Spatial Economics – Assignment 3

Elia Di Gregorio (h12039313@s.wu.ac.at)   Max Heinze (h11742049@s.wu.ac.at)
Sophia Ludescher (h11801892@s.wu.ac.at)

May 10, 2024

Contents

> *The executable code that was used in compiling the assignment is available on GitHub at*
> *https://github.com/maxmheinze/spatial.*

Task A

Naïve Panel Model and Specification Search

We begin by fitting the spatially naïve panel model using the `plm` function. The output is printed below.

```
cm1 <- plm(logc ~ logp + logy, data = cigs, effect = "twoway", model = "within",
    index = c("state", "year"))
```

	Dependent variable:
	logc
logp	−1.035***
	(0.042)
logy	0.529***
	(0.047)
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

As the true spatial econometricians we are, we now ignore everything that is reminiscent of economic theory and venture on a standard specification testing path. We begin by performing a Lagrange Multiplier test for spatial lags in both the error and the dependent. The results are printed below.

```
slmtest(cm1, cigm, test = "lme")
```

```
##
##  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
```

```
slmtest(cm1, cigm, test = "lml")
```

```
##
##  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
```

As we can see, both tests return significant p-values, meaning that we reject the null hypotheses of spatial dependence being present in the errors or the dependent, respectively. We thus continue by performing robust LM tests for spatial dependence in both the errors and the dependent which account for the respectively other type of spatial dependence potentially being present. The results of these two tests are printed below.

```
slmtest(cm1, cigm, test = "rlme")
```

```
##
##  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
```

```
slmtest(cm1, cigm, test = "rlml")
```

```
##
##  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
```

Since this time, testing for spatial dependence in the error given spatial dependence in the dependent returns a significant result, and testing for spatial dependence in the dependent given spatial dependence in the errors does not, we conclude that based on these test results, a **SEM model** is the most appropriate specification.

Estimating a SEM Model and Comparing it to an SLX Model

We use the `spml` function from the `splm` package to estimate a spatial panel model using Maximum Likelihood. Consistent with our previous answer, we estimate a **SEM model**, and we do this by setting `lag = FALSE` as well as `spatial.error = "b"`. Written down, the model equation is identical to the one presented in the assignment question, except for the important distinction that

$$\boldsymbol{\varepsilon} = \rho \boldsymbol{W}\boldsymbol{\varepsilon} + \boldsymbol{u}, \quad \boldsymbol{u} \sim \mathrm{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}).$$

```
cm2 <- spml(logc ~ logp + logy, data = cigs, listw = cigm, effect = "twoways", model =
"within",
    index = c("state", "year"), lag = FALSE, spatial.error = "b")
```

The results of this estimation are printed in the following.

	Dependent variable:
	logc
logp	−1.004***
	(0.040)
logy	0.554***
	(0.049)
Spatial error rho	0.240***
	(0.033)
Observations	1,380
Note:	*p<0.1; **p<0.05; ***p<0.01

Next, we estimate an **SLX model** using the `plm` function in combination with the `slag` function to create spatial lags. Since there neither is a lag of the dependent nor are there spatial errors, there is no need to use a function for a dedicatedly spatial panel model. The model we estimate below amounts to

$$\boldsymbol{y}_t = \boldsymbol{X}_t \boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{X}_t \boldsymbol{\gamma} + \boldsymbol{\mu} + \phi_t \boldsymbol{\iota} + \boldsymbol{\varepsilon}_t,$$

where, diverting from the original notation for ease of reading, $\boldsymbol{y}_t$ is a $n \times 1$ vector of stacked $\log(C_{it})$, $\boldsymbol{X}_t$ is a $n \times 2$ matrix consisting of stacked $\log(P_{it})$ and $\log(I_{it})$ as columns, $\boldsymbol{W}$ is the provided weights matrix, $\boldsymbol{\beta} = (\beta_1, \beta_2)'$, $\boldsymbol{\gamma} = (\gamma_1, \gamma_2)'$, and both $\boldsymbol{\mu}$ and $\boldsymbol{\varepsilon}$ are stacked over individuals and thus $n \times 1$ vectors.

```
cm3 <- plm(logc ~ logp + logy + slag(logp, listw = cigm) + slag(logy, listw = cigm),
    data = cigs, effect = "twoways", model = "within", index = c("state", "year"))
```

Again, the results are printed below.

	Dependent variable:
	logc
logp	−1.017***
	(0.042)
logy	0.608***
	(0.060)
slag(logp, listw = cigm)	−0.220***
	(0.077)
slag(logy, listw = cigm)	−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

We can see that there is positive spatial autocorrelation of errors in the SEM model, and a negative spillover effect of price changes in the SLX model. The **positive spatial autocorrelation of errors** can be interpreted as that states that are contiguous are likely to experience similar shocks. This can be explained as spillover effects of an unobserved variable affecting cigarette demand; however, it is not possible to deduce information about the spillover effect of price changes, one of the explanatory variables in the model, from this error autocorrelation.

In contrast, the SLX model allows for directly examining and interpreting spillover effects, which could be the rationale for choosing it over the SEM (or a SAR) model. The coefficient we receive for **the effect of adjacent states' price changes**, γ_1, is $-\$0.220. This can directly be interpreted as that given a one-unit increase in log price of cigarettes, log demand in adjacent states decreases by an average of 0.22 (or, if you want to be imprecise, the price of cigarettes increasing by one percent is associated with demand in adjacent states decreasing by an average of 0.22 percent), ceteris paribus.

Of course, this is really weird, or should at least strike us as such if we hold any belief in our original bootlegging hypothesis of people traveling to other states if cigarettes are cheaper there. This is because the above mentioned result indicates the exact opposite, i.e., if the price of cigarettes in State A increases, *less* people will demand cigarettes in neighboring State B, where the price has *not* increased.

Distance Decay SLX

Since we know that results depend quite strongly on our choice of W, it makes sense that Halleck Vega and Elhorst (2015) try a different W and we are asked to follow after them. So, in the following, we estimate an **SLX model with a distance decay specification**, where $w_{ij} = d_{ij}^{-\gamma}$ and $\gamma = 3$.

```
cigd <- read_excel("./assignment3/data/cigarettes/cigar_states.xls") %>%
    select(longitude, latitude) %>%
    as.matrix() %>%
    `rownames<-`(cign)


cigi <- distm(cigd, fun = distVincentyEllipsoid)


cigj <- (cigi/1e+06)^(-3) %>%
    `diag<-`(0) %>%
    mat2listw(style = "B")


cm4 <- plm(
  logc ~ logp + logy + slag(logp, listw = cigj) + slag(logy, listw = cigj), #SLX w/ distance
decay
  data = cigs,
  effect = "twoways",
```

```
  model = "within",
  index = c("state", "year")
)
```

Again, the model results are printed below.

|	*Dependent variable:*
	logc
logp	−0.900***
	(0.038)
logy	0.642***
	(0.042)
slag(logp, listw = cigj)	0.00004***
	(0.00001)
slag(logy, listw = cigj)	−0.0001***
	(0.00001)
Observations	1,380
R^2	0.520
Adjusted R^2	0.491
F Statistic	351.699*** (df = 4; 1301)
Note:	*p<0.1; **p<0.05; ***p<0.01

We can see that this time, we actually get the suspected bootlegging effect, as γ_1 is positive. This may indicate that ~~you get your desired results if you just try a sufficiently large number of specifications~~ the distance decay matrix fits the true DGP better.

