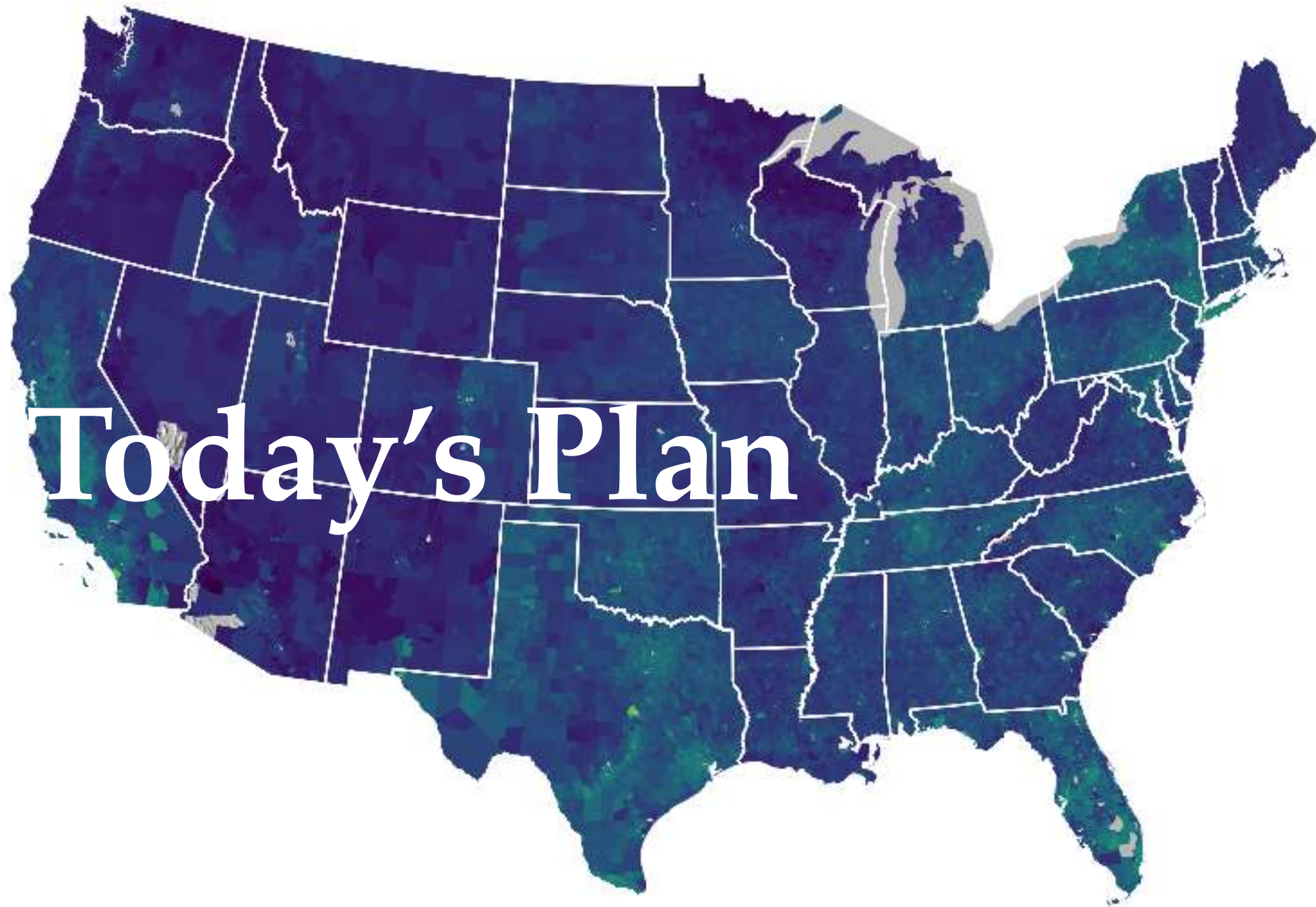


Building Spatial Databases with Attributes

HES 505 Fall 2022: Session 14

Matt Williamson



Objectives

- By the end of today, you should be able to:
 - Define *spatial analysis*
 - Describe the steps in planning a spatial analysis
 - Understand the structure of relational databases
 - Begin building a database for spatial analysis

What is spatial analysis?

