Spatial Economics – Assignment 2

Gustav Pirich (h11910449) Peter Prlleshi () Filip Lukijanovic ()

April 2, 2024

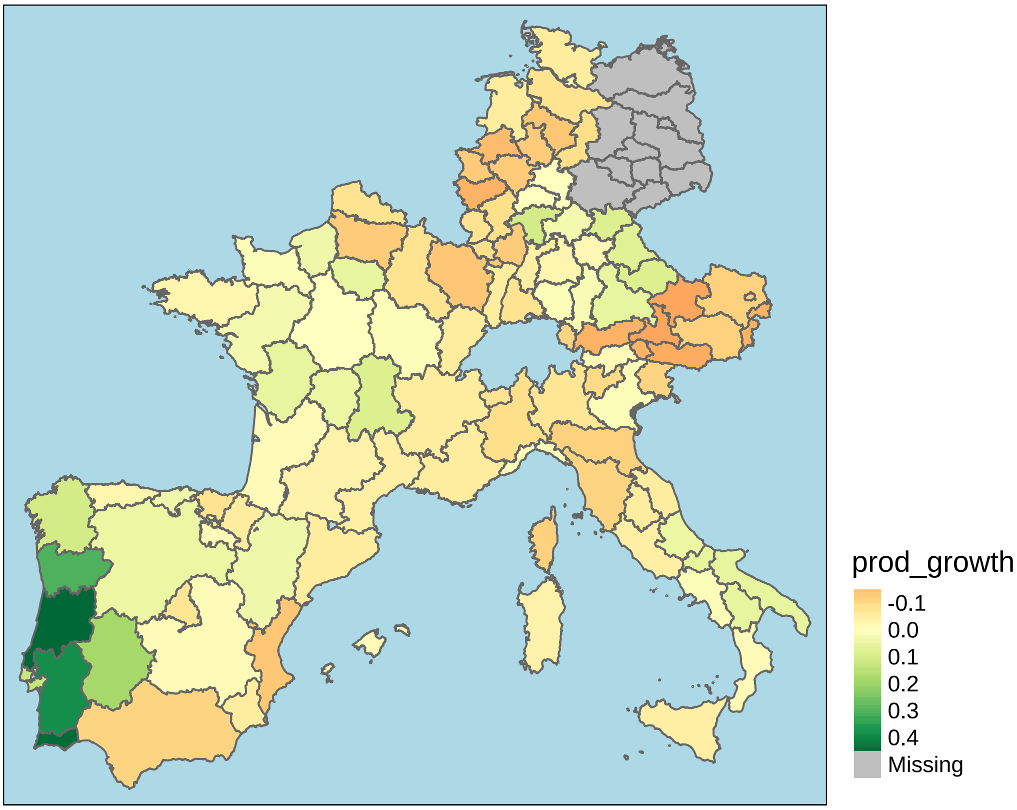
Contents

The code that was used in compiling the assignment is available on GitHub at https://github.com/gustavpirich/spatial_econ/blob/main/02_assignment/02_assignmnet.Rmd.

Exercise A

Calculate the growth rate of productivity from 1980 to 2013 and create a map that shows the productivity growth for each region.

The map shows the productivity growth rates in the NUTS-2 regions for the selected countries. We can see that many regions especially in Germany, Austria, and France exhibited negative productivity growth over the selected time period. Notably, Portugal's productivity has been growing the fastest. We suspect that the negative growth rates can be explained by the fact that high-income countries had a high baseline productivity to being with, while Portugal had a low baseline productivity. This could be evidence of convergence among productivity differences across Europe.



Generate three different spatial weights matrixes using (i) a distance threshold, (ii) smooth distance-decay, and iii) a contiguity-based measure.

(i) Distance Threshold

We first create a spatial weights matrix based on the distance threshold:

```
coords <- st_coordinates(st_centroid(EU27))

distw <- dnearneigh(coords, 0, 10, row.names=EU27$Id)
summary(distw)
```

```
## Neighbour list object:
## Number of regions: 114
## Number of nonzero links: 7386
## Percentage nonzero weights: 56.83287
## Average number of links: 64.78947
## Link number distribution:
##
## 18 19 22 24 25 27 29 30 31 33 35 36 37 39 40 41 42 45 48 49 50 51 55 57 59 61
##  2  1  1  3  2  1  1  2  2  2  1  1  1  2  1  1  1  1  1  2  1  1  2  1  3  1
## 65 66 67 68 69 70 71 72 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91
##  4  1  2  3  2  2  5  5  2  2  3  5  2  4  1  5  2  4  3  5  3  4  1  1  2  2
## 94
##  1
## 2 least connected regions:
## PT15 PT17 with 18 links
## 1 most connected region:
## ITC1 with 94 links
```

```
#plot(coords)
#plot(distw, coords, add=TRUE, col="green", cex=0.1)

# Create a distance threshold spatial weights matrix
thresh_dist <- 10  # Threshold distance in km
dist_w <- dnearneigh(coords, 0, thresh_dist)
dist_w_matrix <- nb2mat(dist_w, style="W", zero.policy=TRUE)
```

(ii) Smooth-Distance Decay

We now create a spatial weights matrix based on a smooth distance-decaqy

```
# Calculate distances between all pairs
distances <- sp::spDists(coords, longlat=TRUE)
# Apply a distance decay function (inverse distance in this case)
decay_function <- function(d) exp(-0.01*d)
decay_weights_matrix <- decay_function(distances)

decay_weights_matrix <- decay_weights_matrix - diag(nrow(decay_weights_matrix))
```

