tp6_direct + tp6_spatial + tp5_ctotal
  63.
208         mata: st_matrix("tp6_direct", tp6_direct)
  64.          mata: st_matrix("tp6_spatial", tp6_spatial)
  65.          mata: st_matrix("tp6_total", tp6_total)
  66.          mata: st_matrix("tp6_ctotal", tp6_ctotal)
  67.
209         mata: tp7_direct = phi :* tp6_total
  68.          mata: tp7_spatial = (theta*Wcoal*tp6_total')'
  69.          mata: tp7_total = tp7_direct + tp7_spatial
  70.          mata: tp7_ctotal = tp7_direct + tp7_spatial + tp6_ctotal
  71.
210         mata: st_matrix("tp7_direct", tp7_direct)
  72.          mata: st_matrix("tp7_spatial", tp7_spatial)
  73.          mata: st_matrix("tp7_total", tp7_total)
  74.          mata: st_matrix("tp7_ctotal", tp7_ctotal)
  75.
211         mata: tp8_direct = phi :* tp7_total
  76.          mata: tp8_spatial = (theta*Wcoal*tp7_total')'
  77.          mata: tp8_total = tp8_direct + tp8_spatial
  78.          mata: tp8_ctotal = tp8_direct + tp8_spatial + tp7_ctotal
  79.
212         mata: st_matrix("tp8_direct", tp8_direct)
  80.          mata: st_matrix("tp8_spatial", tp8_spatial)
  81.          mata: st_matrix("tp8_total", tp8_total)
  82.          mata: st_matrix("tp8_ctotal", tp8_ctotal)
  83.
213         mata: tp9_direct = phi :* tp8_total
  84.          mata: tp9_spatial = (theta*Wcoal*tp8_total')'
  85.          mata: tp9_total = tp9_direct + tp9_spatial
  86.          mata: tp9_ctotal = tp9_direct + tp9_spatial + tp8_ctotal
  87.
```

```
214         mata: st_matrix("tp9_direct", tp9_direct)
   88.         mata: st_matrix("tp9_spatial", tp9_spatial)
   89.         mata: st_matrix("tp9_total", tp9_total)
   90.         mata: st_matrix("tp9_ctotal", tp9_ctotal)
   91.
215         mata: tp10_direct = phi :* tp9_total
   92.         mata: tp10_spatial = (theta*Wcoal*tp9_total')'
   93.         mata: tp10_total = tp10_direct + tp10_spatial
   94.         mata: tp10_ctotal = tp10_direct + tp10_spatial + tp9_ctotal
   95.
216         mata: st_matrix("tp10_direct", tp10_direct)
   96.         mata: st_matrix("tp10_spatial", tp10_spatial)
   97.         mata: st_matrix("tp10_total", tp10_total)
   98.         mata: st_matrix("tp10_ctotal", tp10_ctotal)
   99.
217         mata: total_effect = tp0_total + tp1_total + tp2_total + tp3_total + tp4_tot
  > al + tp5_total + tp6_total + tp7_total + tp8_total + tp9_total + tp10_total
  100.         mata: st_matrix("total_effect", total_effect)
  101.
218         foreach p of numlist 1(1)4 {
  102.                 foreach t of numlist 0(1)10 {
  103.                         mat tp`t'_direct_draws[`d',`p'] = tp`t'_direct[1,`p']
  104.                         mat tp`t'_spatial_draws[`d',`p'] = tp`t'_spatial[1,`p']
  105.                         mat tp`t'_ctotal_draws[`d',`p'] = tp`t'_ctotal[1,`p']
  106.                         mat tp`t'_total_draws[`d',`p'] = tp`t'_total[1,`p']
  107.                 }
  108.                 mat total_draws[`d',`p'] = total_effect[1,`p']
  109.         }
  110.
219         if mod(`d',10) == 0 {
  111.                 nois display "." _c
  112.                 if mod(`d',100) == 0 {
  113.                         nois display ""
  114.                 }
  115.         }
  116.         local d = `d' + 1
  117. }
  .........
  .........
  .........
  .........
  .........
  .........
  .........
  .........
  .........
  .........

