Spatial Economics – Assignment 2

Max Heinze (h11742049@s.wu.ac.at) Jaime Miravet (h12235992@s.wu.ac.at)
Jan Trimmel (h11809096@s.wu.ac.at)

April 15, 2024

Contents

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

Task A

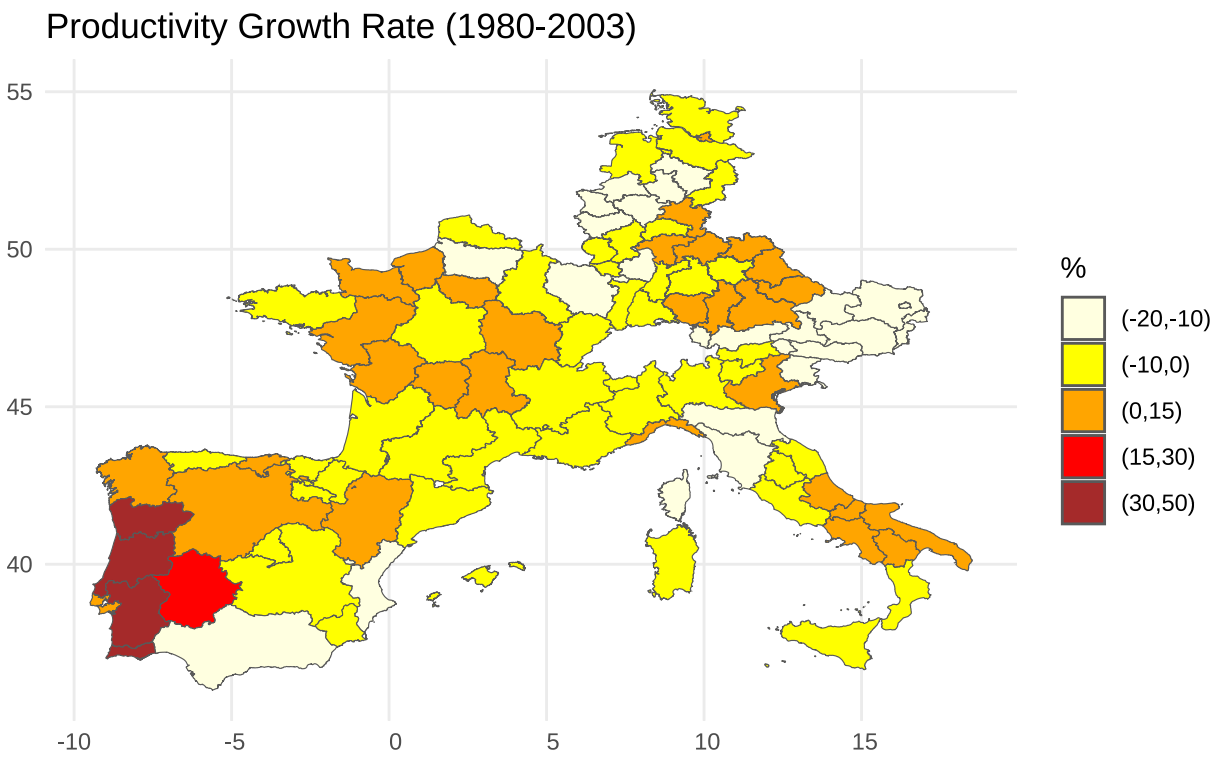
Productivity Growth Map

First, we load both the datafile and shapefile, select the countries of interest, and create the productivity growth rate variable.

```
pacman::p_load(dplyr, ggplot2, sf, spdep, raster)

# Load dataset and shapefile
load("./assignment2/data/data1.rda")
shp <- st_read(dsn = "./assignment2/data/EU27.shp", quiet = TRUE)

# Select variables of interest
vars <- c("IDb", "pr80b", "pr103b", "lninv1b", "lndens.empb")
data <- data1[vars]

# Create productivity growth variable for countries of interest
merged_shp <- left_join(shp, data, by = c(Id = "IDb"))
countries <- c("AT", "DE", "ES", "FR", "IT", "PT")
df <- merged_shp[grep(paste(countries, collapse = "|"), merged_shp$Id), ]
df <- na.omit(df)
df$prgrowth <- ((df$pr103b - df$pr80b)/df$pr80b) * 100
```

We create the productivty growth rate map:

```
# Cut the productivity growth values into five segments to plot them
breaks <- c(-20, -10, 0, 15, 30, 50)
labels <- c("(-20,-10)", "(-10,0)", "(0,15)", "(15,30)", "(30,50)")
df$prgrowth_bins <- cut(df$prgrowth, breaks = breaks, labels = labels)

# Visualization of productivity growth rate
bin_colors <- c("lightyellow", "yellow", "orange", "red", "brown")
ggplot() + geom_sf(data = df, aes(fill = prgrowth_bins)) + scale_fill_manual(name = "%",
    breaks = labels, values = bin_colors) + labs(title = "Productivity Growth Rate
    (1980-2003)") +
    theme_minimal()
```



Creating the weight matrices

(1) Distance threshold matrix

```
# Create centroids of regions
centr <- st_centroid(df)
coord <- st_coordinates(centr)
dist <- dnearneigh(coord, 0, 10000, row.names = df$Id)

# Minimum distance threshold so there are no disconnected regions
k1 <- knearneigh(coord, k = 1)
k1 <- knn2nb(k1)
min.max <- max(unlist(nbdists(k1, coords = coord)))
min.max
```

```
## [1] 2.652709
```

```
# We use the obtained maximum value of the threshold
dist <- dnearneigh(coord, 0, min.max, row.names = df$Id)

# With this new distance threshold, each region has an average of 8.3 links.

w1 <- nb2mat(dist, style = "W")
```

It appears that setting a large maximum value in the distance threshold yields too many links per region. Therefore, we compute the minimum value of the max. distance threshold so that no region is isolated. We create a weight matrix in which each region has a positive value if its kth neighbor is below the maximum distance threshold (around 2.65) and 0 if it is above.

(2) Smooth distance-decay matrix

We create a smooth distance-decay function so that the value decreases as distance between regions increases by setting δ = 0 in the exponential function $\exp[\delta d(i,j)]$. Then, we create a matrix by transforming the distance matrix that contains the distances between the centroids of the regions with the smooth distance-decay function

```
function0 <- function(x) {
    exp(-0.4 * x)
}


distance <- st_distance(centr)
g2 <- matrix(NA, nrow = nrow(distance), ncol = ncol(distance))

for (i in seq_along(distance)) {
    g2[i] <- function0(distance[i])
}

# Finally, we row-normalize the matrix and set its diagonal to 0 to create a
# weight matrix.
w2 <- g2/rowSums(g2)
diag(w2) <- 0
```

(3) Contiguity-based matrix

We construct a matrix such that in the row of each region each element will be positive for contiguous neighbors and 0 otherwise.

```
contiguity <- poly2nb(df, row.names = df$Id, queen = TRUE)

w3 <- nb2mat(contiguity, style = "W", zero.policy = TRUE)
# zero.policy is set to TRUE since there are regions with no contiguous
# neighbors
```

Comparing the matrices

First, we can compute the sparsity of each matrix (proportion of zero elements).

```
sp1 <- sum(w1 == 0)/length(w1)
sp2 <- sum(w2 == 0)/length(w2)
sp3 <- sum(w3 == 0)/length(w3)
sparsity <- c(sp1, sp2, sp3)
names(sparsity) <- c("W1", "W2", "W3")
print(sparsity)
```

```
##          W1          W2          W3
## 0.918559713 0.009708738 0.958148742
```

We observe that the elements of the matrices based on a distance threshold and on contiguity are mostly zero.The matrix based on a smooth distance-decay function has a much smaller proportion of zero elements. On the one hand, given the small distance threshold imposed and the large number of regions, it's not surprising that W_1 and W_3 have mostly zero elements, representing no link with most neighbors. On the other hand, since we use a decaying exponential function to construct W_2, even the regions that are furthest away from each other have a positive value. Therefore, the only zero elements are diagonal elements.

Then we compute their eigenvalues.

```
eigenw1 <- eigen(w1)$value
eigenw2 <- eigen(w2)$value
eigenw3 <- eigen(w3)$value

# Since each matrix has 103 eigenvalues, we focus on the largest 5.
eigentop1 <- eigenw1[1:5]
eigentop2 <- eigenw2[1:5]
eigentop3 <- eigenw3[1:5]

eigenvalues <- cbind(eigentop1, eigentop2, eigentop3)
print(eigenvalues)
```

```
##      eigentop1 eigentop2 eigentop3
## [1,] 1.0000000 0.9136859 1.0000000
## [2,] 0.9909479 0.7346632 0.9877825
## [3,] 0.9560398 0.5670454 0.9787316
## [4,] 0.9471914 0.4869174 0.9596820
## [5,] 0.9090106 0.3743769 0.9451354
```

We observe that the top 5 eigenvalues of W_1 and W_3 are larger than those of W_2. This shows there is stronger spatial autocorrelation in W_1 and W_3. On the other hand, the eigenvalues of W_1 and W_3 appear to decrease gradually, whereas those of W_2 decrease more rapidly. This indicates a faster decay in spatial autocorrelation with distance in W_2 compared to the other two matrices.

