Lecture 05: Spatial Data

Theory and Tools (a.k.a. GIS Tools Lab.)

Bruno Conte

02/Oct/2023

Spatial data in economics: schedule

1. ~~Introduction to (spatial) data and programming in R~~ [18.Sep.2023]
2. ~~Spatial data basics: vector data + assignment~~ [21.Sep.2023]
3. ~~Basic operations with vector data + assignment~~ [25.Sep.2023]
4. ~~Geometry operations and miscelanea + follow-up + assignment~~ [28.Sep.2023]
5. Raster data and operations + assignment [02.Oct.2023]
   - Raster basics: creating and loading rasters with `terra`
   - Operations: unary and vector-raster tools
   - Students' feedback survey
6. Take-home exam [03.Nov.2023]

Main references for this class

1. Lovelace, R., Nowosad, J. and Muenchow, J., 2019. **Geocomputation with R.** Chapman and Hall/CRC.
    - Chapters 2.3, 3.3, 4.3, 5.3, and 6
2. Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446
3. Wickham, H. and Grolemund, G., 2016. R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc.".

Raster data: basics

- GIS systems represent raster data as an "image":
    - Geography as continuum of pixels (gridcells) with associated values
    - Normally represents high resolution features of the geography (like an image)

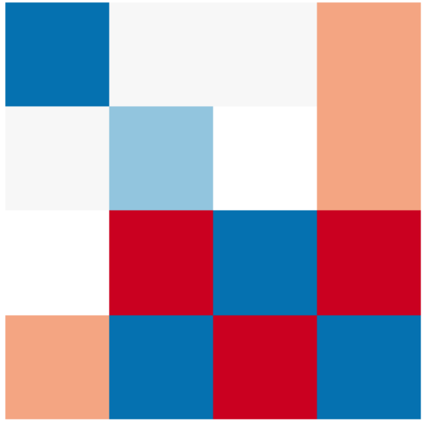
A. Cell IDs

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

B. Cell values

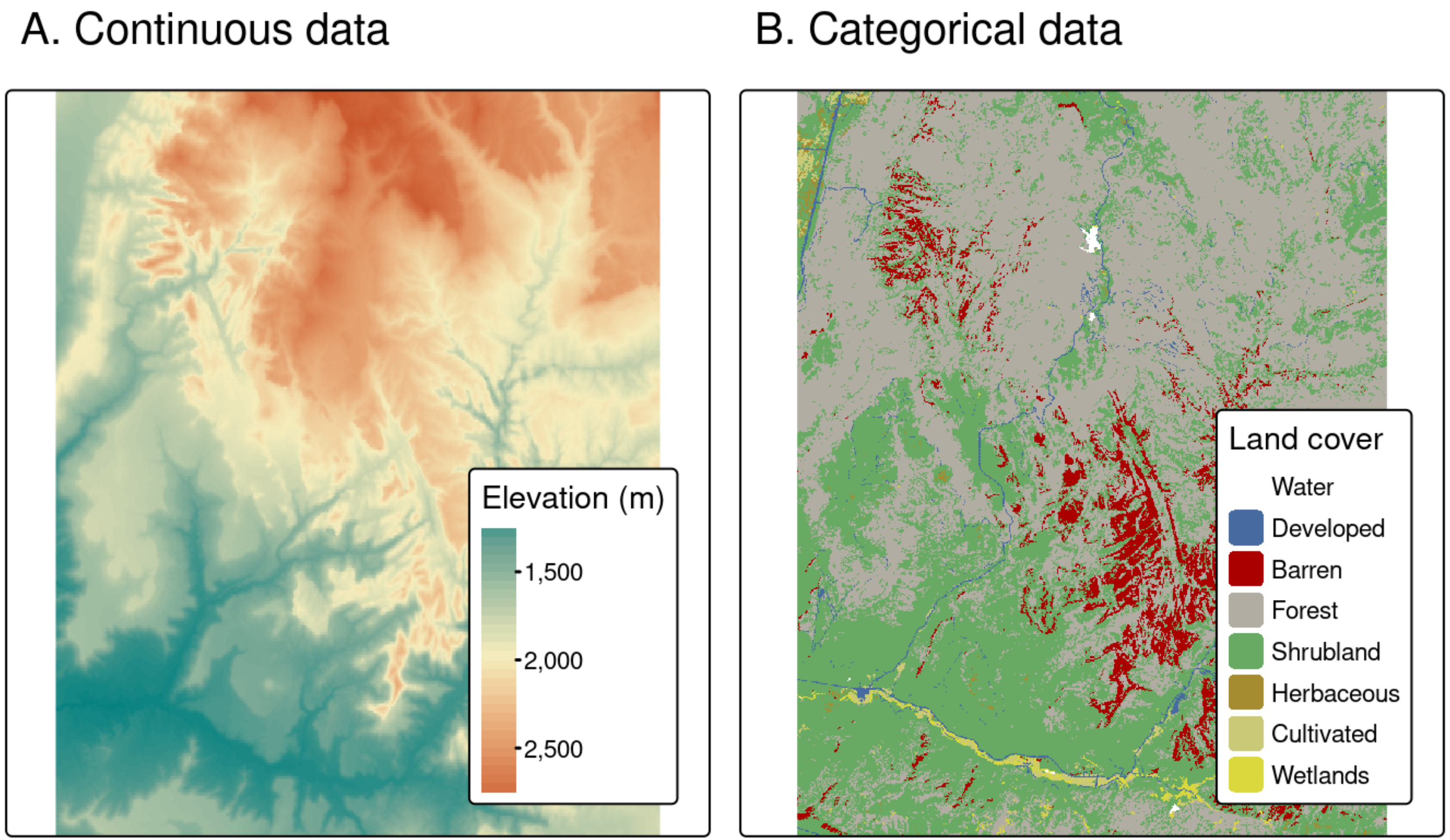
92	55	48	21
58	70	NA	37
NA	12	94	11
36	83	4	88