220
221 mat effect = J(45, 20, 0)

222
223 local b = 1

224 foreach t of numlist 0(1)10 {
    2.         svmat tp`t'_direct_draws
    3.         foreach p of numlist 1(1)4 {
    4.                 local c = (`p' * 5) - 4
    5.                 local cp1 = `c' + 1
    6.                 local cp2 = `c' + 2
    7.                 local cp3 = `c' + 3
    8.                 local cp4 = `c' + 4
    9.
```

```
225                 sum tp`t'_direct_draws`p', meanonly
   10.                  mat effect[`b',`c'] = r(mean)
   11.
226                 _pctile tp`t'_direct_draws`p', p(2.5 5 95 97.5)
   12.                  mat effect[`b',`cp1'] = r(r2)
   13.                  mat effect[`b',`cp2'] = r(r3)
   14.                  mat effect[`b',`cp3'] = r(r1)
   15.                  mat effect[`b',`cp4'] = r(r4)
   16.           }
   17.           local b = `b' + 1
   18.
227           svmat tp`t'_spatial_draws
   19.           foreach p of numlist 1(1)4 {
   20.                   local c = (`p' * 5) - 4
   21.                   local cp1 = `c' + 1
   22.                   local cp2 = `c' + 2
   23.                   local cp3 = `c' + 3
   24.                   local cp4 = `c' + 4
   25.
228                 sum tp`t'_spatial_draws`p', meanonly
   26.                  mat effect[`b',`c'] = r(mean)
   27.
229                 _pctile tp`t'_spatial_draws`p', p(2.5 5 95 97.5)
   28.                  mat effect[`b',`cp1'] = r(r2)
   29.                  mat effect[`b',`cp2'] = r(r3)
   30.                  mat effect[`b',`cp3'] = r(r1)
   31.                  mat effect[`b',`cp4'] = r(r4)
   32.           }
   33.           local b = `b' + 1
   34.
230           svmat tp`t'_total_draws
   35.           foreach p of numlist 1(1)4 {
   36.                   local c = (`p' * 5) - 4
   37.                   local cp1 = `c' + 1
   38.                   local cp2 = `c' + 2
   39.                   local cp3 = `c' + 3
   40.                   local cp4 = `c' + 4
   41.
231                 sum tp`t'_total_draws`p', meanonly
   42.                  mat effect[`b',`c'] = r(mean)
   43.
232                 _pctile tp`t'_total_draws`p', p(2.5 5 95 97.5)
   44.                  mat effect[`b',`cp1'] = r(r2)
   45.                  mat effect[`b',`cp2'] = r(r3)
   46.                  mat effect[`b',`cp3'] = r(r1)
   47.                  mat effect[`b',`cp4'] = r(r4)
   48.           }
   49.           local b = `b' + 1
   50.
233           svmat tp`t'_ctotal_draws
   51.           foreach p of numlist 1(1)4 {
   52.                   local c = (`p' * 5) - 4
   53.                   local cp1 = `c' + 1
   54.                   local cp2 = `c' + 2
   55.                   local cp3 = `c' + 3
   56.                   local cp4 = `c' + 4
   57.
234                 sum tp`t'_ctotal_draws`p', meanonly
   58.                  mat effect[`b',`c'] = r(mean)
   59.
```

```
235                 _pctile tp`t'_ctotal_draws`p', p(2.5 5 95 97.5)
   60.                 mat effect[`b',`cp1'] = r(r2)
   61.                 mat effect[`b',`cp2'] = r(r3)
   62.                 mat effect[`b',`cp3'] = r(r1)
   63.                 mat effect[`b',`cp4'] = r(r4)
   64.         }
   65.         local b = `b' + 1
   66. }

236
237 svmat total_draws

238 foreach p of numlist 1(1)4 {
    2.         local c = (`p' * 5) - 4
    3.         local cp1 = `c' + 1
    4.         local cp2 = `c' + 2
    5.         local cp3 = `c' + 3
    6.         local cp4 = `c' + 4
    7.
239         sum total_draws`p', meanonly
    8.         mat effect[45,`c'] = r(mean)
    9.
240         _pctile total_draws`p', p(2.5 5 95 97.5)
   10.         mat effect[45,`cp1'] = r(r2)
   11.         mat effect[45,`cp2'] = r(r3)
   12.         mat effect[45,`cp3'] = r(r1)
   13.         mat effect[45,`cp4'] = r(r4)
   14. }

241
242 matrix rownames effect = tp0_direct tp0_spatial tp0_total tp0_ctotal tp1_direct tp1_
  > spatial tp1_total tp1_ctotal tp2_direct tp2_spatial tp2_total tp2_ctotal tp3_direct
  > tp3_spatial tp3_total tp3_ctotal tp4_direct tp4_spatial tp4_total tp4_ctotal tp5_dir
  > ect tp5_spatial tp5_total tp5_ctotal tp6_direct tp6_spatial tp6_total tp6_ctotal tp7
  > _direct tp7_spatial tp7_total tp7_ctotal tp8_direct tp8_spatial tp8_total tp8_ctotal
  >  tp9_direct tp9_spatial tp9_total tp9_ctotal tp10_direct tp10_spatial tp10_total tp1
  > 0_ctotal total

