

Spatial Economics – Assignment 3

Gustav Pirich (h11910449)

Gabriel Konecny (h11775903)

May 4, 2024

Contents

Exercise A	2
Estimate the demand model, and test for spatial dependence using the procedures discussed in class, and the provided weights matrix. Suppress any theoretical considerations	2
Stage 1	2
Stage 2	2
Estimate the model implied by the specification search and contrast the effect estimates with a SLX model specification:	3
What could be the rationale to use the SLX model as opposed to other spatial specifications?	4
Which type(s) of spillover(s) would you expect in the model for cigarette demand?	5
Re-run the analysis using a SLX model with a distance decay specification (between centers) for the weights matrix, i.e. $w_{ij} = \frac{1}{\gamma_{ij}}$ following Halleck Vega and Elhorst (2015). Use a distance decay parameter of $\gamma = 3$	5
Bonus	7
Exercise B	7
Unit of Observation	7
Interpretation of Coefficients	7
Replication	9
Summarize and visualize the weights matrixes and briefly explain what they imply	10
Visualization of Spillover Effects	12
Appendix	13

Loading required package: sysfonts

Loading required package: showtextdb

The code that was used in compiling the assignment is available on GitHub at
https://github.com/gustavpirich/spatial_econ/blob/main/03_assignment/03_assignmnet.Rmd.

Exercise A

Estimate the demand model, and test for spatial dependence using the procedures discussed in class, and the provided weights matrix. Suppress any theoretical considerations

Stage 1

The demand model without spatial effects can be estimated by using the package `plm`:

Table 1:	
Dependent variable:	
	logc
logp	-1.035*** (0.042)
logy	0.529*** (0.047)
State FE	X
Year FE	X
Observations	1,380
R ²	0.394
Adjusted R ²	0.359
F Statistic	424.344*** (df = 2; 1303)
Note:	*p<0.1; **p<0.05; ***p<0.01

The demand for cigarettes depends negatively on the price of the cigarettes and positively on the real disposable income.

We now test for spatial autocorrelation. Global Moran's I is highly statistically significant, presenting evidence that there is indeed spatial autocorrelation in the residuals.

```
##
## Global Moran I for regression residuals
##
## data:
## model: lm(formula = logc ~ ., data = data_big)
## weights: mat2listw(W_, style = "W")
##
## Moran I statistic standard deviate = 8.5843, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Observed Moran I      Expectation      Variance
##      0.1488187071    -0.0227212242    0.0003993162
```

Stage 2

We now run the testing procedures as we learnt in class;

```
# Spatial lag structure?
slmtest(fm1, data = data, listw = 1W, test = "lm1", model = "within", effect = "twoways")

##
## LM test for spatial lag dependence
##
## data: formula (within transformation)
## LM = 46.901, df = 1, p-value = 7.468e-12
## alternative hypothesis: spatial lag dependence
```

```

# Spatial error structure?
slmtest(fm1, data = data, listw = lW, test = "lme", model = "within", effect = "twoways")

##
## LM test for spatial error dependence
##
## data: formula (within transformation)
## LM = 54.655, df = 1, p-value = 1.437e-13
## alternative hypothesis: spatial error dependence

# Robust versions: Spatial lag allowing for spatial error?
slmtest(fm1, data = data, listw = lW, test = "rlml", model = "within", effect = "twoways")

##
## Locally robust LM test for spatial lag dependence sub spatial error
##
## data: formula (within transformation)
## LM = 1.1563, df = 1, p-value = 0.2822
## alternative hypothesis: spatial lag dependence

# Spatial error allowing for spatial lag?
slmtest(fm1, data = data, listw = lW, test = "rlme", model = "within", effect = "twoways")

##
## Locally robust LM test for spatial error dependence sub spatial lag
##
## data: formula (within transformation)
## LM = 8.9106, df = 1, p-value = 0.002835
## alternative hypothesis: spatial error dependence

```

The first two tests present evidence for the presence of spatial error dependence and spatial lag dependence. However, once we run the robust versions of the test, we find that there is no evidence for a spatial lag dependence. Thus according to the classical specification search procedure, we should estimate a **SEM model**.

Estimate the model implied by the specification search and contrast the effect estimates with a SLX model specification:

Let us proceed by using the `spml` function. We can estimate the SEM model by setting the spatial lag of dependent variable to false (`lag=FALSE`):

The SLX model can be estimated by using OLS, i.e. by using `plm()` function from before where we include the spatially lagged values of the explanatory variables:

```

slx_OLS <- function(W) {
  W_ <- kronecker(diag(nrow(unique(cigarette_2var[, "year"]))), W)
  W_logp <- W_ %*% data[, "logp"]
  W_logy <- W_ %*% data[, "logy"]
  data_slx <- cbind(data, W_logp, W_logy)
  slx <- plm(logc ~ logp + logy + W_logp + W_logy, data = data_slx, model = "within",
    effect = "twoways")
  return((slx))
}

```

Now, to compare both let's look at the regression outputs:

```

## Spatial panel fixed effects error model
##
##
## Call:
## spml(formula = fm1, data = data, listw = lW, model = "within",
##   effect = "twoways", lag = FALSE, spatial.error = "b")
##
## Residuals:

```

```
##           Min.       1st Qu.       Median       3rd Qu.       Max.
## -0.43226537 -0.03661431 -0.00098061  0.04223100  0.50712114
##
## Spatial error parameter:
##      Estimate Std. Error t-value Pr(>|t|)
## rho  0.24004   0.03267   7.3474 2.021e-13 ***
##
## Coefficients:
##      Estimate Std. Error t-value Pr(>|t|)
## logp -1.004297   0.040011 -25.101 < 2.2e-16 ***
## logy  0.553849   0.049217  11.253 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 2: SLX Model

Dependent variable:	
logc	
logp	-1.017*** (0.042)
logy	0.608*** (0.060)
W_logp	-0.220*** (0.077)
W_logy	-0.219*** (0.080)
Observations	1,380
R ²	0.400
Adjusted R ²	0.364
F Statistic	217.105*** (df = 4; 1301)
Note:	*p<0.1; **p<0.05; ***p<0.01

The coefficient estimates above correspond to those reported in table 2 of Vega and Elhorst (2015). Both the SEM and SLX suggest, on average, a decrease in domestic demand for cigarettes of around 1% following an increase in domestic prices by 1%, ceteris paribus. Similarly both models suggest an increase in demand for cigarettes given an increase in domestic income. The SLX coefficient estimate for this is higher. Additionally, the SLX model provides effects of lagged independent variables of cigarette demand. The SEM model does not provide any additional meaningful parameters since its parameter for spatial autocorrelation of errors is a *nuisance parameter*.

What could be the rationale to use the SLX model as opposed to other spatial specifications?

The choice of the model should be based on theory. If we expect spillovers of exogenous variables and spillovers of endogenous variables or correlation of error terms is expected to be less pronounced or absent, it might be a good idea to consider the SLX model. However, there are also reasons to consider SLX as a point of departure even if the theory is not clear about which model to choose. First, when using the SLX model, the estimation and interpretation of spillover effects is more straightforward. In contrast to endogenous models, SLX allows us to parameterize W and its parameters can be estimated. An undesirable property of SAR and SAC models is that the ratio between the spillover and direct effects is the same for every explanatory variable. This is not the case for SLX model, where we get flexible estimates of both by construction. Thus SLX can be considered less imposing in this perspective.

Which type(s) of spillover(s) would you expect in the model for cigarette demand?

We would expect no endogenous spillover effects, since we consider it unlikely that demand for cigarettes in a country should determine demand for cigarettes in neighboring country if there are no shortages. If however, the price of cigarettes in neighboring country is lower, people living near the border could be buying cigarettes in the neighboring country. This is called *bootlegging effect*. On other hand, if the income in our country increases, the opportunity costs of time of our citizens increase and they might then prefer to buy the more expensive cigarettes in home country instead. Thus we would expect spatially lagged exogenous variables to be relevant: An increase in price of cigarettes in neighboring country should have positive effect on the demand for cigarettes in home country. Increase in income of neighboring country should have negative effect on the demand for cigarettes in home.

Re-run the analysis using a SLX model with a distance decay specification (between centers) for the weights matrix, i.e. $w_{ij} = \frac{1}{\gamma_{ij}}$ following Halleck Vega and Elhorst (2015). Use a distance decay parameter of $\gamma = 3$

For this we first create a neighbour list, where all regions are neighbors by setting the distance threshold to big enough value and then compute distances using `nbdists`. They are then transformed as suggested above and scaled by maximum eigenvalue as in the paper.

```
coords <- cigar_states[, 2:3]

decay <- function(gamma) {
  distw <- dnearneigh(coords, 0, 999999, row.names = cigar_states$Name, longlat = FALSE)
  dists <- nbdists(distw, coords, longlat = FALSE)
  ids <- lapply(dists, function(x) lapply(x, function(y) 1/(y^gamma)))
  lWd <- nb2listw(distw, glist = ids, style = "B", zero.policy = TRUE)
  Wd1 <- listw2mat(lWd)
  Wd <- Wd1/max(abs(eigen(Wd1)$values))
  return(Wd)
}
Wd <- decay(3)
colnames(Wd) <- rownames(Wd)
```

Using gamma similar to the paper, we get coefficient estimates which are very close to those in the paper:

Table 3: Distance Decay SLX	
	Dependent variable:
	logc
logp	−0.907*** (0.038)
logy	0.652*** (0.043)
W_logp	0.259*** (0.061)
W_logy	−0.824*** (0.049)
Observations	1,380
R ²	0.514
Adjusted R ²	0.484
F Statistic	343.407*** (df = 4; 1301)
Note:	*p<0.1; **p<0.05; ***p<0.01

While the interpretation is non-lagged independent variables is similar as in regressions above, we might want to focus on spatially lagged variables. The model suggests that increase in price of cigarettes in neighboring

regions increases demand for cigarettes in home region, ceteris paribus. This is consistent with the bootlegging behavior discussed in the paper, which predicts that some people might buy cigarettes in/from other regions if the price there is lower. The coefficient estimate for income in neighboring regions suggest lower demand for cigarettes if the real income in neighboring regions increases. This suggests that if people have higher income their opportunity cost of time is higher and they might prefer the convenience of buying cigarettes in their home region.

- Describe the network recovered by Halleck Vega and Elhorst (2015):