C. Colored values



Raster data: basics

- Normally represents high resolution features of the geography (like an image)



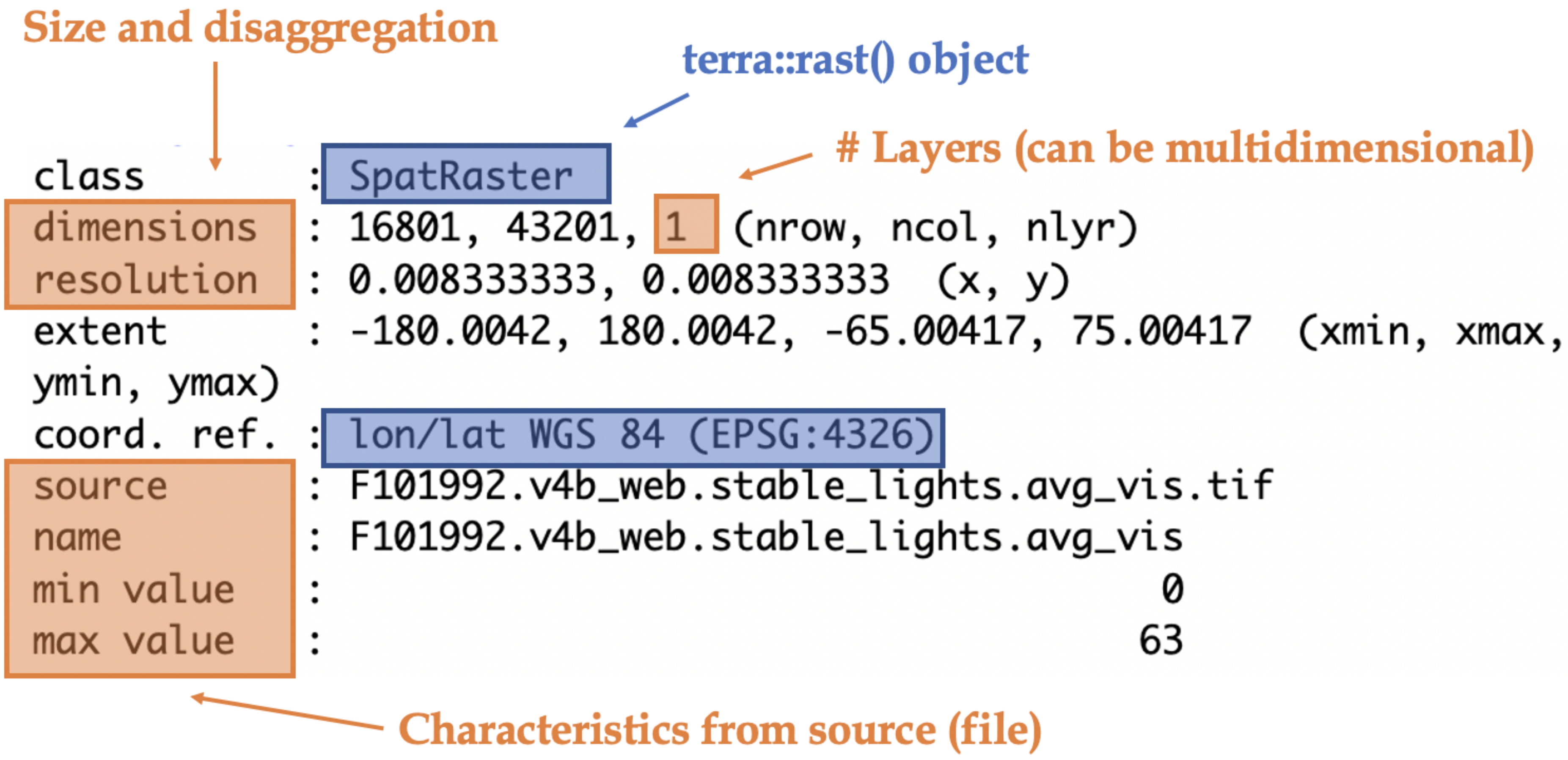
Raster data (and other operations with rasters) in R