243 *matrix colnames effect = G G_lo90 G_hi90 G_lo95 G_hi95 SPD SPD_lo90 SPD_hi90 SPD_lo
  > 95 SPD_hi95 FDP FDP_lo90 FDP_hi90 FDP_lo95 FDP_hi95 CDU CDU_lo90 CDU_hi90 CDU_lo95 C
  > DU_hi95
244
245 preserve

246         clear

247
248         matrix colnames effect = effect1 lo901 hi901 lo951 hi951 effect2 lo902 hi902
  >  lo952 hi952 effect3 lo903 hi903 lo953 hi953 effect4 lo904 hi904 lo954 hi954

249         svmat effect, names(col)
  number of observations will be reset to 45
  Press any key to continue, or Break to abort
  number of observations (_N) was 0, now 45

250
251         gen str12 effecttype = ""
  (45 missing values generated)

252         local b = 1
```

```
253         qui foreach v in tp0_direct tp0_spatial tp0_total tp0_ctotal tp1_direct tp1_
  > spatial tp1_total tp1_ctotal tp2_direct tp2_spatial tp2_total tp2_ctotal tp3_direct
  > tp3_spatial tp3_total tp3_ctotal tp4_direct tp4_spatial tp4_total tp4_ctotal tp5_dir
  > ect tp5_spatial tp5_total tp5_ctotal tp6_direct tp6_spatial tp6_total tp6_ctotal tp7
  > _direct tp7_spatial tp7_total tp7_ctotal tp8_direct tp8_spatial tp8_total tp8_ctotal
  >  tp9_direct tp9_spatial tp9_total tp9_ctotal tp10_direct tp10_spatial tp10_total tp1
  > 0_ctotal total {

254
255         order effecttype

256
257         foreach p of numlist 1(1)4 {
    2.                  di _newline(2) "Party = `p' "
    3.                  foreach t of numlist 0(1)3 {
    4.                          list effectt *`p' if effecttype == "tp`t'_direct"
    5.                          list effectt *`p' if effecttype == "tp`t'_spatial"
    6.                  }
    7.
258                 list effectt *`p' if effecttype == "total"
    8.          }
```

Party = 1

	effecttype	effect1	lo901	hi901	lo951	hi951
1.	tp0_direct	.548045	.1059086	1.035595	.0271825	1.138477

	effecttype	effect1	lo901	hi901	lo951	hi951
2.	tp0_spatial	.0201592	.000333	.049847	-.001854	.0589406

	effecttype	effect1	lo901	hi901	lo951	hi951
5.	tp1_direct	.016554	-.1016289	.1338851	-.1344817	.1710894

	effecttype	effect1	lo901	hi901	lo951	hi951
6.	tp1_spatial	.0723461	-.0021458	.1769517	-.0145571	.1991124

	effecttype	effect1	lo901	hi901	lo951	hi951
9.	tp2_direct	.0088645	-.0055384	.0425522	-.007957	.0598758

	effecttype	effect1	lo901	hi901	lo951	hi951
10.	tp2_spatial	.0241278	-.0006233	.0623693	-.0036089	.0779397

	effecttype	effect1	lo901	hi901	lo951	hi951
13.	tp3_direct	.0018765	-.0053475	.0124186	-.0076786	.0201806

	effecttype	effect1	lo901	hi901	lo951	hi951
14.	tp3_spatial	.0069979	-.0001298	.0213236	-.0008555	.0261765

	effect~e	effect1	lo901	hi901	lo951	hi951
45.	total	.7039601	.1541549	1.30822	.0401815	1.467235

Party = 2

	effecttype	effect2	lo902	hi902	lo952	hi952
1.	tp0_direct	.5489031	.1060398	1.037227	.0272763	1.13924

	effecttype	effect2	lo902	hi902	lo952	hi952
2.	tp0_spatial	.059449	.0009982	.1454002	-.005536	.1726035

	effecttype	effect2	lo902	hi902	lo952	hi952
5.	tp1_direct	.1192949	.0159933	.2636255	.0012007	.316614

	effecttype	effect2	lo902	hi902	lo952	hi952
6.	tp1_spatial	.0676511	-.0019614	.1663941	-.0137815	.1896828

	effecttype	effect2	lo902	hi902	lo952	hi952
9.	tp2_direct	.0399449	.0036021	.105576	.0002918	.1270289

	effecttype	effect2	lo902	hi902	lo952	hi952
10.	tp2_spatial	.0124581	-.0018148	.0431122	-.0037426	.0521486

	effecttype	effect2	lo902	hi902	lo952	hi952
13.	tp3_direct	.0120535	.0005531	.0372646	.0000233	.0480911

	effecttype	effect2	lo902	hi902	lo952	hi952
14.	tp3_spatial	.0049659	6.07e-06	.0175182	-.0001496	.0211304

	effect~e	effect2	lo902	hi902	lo952	hi952
45.	total	.8732105	.1749373	1.647423	.043557	1.82741

Party = 3

	effecttype	effect3	lo903	hi903	lo953	hi953
1.	tp0_direct	.5489031	.1060398	1.037227	.0272763	1.13924

	effecttype	effect3	lo903	hi903	lo953	hi953
2.	tp0_spatial	.059449	.0009982	.1454002	-.005536	.1726035

	effecttype	effect3	lo903	hi903	lo953	hi953
5.	tp1_direct	.1281272	.0147597	.2853727	.0038902	.3253319

	effecttype	effect3	lo903	hi903	lo953	hi953
6.	tp1_spatial	0	0	0	0	0

	effecttype	effect3	lo903	hi903	lo953	hi953
9.	tp2_direct	.0299883	.0014414	.0834886	.000186	.1013005

	effecttype	effect3	lo903	hi903	lo953	hi953
10.	tp2_spatial	0	0	0	0	0

	effecttype	effect3	lo903	hi903	lo953	hi953
13.	tp3_direct	.0075927	.0001848	.0263609	8.57e-06	.0325354

	effecttype	effect3	lo903	hi903	lo953	hi953
14.	tp3_spatial	0	0	0	0	0

	effect~e	effect3	lo903	hi903	lo953	hi953
45.	total	.776943	.1503181	1.492778	.040286	1.642545

Party = 4

	effecttype	effect4	lo904	hi904	lo954	hi954
1.	tp0_direct	.548045	.1059086	1.035595	.0271825	1.138477

	effecttype	effect4	lo904	hi904	lo954	hi954
2.	tp0_spatial	.0201592	.000333	.049847	-.001854	.0589406

	effecttype	effect4	lo904	hi904	lo954	hi954
5.	tp1_direct	.0761249	-.0224623	.2254347	-.044933	.2630578

	effecttype	effect4	lo904	hi904	lo954	hi954
6.	tp1_spatial	0	0	0	0	0

	effecttype	effect4	lo904	hi904	lo954	hi954
9.	tp2_direct	.016285	.0000627	.0600193	4.54e-06	.0805576

	effecttype	effect4	lo904	hi904	lo954	hi954
10.	tp2_spatial	0	0	0	0	0

	effecttype	effect4	lo904	hi904	lo954	hi954
13.	tp3_direct	.0038167	-.0000507	.0173295	-.0003532	.0235385

	effecttype	effect4	lo904	hi904	lo954	hi954
14.	tp3_spatial	0	0	0	0	0

	effect~e	effect4	lo904	hi904	lo954	hi954
45.	total	.6659172	.1251625	1.289984	.0316565	1.389699