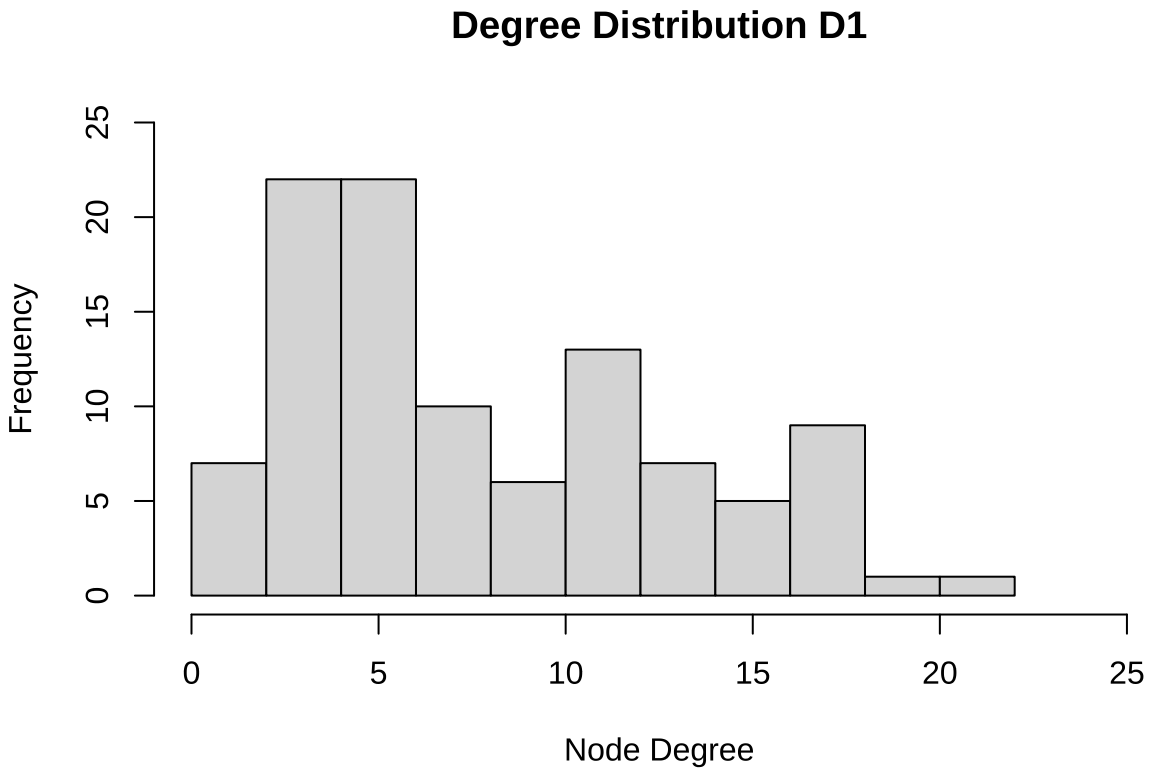
Now, in order to analyze the matrices from a graph theory perspective, we convert the weight matrices into adjacency (binary) matrices.

```
d1 <- ifelse(w1 > 0, 1, 0)
d2 <- ifelse(w2 > 0, 1, 0)
d3 <- ifelse(w3 > 0, 1, 0)
```

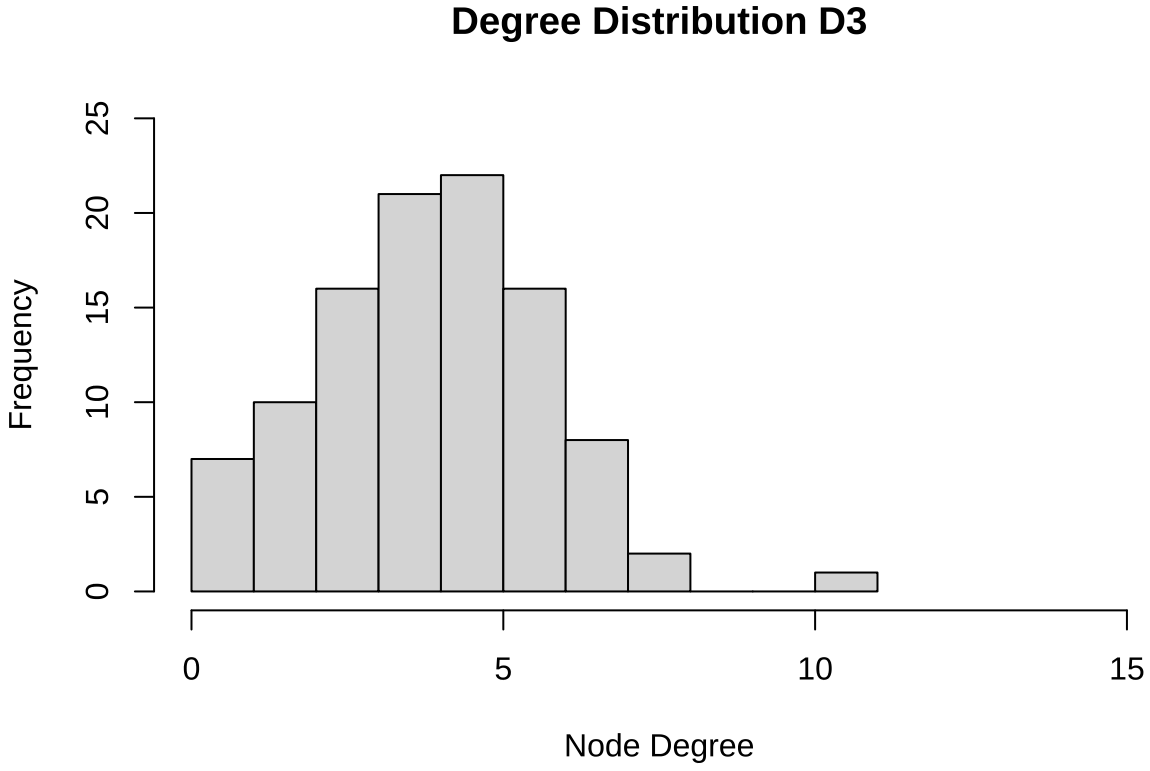
Once we have adjacency matrices, we can compute their degree distribution. Note that since W_2 has no zero off-diagonal elements, every region is connected to each other, so there is no point on computing the degree distribution of D_2.

```
node_degrees1 <- rowSums(d1)
node_degrees3 <- rowSums(d3)
```

```r
hist(node_degrees1, main = "Degree Distribution D1", xlab = "Node Degree", ylab =
"Frequency",
    xlim = c(0, 25), ylim = c(0, 25))
```



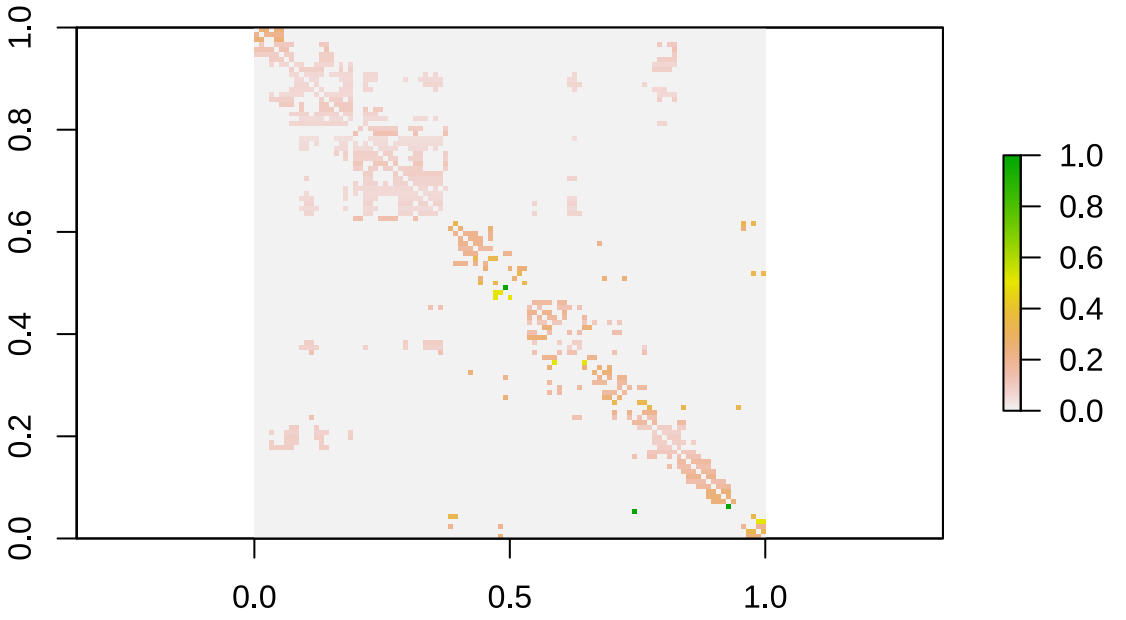

```r
hist(node_degrees3, main = "Degree Distribution D3", xlab = "Node Degree", ylab =
"Frequency",
    xlim = c(0, 15), ylim = c(0, 25))
```



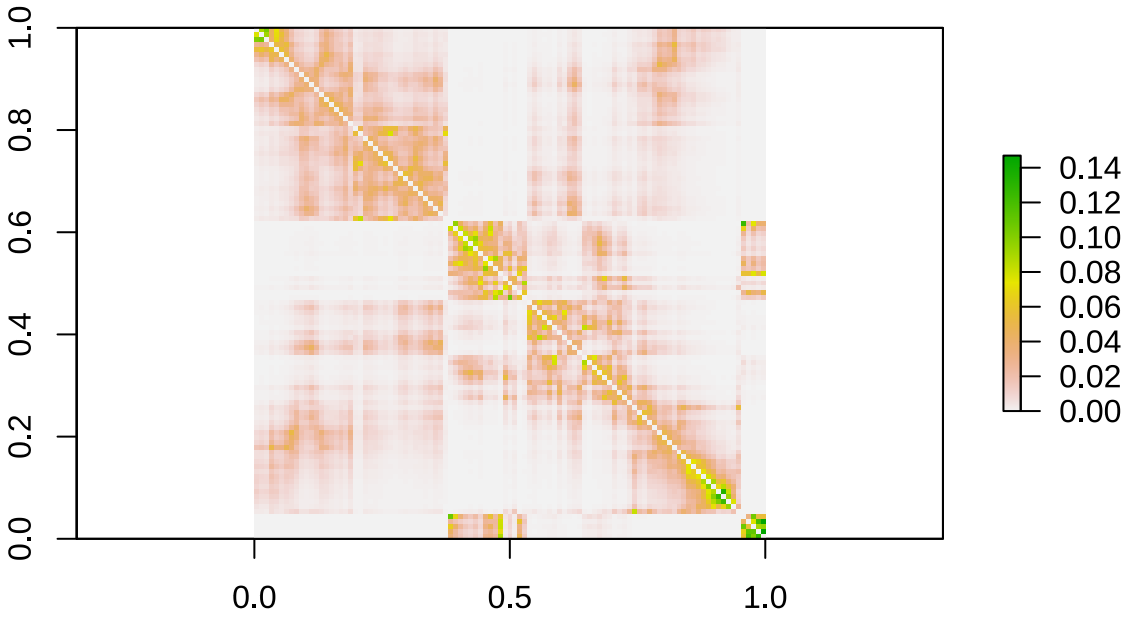
The adjacency matrix obtained when transforming W_1, D_1, shows a degree distribution more spread and skewed to the right, whereas D_3 (transformed W_3) shows a more concentrated degree distribution with hardly no skeewness.