Requires additional libraries/packages than `sf`

1. `terra`: contains most of the raster-related functions
2. `exactextractr`: performs high-performance zonal statistics
3. `gdistance`: used to calculate distances over raster

Raster basics: loading and creating with terra

- **Raster data:** represented with `terra`'s `SpatRaster` object

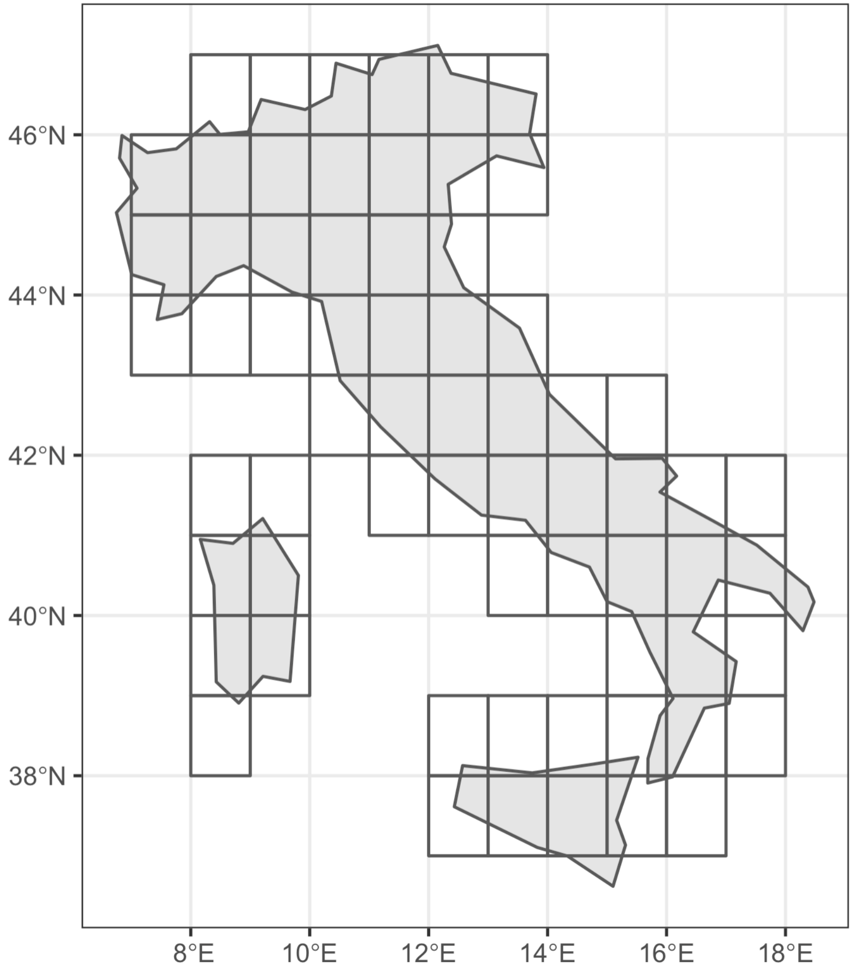


Your turn: Hands-in

Hands-in: your turn! (1/3)