```
259
260         *** Reshape for better ggplot-ing
261         reshape long effect lo90 hi90 lo95 hi95, i(effecttype) j(party)
(note: j = 1 2 3 4)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                       45   ->     180
Number of variables                  21   ->       7
j variable (4 values)                     ->   party
xij variables:
            effect1 effect2 ... effect4   ->   effect
                  lo901 lo902 ... lo904   ->   lo90
                  hi901 hi902 ... hi904   ->   hi90
                  lo951 lo952 ... lo954   ->   lo95
                  hi951 hi952 ... hi954   ->   hi95
-----------------------------------------------------------------------------
```

```
262
263         drop if effecttype == "total"
  (4 observations deleted)

264
265         *** Generate party name
266         gen str11 partyname = "Greens" if party == 1
  (132 missing values generated)

267         replace partyname = "SPD" if party == 2
  (44 real changes made)

268         replace partyname = "FDP" if party == 3
  (44 real changes made)

269         replace partyname = "CDU" if party == 4
  (44 real changes made)

270
271         *** Generate time and effect type variables
272         split effecttype, p("tp" "_")
  variables created as string:
  effecttype1  effecttype2  effecttype3

273         drop effecttype effecttype1

274         rename effecttype2 order

275         rename effecttype3 effectname

276         destring order, force replace
  order: all characters numeric; replaced as byte

277
278         sort effectname party order

279
280         saveold "Results\Data\Spatial-Long-Term Effects.dta", replace version(12)
  (saving in Stata 12 format, which can be read by Stata 11 or 12)
  file Results\Data\Spatial-Long-Term Effects.dta saved

281 restore

282
283 log close
       name:  <unnamed>
        log:  C:\Users\williamslaro\Documents\Research\Projects\Spatial Methods\Robustn
> ess\Spring 2018\R&R\Results\Spatial Issue Competition--Replication.smcl
   log type:  smcl
  closed on:   5 Jun 2018, 12:37:02
```