Plotting the matrices

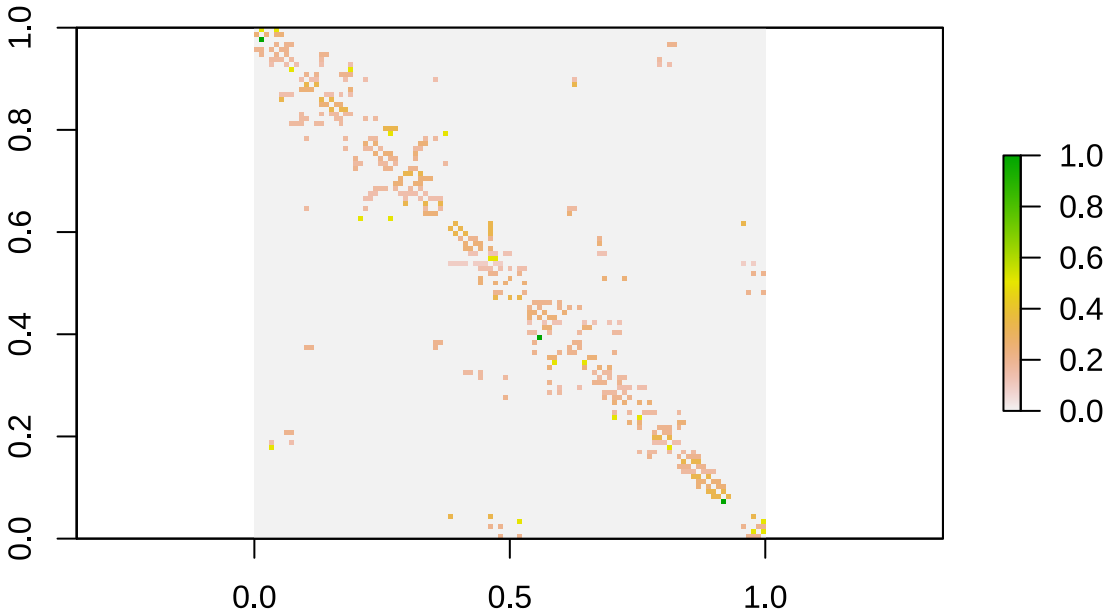
```
v1 <- raster(w1)
v2 <- raster(w2)
v3 <- raster(w3)
plot(v1)
```




```
plot(v2)
```




```
plot(v3)
```



We observe that W_1 and W_3 represent a network there are less connections between regions, but the existing links are stronger than those of W_2.The smooth distance-decay matrix W_2, on the other hand, represents a much more interconnected network, where each region is connected to each other but the links are weaker on average.

W_2 can help us visualize the clusters in the network. To do so, we must take into account how the countries are ordered in the matrix. The order (from first rows(columns) to last) is Austria, Germany, Spain, France, Italy and Portugal. (This is because the region codes are ordered alphabetically (AT, DE, ES, FR, IT, PT). Looking at W_2 we observe the biggest cluster is located among the Austrian and German regions (top-left corner). Then, the smaller cluster in the middle represents Spain. As we can see, Spanish regions have a almost non-existing link with Austria and Germany, and their stronger links are related to Portugal (middle of the bottom (or right) part). Then, in the diagonal below Spain we see France. We can see how it is the country with more connections to the rest of countries. Below in the diagonal we observe the cluster formed by the Italian regions. We see that their strongest foreign links are with Austrian regions. Finally, in the very bottom-right corner we see Portugal. It is clear how Portuguese regions are isolated from all countries except Spain.

Computing a measure of spatial autocorrelation for productivity growth

To measure spatial autocorrelation, we will compute a Global Moran's I statistic for each matrix.

```
# (1) Distance threshold matrix
w1listw <- mat2listw(w1, style = "W")  #We transform w1 into a listw so that moran.test()
works
i1 <- moran.test(df$prgrowth, w1listw)
i1 <- i1$estimate["Moran I statistic"]
# (2) Smooth distance-decay matrix
w2listw <- mat2listw(w2, style = "W")
i2 <- moran.test(df$prgrowth, w2listw)
i2 <- i2$estimate["Moran I statistic"]
# (3) Contiguity matrix
w3listw <- mat2listw(w3, style = "W", zero.policy = TRUE)
i3 <- moran.test(df$prgrowth, w3listw)
i3 <- i3$estimate["Moran I statistic"]

# I's statistics comparison
I_score_comparison <- c(i1, i2, i3)
names(I_score_comparison) <- c("w1", "w2", "w3")
print(I_score_comparison)
```

```
##        w1        w2        w3
## 0.5748219 0.3645441 0.5379854
```

The Global Moran's I score can range from -1 to 1, and it measures spatial autocorrelation, indicating the degree

of similarity between neighboring regions in terms of productivity growth rate. We observe that W_1 and W_3 have a similar value around 0.55, indicating a moderate positive spatial autocorrelation between the regions. The I statistic of matrix W_2 is somewhat smaller, implying a lower spatial autocorrelation, though it still shows signs of clustering.

OLS regression

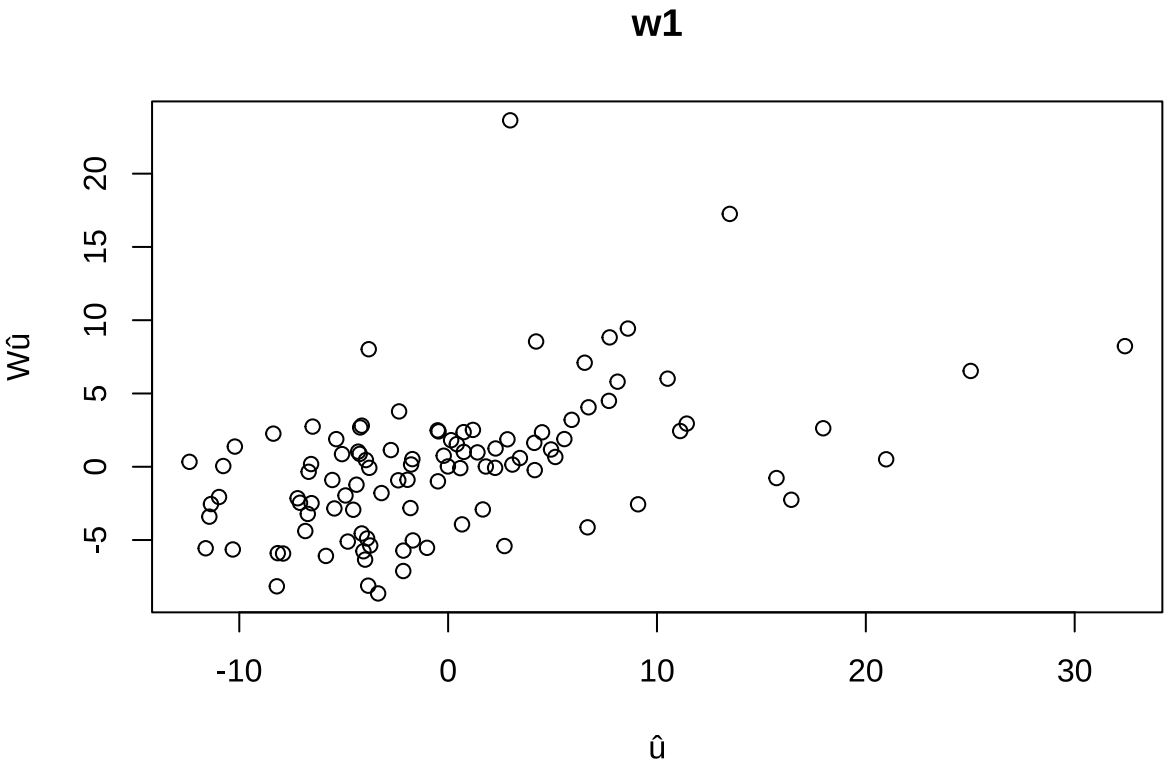
```
# pr80b, pr103b: Productivity of the region in 1980 and 2003 lninv1b: log of
# investment lndens.empb: log of densitiy of employment

# We run the regression and store the residuals.
ols <- lm(prgrowth ~ pr80b + lninv1b + lndens.empb, data = df)
residuals <- residuals(ols)
```

We want to observe if errors are spatially autocorrelated, i.e., if errors of similar size come from regions that are closer in space. To do that, we can compute the spatially lagged errors of the regression. That is, we can multiply our desired weight matrix by the vector of residuals. Note that $W\hat{u}$ shows the average value of the residuals of the neighbors of each region. This can show how clustered residuals are in space.