```
distw
```

```
## Neighbour list object:
## Number of regions: 46
## Number of nonzero links: 2070
## Percentage nonzero weights: 97.82609
## Average number of links: 45
```

```
46 * 46 - 46
```

```
## [1] 2070
```

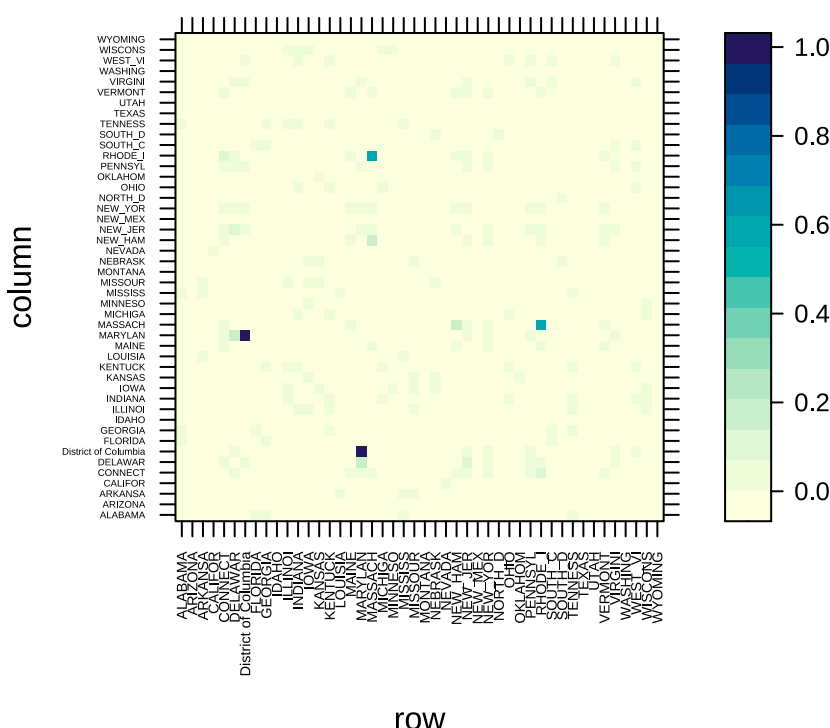
```
(46 * 46 - 46)/46 * 46
```

```
## [1] 2070
```

In this network all agents are connected, only the weights differ - which are given by the distance decay formula above. This corresponds to average number of links being 45 i.e. all regions except the region itself. Thus the percentage of nonzero weights simply gives the share of diagonal elements of a 46*46 matrix.

Below we visualize the spatial weights matrix:

Distance Decay Spatial Weights Matrix



Reminding ourselves that all regions except the diagonal are neighbors, we can see that the definition of distance decay parameter used results in few strongly connected neighbors and many weaker connections.

- Who are the central agents and where do spillover effects occur?

We could define the most central agent as one, who has the strongest connection to other regions overall. This could correspond to maximizing the nonlinear transformation of distance which we are using. Thus an agent is most central, if the sum of its weights to other agents is highest, i.e. if the sum of its row is highest.

```
##                               value
## MARYLAN                      1.2524425
## District of Columbia        1.1439427
## MASSACH                     0.7947205
## RHODE_I                     0.7545230
## DELAWAR                     0.3361666
## NEW_HAM                     0.3198986
```

Maryland is most central agent. It could be that since multiple small regions from around Maryland are included, Maryland has very small distance to them and thus its inverse is large, contributing a lot to maximization of criterion made up above. Thus the most central agent is the one who has many close neighbors. From the table above we see that other important agents are the District of Columbia, Massachusetts and Rhode Island.

We get similar results when using eigenvector centrality:

- What is the average partial effect of increasing income?

We compute APE as follows: $\beta_{logy} + \theta_{logy} \frac{1}{N} \sum_j \sum_i w_{ij}$ where I is number of rows of W and J number of columns.

```
beta <- coef(slx_OLS(Wd))["logy"]
theta <- coef(slx_OLS(Wd))["W_logy"]
N <- nrow(Wd) * ncol(Wd)
beta + theta * sum(Wd)/N
```