From a network perspective, this distance decay matrix has interesting implications. Our idea was that if there is a large number of edges, and these edges are weighted by the inverse of a power of the distance, then plotting these network must yield something that vaguely looks like the U.S. And it actually did:

```
plot(cigg, vertex.size = 10, vertex.color = "#BBBBBB", vertex.label.cex = 0.7,
vertex.label.font = 2,
    vertex.label.family = "Lato", vertex.label.color = "black", edge.color = "black",
    edge.width = 0.5, asp = 0)
```
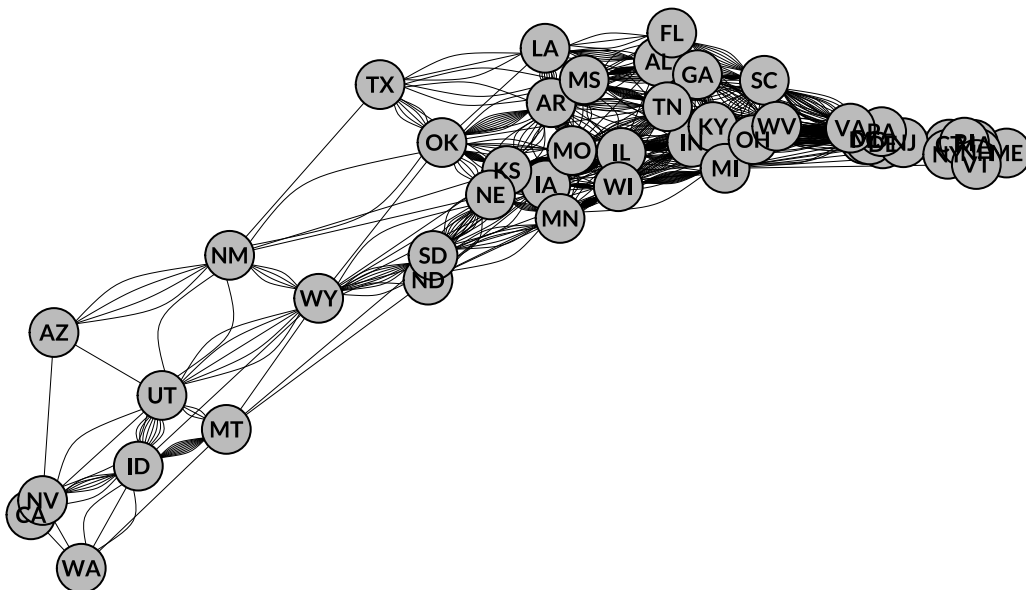

Since the plot is constructed anew every time we compile the document, we do not know how exactly it looks in our final submission. However, it should always resemble the true spatial position of the states recognizably—it

may be flipped, it may be mirrored, it may not have north at the top, but the result should resemble the continental United States.

Then we thought about printing the weights matrix, only to find out that this is all but impossible on an A4 sheet of paper. So instead, here is a heatmap of the matrix (with logged values so that the scale appears more clearly):

Spatial Weights Matrix Heatmap
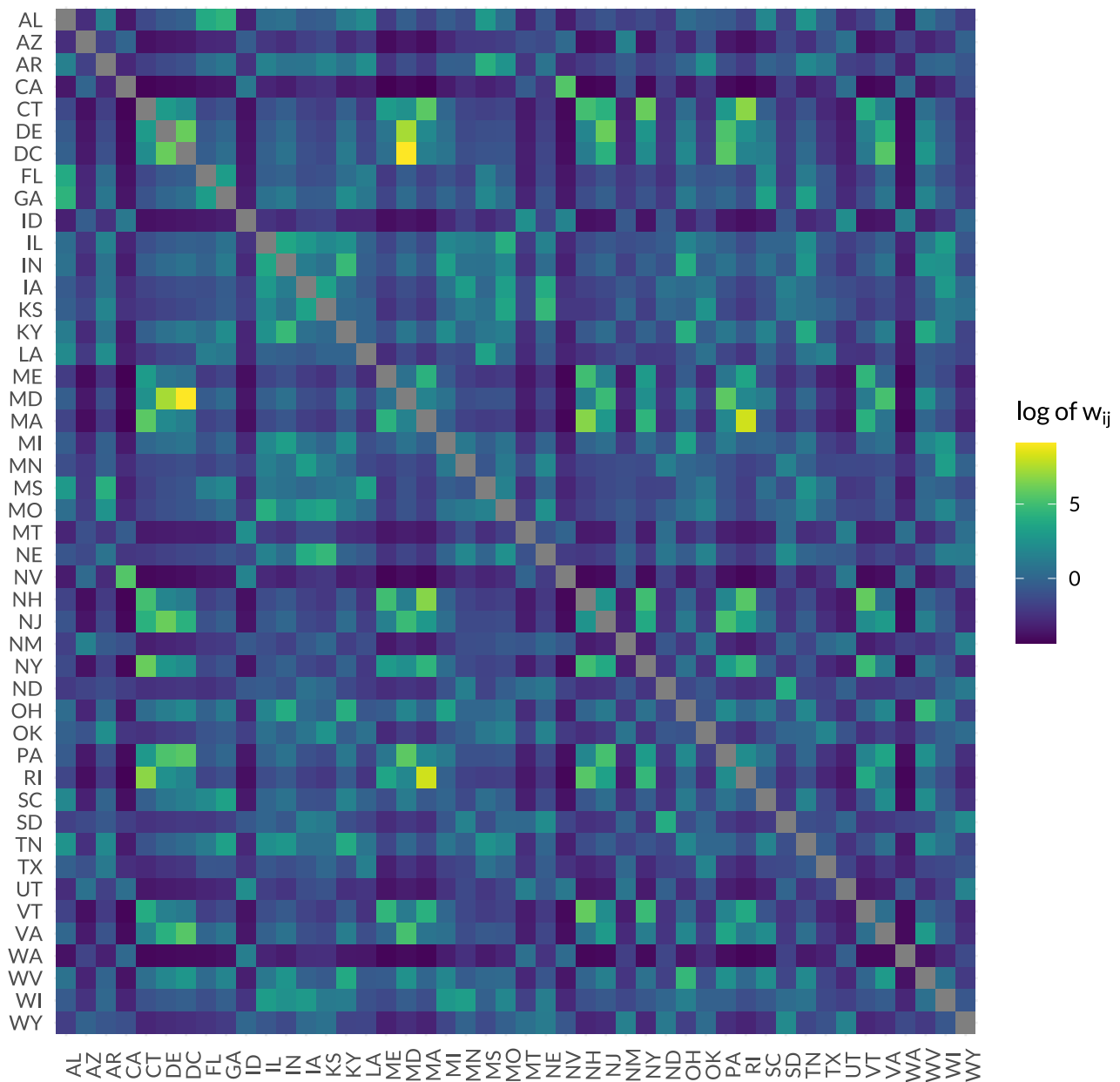
Regarding centrality, we calculated two well-known measures: row sums (since our W is not row-stochastic) and eigenvector centrality. The results are printed below.

```r
rowSums(cigk) %>%
    sort(decreasing = TRUE) %>%
    knitr::kable(col.names = "Row Sum")
```