Dividing Italy in **gridcells**

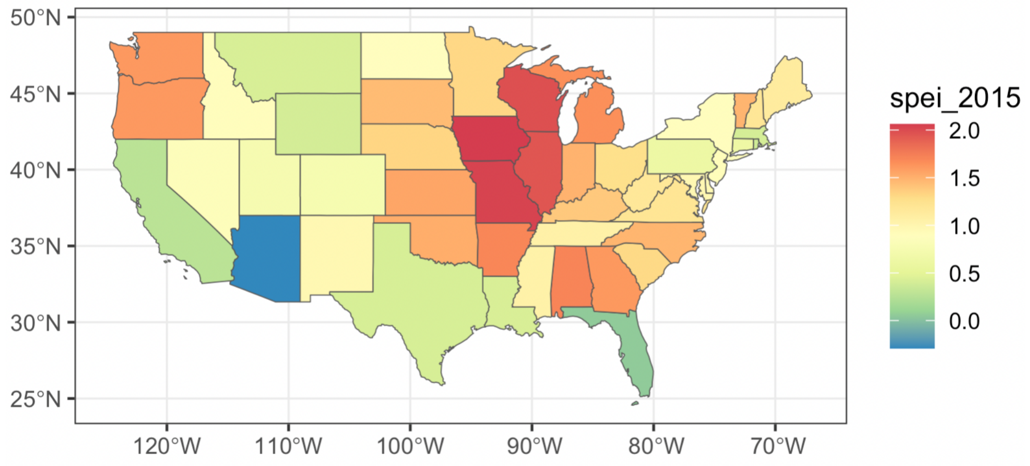
- Create a 1 x 1 degree raster
- Convert it to polygon (i.e. create the grid)
- Use `world` data filtered to Italy, keep gridcells that **intersect** with Italy
- Visualize it:

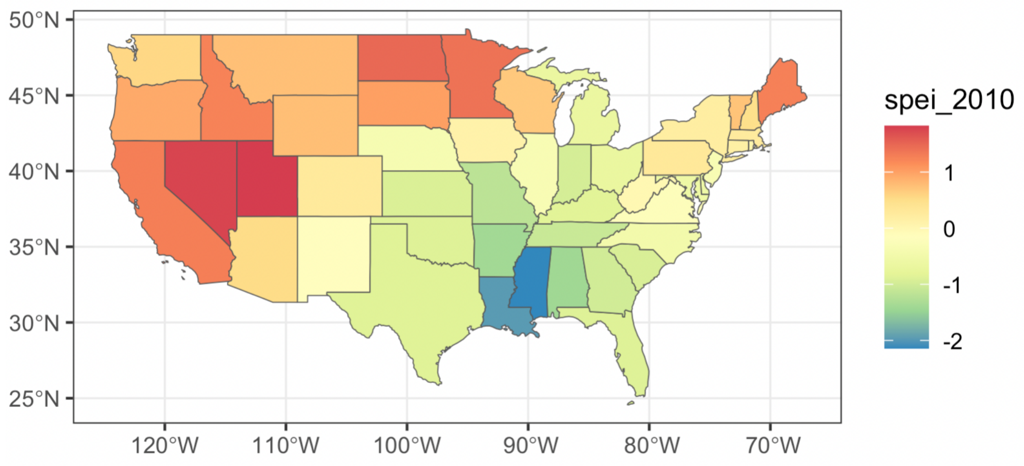


Hands-in: your turn! (2/3)

Calculating **climate change** in USA

- Use the `us_states` data on the geography of US states
- Combine it with the SPEI index:
  - Retrieve average SPEI index across states
  - Do so for 3-4 different years
- Visualize it:

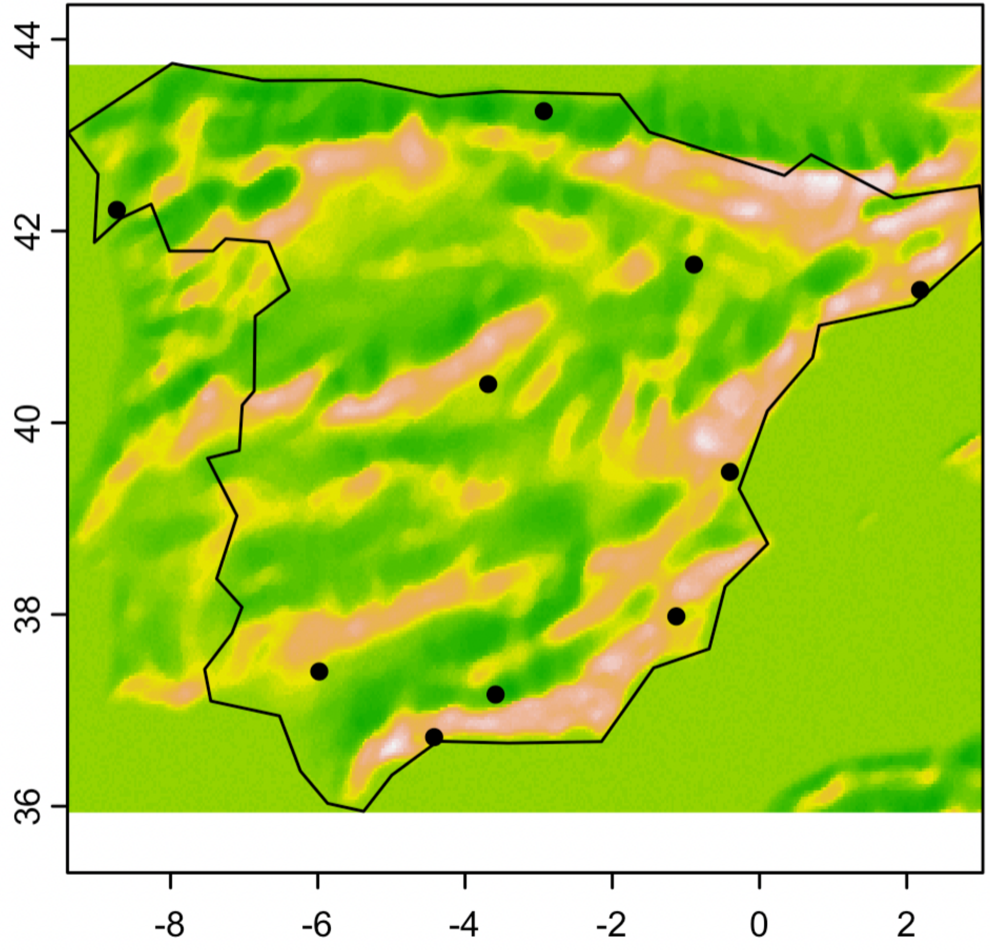




Hands-in: your turn! (3/3)

Geography and **bilateral distances** in Spain

- Use the `ne_10m_populated_places` shapefile to retrieve the **10 top-populated places** in Spain
- Crop the elevation data from `MSR_50M.tif` raster with Spain
- Visualize them together with `plot()` function
- Calculate the path and distance between Madrid and Vigo
  - Hint: approx. 640 km!