```
## [1] 0.6496153
```

Thus, the average partial effect of increasing income of is very close to the effect of increasing income in home only. This is because using the specification of distance decay above, the links between the agents are mostly weak.

Bonus

Exercise B

Unit of Observation

Cells	Stats
Min	9785713765 [m^2]
Mean	11698674731 [m^2]
Median	11904427037 [m^2]
Max	12364312149 [m^2]

The unit of observation are cells in a raster representing a 1×1 degree latitude longitude grid cover. At the equator this corresponds to a side length of 110 km. The areal extension of the cells varies with latitude. Further away from the equator, the area of a cell decreases. The table shows the area of the cells. The smallest cell has an area of about 9.785 km^2 , the largest 12.364 km^2 . On average the size of the cells is 11.900 km^2 . Thus the gap between the smallest and largest cell amounts to about 20%. They differ because of the distortions induced by the projection of the surface of the earth, onto a 2 dimensional raster grid.

Interpretation of Coefficients

The binary contiguity matrix is *not* row normalized. This implies that the coefficients for the spatially lagged dependent and independent variables need to be interpreted as the direct impact of marginal changes in only one of the neighboring cells.

Table 4: Eigenvector Centrality

	state	ec
MARYLAN	MARYLAN	1.0000000
District of Columbia	District of Columbia	0.9873118
DELAWAR	DELAWAR	0.2168323
PENNSYL	PENNSYL	0.1330974
VIRGINI	VIRGINI	0.0898499
NEW_JER	NEW_JER	0.0442486
RHODE_I	RHODE_I	0.0060478
MASSACH	MASSACH	0.0057002
CONNECT	CONNECT	0.0050694
NEW_YOR	NEW_YOR	0.0049031
WEST_VI	WEST_VI	0.0034141
NEW_HAM	NEW_HAM	0.0029655
VERMONT	VERMONT	0.0018586
OHIO	OHIO	0.0016525
SOUTH_C	SOUTH_C	0.0015467
MAINE	MAINE	0.0008187
KENTUCK	KENTUCK	0.0007869
MICHIGA	MICHIGA	0.0006374
GEORGIA	GEORGIA	0.0005251
INDIANA	INDIANA	0.0005125
TENNESS	TENNESS	0.0003740
FLORIDA	FLORIDA	0.0002843
ALABAMA	ALABAMA	0.0002729
ILLINOI	ILLINOI	0.0001955
WISCONS	WISCONS	0.0001770
MISSISS	MISSISS	0.0001352
MISSOUR	MISSOUR	0.0001145
ARKANSA	ARKANSA	0.0000996
LOUISIA	LOUISIA	0.0000893
IOWA	IOWA	0.0000847
MINNESO	MINNESO	0.0000807
KANSAS	KANSAS	0.0000627
NEBRASK	NEBRASK	0.0000532
OKLAHOM	OKLAHOM	0.0000445
TEXAS	TEXAS	0.0000352
SOUTH_D	SOUTH_D	0.0000295
NORTH_D	NORTH_D	0.0000258
WYOMING	WYOMING	0.0000186
NEW_MEX	NEW_MEX	0.0000162
UTAH	UTAH	0.0000096
ARIZONA	ARIZONA	0.0000090
MONTANA	MONTANA	0.0000088
IDAHO	IDAHO	0.0000066
NEVADA	NEVADA	0.0000053
CALIFOR	CALIFOR	0.0000048
WASHING	WASHING	0.0000040

Replication

Dependent Variable: Model:	ANY_EVENT_ACLED	
	(1)	(2)
<i>Variables</i>		
SPEI4pg	0.0346*** (0.0053)	0.0103 (0.0149)
L1_SPEI4pg	0.0026 (0.0049)	0.0131 (0.0135)
L2_SPEI4pg	0.0104** (0.0049)	-0.0097 (0.0140)
GSmain_ext_SPEI4pg	-0.0338*** (0.0088)	-0.0058 (0.0158)
L1_GSmain_ext_SPEI4pg	-0.0321*** (0.0081)	-0.0433*** (0.0158)
L2_GSmain_ext_SPEI4pg	-0.0348*** (0.0085)	-0.0240 (0.0156)
W_L2_GSmain_ext_SPEI4pg		-0.0026 (0.0026)
W_L1_GSmain_ext_SPEI4pg		0.0021 (0.0026)
W_GSmain_ext_SPEI4pg		-0.0056** (0.0027)
W_SPEI4pg		0.0043* (0.0023)
W_L1_SPEI4pg		-0.0017 (0.0021)
W_L2_SPEI4pg		0.0036* (0.0021)
<i>Fixed-effects</i>		
as.factor(year)	Yes	Yes
<i>Fit statistics</i>		
Observations	35,042	35,042
R ²	0.18918	0.19614
Within R ²	0.18655	0.19353
<i>Clustered (cell) standard-errors in parentheses</i>		
<i>Signif. Codes: ***: 0.01, **: 0.05, *: 0.1</i>		

Table 5: Replication Model 3

	Estimate	Std. Error	t-value	Pr(> t)
lambda	0.0458	0.0010	45.3304	0
lag(ANY_EVENT_ACLED)	0.3373	0.0050	67.4562	0
SPEI4pg	-0.0017	0.0134	-0.1264	0.8994
L1_SPEI4pg	0.0138	0.0139	0.9895	0.3224
L2_SPEI4pg	-0.0160	0.0138	-1.1537	0.2486
GSmain_ext_SPEI4pg	0.0004	0.0141	0.0265	0.9789
L1_GSmain_ext_SPEI4pg	-0.0469	0.0147	-3.1841	0.0015
L2_GSmain_ext_SPEI4pg	-0.0110	0.0148	-0.7466	0.4553
W_GSmain_ext_SPEI4pg	-0.0028	0.0024	-1.1698	0.2421