	Row Sum
MD	11026.67198
DC	9945.46534
RI	4756.35048
MA	4728.87454
DE	2663.12139
CT	1913.64955

	Row Sum
NH	1878.45853
PA	1082.00099
NJ	983.52610
NY	968.47870
VT	746.96390
VA	615.20596
ME	386.75814
KY	333.66481
OH	308.21679
IN	303.99804
WV	271.93286
NV	253.46415
CA	248.41830
KS	215.91668
NE	209.03832
IL	207.81238
GA	201.84045
AL	193.97210
MO	191.43001
IA	185.93943
TN	176.79008
MS	160.89378
AR	140.08363
WI	127.73114
MI	123.99357
SC	100.11099
FL	98.10254
SD	90.44541
MN	85.93133
ND	71.63010
LA	70.08663
OK	60.44297
WY	38.15529
ID	35.44483
UT	32.25951
TX	25.43502
MT	25.15952
NM	24.47703
AZ	14.19491
WA	10.75984

	Eigenvector Centrality
MD	1.0000000
DC	0.9871728
DE	0.2112788
PA	0.0682453
VA	0.0486044
NJ	0.0310696
CT	0.0033553
WV	0.0030710
RI	0.0028691
NY	0.0025882
MA	0.0023642
OH	0.0013980
NH	0.0013860
VT	0.0008475
SC	0.0007672

	Eigenvector Centrality
KY	0.0006332
IN	0.0003951
MI	0.0003824
ME	0.0003685
GA	0.0002656
TN	0.0002394
AL	0.0000095
IL	0.0000043
WI	0.0000021
FL	0.0000017
MO	0.0000013
MS	0.0000012
IA	0.0000005
AR	0.0000005
MN	0.0000003
KS	0.0000002
LA	0.0000001
NE	0.0000001
OK	0.0000000
SD	0.0000000
TX	0.0000000
ND	0.0000000
WY	0.0000000
NM	0.0000000
MT	0.0000000
UT	0.0000000
AZ	0.0000000
ID	0.0000000
NV	0.0000000
CA	0.0000000
WA	0.0000000

Using both measures of centrality, Maryland is the most central state, followed by DC as the second-most central entity. The then following positions differ a bit between the two measures of centrality.

Regarding the question where spillover effects occur, let us take another look at the model equation of the SLX model:

$$\boldsymbol{y}_t = \boldsymbol{X}_t\boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{X}_t\boldsymbol{\gamma} + \boldsymbol{\mu} + \phi_t\boldsymbol{\iota} + \boldsymbol{\varepsilon}_t,$$

The **spillover effect** we are interested in is described by γ_1, which in the equation is pre-multiplied by the first column of $\boldsymbol{W}\boldsymbol{X}_t$. In a sense, the first column of $\boldsymbol{W}\boldsymbol{X}_t$ determines where, or how strongly, spillover effects arise. Given a one-unit increase in every state's $\log(P_{it})$, this means that spillover effects of state i on state j will arise where w_{ij} is large. For an illustration, you can thus refer to the heatmap above.

The **average partial effect** is the effect that a (say, one-unit) change in an explanatory variable exerts on the dependent, averaged across all units. In a normal regression setting without spatial lags, this would be represented by the coefficient β, which is constant over all i. However, in this case, changing the log income vector by one unit for all states i would exert influence over the demand for cigarettes via two channels: the "direct" channel, represented by the parameter β_2, and the "indirect" channel, represented by γ_2 times the i-th row of $\boldsymbol{W}$. The partial effect of a one-unit change in $\log(I_{it})$ for a certain i is thus

$$\mathrm{PE}_i = \beta_2 + \gamma_2 \sum_j w_{ij},$$

and the average partial effect of a one-unit change in $\log(I_{it})$ is then

$$\mathrm{APE} = \beta_2 + \gamma_2 n^{-1} \sum_i \sum_j w_{ij}.$$

Bonus question

First of all, it should be noted that the bonus question largely refers to the paper by Kuschnig (2022). Thereby, parts of it are replicated in this exercise to answer the question asked in the task assignment. In order to have the same initial datasets, the first step is therefore to reload the data used in this exercise in a way that is identical to the specifications used in the paper.

The following code is used to replicate the results, and therefore also the distance decay parameter, from the $SLX(\delta)$ model in the paper. δ thereby refers to the distance decay parameter γ. In order to maintain parity with the paper, we stick to naming the distance decay parameter as δ in the bonus question.

```r
years_numb <- length(unique(cigs$year))

W_cont <- kronecker(diag(years_numb), cigw/rowSums(cigw))

dist <- as.matrix(dist(cigd))
diag(dist) <- Inf

Psi <- function(delta) {
    W_dist <- dist^(-delta)
    W_dist <- W_dist/max(eigen(W_dist, symmetric = TRUE)$values)
    kronecker(diag(years_numb), W_dist)
}

y <- cbind(logc = cigs$logc)
X <- model.matrix(logc ~ logp + logy + year + state, data = cigs)
X_lag <- X[, c("logp", "logy")]
colnames(X_lag) <- c("w_logp", "w_logy")

X_cont <- W_cont %*% X_lag

n_save <- 50000L
n_burn <- 10000L

out_slxdx <- bslx(y ~ X, W = Psi, X_SLX = X_lag, n_save = n_save, n_burn = n_burn,
    options = set_options(SLX = set_SLX(delta = 3, delta_scale = 0.05, delta_a = 2,
        delta_b = 2)))

saveRDS(out_slxdx, "./assignment3/intermediate_output/out_slxdx.RDS")
```

Following the paper and the R code provided on github, we calculate the number of unique years in the dataset to then create a spatial contiguity matrix by applying the Kronecker product to the identity matrix of the years (diag(years_numb)) and the row-normalized weights. To create the inverse-distance decay function, we compute the pairwise distances between the states using the Euclidean distance, where then the diagonal is set to infinity (diagonal elements will be 0) to prevent self-connections. The function *Psi* then takes delta as a parameter and defines the weights matrix $1/dist^{\delta}$, scales the matrix by the maximum eigenvalue and applies the Kronecker product with the identity matrix of the years to expand for the panel data.