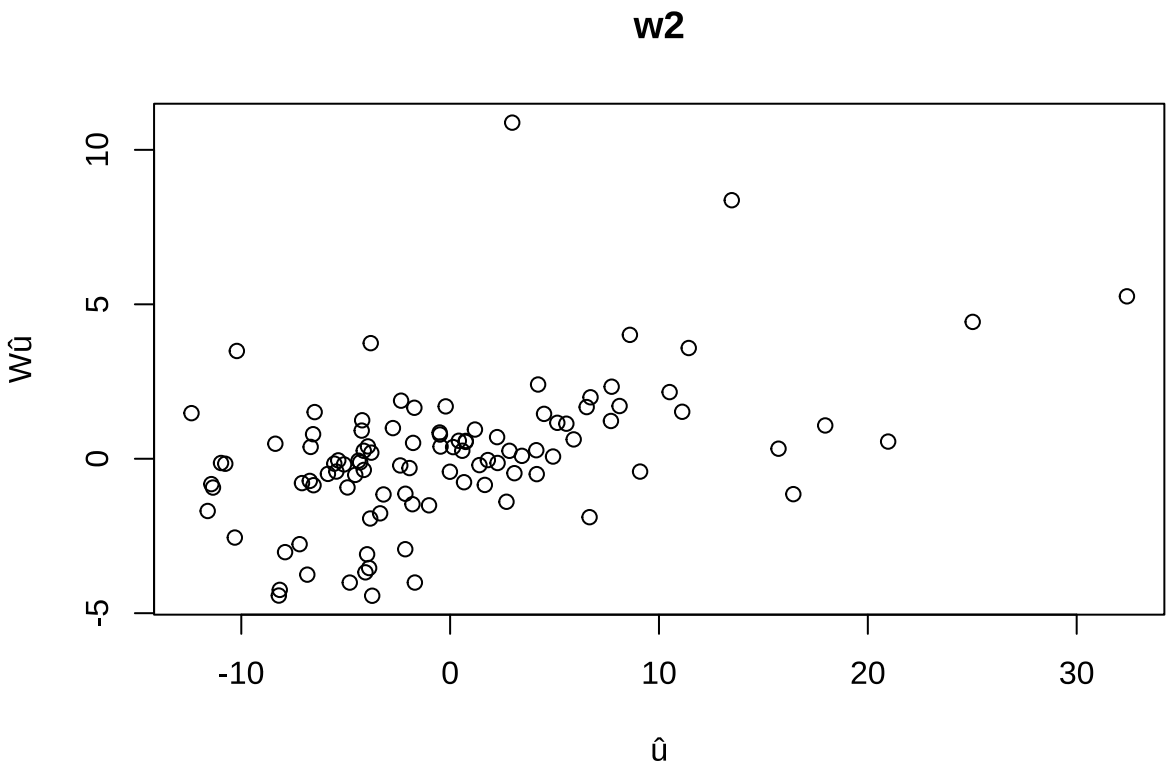
```
lagged_residuals_w1 <- lag.listw(w1listw, residuals)
lagged_residuals_w2 <- lag.listw(w2listw, residuals)
lagged_residuals_w3 <- lag.listw(w3listw, residuals)
```

To visually check if the residuals are autocorrelated, we can create scatter plots which relate each residuals to its spatially lagged residual. The intuition behind this is that for each region, its residual is compared to the average residual value of its neighbors. If there is spatial autocorrelation, we would expect each residual ($\hat{u}_i$) to be of similar size to the the average residual value of its neighbors ($W\hat{u}_i$).

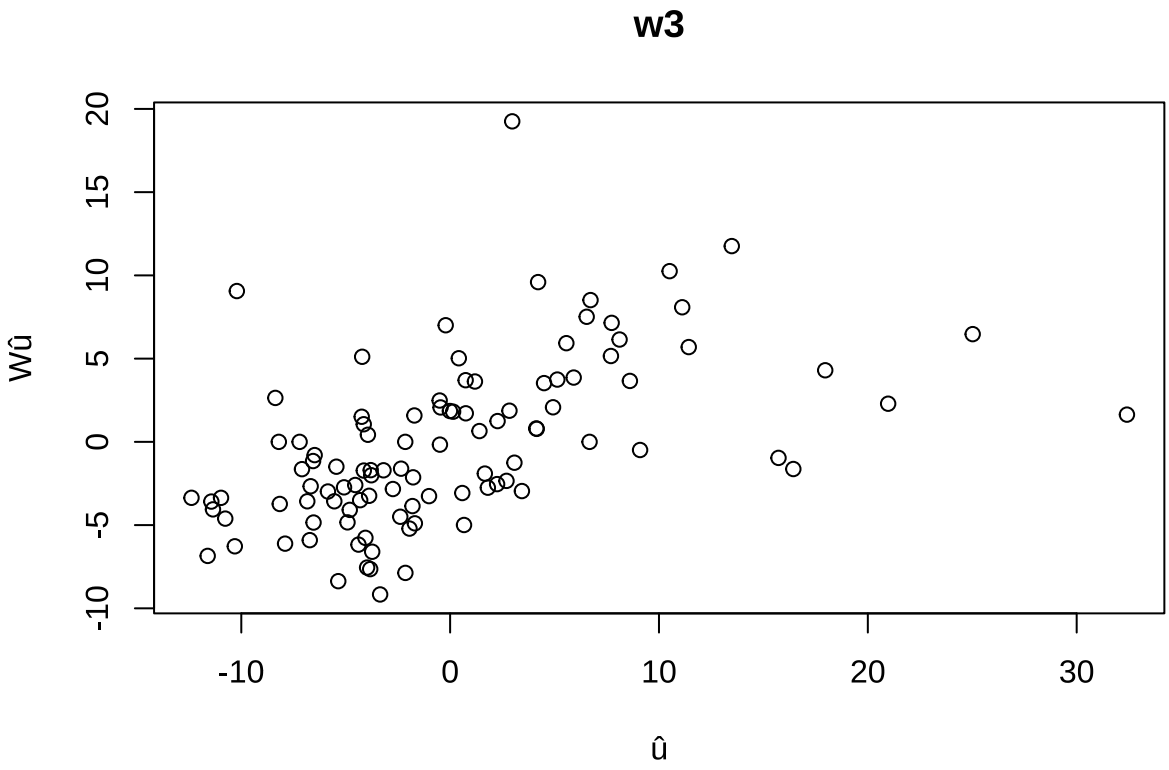
```
plot(residuals, lagged_residuals_w1, main = "w1", xlab = "û", ylab = "Wû")
```




```
plot(residuals, lagged_residuals_w2, main = "w2", xlab = "û", ylab = "Wû")
```




```
plot(residuals, lagged_residuals_w3, main = "w3", xlab = "û", ylab = "Wû")
```



For all three weight matrices, we observe a slight positive relationship between residuals and their spatially lagged counterparts, specially when using W_3, implying certain spatial autocorrelation between residuals.

To statistically check for spatial autocorrelation, we can compute once again Global Moran's I using each weight matrix, this time analyzing the residuals instead of productivity growth.

```
iw1 <- moran.test(residuals, w1listw)
iw1 <- iw1$estimate["Moran I statistic"]

iw2 <- moran.test(residuals, w2listw)
iw2 <- iw2$estimate["Moran I statistic"]
```

```r
iw3 <- moran.test(residuals, w3listw)
iw3 <- iw3$estimate["Moran I statistic"]

I_score_comparison2 <- c(iw1, iw2, iw3)
names(I_score_comparison2) <- c("w1", "w2", "w3")
print(I_score_comparison2)
```

```
##        w1        w2        w3
## 0.2930808 0.1380220 0.3225208
```

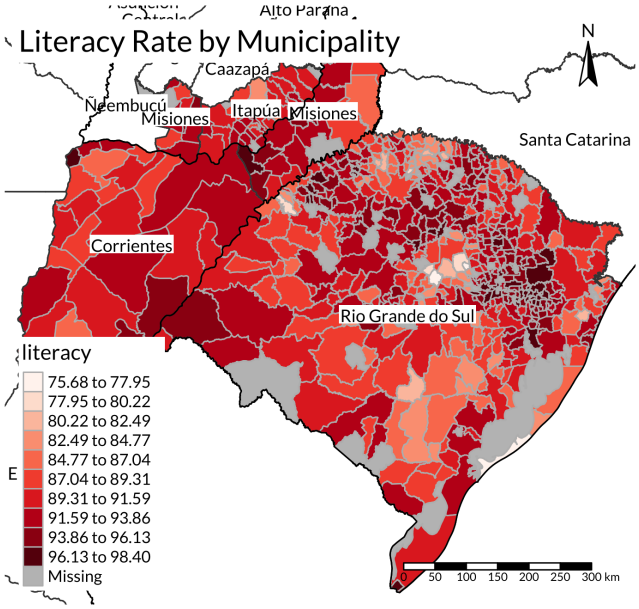
The statistics related to W_1 and W_3 imply a moderate level of clustering of errors, whereas if we use W_2 to check for spatial autocorrelation, we would obtain a value indicating less clustering. The presence of spatial dependence across errors implies that the assumption of independent errors is violated. If we were to run an OLS regression, our estimates would be biased and inconsistent.If we suspect that the errors are spatially autocorrelated, we can use the spatial error model (SEM), using a weight matrix to account for the autocorrelation.

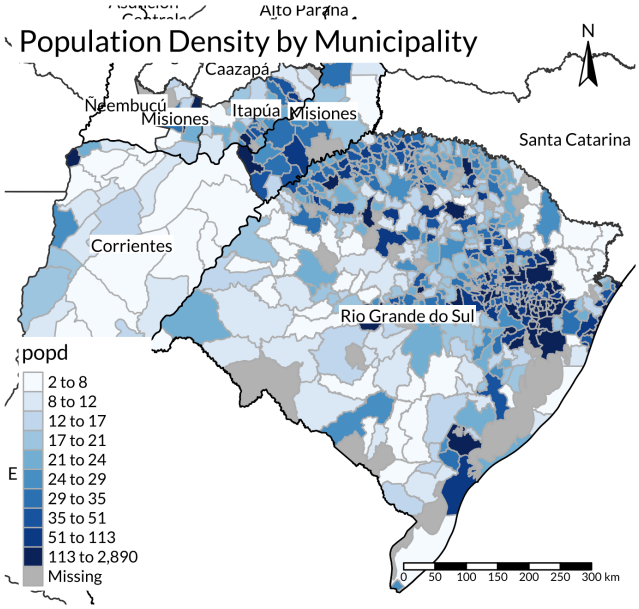
On the other hand, if there is spatial dependence between the observations of interest (in our case productivity growth rate), the estimates obtained from a regression will not accurately capture the true effect of the independent variable(s), since effects related to spatial characteristics will be mistakenly attributed to the parameter. In that case, we could estimate a spatial autoregressive model (SAR), in which the neighbors' characteristics are also considered when estimating the parameters.