W_L1_GSmain_ext_SPEI4pg	0.0066	0.0025	2.6670	0.0077
W_L2_GSmain_ext_SPEI4pg	-0.0011	0.0025	-0.4346	0.6639
W_SPEI4pg	0.0023	0.0021	1.1129	0.2658
W_L1_SPEI4pg	-0.0028	0.0021	-1.3146	0.1886
W_L2_SPEI4pg	0.0043	0.0021	2.0831	0.0372

Next, we create an adjacency matrix based on horizontal contiguity. To that end we write a function which takes

Table 6: Replication Model 4

	Estimate	Std. Error	t-value	Pr(> t)
lambda	0.0331	0.0011	29.8567	0
lag(ANY_EVENT_ACLED)	0.3453	0.0051	67.3353	0
SPEI4pg	-0.0003	0.0135	-0.0210	0.9832
L1_SPEI4pg	0.0184	0.0140	1.3074	0.1911
L2_SPEI4pg	-0.0139	0.0139	-0.9961	0.3192
GSmain_ext_SPEI4pg	0.0004	0.0140	0.0264	0.9789
L1_GSmain_ext_SPEI4pg	-0.0516	0.0146	-3.5242	0.0004
L2_GSmain_ext_SPEI4pg	-0.0050	0.0147	-0.3429	0.7317
W_GSmain_ext_SPEI4pg	-0.0035	0.0026	-1.3479	0.1777
W_L1_GSmain_ext_SPEI4pg	0.0071	0.0027	2.6243	0.0087
W_L2_GSmain_ext_SPEI4pg	-0.0017	0.0027	-0.6222	0.5338
W_SPEI4pg	0.0012	0.0022	0.5354	0.5924
W_L1_SPEI4pg	-0.0018	0.0023	-0.7936	0.4274
W_L2_SPEI4pg	0.0050	0.0023	2.1730	0.0298

the centroids and maximum distance for the contiguity cutoff as inputs and produces a neighbors list for each cells based on horizontal contiguity. We assign a neighbors status based on two conditions. First a cell needs to have the same latitude (or longitude in the case of vertical contiguity) and the centroid of the cell needs to be closer than a pre-specified distance cutoff. We check this condition for all cells. We then rerun specification 3 and 4 based on those contiguity matrices.

We can now rerun our analysis:

Table 7: Model 3 Vertical

	Estimate	Std. Error	t-value	Pr(> t)
lambda	0.0798	0.0026	30.5563	0
lag(ANY_EVENT_ACLED)	0.3623	0.0051	71.1865	0
SPEI4pg	0.0002	0.0136	0.0173	0.9862
L1_SPEI4pg	0.0163	0.0142	1.1513	0.2496
L2_SPEI4pg	-0.0144	0.0141	-1.0231	0.3063
GSmain_ext_SPEI4pg	-0.0029	0.0144	-0.2022	0.8398
L1_GSmain_ext_SPEI4pg	-0.0438	0.0150	-2.9176	0.0035
L2_GSmain_ext_SPEI4pg	-0.0089	0.0150	-0.5943	0.5523
W_GSmain_ext_SPEI4pg	-0.0035	0.0024	-1.4550	0.1457
W_L1_GSmain_ext_SPEI4pg	0.0055	0.0025	2.1703	0.0300
W_L2_GSmain_ext_SPEI4pg	-0.0024	0.0025	-0.9324	0.3511
W_SPEI4pg	0.0028	0.0021	1.3614	0.1734
W_L1_SPEI4pg	-0.0029	0.0021	-1.3588	0.1742
W_L2_SPEI4pg	0.0047	0.0021	2.2263	0.0260

We reproduced both specification 3, and 4 for both contiguity matrices. Now that we have produced a lot of tables, what do we make of that?

The bottom line is that the results are pretty robust to different specifications of the spatial weights matrix. The main coefficient of interest (the lag of SEPI Growing Season) is statistically significant and of similar magnitude for all regressions. This is reassuring for the validity and robustness of the results. The coefficient for the (spatial) autoregressive parameter λ varies somewhat depending on the specification and spatial weights matrix being used. For Model 4 λ amounts to 0.0331, while for model 3 with the vertical spatial weights matrix λ amounts to about 0.08. Due to the amplification of those effects this implies rather large differences in the amplification of conflict events. Nonetheless overall the results are all remarkably similar.

Summarize and visualize the weights matrixes and briefly explain what they imply

In both the horizontal and the vertical contiguity matrix the average degree is about 1.86 and 2.09. Thus most cells have (as expected) two neighbors but some cells, at the edge, have only one neighbors. Moreover, some

Table 8: Model 3 Horizontal

	Estimate	Std. Error	t-value	Pr(> t)
lambda	0.0630	0.0024	26.2314	0
lag(ANY_EVENT_ACLED)	0.3677	0.0051	71.8583	0
SPEI4pg	-0.0007	0.0137	-0.0523	0.9583
L1_SPEI4pg	0.0138	0.0142	0.9698	0.3321
L2_SPEI4pg	-0.0156	0.0141	-1.1054	0.2690
GSmmain_ext_SPEI4pg	-0.0029	0.0144	-0.1985	0.8426
L1_GSmmain_ext_SPEI4pg	-0.0447	0.0151	-2.9632	0.0030
L2_GSmmain_ext_SPEI4pg	-0.0109	0.0151	-0.7224	0.4701
W_GSmmain_ext_SPEI4pg	-0.0037	0.0024	-1.5031	0.1328