We then create a response variable matrix *y* with the log of cigarette consumption as well as construct the model matrix *X*, which shows the model specification / regression equation, including logged cigarette prices, logged income, and fixed effects for year and state (as stated in the task assignment). Creating *X_lag*, we extract the relevant columns for the spatial lags of prices and income and calculate the spatially lagged explanatory variables by performing a matrix multiplication of *W_cont* with *X_lag*. In a next step, we fit the SLX model using the bslx function. We define the model formula, provide the function for the distance decay weights matrix, specify the explanatory variables for spatial lags, set the number of posterior draws, as well as provide settings/options for the SLX model (including prior parameters for the distance decay parameter) following the specifications from the github code. We specify the starting value for the decay parameter in the Markov Chain Monte Carlo sampler, which then iteratively updates this value based on the posterior distribution obtained through sampling. We also specify the prior distribution for δ, indicating the shape of the inverse-gamma prior distribution.

The following table shows the estimate (mean) for the distance decay parameter δ as well as a 99% confidence interval derived with the *coda* package:

```
## # A tibble: 1 x 4
##   Parameter              Estimate `Lower Limit (99%)` `Upper Limit (99%)`
##   <chr>                     <dbl>               <dbl>               <dbl>
## 1 Distance Decay (Delta)     3.00                2.46                3.74
```

The distance decay parameter derived using the out_slxdx output and therefore Bayesian inference quantifies the rate at which the spatial influence of a spatial variable decreases/decays over distance. This parameter is critical for modeling how changes in one location affect changes in another. In the context of the cigarette consumption model, the distance decay parameter indicates how cigarette sales or consumption in one state influence neighboring states, considering the spatial distribution of these states. A higher δ means that spatial influence decays more rapidly with distance, meaning that spatially close states have a much stronger influence on each other than those further away. We cannot be 100% sure what the exact value of the estimate will be when knitting the final Rmd file. The estimate, however, matches quite well the value from the paper by Kuschnig (2022). The estimated mean distance decay parameter ($\delta = 3.00$ when we last ran code and therefore close to the 3.01 from the paper) thereby indicates the average of the posterior distribution of the parameter. The distance decay estimate signifies a relatively steep decline in spatial influence with distance, indicating that regions closer to each other have a significantly stronger effect on one another. This also makes sense intuitively in the context of cigarette consumption.

Task B

Describe the Unit of Observation

The unit of observation is defined in the paper as *grid cells* at a sub-national level in Africa, with each single cell as one unit of observation. Each of these cells within the grid system represents a certain area on the African map. For the analysis, a panel data set is used that comprises around 2700 cells in 46 African countries from 1997 to 2011, including various geo-referenced covariates. Within the units of observation, different types of data are aggregated (conflict data, climate data, as well as supplementary data on infrastructure and ethnic fractionalization, to give just a few examples).

What are their areas?

Each of these cells has a resolution of 1 degree latitude × 1 degree longitude (which corresponds to about 110 km × 110 km at the equator, see also the next point on relative differences in sizes of the cells). The validation of the resolution (1 degree resolution) is carried out by the authors through a robustness analysis with higher and lower spatial scales, in which they look at both a higher resolution 0.5 × 0.5 degree grid as well as a lower resolution with a 2 × 2 grid in addition to the chosen 1 degree resolution. The aim is to find a good balance between the degree of localization of agricultural shocks and the detection of the extent of conflict spillovers. As can be seen from the online appendix, and what is interesting for the interpretation of the area of the units of observations, the authors in their analysis also calculate the area of each cell in square kilometers corresponding to the land, excluding seas and lakes. Furthermore, cell-level controls such as elevation, roughness, or altitude are also taken into account.

Why and to which (relative) extent do they differ?

One possible reason for potential relative differences in the areas of each cell is the spherical shape of the Earth. Since the earth is a sphere and not a two-dimensional surface, a 1 × 1-degree grid cell does not correspond to an area of 110 km × 110 km everywhere on the globe. As the Earth is a globe, the longitudes converge at the poles. The horizontal distance (from East to West, i.e. the width of a cell) from one longitude to the next longitude becomes smaller the further you move from the equator to the poles (to the North or South). However, the distance between the lines of latitude, which run parallel, remains roughly constant across the globe. So while around the equator a 1 degree × 1 degree cell area has about 12100 squared kilometers (110 km × 110 km), the area of a 1 degree × 1 degree cell deviates and decreases the further you move towards the poles (as the East-West distance of the cell decreases). As the equator runs through Africa, the areas of 1 × 1 degree cells in Central Africa are larger than those in North and South Africa (as these cells are not located on/around the equator but located further towards the poles, where eventually the width of a cell would converge to zero).

Furthermore, as taken into account in the analysis, the area of each cell in squared kilometers of land (and thus excluding seas and lakes), depends on the occurrence/proportion of standing or flowing waters in a cell. The presence of roughness, hills, or mountains can also influence the area surface. The areas can therefore differ depending on various conditions and environmental factors of a cell. The area of a 110 km $times$ 110 km square can, additionally and to mention a third reason, in general vary depending on the map and projection used. This is because distances and areas can deviate according to the projection used in an analysis due to distortion.

Un-normalized weights matrix and its effects on interpretations

The weights matrix *W* used in the paper is a binary contiguity matrix, which is not normalized. All cells surrounding the cell of interest (neighboring cells being defined as being within a distance threshold of 180km) are assigned a value/weight of 1 and all other cells are assigned a value/weight of 0. In this case, the 8 adjacent cells are effectively considered neighbors (this is also called queen contiguity). More precisely, the average cell in the sample has 7.4 neighbors (since the cells on the edges no longer have 8 neighbors). Due to the binarity of the matrix, there is no further variation in influence based on the exact distance (either a cell is a neighbor of the cell of interest and thus has influence or not). As the matrix is not normalized, the sum of the values/weights in each row is not equal to 1, as is the case in the usual row-normalized matrix. The influence of each adjacent cell is taken into account equally and the cumulative effects of all neighbors are summed directly in the analysis. The influence of a certain cell on the cell of interest therefore does not depend on how many other neighbors the cell of interest has. The decision to use a non-normalized weights matrix was made by the authors to simplify the interpretation, as the coefficients of the spatial lags can be directly interpreted as the effects of a marginal change in a particular variable in one of the neighbors.

The spatial econometric model thus captures the total influence of those units that are considered neighboring, whereby each coefficient of a neighboring cell with a change can be interpreted as the total additional effect of this one other neighboring cell. Therefore, the coefficients on spatially lagged variables show cumulative and not averaged effects of changes in neighboring cells. In contrast, a normalized weights matrix (more precisely, a row-normalized weights matrix) shows the average influence of all neighbors. The coefficients for spatial lags can thus be interpreted directly as the effects of a change in a neighbor on the dependent variable in the cell of interest, and hence the authors' decision to use the chosen matrix *W* for simplicity reasons. Specifically, the coefficient for spatially lagged variables gives a direct interpretation for the effect on the cell of interest of a change in, say, conflict frequency or weather shocks in one or more neighboring cells.