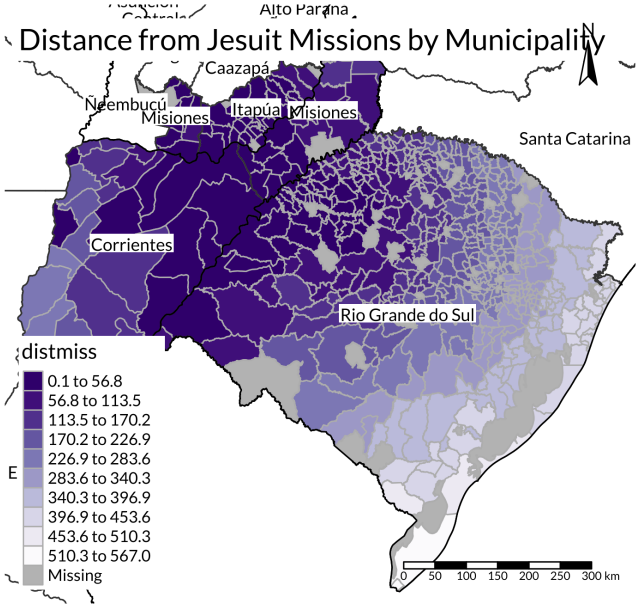
Task B

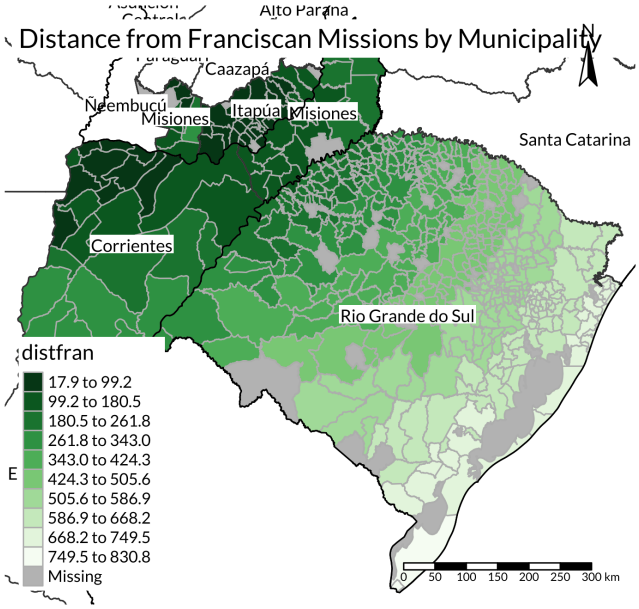
Nice Maps

We create four maps:









Replication of Table 2

Below are code and results of our replication of Table 2 from Valencia Caicedo (2019). Since the original author used Stata's robust standard errors, a notorious problem for replication in R, we specifically report Stata-style standard errors in the table below using the `starprep` function from the `estimatr` package. Using these standard errors, we can reproduce both coefficients and standard errors exactly.

```
# Replicate Results ------------------------------------------------------
col1 <- lm(illiteracy ~ distmiss + lati + longi + corr + ita + mis + mis1, data = litr)
col2 <- lm(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci + rugg +
    river + coast + corr + ita + mis + mis1, data = litr)

litr_bra <- subset(litr, country == "BRA")
col3 <- lm(illiteracy ~ distmiss + lati + longi + as.factor(mesorregi), data = litr_bra)
col4 <- lm(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci + rugg +
    river + coast + as.factor(mesorregi), data = litr_bra)

litr_arg <- subset(litr, country == "Argentina")
col5 <- lm(illiteracy ~ distmiss + lati + longi + corr, data = litr_arg)
col6 <- lm(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci + rugg +
```

```
    river + coast + corr, data = litr_arg)

litr_pry <- subset(litr, country == "Paraguay")
col7 <- lm(illiteracy ~ distmiss + ita, data = litr_pry)
col8 <- lm(illiteracy ~ distmiss + area + tempe + alti + preci + rugg + river + coast +
    ita, data = litr_pry)
```

	Dependent variable:			
	illiteracy			
	(1)	(2)	(3)	(4)
distmiss	0.011***	0.011**	0.016*	0.030***
	(0.004)	(0.005)	(0.009)	(0.010)
lati	0.556**	0.070	0.408	4.575**
	(0.238)	(0.781)	(0.553)	(1.807)
longi	−1.108***	−1.007*	−1.022	−5.694***
	(0.257)	(0.556)	(0.689)	(1.811)
area		0.0001		−0.0002
		(0.0002)		(0.0004)
tempe		0.059		−0.062
		(0.077)		(0.124)
alti		0.006		0.001
		(0.004)		(0.005)
preci		−0.003		0.001
		(0.002)		(0.003)
rugg		−0.00000		−0.00000
		(0.00000)		(0.00000)
river		1.470**		1.723*
		(0.712)		(0.893)
coast		0.209		−4.976**
		(0.894)		(2.147)
corr	−5.341***	−6.032***		
	(1.286)	(1.583)		
ita	−3.187***	−2.409***		
	(0.728)	(0.833)		
mis	−4.324***	−4.734***		
	(1.122)	(1.488)		
mis1	−3.279***	−2.299**		
	(0.860)	(0.974)		
as.factor(mesorregi)4302			−2.720***	−2.543**
			(0.863)	(1.037)
as.factor(mesorregi)4303			−0.483	−0.383
			(1.133)	(1.228)
as.factor(mesorregi)4304			−0.771	0.196
			(1.150)	(1.403)
as.factor(mesorregi)4305			−3.023**	−1.290
			(1.312)	(1.541)
as.factor(mesorregi)4306			−1.724	−3.421
			(2.015)	(2.184)
as.factor(mesorregi)4307			−0.437	0.327
			(2.553)	(2.634)
Constant	−35.328***	−53.741*	−35.274	−143.869**
	(11.797)	(32.497)	(38.125)	(66.532)
Observations	549	548	467	467
R^2	0.042	0.073	0.094	0.135
Adjusted R^2	0.029	0.049	0.076	0.104
Residual Std. Error	3.948 (df = 541)	3.912 (df = 533)	4.040 (df = 457)	3.978 (df = 450)
Note:				*p<0.1; **p<0.05; ***p<0.01

	Dependent variable:			
	illiteracy			
	(5)	(6)	(7)	(8)
distmiss	0.016**	0.067***	0.005	0.014
	(0.007)	(0.022)	(0.012)	(0.027)
lati	0.084	−9.338**		
	(0.758)	(3.831)		
longi	1.095	7.186***		
	(0.803)	(2.676)		
area		−0.0001		0.0004
		(0.0003)		(0.001)
tempe		0.968***		0.360
		(0.237)		(0.220)
alti		0.065***		0.016
		(0.011)		(0.014)
preci		−0.017**		0.0001
		(0.008)		(0.005)
rugg		−0.00005***		0.0001
		(0.00002)		(0.00005)
river		9.795***		0.983
		(2.173)		(5.492)
coast		1.889		0.826
		(3.522)		(4.557)
corr	3.771**	−3.043		
	(1.850)	(3.644)		
ita			−0.231	0.829
			(0.832)	(2.316)
Constant	69.263*	−41.058	8.673***	−80.723*
	(36.792)	(54.628)	(0.694)	(43.888)
Observations	42	42	40	39
R^2	0.165	0.669	0.004	0.251
Adjusted R^2	0.075	0.547	−0.050	0.019
Residual Std. Error	2.924 (df = 37)	2.045 (df = 30)	2.150 (df = 37)	2.101 (df = 29)
Note:				*p<0.1; **p<0.05; ***p<0.01

Next, we try to reproduce the Conley standard errors. We try two different approaches, first using the `conleyreg` package and then using the `fixest` package. Valencia Caicedo (2019) specifies a cutoff distance of 0.1 degrees. Both packages we use only allow us to specify the cutoff distance in kilometers, so for the sake of simplicity, we use the distance that 0.1 degrees equal at the equator, which is 6 nautical miles, or approximately 11.112 kilometers. We first print the results from the `conleyreg` package and then from the `fixest` package.

```
lit1 <- litr %>%
    drop_na(lati, longi) %>%
    mutate(lat = lati, lon = longi)

col1c <- conleyreg(illiteracy ~ distmiss + lati + longi + corr + ita + mis + mis1,
    data = lit1, dist_cutoff = 11.112, lat = "lat", lon = "lon")
col2c <- conleyreg(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr + ita + mis + mis1, data = lit1, dist_cutoff = 11.112,
    lat = "lat", lon = "lon")

lit1_bra <- subset(lit1, country == "BRA")
col3c <- conleyreg(illiteracy ~ distmiss + lati + longi + as.factor(mesorregi), data =
lit1_bra,
    dist_cutoff = 11.112, lat = "lat", lon = "lon")
col4c <- conleyreg(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + as.factor(mesorregi), data = lit1_bra, dist_cutoff = 11.112,
    lat = "lat", lon = "lon")

lit1_arg <- subset(lit1, country == "Argentina")
col5c <- conleyreg(illiteracy ~ distmiss + lati + longi + corr, data = lit1_arg,
    dist_cutoff = 11.112, lat = "lat", lon = "lon")
col6c <- conleyreg(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
```

```
    rugg + river + coast + corr, data = lit1_arg, dist_cutoff = 11.112, lat = "lat",
    lon = "lon")

lit1_pry <- subset(lit1, country == "Paraguay")
col7c <- conleyreg(illiteracy ~ distmiss + ita, data = lit1_pry, dist_cutoff = 11.112,
    lat = "lat", lon = "lon")
col8c <- conleyreg(illiteracy ~ distmiss + area + tempe + alti + preci + rugg + river +
    coast + ita, data = lit1_pry, dist_cutoff = 11.112, lat = "lat", lon = "lon")
```