W_L1_GSmmain_ext_SPEI4pg	0.0055	0.0025	2.1782	0.0294
W_L2_GSmmain_ext_SPEI4pg	-0.0024	0.0025	-0.9410	0.3467
W_SPEI4pg	0.0031	0.0021	1.4677	0.1422
W_L1_SPEI4pg	-0.0026	0.0022	-1.1924	0.2331
W_L2_SPEI4pg	0.0050	0.0021	2.3475	0.0189

Table 9: Model 4 Vertical

	Estimate	Std. Error	t-value	Pr(> t)
lambda	0.0556	0.0027	20.9278	0
lag(ANY_EVENT_ACLED)	0.3597	0.0052	69.5995	0
SPEI4pg	0.0008	0.0136	0.0559	0.9555
L1_SPEI4pg	0.0209	0.0142	1.4760	0.1400
L2_SPEI4pg	-0.0124	0.0141	-0.8850	0.3762
GSmmain_ext_SPEI4pg	-0.0019	0.0141	-0.1361	0.8918
L1_GSmmain_ext_SPEI4pg	-0.0494	0.0148	-3.3487	0.0008
L2_GSmmain_ext_SPEI4pg	-0.0031	0.0148	-0.2123	0.8319
W_GSmmain_ext_SPEI4pg	-0.0040	0.0026	-1.5148	0.1298
W_L1_GSmmain_ext_SPEI4pg	0.0062	0.0027	2.2667	0.0234
W_L2_GSmmain_ext_SPEI4pg	-0.0023	0.0027	-0.8605	0.3895
W_SPEI4pg	0.0016	0.0022	0.7207	0.4711
W_L1_SPEI4pg	-0.0019	0.0023	-0.8059	0.4203
W_L2_SPEI4pg	0.0051	0.0023	2.2213	0.0263

Table 10: Model 4 Horizontal

	Estimate	Std. Error	t-value	Pr(> t)
lambda	0.0398	0.0025	16.0918	0
lag(ANY_EVENT_ACLED)	0.3642	0.0052	70.1912	0
SPEI4pg	0.0003	0.0137	0.0188	0.9850
L1_SPEI4pg	0.0192	0.0142	1.3488	0.1774
L2_SPEI4pg	-0.0131	0.0141	-0.9300	0.3524
GSmmain_ext_SPEI4pg	-0.0018	0.0142	-0.1280	0.8981
L1_GSmmain_ext_SPEI4pg	-0.0501	0.0148	-3.3829	0.0007
L2_GSmmain_ext_SPEI4pg	-0.0041	0.0148	-0.2790	0.7803
W_GSmmain_ext_SPEI4pg	-0.0042	0.0026	-1.5973	0.1102
W_L1_GSmmain_ext_SPEI4pg	0.0061	0.0027	2.2368	0.0253
W_L2_GSmmain_ext_SPEI4pg	-0.0024	0.0027	-0.8777	0.3801
W_SPEI4pg	0.0017	0.0023	0.7706	0.4409
W_L1_SPEI4pg	-0.0015	0.0023	-0.6518	0.5146
W_L2_SPEI4pg	0.0054	0.0023	2.3226	0.0202

cells even have more than two neighbors. This is because the condition that any cells is being assigned neighborhood status who's centroid is less than 180 kilometers away can include more than one neighbor. This implies that the networks are sparser and less dense than the adjacency matrix used by the authors. This is also reflected in lower 'graph density' for the horizontal and vertical contiguity matrix as compared to the baseline adjacency matrix. Visualizing the spatial weight matrices further corroborates the sparser structure.

More generally, the horizontal/vertical contiguity matrix implies that only cells which share the same latitude/longitude and are directly adjacent are considered to be affecting each other directly. It is hard to make a coherent case for why this should be the case in the real world. There is no inherent reason to assume that effects only propagate in one 'direction'.

Still due to the nature of dynamic multipliers in the form of the spatial autoregressive terms, even if the network appears relatively sparse and two cells are not directly linked the cells are still connected. However, the restriction of horizontal and vertical contiguity significantly restricts the dynamic propagation of the effects in this context.

Visualization of Spillover Effects

For this exercise we simulate the effect of a one-standard deviation SEPI growing season shock holding everything else constant. We multiply the shock times -1 to obtain the negative impact and thus the *increase* in conflict incidence. We exogenously shock a cell and calculate 'manually', based on the coefficients of model 4, the magnitude of the shock propagation. We take into account both the dynamic, and spatial autoregressive structure of the model. We see that the shock reaches its peak effect size at $t = 1$, consistent with the main insight of the paper (the lagged effect of growing season shocks on conflict). A negative one standard deviation growing season shock increases conflict incidence by about 2.1 percentage points for the affected cell. Overall the spillover effects both in absolute magnitude and visually as reflected in the figure are rather small. The maximum conflict increase in a neighboring cell only amounts to about 0.5 percentage points.

COMPUTATION OF SHOCKS