What is also important to mention is that this un-weighted weights matrix tends to be sensitive to the number of identified neighbors. The cumulative influence of neighboring cells can potentially be more significant and the coefficients larger if the cell of interest has a higher number neighbors (more neighbors means higher cumulative influence, which then says very little about the average effect of a change in a neighbor). However, since the weights matrix *W* is binary (no further weighting) and most cells have exactly 8 neighbors (with the exception of border cells) as well as the fact that the average number of neighbors is quite close to the 8 neighbors at 7.4, this possible distortion due to different numbers of neighbors is moderate, a comparison of the cells is largely possible or the issue only needs to be taken into account, if at all, when a comparison is made with border cells.

Replication Tables

For the replication of Table 2 from the paper we prepare the dataset by removing one year of observation to have the same size of theirs, besides transforming into factors some key variables.

For the replication of the first column we employed the `plm()` regression given we are dealing with observations on multiple countries observed over multiple time periods.

```
model_1 <- plm(ANY_EVENT_ACLED ~ SPEI4pg + L1_SPEI4pg + L2_SPEI4pg + GSmain_ext_SPEI4pg +
    L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell +
    area_cell + as.factor(use_primary) + dis_river_cell + as.factor(shared) +
as.factor(border) +
    as.factor(any_mineral) + ELF + as.factor(year_) + i(country_, as.numeric(year_)),
    data = df_1, index = c("cell", "year_"), model = "pooling")

saveRDS(model_1, "./assignment3/intermediate_output/model_1.RDS")
```

For the replication of the second column, we extended the first mode to include also the vector of observations weighted by the matrix W.

```
model_2 <- plm(ANY_EVENT_ACLED ~ SPEI4pg + L1_SPEI4pg + L2_SPEI4pg + GSmain_ext_SPEI4pg +
    L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg + W_SPEI4pg + W_L1_SPEI4pg + W_L2_SPEI4pg
+
    W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg + W_L2_GSmain_ext_SPEI4pg +
elevation_cell +
    rough_cell + area_cell + as.factor(use_primary) + dis_river_cell + as.factor(shared) +
    as.factor(border) + as.factor(any_mineral) + ELF + W_elevation_cell + W_rough_cell +
    W_area_cell + W_use_primary + W_dis_river_cell + as.factor(W_shared) +
as.factor(W_border) +
    as.factor(W_any_mineral) + W_ELF + as.factor(year_) + i(country_, as.numeric(year_)),
    data = df_1, index = c("cell", "year_"), model = "pooling")

saveRDS(model_2, "./assignment3/intermediate_output/model_2.RDS")
```

For the replication of the third column we start by building the contiguity matrix with the paper's threshold of 180 km radius.

```
coor <- df_1 %>%
    filter(year_ == 2000) %>%
    select(cell, geometry) %>%
    st_as_sf(crs = "WGS84")

distw <- dnearneigh(st_centroid(coor), 0, 180, row.names = coor$cell)
W <- nb2listw(distw, style = "B", zero.policy = TRUE)
```

```r
model_3 <- spml(ANY_EVENT_ACLED ~ lag(ANY_EVENT_ACLED) + SPEI4pg + L1_SPEI4pg + L2_SPEI4pg +
    GSmain_ext_SPEI4pg + L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg + W_SPEI4pg +
    W_L1_SPEI4pg + W_L2_SPEI4pg + W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg +
    W_L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell + area_cell +
as.factor(use_primary) +
    dis_river_cell + as.factor(shared) + as.factor(border) + as.factor(any_mineral) +
    ELF + W_elevation_cell + W_rough_cell + W_area_cell + as.factor(W_border) +
as.factor(W_use_primary) +
    W_dis_river_cell + as.factor(W_shared) + as.factor(W_any_mineral) + W_ELF +
as.factor(country_):as.numeric(year_),
    data = df_1, index = c("cell", "year_"), listw = W, model = "pooling", effect = "time",
    spatial.error = "none", zero.policy = TRUE, lag = TRUE, dynamic = TRUE, local =
    list(parallel = T))

saveRDS(model_3, "./assignment3/intermediate_output/model_3.RDS")
```

For the last column 4, we include the country*year fixed effect, and leave out the year and country-specific time-trend controls. Differently from before in Model 3, `year_` is given in levels, rather than in numeric form. In this way one obtains the interaction term FE whereas in the other way one makes the country-specific time-trend.

```r
df_2 <- df_1 %>%
    mutate(countryyear = paste0(country_, "_", year_))


## Model 4
model_4 <- spml(ANY_EVENT_ACLED ~ lag(ANY_EVENT_ACLED) + SPEI4pg + L1_SPEI4pg + L2_SPEI4pg +
    GSmain_ext_SPEI4pg + L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg + W_SPEI4pg +
    W_L1_SPEI4pg + W_L2_SPEI4pg + W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg +
    W_L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell + area_cell +
as.factor(use_primary) +
    dis_river_cell + as.factor(shared) + as.factor(border) + as.factor(any_mineral) +
    ELF + W_elevation_cell + W_rough_cell + W_area_cell + as.factor(W_use_primary) +
    W_dis_river_cell + as.factor(W_shared) + as.factor(W_border) + as.factor(W_any_mineral)
+
    W_ELF + as.factor(countryyear), data = df_2, index = c("cell", "year_"), listw = W,
    model = "pooling", effect = "individual", spatial.error = "none", zero.policy = TRUE,
    dynamic = TRUE, lag = TRUE, Hess = TRUE, local = list(parallel = T))

saveRDS(model_4, "./assignment3/intermediate_output/model_4.RDS")
```

	Dependent variable:	
	ANY_EVENT_ACLED	
	(1)	(2)
SPEI4pg	0.035***	0.010
	(0.006)	(0.015)
L1_SPEI4pg	0.003	0.013
	(0.006)	(0.015)
L2_SPEI4pg	0.010*	−0.010
	(0.006)	(0.015)
GSmain_ext_SPEI4pg	−0.034***	−0.006
	(0.009)	(0.015)
L1_GSmain_ext_SPEI4pg	−0.032***	−0.043***
	(0.009)	(0.016)
L2_GSmain_ext_SPEI4pg	−0.035***	−0.024
	(0.009)	(0.016)
W_SPEI4pg		0.004*
		(0.002)
W_L1_SPEI4pg		−0.002
		(0.002)
W_L2_SPEI4pg		0.004
		(0.002)
W_GSmain_ext_SPEI4pg		−0.005**
		(0.003)
W_L1_GSmain_ext_SPEI4pg		0.002
		(0.003)
W_L2_GSmain_ext_SPEI4pg		−0.003
		(0.003)
Observations	35,042	35,042
R^2	0.189	0.195
Adjusted R^2	0.187	0.193
F Statistic	108.780*** (df = 75; 34966)	77.869*** (df = 109; 34932)

Note: *p<0.1; **p<0.05; ***p<0.01

Overall, the replication is satisfactory for the OLS columns, but in the case of the spatial dynamic models, results are not comparable to those of the Authors'

Table 3: Replication of Column 3.