	Dependent variable:			
	(1)	(2)	(3)	(4)
distmiss	0.011***	0.011**	0.016*	0.030***
	(0.004)	(0.005)	(0.009)	(0.010)
lati	0.556**	0.070	0.408	4.575**
	(0.247)	(0.773)	(0.570)	(1.792)
longi	−1.108***	−1.007*	−1.022	−5.694***
	(0.266)	(0.549)	(0.696)	(1.780)
area		0.0001		−0.0002
		(0.0002)		(0.0003)
tempe		0.059		−0.062
		(0.080)		(0.130)
alti		0.006		0.001
		(0.004)		(0.006)
preci		−0.003		0.001
		(0.002)		(0.003)
rugg		−0.00000		−0.00000
		(0.00000)		(0.00000)
river		1.470**		1.723*
		(0.731)		(0.904)
coast		0.209		−4.976**
		(0.885)		(2.125)
corr	−5.341***	−6.032***		
	(1.325)	(1.612)		
ita	−3.187***	−2.409***		
	(0.758)	(0.848)		
mis	−4.324***	−4.734***		
	(1.159)	(1.519)		
mis1	−3.279***	−2.299**		
	(0.876)	(0.980)		
as.factor(mesorregi)4302			−2.720***	−2.543**
			(0.896)	(1.067)
as.factor(mesorregi)4303			−0.483	−0.383
			(1.150)	(1.250)
as.factor(mesorregi)4304			−0.771	0.196
			(1.215)	(1.448)
as.factor(mesorregi)4305			−3.023**	−1.290
			(1.393)	(1.659)
as.factor(mesorregi)4306			−1.724	−3.421
			(2.027)	(2.190)
as.factor(mesorregi)4307			−0.437	0.327
			(2.576)	(2.654)
Constant	−35.328***	−53.741	−35.274	−143.869**
	(12.074)	(33.356)	(38.313)	(66.608)
Note:				*p<0.1; **p<0.05; ***p<0.01

	Dependent variable:			
	(5)	(6)	(7)	(8)
distmiss	0.016**	0.067***	0.005	0.014
	(0.007)	(0.019)	(0.011)	(0.023)
lati	0.084	−9.338***		
	(0.711)	(3.238)		
longi	1.095	7.186***		
	(0.754)	(2.262)		
area		−0.0001		0.0004
		(0.0002)		(0.001)
tempe		0.968***		0.360*
		(0.200)		(0.190)
alti		0.065***		0.016
		(0.010)		(0.012)
preci		−0.017**		0.0001
		(0.007)		(0.004)
rugg		−0.00005***		0.0001
		(0.00002)		(0.00004)
river		9.795***		0.983
		(1.837)		(4.738)
coast		1.889		0.826
		(2.976)		(3.941)
corr	3.771**	−3.043		
	(1.736)	(3.080)		
ita			−0.231	0.829
			(0.794)	(1.998)
Constant	69.263*	−41.058	8.673***	−80.723**
	(34.532)	(46.169)	(0.666)	(37.732)
Note:				*p<0.1; **p<0.05; ***p<0.01

```
col1cf <- feols(illiteracy ~ distmiss + lati + longi + corr + ita + mis + mis1, data = litr,
    vcov_conley(lat = "lati", lon = "longi", cutoff = 11.112, distance = "spherical"))
col2cf <- feols(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr + ita + mis + mis1, data = litr, vcov_conley(lat = "lati",
    lon = "longi", cutoff = 11.112, distance = "spherical"))

col3cf <- feols(illiteracy ~ distmiss + lati + longi + as.factor(mesorregi), data =
litr_bra,
    vcov_conley(lat = "lati", lon = "longi", cutoff = 11.112, distance = "spherical"))
col4cf <- feols(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + as.factor(mesorregi), data = litr_bra, vcov_conley(lat = "lati",
    lon = "longi", cutoff = 11.112, distance = "spherical"))

col5cf <- feols(illiteracy ~ distmiss + lati + longi + corr, data = litr_arg,
vcov_conley(lat = "lati",
    lon = "longi", cutoff = 11.112, distance = "spherical"))
col6cf <- feols(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr, data = litr_arg, vcov_conley(lat = "lati", lon = "longi",
    cutoff = 11.112, distance = "spherical"))

col7cf <- feols(illiteracy ~ distmiss + ita, data = litr_pry, vcov_conley(lat = "lati",
    lon = "longi", cutoff = 11.112, distance = "spherical"))
col8cf <- feols(illiteracy ~ distmiss + area + tempe + alti + preci + rugg + river +
    coast + ita, data = litr_pry, vcov_conley(lat = "lati", lon = "longi", cutoff = 11.112,
    distance = "spherical"))
```

Dependent Variable:	illiteracy							
Model:	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Variables								
Constant	-35.33***	-53.74	-35.27	-143.9*	69.26*	-41.06	8.673***	-80.72*
	(12.96)	(38.28)	(40.26)	(73.54)	(36.35)	(53.97)	(0.6825)	(43.22)
distmiss	0.0105***	0.0112*	0.0164*	0.0297**	0.0157**	0.0669***	0.0045	0.0138
	(0.0040)	(0.0057)	(0.0092)	(0.0115)	(0.0073)	(0.0217)	(0.0112)	(0.0261)
lati	0.5561*	0.0698	0.4078	4.575**	0.0837	-9.338**		
	(0.2841)	(0.7983)	(0.6512)	(1.898)	(0.7486)	(3.785)		
longi	-1.108***	-1.007*	-1.022	-5.694***	1.095	7.186**		
	(0.2957)	(0.5727)	(0.7468)	(1.857)	(0.7939)	(2.644)		
corr	-5.341***	-6.032***			3.771**	-3.043		
	(1.462)	(1.776)			(1.828)	(3.600)		
ita	-3.187***	-2.409**					-0.2311	0.8290
	(0.8686)	(0.9485)					(0.8123)	(2.294)
mis	-4.324***	-4.734***						
	(1.289)	(1.695)						
mis1	-3.279***	-2.299**						
	(0.9438)	(1.042)						
area		0.0001		-0.0002		-8.9×10^{-5}		0.0004
		(0.0002)		(0.0004)		(0.0003)		(0.0007)
tempe		0.0587		-0.0625		0.9675***		0.3598
		(0.0937)		(0.1558)		(0.2343)		(0.2205)
alti		0.0057		0.0006		0.0654***		0.0160
		(0.0042)		(0.0067)		(0.0112)		(0.0137)
preci		-0.0026		0.0010		-0.0171**		0.0001
		(0.0027)		(0.0037)		(0.0083)		(0.0051)
rugg		-3.56×10^{-6}		-3.09×10^{-6}		-4.8×10^{-5}**		6.93×10^{-5}
		(4.8×10^{-6})		(4.71×10^{-6})		(1.82×10^{-5})		(4.85×10^{-5})
river		1.470*		1.723*		9.795***		0.9834
		(0.8202)		(1.011)		(2.147)		(5.445)
coast		0.2086		-4.976**		1.889		0.8264
		(0.9177)		(2.244)		(3.480)		(4.543)
as.factor(mesorregi)4302			-2.720**	-2.543**				
			(1.065)	(1.256)				
as.factor(mesorregi)4303			-0.4829	-0.3830				
			(1.232)	(1.382)				
as.factor(mesorregi)4304			-0.7711	0.1960				
			(1.426)	(1.686)				
as.factor(mesorregi)4305			-3.023*	-1.290				
			(1.648)	(1.987)				
as.factor(mesorregi)4306			-1.724	-3.421				
			(2.130)	(2.350)				
as.factor(mesorregi)4307			-0.4368	0.3267				
			(2.751)	(2.881)				
Fit statistics								
Observations	549	548	467	467	42	42	40	39
R^2	0.04178	0.07299	0.09402	0.13481	0.16514	0.66887	0.00359	0.25135
Adjusted R^2	0.02938	0.04864	0.07618	0.10405	0.07488	0.54745	-0.05027	0.01901

Conley (11.112km) standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

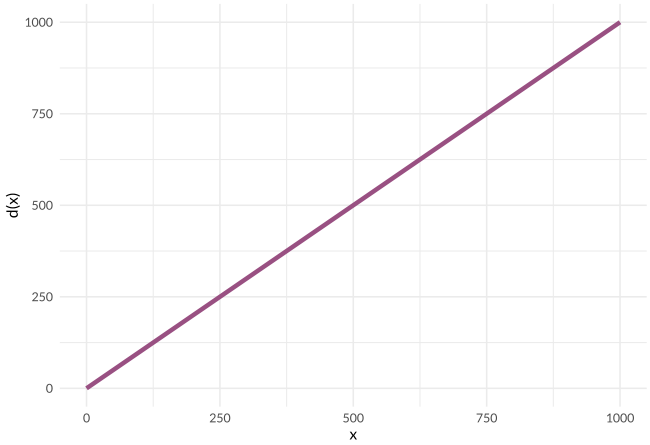
Evidently, we could not reproduce the exact Conley standard errors reported in Valencia Caicedo (2019). And the two packages yielded different errors even though we specified the same cutoff (11.112 kilometers) and the same method of distance calculation (spherical). Unfortunately, we could not pin down what the reason for those differences was.

Ways to Use Distance

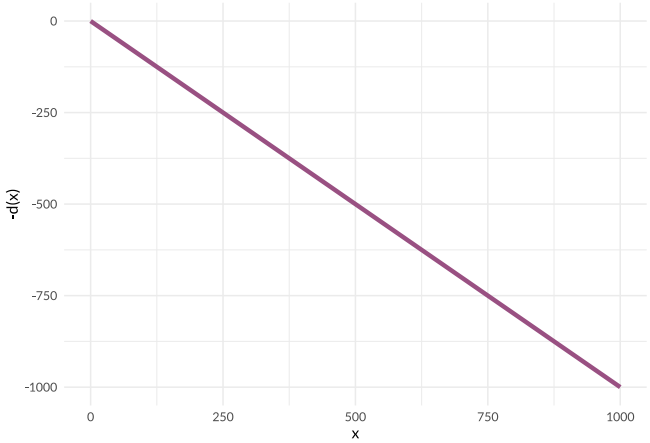
In his specification, Felipe Valencia Caicedo (2019) uses the following specification (Eq. 1 in the paper):

$$Y_{2000,ij} = \alpha + \beta d(M_{ij}) + \gamma GEO_{ij} + \mu_j + \varepsilon_{ij},$$

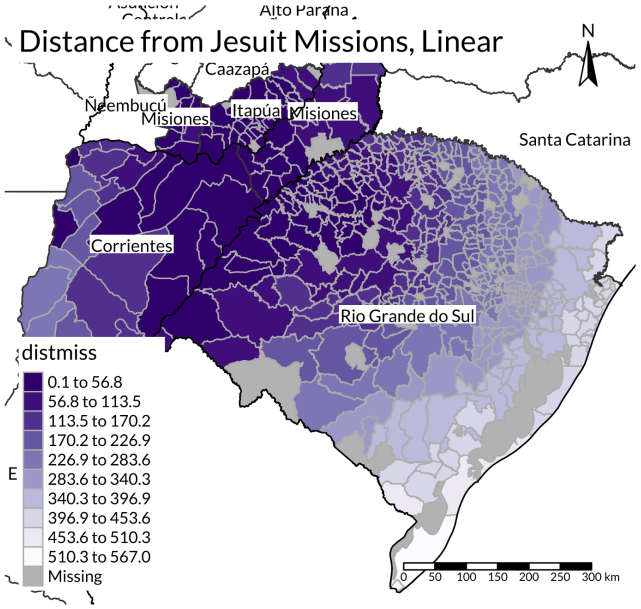
where $d(\cdot)$ is described as either being a dummy for the presence of a mission or the plain (i.e., untransformed) distance, although we do not find examples of the former in any of the printed tables in the paper, so we will just be calling it the distance. The decaying impact of a mission can be plotted against distance as follows:



If we view $d(\cdot)$ as *impact* exerted by a mission, it might be more intuitive to plot the negative of it, so that a higher value corresponds to more impact.