```
# first number denotes period second number denotes neighbor shock in period 0
# of standard deviation in growing season SEPI (without contemporaneously
# affecting SEPI over the year)
risk_cell_0_0 <- -1 * mod4$coefficients["GSmain_ext_SPEI4pg"] *
sd(geoconflict_main_weights$GSmain_ext_SPEI4pg)/(1 -
  mod4$arcoef)

risk_cell_0_1 <- -1 * mod4$coefficients["W_GSmain_ext_SPEI4pg"] *
sd(geoconflict_main_weights$GSmain_ext_SPEI4pg) +
  risk_cell_0_0 * mod4$arcoef

risk_cell_0_2 <- -1 * mod4$coefficients["W_GSmain_ext_SPEI4pg"] *
mod4$coefficients["W_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg) + mod4$arcoef * risk_cell_0_1 +
  mod4$arcoef * mod4$arcoef * risk_cell_0_0

risk_cell_1_0 <- (risk_cell_0_0 * mod4$coefficients["lag(ANY_EVENT_ACLED)"] + -1 *
  mod4$coefficients["L1_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg))/(1 -
  mod4$arcoef)

risk_cell_1_1 <- risk_cell_0_1 * mod4$coefficients["lag(ANY_EVENT_ACLED)"] + -1 *
  mod4$coefficients["W_L1_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg) +
  risk_cell_1_0 * mod4$arcoef

risk_cell_1_2 <- risk_cell_0_2 * mod4$coefficients["lag(ANY_EVENT_ACLED)"] + -1 *
  mod4$coefficients["W_L1_GSmain_ext_SPEI4pg"] *
  mod4$coefficients["W_L1_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg) + mod4$arcoef * risk_cell_1_1 +
  mod4$arcoef * mod4$arcoef * risk_cell_1_0
```

```

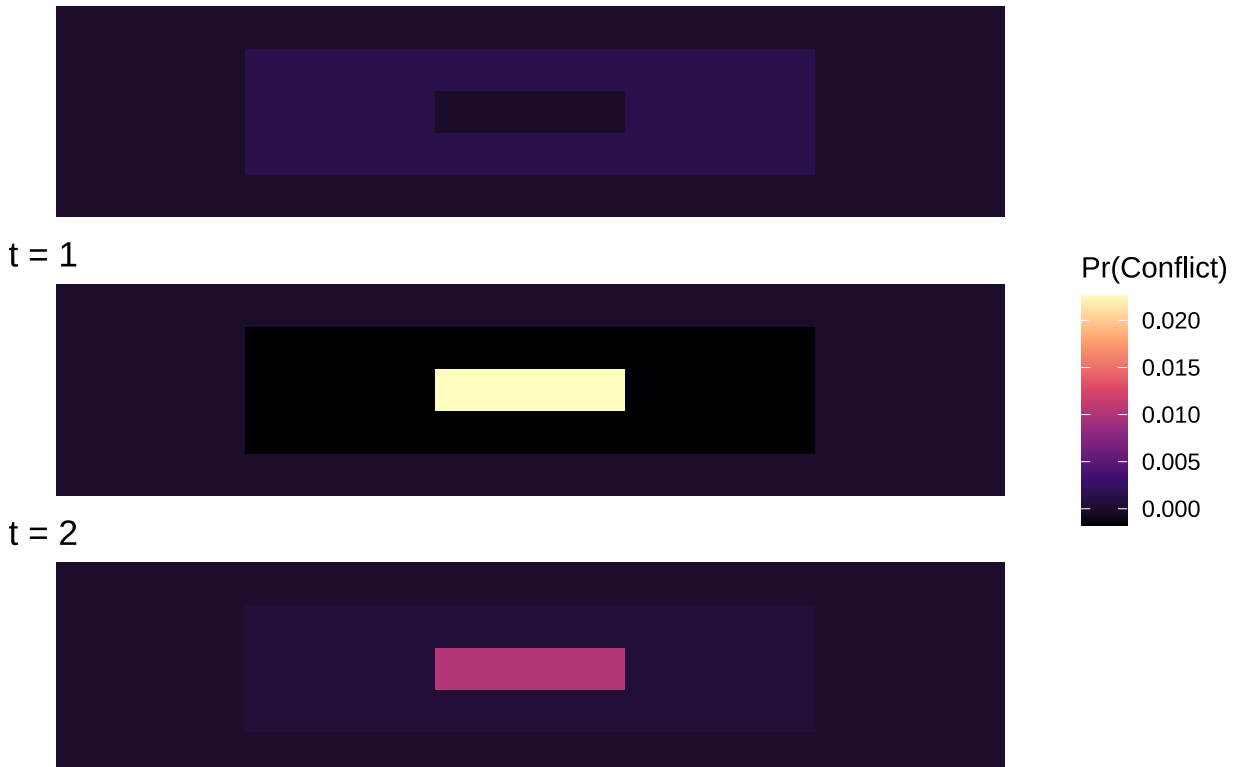
risk_cell_2_0 <- (risk_cell_1_0 * mod4$coefficients["lag(ANY_EVENT_ACLED)"] + risk_cell_0_0
*
  mod4$coefficients["lag(ANY_EVENT_ACLED)"] * mod4$coefficients["lag(ANY_EVENT_ACLED)"] +
-1 * mod4$coefficients["L2_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg))/(1 -
  mod4$arcoef)