	Estimate	Std. Error	t-value	Pr(>|t|)
lambda	0.0217055	0.0011541	18.8080409	0.0000000
(Intercept)	0.0480665	0.0166628	2.8846591	0.0039184
lag(ANY_EVENT_ACLED)	0.3773143	0.0051549	73.1948267	0.0000000
SPEI4pg	-0.0037283	0.0138138	-0.2698975	0.7872391
L1_SPEI4pg	0.0134666	0.0143522	0.9382955	0.3480926
L2_SPEI4pg	-0.0139103	0.0142513	-0.9760725	0.3290285
GSmain_ext_SPEI4pg	-0.0009407	0.0145393	-0.0646972	0.9484151
L1_GSmain_ext_SPEI4pg	-0.0455605	0.0151988	-2.9976418	0.0027208

	Estimate	Std. Error	t-value	Pr(>|t|)
L2_GSmain_ext_SPEI4pg	-0.0124235	0.0152197	-0.8162721	0.4143445
W_SPEI4pg	0.0036935	0.0021146	1.7466554	0.0806971
W_L1_SPEI4pg	-0.0024722	0.0021772	-1.1354718	0.2561777
W_L2_SPEI4pg	0.0047619	0.0021500	2.2147989	0.0267739
W_GSmain_ext_SPEI4pg	-0.0042563	0.0024663	-1.7257848	0.0843861
W_L1_GSmain_ext_SPEI4pg	0.0055888	0.0025607	2.1825588	0.0290683
W_L2_GSmain_ext_SPEI4pg	-0.0020898	0.0025552	-0.8178655	0.4134340

Table 4: Replication of Column 4.

	Estimate	Std. Error	t-value	Pr(>|t|)
lambda	0.0131011	0.0011795	11.1072420	0.0000000
(Intercept)	0.0230979	0.0270092	0.8551844	0.3924491
lag(ANY_EVENT_ACLED)	0.3689296	0.0052048	70.8830615	0.0000000
SPEI4pg	-0.0012055	0.0137063	-0.0879539	0.9299133
L1_SPEI4pg	0.0193334	0.0142505	1.3566801	0.1748829
L2_SPEI4pg	-0.0126191	0.0141483	-0.8919108	0.3724407
GSmain_ext_SPEI4pg	-0.0006339	0.0142053	-0.0446232	0.9644077
L1_GSmain_ext_SPEI4pg	-0.0505173	0.0148576	-3.4001055	0.0006736
L2_GSmain_ext_SPEI4pg	-0.0049433	0.0148883	-0.3320249	0.7398705
W_SPEI4pg	0.0019647	0.0022584	0.8699345	0.3843362
W_L1_SPEI4pg	-0.0016102	0.0023451	-0.6866108	0.4923280
W_L2_SPEI4pg	0.0053523	0.0023122	2.3148317	0.0206221
W_GSmain_ext_SPEI4pg	-0.0044642	0.0026322	-1.6959727	0.0898910
W_L1_GSmain_ext_SPEI4pg	0.0062899	0.0027355	2.2993863	0.0214830
W_L2_GSmain_ext_SPEI4pg	-0.0022051	0.0027343	-0.8064515	0.4199826

Vertical - Horizontal Contiguity

For the second part of the Assignment, when asked to build the vertical-horizontal contiguity matrix, we set up the following functions, `neighbors_horizontal` and `neighbors_vertical`. The former (latter) computes horizontal (vertical) neighbors for a set of points represented by their centroids. It starts by extracting the latitudes (longitudes) of the centroids and calculating the distances between all pairs of centroids. Then, it iterates over each centroid, identifying other centroids within the same latitude (longitude) band and within a specified distance threshold. It stores this information in a matrix where each row represents a centroid and each column indicates whether another centroid is a horizontal (vertical) neighbor. Finally, it converts this matrix into a spatial weights list object and returns it.

```
neighbors_horizontal <- function(centroids) {

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

    # Initialize a matrix to store horizontal neighbors
    num_points <- nrow(centroids)
    horizontal_neighbors <- matrix(0, nrow = num_points, ncol = num_points)

    for (i in seq_len(num_points)) {
        # Find points within the same latitude band
        neighbor_ids <- which(round(latitudes, 2) == round(latitudes[i], 2) & distances[i,
            ] < set_units(180000, "m"))  # keep threshold from text

        # Store neighbor ids in the matrix
        horizontal_neighbors[i, neighbor_ids] <- 1

    }
```

```r
    # Convert the matrix to a listw object
    listw <- mat2listw(horizontal_neighbors, style = "B", zero.policy = TRUE)

    return(listw)
}

neighbors_vertical <- function(centroids) {

    longitudes <- st_coordinates(centroids)[, 1]
    distances <- st_distance(centroids)

    # Initialize a matrix to store vertical neighbors
    num_points <- nrow(centroids)
    vertical_neighbors <- matrix(0, nrow = num_points, ncol = num_points)

    for (i in seq_len(num_points)) {

        neighbor_ids <- which(round(longitudes, 2) == round(longitudes[i], 2) & distances[i,
            ] < set_units(180000, "m"))

        # Store neighbor ids in the matrix
        vertical_neighbors[i, neighbor_ids] <- 1
    }

    # Convert the matrix to a listw object
    listw <- mat2listw(vertical_neighbors, style = "B", zero.policy = TRUE)

    return(listw)
}

# Apply the functions
horizontal <- neighbors_horizontal(st_centroid(coor))
vertical <- neighbors_vertical(st_centroid(coor))

# Visualization to check correctness
plot(st_geometry(coor))
coords <- st_coordinates(st_centroid(coor))
plot(horizontal, coords, add = TRUE, col = "blue", cex = 0.5)
plot(vertical, coords, add = TRUE, col = "red", cex = 0.5)  #all good!
```

The concept of a horizontal/vertical contiguity matrix suggests that only cells sharing the same latitude or longitude and directly neighboring each other affect one another. This assumption lacks a clear rationale when considering real-world dynamics, as there's no inherent reason to believe effects propagate in such a uni-directional manner, except that of limiting propagation of an effect.