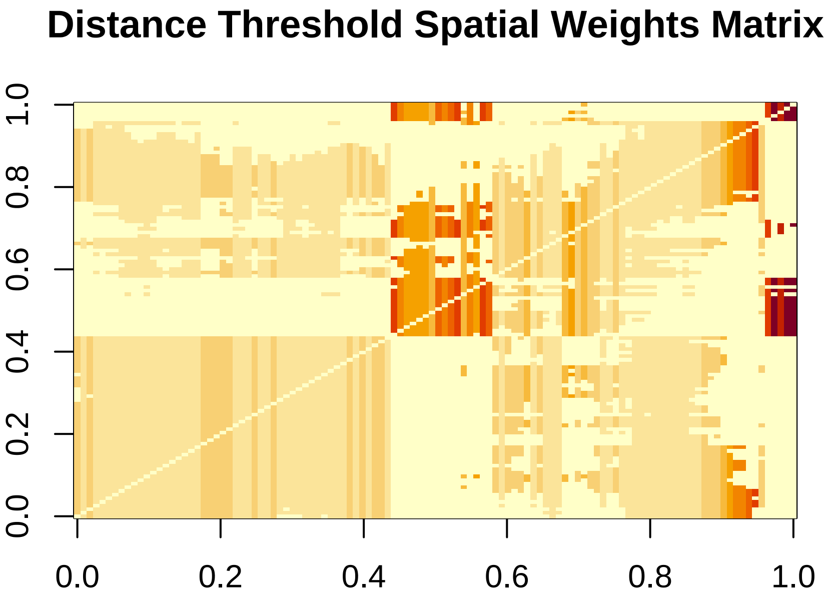
(iii) Contiguity-based measure

```
# Create a contiguity-based spatial weights matrix
contig_w <- poly2nb(EU27)
contig_w_matrix <- nb2mat(contig_w, style="W", zero.policy=TRUE)
```

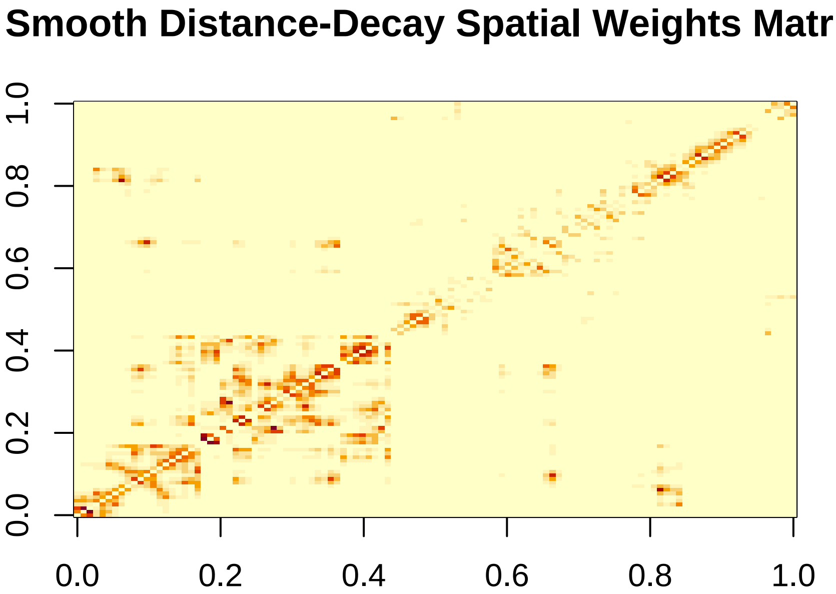
Compare teh matrices; use your knowledge of graph theory and linear algebra

Plot the matrix

```
image(dist_w_matrix, main="Distance Threshold Spatial Weights Matrix")
```

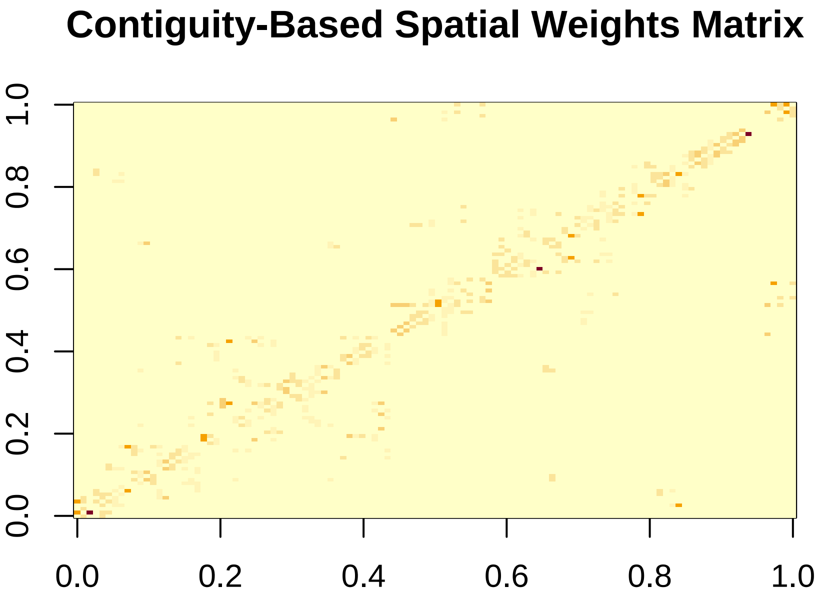


```r
image(decay_weights_matrix, main="Smooth Distance-Decay Spatial Weights Matrix")
```



Smooth Distance-Decay Spatial Weights Matrix

```r
image(contig_w_matrix, main="Contiguity-Based Spatial Weights Matrix")
```



Contiguity-Based Spatial Weights Matrix