If we draw a map and plot distance from a mission using this function, we get the same map as above.



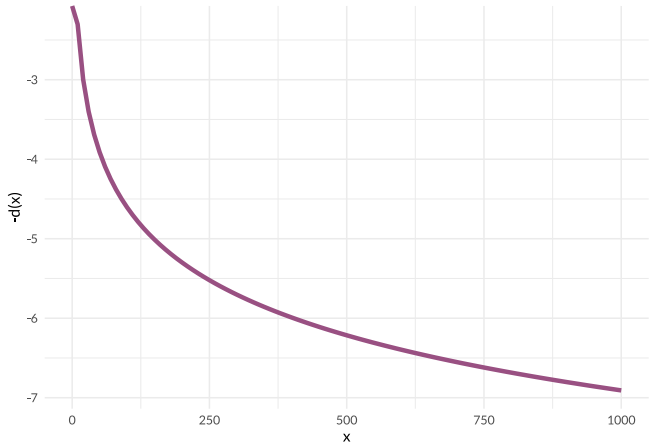
Letting d be

$$d : \mathbb{R} \to \mathbb{R}, \quad x \mapsto x$$

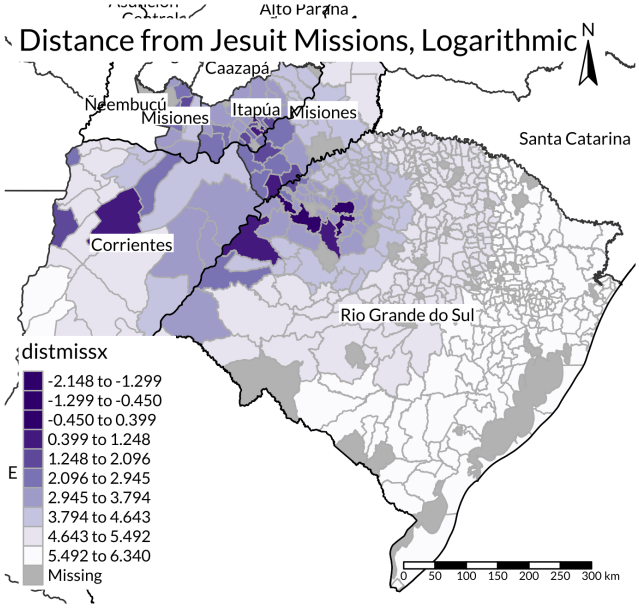
means that we attribute the same change in impact to moving from 0km from the treatment position to 100km from it, as to moving from 900km distance to 1000km. This is quite an assumption. Alternatively, we could use the **logarithm** of the distance,

$$d : \mathbb{R} \to \mathbb{R}, \quad x \mapsto \log(x),$$

like this:



This gives the following map.

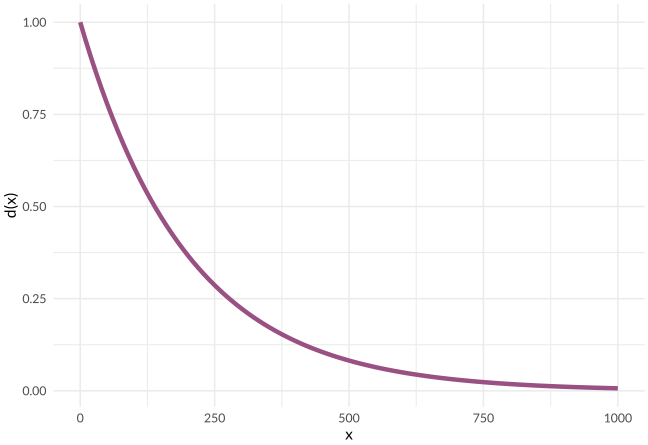


We can see that unlike in the linear case, the differences between distances at a greater length from the locations of treatment are weighted down, whereas distance differences closer to the mission are exaggerated. It could make sense to use such a specification instead of a linear one if we suspect impacts of a mission to be roughly comparable between places that are 400 or 500 kilometers from the closest mission, or if we are concerned that portions of Rio Grande do Sul are included in the analysis that are much farther from where Jesuits went than portions of Argentinian or Paraguayan states that were not included.

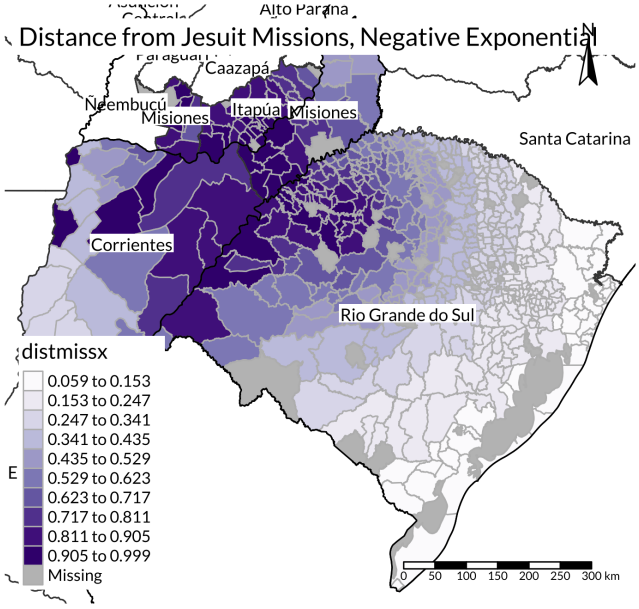
Alternatively, we could consider a **negative exponential** specification,

$$d : \mathbb{R} \to \mathbb{R}, \quad x \mapsto \exp(-\lambda x),$$

which is plotted below for $\lambda = \frac{1}{500}$:



Applying to our data, it looks like this:

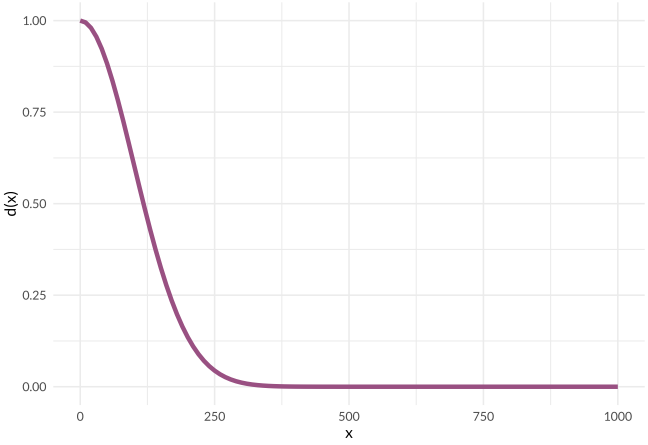


By adjusting λ, we can make the decay more or less "aggressive." With our present specification, it is "between" the linear and logarithmic cases.

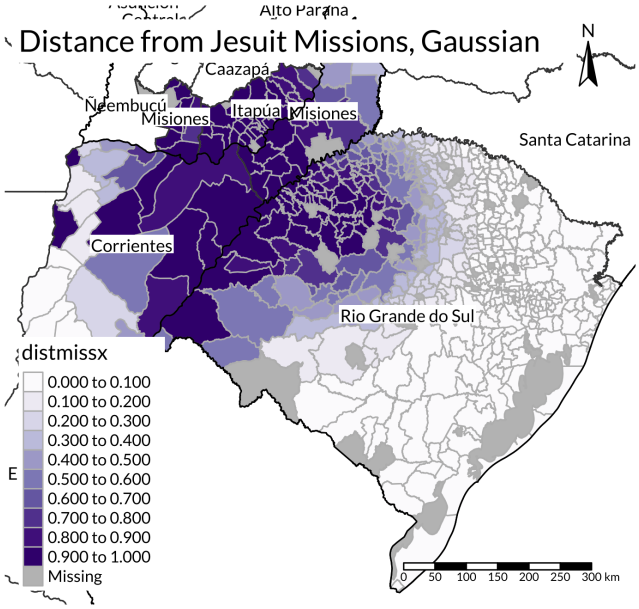
Lastly, we could consider using a **Gaussian** decay function, like this:

$$d : \mathbb{R} \to \mathbb{R}, \quad x \mapsto \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Letting $\sigma = 100$ gives the following:



We can see that by using the Gaussian decay function, we can model something that is closer to a "cutoff," i.e., the decay happens much more abruptly. Applying this function to the data gives the following map:



We can clearly see that different functions yield vastly different distance measures across municipalities. So it is quite sensible to suspect that estimation results are subject to the choice of distance measure. To check this, we run the regression from column (2) of Table 2 again – but each time with a different distance measure.

```
litrd <- litr %>%
    mutate(distmiss_log = log(distmiss), distmiss_exp = exp(-distmiss/200), distmiss_gau =
    exp(-(distmiss^2)/(2 *
        100^2)))

col2_reg <- lm(illiteracy ~ distmiss + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr + ita + mis + mis1, data = litrd)
col2_log <- lm(illiteracy ~ distmiss_log + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr + ita + mis + mis1, data = litrd)
col2_exp <- lm(illiteracy ~ distmiss_exp + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr + ita + mis + mis1, data = litrd)
col2_gau <- lm(illiteracy ~ distmiss_gau + lati + longi + area + tempe + alti + preci +
    rugg + river + coast + corr + ita + mis + mis1, data = litrd)
```

	Dependent variable:			
	illiteracy			
	(1)	(2)	(3)	(4)
distmiss	0.011**			
	(0.005)			
distmiss_log		0.078		
		(0.289)		
distmiss_exp			−3.293	
			(2.082)	
distmiss_gau				−1.631
				(1.246)
lati	0.070	−0.065	−0.216	−0.182
	(0.781)	(0.795)	(0.805)	(0.796)
longi	−1.007*	−0.360	−0.580	−0.454
	(0.556)	(0.536)	(0.535)	(0.539)
area	0.0001	0.0001	0.0001	0.0001
	(0.0002)	(0.0002)	(0.0002)	(0.0002)
tempe	0.059	−0.028	0.014	−0.009
	(0.077)	(0.071)	(0.075)	(0.070)
alti	0.006	0.002	0.003	0.002
	(0.004)	(0.003)	(0.003)	(0.003)
preci	−0.003	−0.003	−0.002	−0.002
	(0.002)	(0.002)	(0.002)	(0.002)
rugg	−0.00000	−0.00000	−0.00000	−0.00000
	(0.00000)	(0.00000)	(0.00000)	(0.00000)
river	1.470**	1.305*	1.292*	1.330*
	(0.712)	(0.717)	(0.710)	(0.713)
coast	0.209	0.564	0.578	0.652
	(0.894)	(0.874)	(0.890)	(0.892)
corr	−6.032***	−3.740***	−4.638***	−4.191***
	(1.583)	(1.419)	(1.451)	(1.451)
ita	−2.409***	−1.922**	−1.776**	−1.823**
	(0.833)	(0.888)	(0.841)	(0.832)
mis	−4.734***	−3.296**	−3.489**	−3.389**
	(1.488)	(1.398)	(1.364)	(1.345)
mis1	−2.299**	−2.212**	−1.970**	−2.072**
	(0.974)	(1.042)	(0.998)	(0.985)
Constant	−53.741*	−3.482	−28.162	−16.605
	(32.497)	(28.517)	(29.882)	(26.957)
Observations	548	548	548	548
R^2	0.073	0.063	0.067	0.066
Adjusted R^2	0.049	0.038	0.043	0.041
Residual Std. Error (df = 533)	3.912	3.934	3.924	3.927
Note:				*p<0.1; **p<0.05; ***p<0.01

Of course, coefficient magnitudes cannot be directly compared, since we wildly scaled and transformed the distance measure. Even the sign changes do not necessarily need to worry us, since we transformed distance measures in a way that they sometimes shrank and sometimes increased with increasing distance. However, we see that using the different distance measures, effect sizes are no longer statistically different from zero.

Persistence and Space, or: Where's Waldo?

Because any text about spatial autocorrelation would be incomplete without reciting Waldo[1] Tobler's First Law of Geography, here it is in all its beauty: "Everything is related to everything else, but near things are more related than distant things." Applied to the present context, this has important implications.

First, the presence of a mission in some given municipality is likely autocorrelated with whether adjacent municipalities were subject to missionary activity too. If the first missionaries went to some place in what we today call Misiones, it is likely that subsequent missionaries built on locally specific knowledge of earlier missions and settled near them. Our distance plots, especially the logarithmic one, are a good indication of that autocorrelation, since we can see that all missions were more or less in the same region.

Second, we can assume literacy rates to be spatially autocorrelated as well. Again, we can see this from the map we originally produced where we plotted literacy rates across the regions analyzed by Valencia Caicedo (2019). We can see that high literacy rates concentrate in one region around the Argentinian state of Misiones and the northwestern part of Rio Grande do Sul, and in one region in the eastern part of Rio Grande do Sul. And if we take a glance at the map below, we can also see that these regions are clusters of higher population density.

Kelly (2021) notes that this covert relation leads to standard errors usually being too low since the presence of spatial clusters in two variables means that whatever correlation exists between the variables in one locality is likely to also be present in adjacent localities. He also notes that different types of available standard error corrections lead to widely different results. Conley standard errors, which Valencia Caicedo (2019) uses as a robustness check, tend to be the most conservative of the corrections. Having standard errors that are too small, of course, does not bias the estimate, but will cause false positive results.

The same can be thought of if there are potential confounders. If there is a unobserved spatial variable that influences the outcome and that is correlated with the covariates that are included in the model, the coefficient estimates will be biased (Dupont et al., 2023). Assuming the variables considered in the model in Valencia Caicedo (2019) behave like most other socioeconomic variables in space, we can assume that any spatial effect will be correlated with the model covariates, leading to the possibility of biased coefficients.

[1] Here he is.

Task C

The Perils of Ignoring Peer Effects

In "The perils of peer effects", Angrist (2014) claims that without covariates, peer effects are vacuous and designs that "manipulate peer characteristics [...] have mostly uncovered little in the way of socially significant causal effects" (Angrist, 2014) While these assumptions may hold in theory and under the assumptions of IV, it is important to consider the potential impact of personal effects. Neglecting to do so could result in implications that may compromise the validity of a method or overlook important social dynamics. Peer groups hold significant power within society, from shaping individual behaviors to influencing broader social outcomes. It is important to acknowledge the complexity of social phenomena and avoid reducing them to mere statistical forces. By doing so, we run the risk of oversimplifying social dynamics and human behavior.

In educational research, peer effects are an important consideration, particularly within experimental contexts. Several studies confirm that peers can have a significant impact on an individual's academic performance, and (often even more pronounced) on different non-academic variables such as drinking behavior (Sacerdote, 2011). In applied econometric research, we are usually interested in inferring what influences certain behavior. In the education example, it may lead to incorrect or even dangerous policy implications if a peer effect is wrongly attributed to some characteristic. Therefore, it is in our interest as researchers to model peer effects where they occur.

In research designs, it is important to pay attention to both internal and external validity. When researchers prioritize internal validity in experimental settings to establish causality, neglecting peer effects can compromise external validity and limit the generalizability of findings to real-world contexts where peer influence is prevalent. For instance, it is possible that the outcomes of an experimental intervention conducted in a controlled laboratory environment may differ from those obtained when implementing the same intervention in an actual classroom setting where peer interactions are present and play an important role.

Another aspect concerns the use of instrumental variables. When using an instrument, we want it to be both *relevant*, i.e., correlated with the endogenous explanatory variable, and *valid*, i.e., exogenous. Spatial (or another kind of) network effects can lead to similar measurements of some instrumental variable "clustering together." If those observations of that instrumental variable are then used to explain another variable (the endogenous variable in what we would call the second stage regression), and if that variable were to exhibit the same kind of spatial dependency, then, by the argument we made at the end of Task C above, the significance of the relation between the instrument and the endogenous variable could be exaggerated. With that, an instrument may incorrectly be identified as *relevant*. In addition, unaccounted autocorrelation would make an argument about the exogeneity of the instrument difficult, as spatial clustering would cause the instrument to be correlated with the error term. As Betz et al. (2020) notes, this would render the instrument *invalid* and yield a biased second-stage regressor.

It is not difficult to imagine circumstances where such problems are prone to arise. Consider a reasercher who wants to estimate the effect of crop yields on education, hypothesizing that more agricultural output makes for better-nourished children who perform better in school. Since a more educated society might influence crop yields positively, the researcher uses precipitation as an instrument, arguing that it is exogenous since it cannot possibly be influenced by the level of education, and that it is relevant since more rain benefits agricultural activity. However, this weather-based instrument is almost certainly autocorrelated since precipitation depends on geographic conditions that cluster adjacent regions, and education is likely spatially autocorrelated in a similar way as the literacy rates considered in Task B. Therefore, through the mechanisms outlined above, the effect of precipitation may be both biased and its standard errors underestimated in such an analysis.

Therefore, while Angrist (2014) raises important points about the perils of *considering* peer effects, there are non-trivial perils that are associated with *ignoring* them, too.

Task D

The image is a screenshot, and those are conventionally stored in PNG format. The photo *contained* in the screenshot is a photograph, and it is difficult to guess which format it was originally saved in. Let's say it's JPEG. Then, someone inserted the image into the assignment PDF, meaning it is technically not stored as a PNG anymore. What all those ways of storing the image have in common is that they are **raster formats**, as they consist of individual pixels. And even if we print the document containing the image, it gets printed as dots, which are not exactly pixels, but certainly form a raster rather than a vector.

References

Angrist, J. D. (2014). The perils of peer effects. *Labour Economics*, *30*, 98–108. https://doi.org/10.1016/j.labeco.2014.05.008

Betz, T., Cook, S. J., & Hollenbach, F. M. (2020). Spatial interdependence and instrumental variable models. *Political Science Research and Methods*, *8*(4), 646–661. https://doi.org/10.1017/psrm.2018.61

Dupont, E., Marques, I., & Kneib, T. (2023). *Demystifying Spatial Confounding* (Working Paper). arXiv. https://doi.org/10.48550/arXiv.2309.16861

Kelly, M. (2021). *Persistence, randomization, and spatial noise* (Working Paper). Dublin: University College Dublin, UCD School of Economics. https://www.econstor.eu/handle/10419/246498

Sacerdote, B. (2011). Peer Effects in Education: How Might They Work, How Big Are They and How Much Do We Know Thus Far? *Handbook of the Economics of Education* (pp. 249–277). Elsevier. https://doi.org/10.1016/B978-0-444-53429-3.00004-1

Valencia Caicedo, F. (2019). The Mission: Human Capital Transmission, Economic Persistence, and Culture in South America. *Q. J. Econ.*, *134*(1), 507–556. https://doi.org/10.1093/qje/qjy024