One could argue that while this approach may seem overly simplistic, it could find utility in contexts characterized by pronounced regional disparities, such as countries with stark differences between north and south, like Italy, or between west and east, like Germany. In such cases, the matrix's structure might help capture and model the distinct dynamics of these regions, still with the acknowledgment of its limitations in representing more complex interaction patterns.

```
model_3_hor <- spml(ANY_EVENT_ACLED ~ lag(ANY_EVENT_ACLED) + SPEI4pg + L1_SPEI4pg +
    L2_SPEI4pg + GSmain_ext_SPEI4pg + L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg +
    W_SPEI4pg + W_L1_SPEI4pg + W_L2_SPEI4pg + W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg
+
    W_L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell + area_cell +
as.factor(use_primary) +
    dis_river_cell + as.factor(shared) + as.factor(border) + as.factor(any_mineral) +
    ELF + W_elevation_cell + W_rough_cell + W_area_cell + as.factor(W_border) +
as.factor(W_use_primary) +
    W_dis_river_cell + as.factor(W_shared) + as.factor(W_any_mineral) + W_ELF +
as.factor(country_):as.numeric(year_),
    data = df_1, index = c("cell", "year_"), listw = horizontal, model = "pooling",
    effect = "time", spatial.error = "none", zero.policy = TRUE, lag = TRUE, dynamic = TRUE,
    local = list(parallel = T))

model_3_ver <- spml(ANY_EVENT_ACLED ~ lag(ANY_EVENT_ACLED) + SPEI4pg + L1_SPEI4pg +
    L2_SPEI4pg + GSmain_ext_SPEI4pg + L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg +
    W_SPEI4pg + W_L1_SPEI4pg + W_L2_SPEI4pg + W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg
+
    W_L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell + area_cell +
as.factor(use_primary) +
```

```r
    dis_river_cell + as.factor(shared) + as.factor(border) + as.factor(any_mineral) +
    ELF + W_elevation_cell + W_rough_cell + W_area_cell + as.factor(W_border) +
as.factor(W_use_primary) +
    W_dis_river_cell + as.factor(W_shared) + as.factor(W_any_mineral) + W_ELF +
as.factor(country_):as.numeric(year_),
    data = df_1, index = c("cell", "year_"), listw = vertical, model = "pooling",
    effect = "time", spatial.error = "none", zero.policy = TRUE, lag = TRUE, dynamic = TRUE,
    local = list(parallel = T))

model_4_hor <- spml(ANY_EVENT_ACLED ~ lag(ANY_EVENT_ACLED) + SPEI4pg + L1_SPEI4pg +
    L2_SPEI4pg + GSmain_ext_SPEI4pg + L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg +
    W_SPEI4pg + W_L1_SPEI4pg + W_L2_SPEI4pg + W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg
+
    W_L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell + area_cell +
as.factor(use_primary) +
    dis_river_cell + as.factor(shared) + as.factor(border) + as.factor(any_mineral) +
    ELF + W_elevation_cell + W_rough_cell + W_area_cell + as.factor(W_use_primary) +
    W_dis_river_cell + as.factor(W_shared) + as.factor(W_border) + as.factor(W_any_mineral)
+
    W_ELF + as.factor(countryyear), data = df_4, index = c("cell", "year_"), listw =
horizontal,
    model = "pooling", effect = "individual", spatial.error = "none", zero.policy = TRUE,
    dynamic = TRUE, lag = TRUE, Hess = TRUE, local = list(parallel = T))

model_4_ver <- spml(ANY_EVENT_ACLED ~ lag(ANY_EVENT_ACLED) + SPEI4pg + L1_SPEI4pg +
    L2_SPEI4pg + GSmain_ext_SPEI4pg + L1_GSmain_ext_SPEI4pg + L2_GSmain_ext_SPEI4pg +
    W_SPEI4pg + W_L1_SPEI4pg + W_L2_SPEI4pg + W_GSmain_ext_SPEI4pg + W_L1_GSmain_ext_SPEI4pg
+
    W_L2_GSmain_ext_SPEI4pg + elevation_cell + rough_cell + area_cell +
as.factor(use_primary) +
    dis_river_cell + as.factor(shared) + as.factor(border) + as.factor(any_mineral) +
    ELF + W_elevation_cell + W_rough_cell + W_area_cell + as.factor(W_use_primary) +
    W_dis_river_cell + as.factor(W_shared) + as.factor(W_border) + as.factor(W_any_mineral)
+
    W_ELF + as.factor(countryyear), data = df_4, index = c("cell", "year_"), listw =
vertical,
    model = "pooling", effect = "individual", spatial.error = "none", zero.policy = TRUE,
    dynamic = TRUE, lag = TRUE, Hess = TRUE, local = list(parallel = T))


saveRDS(model_3_hor, "./assignment3/intermediate_output/model_3_hor.RDS")
saveRDS(model_3_ver, "./assignment3/intermediate_output/model_3_ver.RDS")
saveRDS(model_4_hor, "./assignment3/intermediate_output/model_4_hor.RDS")
saveRDS(model_4_ver, "./assignment3/intermediate_output/model_4_ver.RDS")
```

Table 5: The model from Column 3 with a horizontal contiguity matrix.

	Estimate	Std. Error	t-value	Pr(>|t|)
lambda	0.0611081	0.0023348	26.1727561	0.0000000
(Intercept)	0.0534644	0.0155503	3.4381589	0.0005857
lag(ANY_EVENT_ACLED)	0.3460228	0.0048108	71.9268737	0.0000000
SPEI4pg	-0.0035601	0.0128915	-0.2761573	0.7824272
L1_SPEI4pg	0.0138972	0.0133940	1.0375712	0.2994697
L2_SPEI4pg	-0.0147183	0.0132998	-1.1066578	0.2684419
GSmain_ext_SPEI4pg	-0.0012276	0.0135686	-0.0904764	0.9279086
L1_GSmain_ext_SPEI4pg	-0.0427716	0.0141840	-3.0154756	0.0025658
L2_GSmain_ext_SPEI4pg	-0.0106836	0.0142036	-0.7521731	0.4519470
W_SPEI4pg	0.0032286	0.0019734	1.6360578	0.1018275

	Estimate	Std. Error	t-value	Pr(>|t|)
W_L1_SPEI4pg	-0.0026051	0.0020319	-1.2820937	0.1998098
W_L2_SPEI4pg	0.0047353	0.0020065	2.3599911	0.0182754
W_GSmain_ext_SPEI4pg	-0.0037730	0.0023016	-1.6392679	0.1011575
W_L1_GSmain_ext_SPEI4pg	0.0052870	0.0023897	2.2124143	0.0269381
W_L2_GSmain_ext_SPEI4pg	-0.0021864	0.0023846	-0.9168576	0.3592173

Table 6: The model from Column 3 with a vertical contiguity matrix.

	Estimate	Std. Error	t-value	Pr(>|t|)
lambda	0.0180750	0.0026640	6.7848825	0.0000000
(Intercept)	0.0666883	0.0164618	4.0510872	0.0000510
lag(ANY_EVENT_ACLED)	0.3783407	0.0050928	74.2900490	0.0000000
SPEI4pg	-0.0004475	0.0136472	-0.0327922	0.9738403
L1_SPEI4pg	0.0147477	0.0141791	1.0401002	0.2982933
L2_SPEI4pg	-0.0140353	0.0140794	-0.9968714	0.3188269
GSmain_ext_SPEI4pg	-0.0029626	0.0143639	-0.2062520	0.8365940
L1_GSmain_ext_SPEI4pg	-0.0437101	0.0150155	-2.9110041	0.0036027
L2_GSmain_ext_SPEI4pg	-0.0113284	0.0150362	-0.7534104	0.4512033
W_SPEI4pg	0.0034746	0.0020891	1.6632275	0.0962669
W_L1_SPEI4pg	-0.0026031	0.0021510	-1.2101958	0.2262038
W_L2_SPEI4pg	0.0050486	0.0021241	2.3768280	0.0174622
W_GSmain_ext_SPEI4pg	-0.0043425	0.0024365	-1.7822253	0.0747125
W_L1_GSmain_ext_SPEI4pg	0.0050842	0.0025298	2.0097409	0.0444586
W_L2_GSmain_ext_SPEI4pg	-0.0027436	0.0025244	-1.0868421	0.2771066

Table 7: The model from Column 4 with a horizontal contiguity matrix.