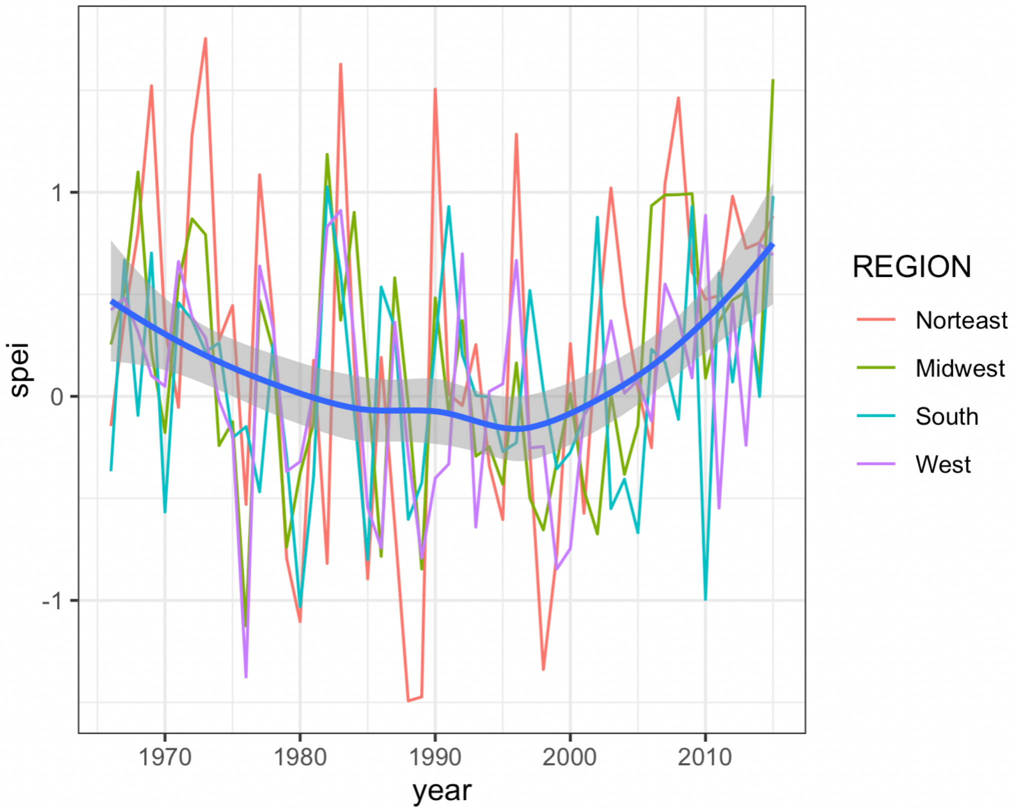


Your turn: Take-home Assignment

Take-home assignment (1/2)

Calculating **climate change** in USA

- Use the `us_states` data on the geography of US states
- Retrieve average SPEI index across **regions** for the past 50 years
- Retrieve the dataset as a panel (time series for each region)
- Plot the evolution of the SPEI index for each region
  - `geom_smooth()`: calculate the average across regions



Take-home assignment (2/2)

Transportation centrality and isolation in Spain

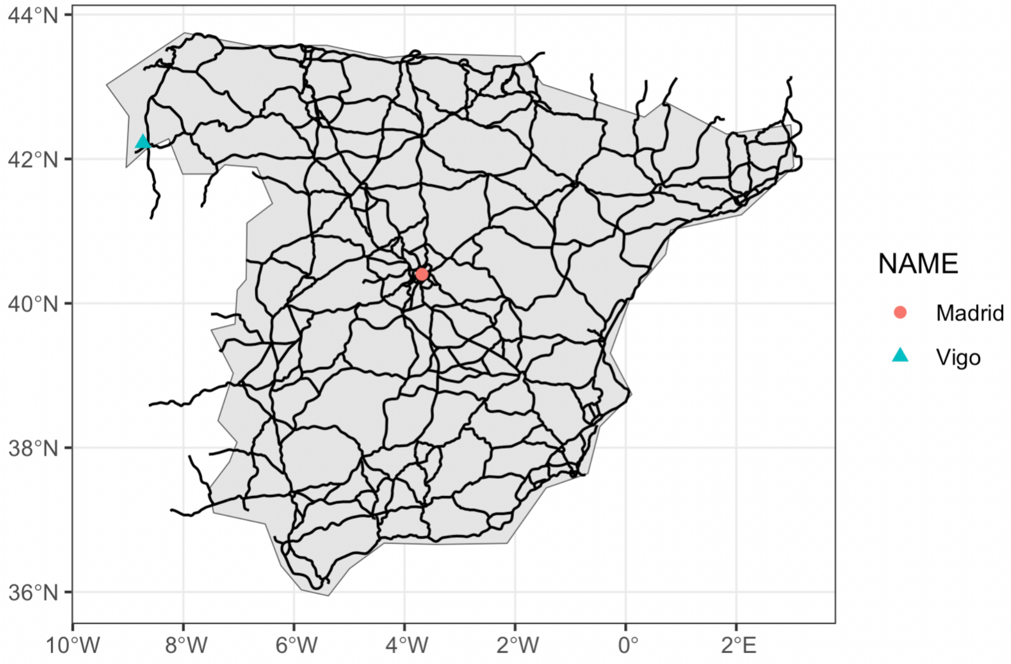
- Use the `ne_10m_populated_places` shapefile to retrieve the **10 top-populated places**

- Crop the `ne_10m_roads` road data within Spain

- Build a raster/friction surface; calculate distances between **all city pairs**

- Bilateral distances if coming from Madrid vs. Vigo: who is more isolated?

    - `geom_density()`: calculates "smoothed" distributions



Take-home assignment (2/2)

Transportation centrality and isolation in Spain

- Use the `ne_10m_populated_places` shapefile to retrieve the **10 top-populated places**

- Crop the `ne_10m_roads` road data within Spain

- Build a raster/friction surface; calculate distances between **all city pairs**

- Bilateral distances if coming from Madrid vs. Vigo: who is more isolated?
    - `geom_density()`: calculates "smoothed" distributions