risk_cell_2_1 <- risk_cell_1_1 * mod4$coefficients["lag(ANY_EVENT_ACLED)"] + risk_cell_0_1 *
  mod4$coefficients["lag(ANY_EVENT_ACLED)"] * mod4$coefficients["lag(ANY_EVENT_ACLED)"] +
-1 * mod4$coefficients["W_L2_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg) +
  risk_cell_2_0 * mod4$arcoef

risk_cell_2_2 <- -1 * mod4$coefficients["W_L2_GSmain_ext_SPEI4pg"] *
  mod4$coefficients["W_L2_GSmain_ext_SPEI4pg"] *
  sd(geoconflict_main_weights$GSmain_ext_SPEI4pg) + mod4$arcoef * risk_cell_2_1 +
  mod4$arcoef * mod4$arcoef * risk_cell_2_0 + risk_cell_2_1 * mod4$arcoef + risk_cell_2_0
*
  mod4$arcoef * mod4$arcoef

```

SEPI Growing Season Shock at t = 0



Appendix

Functions to create horizontal and vertical contiguity neighbors list

```

### Lets first create teh bianry contiguity matrix based on horizontal
### contiguity Extract centroids of the polygons if not already point data

# Calculate distances and filter horizontally contiguous points Define a
# function to find neighbors based on conditions
find_neighbors_horizontal <- function(centroids, max_distance = 150000, lat_tolerance =
0.01) {
  # Create a matrix to store distances
  distances <- st_distance(centroids)

```

```

latitudes <- st_coordinates(centroids)[, 2] # Extract latitudes

# Initialize neighbors list
neighbors_list <- vector("list", nrow(centroids))

for (i in seq_len(nrow(centroids))) {
  # Find points within the same latitude band and within distance
  lat_condition <- abs(latitudes - latitudes[i]) < lat_tolerance
  dist_condition <- distances[i, ] < set_units(max_distance, "meters")

  # Combine conditions
  neighbor_ids <- which(lat_condition & dist_condition)

  # Exclude the point itself from its list of neighbors
  neighbor_ids <- neighbor_ids[neighbor_ids != i]

  # Store neighbor ids
  neighbors_list[[i]] <- as.integer(neighbor_ids)
}

return(neighbors_list)
}

find_neighbors_vertical <- function(centroids, max_distance = 150000, lon_tolerance = 0.01)
{
  # Create a matrix to store distances
  distances <- st_distance(centroids)
  longitudes <- st_coordinates(centroids)[, 1] # Extract longitudes

  # Initialize neighbors list
  neighbors_list <- vector("list", nrow(centroids))

  for (i in seq_len(nrow(centroids))) {
    # Find points within the same longitude band and within distance
    lon_condition <- abs(longitudes - longitudes[i]) < lon_tolerance
    dist_condition <- distances[i, ] < set_units(max_distance, "meters")

    # Combine conditions
    neighbor_ids <- which(lon_condition & dist_condition)

    # Exclude the point itself from its list of neighbors
    neighbor_ids <- neighbor_ids[neighbor_ids != i]

    # Store neighbor ids
    neighbors_list[[i]] <- as.integer(neighbor_ids)
  }

  return(neighbors_list)
}

# Apply the function
neighbors_vertical <- find_neighbors_vertical(geoconflict_main_weights)

neighbors_horizontal <- find_neighbors_horizontal(geoconflict_main_weights)

# Number of elements
n <- length(neighbors_vertical)

```

```

# Initialize an adjacency matrix with 0s
adj_matrix_vertical <- matrix(0, nrow = n, ncol = n)
adj_matrix_horizontal <- matrix(0, nrow = n, ncol = n)

# Populate the adjacency matrix
for (i in seq_len(n)) {
  # Ensure indices are within the valid range
  valid_indices_vertical <- neighbors_vertical[[i]][neighbors_vertical[[i]] <=
    n]
  valid_indices_horizontal <- neighbors_horizontal[[i]][neighbors_horizontal[[i]] <=
    n]

  # Set matrix elements to 1
  adj_matrix_vertical[i, valid_indices_vertical] <- 1
  adj_matrix_horizontal[i, valid_indices_horizontal] <- 1
}

# Turn the matrices into listw objects
vertical_listw <- mat2listw(adj_matrix_vertical, style = "B", zero.policy = TRUE)
horizontal_listw <- mat2listw(adj_matrix_horizontal, style = "B", zero.policy = TRUE)

```