	Estimate	Std. Error	t-value	Pr(>|t|)
lambda	0.0401560	0.0024415	16.4469772	0.0000000
(Intercept)	0.0229292	0.0258507	0.8869878	0.3750855
lag(ANY_EVENT_ACLED)	0.3495915	0.0049815	70.1778988	0.0000000
SPEI4pg	-0.0015408	0.0131184	-0.1174564	0.9064984
L1_SPEI4pg	0.0190033	0.0136393	1.3932821	0.1635345
L2_SPEI4pg	-0.0129479	0.0135414	-0.9561689	0.3389869
GSmain_ext_SPEI4pg	-0.0008919	0.0135960	-0.0655966	0.9476990
L1_GSmain_ext_SPEI4pg	-0.0485602	0.0142202	-3.4148603	0.0006381
L2_GSmain_ext_SPEI4pg	-0.0041968	0.0142496	-0.2945227	0.7683585
W_SPEI4pg	0.0017933	0.0021615	0.8296510	0.4067361
W_L1_SPEI4pg	-0.0016488	0.0022446	-0.7345824	0.4625939
W_L2_SPEI4pg	0.0051592	0.0022130	2.3313106	0.0197370
W_GSmain_ext_SPEI4pg	-0.0041277	0.0025193	-1.6384248	0.1013331
W_L1_GSmain_ext_SPEI4pg	0.0059584	0.0026181	2.2758191	0.0228568
W_L2_GSmain_ext_SPEI4pg	-0.0022446	0.0026170	-0.8577002	0.3910580

Table 8: The model from Column 4 with a vertical contiguity matrix.

	Estimate	Std. Error	t-value	Pr(>|t|)
lambda	0.0106180	0.0022473	4.7247039	0.0000023
(Intercept)	0.0304882	0.0267803	1.1384550	0.2549305
lag(ANY_EVENT_ACLED)	0.3689372	0.0051607	71.4903724	0.0000000
SPEI4pg	0.0003764	0.0135902	0.0276972	0.9779037
L1_SPEI4pg	0.0203037	0.0141298	1.4369450	0.1507336
L2_SPEI4pg	-0.0122159	0.0140284	-0.8707977	0.3838646

	Estimate	Std. Error	t-value	Pr(>|t|)
GSmain_ext_SPEI4pg	-0.0018767	0.0140850	-0.1332382	0.8940050
L1_GSmain_ext_SPEI4pg	-0.0495898	0.0147317	-3.3662093	0.0007621
L2_GSmain_ext_SPEI4pg	-0.0042727	0.0147621	-0.2894392	0.7722453
W_SPEI4pg	0.0019726	0.0022393	0.8809031	0.3783703
W_L1_SPEI4pg	-0.0015942	0.0023253	-0.6856156	0.4929555
W_L2_SPEI4pg	0.0053943	0.0022926	2.3529422	0.0186255
W_GSmain_ext_SPEI4pg	-0.0045569	0.0026099	-1.7459908	0.0808126
W_L1_GSmain_ext_SPEI4pg	0.0058216	0.0027123	2.1463729	0.0318432
W_L2_GSmain_ext_SPEI4pg	-0.0025387	0.0027111	-0.9364225	0.3490557

Overall, these different ways of defining the contiguity matrix do not change radically the effect of the variables of interest. This seems to favor the argumentation of LeSage and Pace (2014) that spatial regression models are not as sensitive as believed to the particular specifications used for the spatial weight structure in these models. The value of the spatial autoregressive parameter λ fluctuates across different specifications and spatial weights matrices. In Model 4, λ stands at 0.04, whereas in Model 3 utilizing the vertical spatial weights matrix, it increases to approximately 0.06. Altough these little differences can translate into large effects because of their amplification, the overall outcomes remain consistent across models.

Visualization of Spillover Effects

Unfortunately, we did not manage to produce such a visualization before the deadline of this assignment. Part of this is due to us being not familiar enough with this kind of dynamic spatial panel model. We will thus comment on the spillover visualization given on page 601 of the paper by Harari and Ferrara (2018) in lieu of an own visualization. For reference, the figure from the paper is printed in the following:

FIGURE 2.—SIMULATED RESPONSES OF SPEI GROWING SEASON SHOCKS

A. Dynamic Impact of a One-Time SPEI Growing Season Shock on Conflict Incidence

B. Spatial Impact of a One-Time SPEI Growing Season Shock on Conflict Incidence
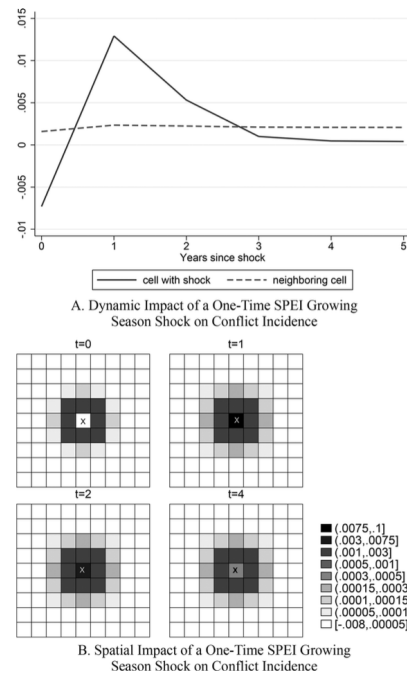
The figure from Harari and Ferrara (2018) shows that the effect of an SPEI growing season shock on conflict events propagates through both time and space. The first panel of their figure shows that expectedly, the effect on the cell experiencing the shock is largest one period after the shock. After three periods, the shock seems to dissipate. In contrast, the effect on neighboring cells is much smaller in magnitude, but does not dissipate in the five periods plotted. The second panel of their figure depicts spillovers beyond the direct neighbors. From that panel, we can see that spillovers to neighbors of neighbors seem to persist similarly to spillovers to neighbors, but that those effects are much weaker. There also does not seem to be any noticable effect on third-degree neighbors and beyond.

References

Halleck Vega, S., & Elhorst, J. P. (2015). The slx model. *Journal of Regional Science*, *55*(3), 339–363. https://doi.org/10.1111/jors.12188

Harari, M., & Ferrara, E. L. (2018). Conflict, climate, and cells: A disaggregated analysis. *The Review of Economics and Statistics*, *100*(4), 594–608. https://doi.org/10.1162/rest_a_00730

Kuschnig, N. (2022). Bayesian spatial econometrics: A software architecture. *Journal of Spatial Econometrics*, *3*(1), 6. https://doi.org/10.1007/s43071-022-00023-w

LeSage, J. P., & Pace, R. K. (2014). The biggest myth in spatial econometrics. *Econometrics*, *2*(4), 217–249. https://doi.org/10.3390/econometrics2040217