What is spatial analysis?

“The process of examining the locations, attributes, and relationships of features in spatial data through overlay and other analytical techniques in order to address a question or gain useful knowledge. Spatial analysis extracts or creates new information from spatial data”.

— ESRI Dictionary

What is spatial analysis?

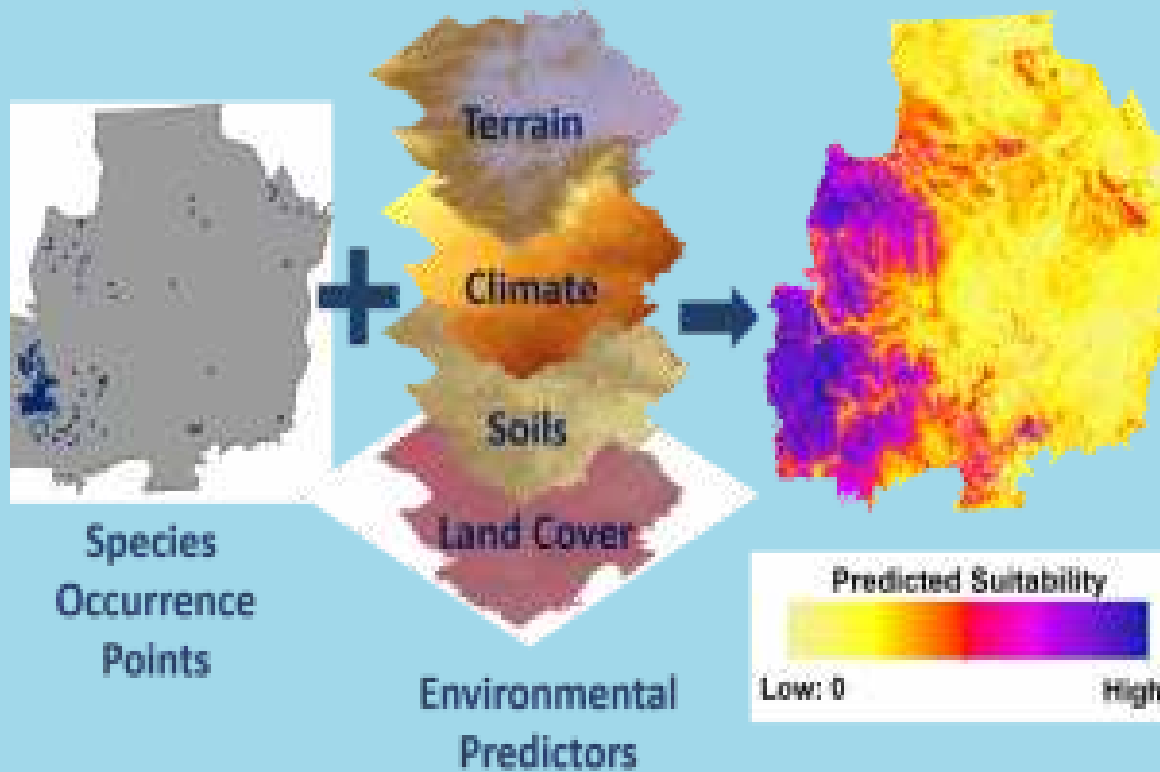
- The process of turning maps into information
- Any- or everything we do with GIS
- The use of computational and statistical algorithms to understand the relations between things that co-occur in space.



John Snow's cholera outbreak map

Common goals for spatial analysis

Building a Model



- Describe and visualize locations or events
- Quantify patterns
- Characterize 'suitability'
- Determine (statistical) relations

courtesy of NatureServe

Common pitfalls of spatial analysis

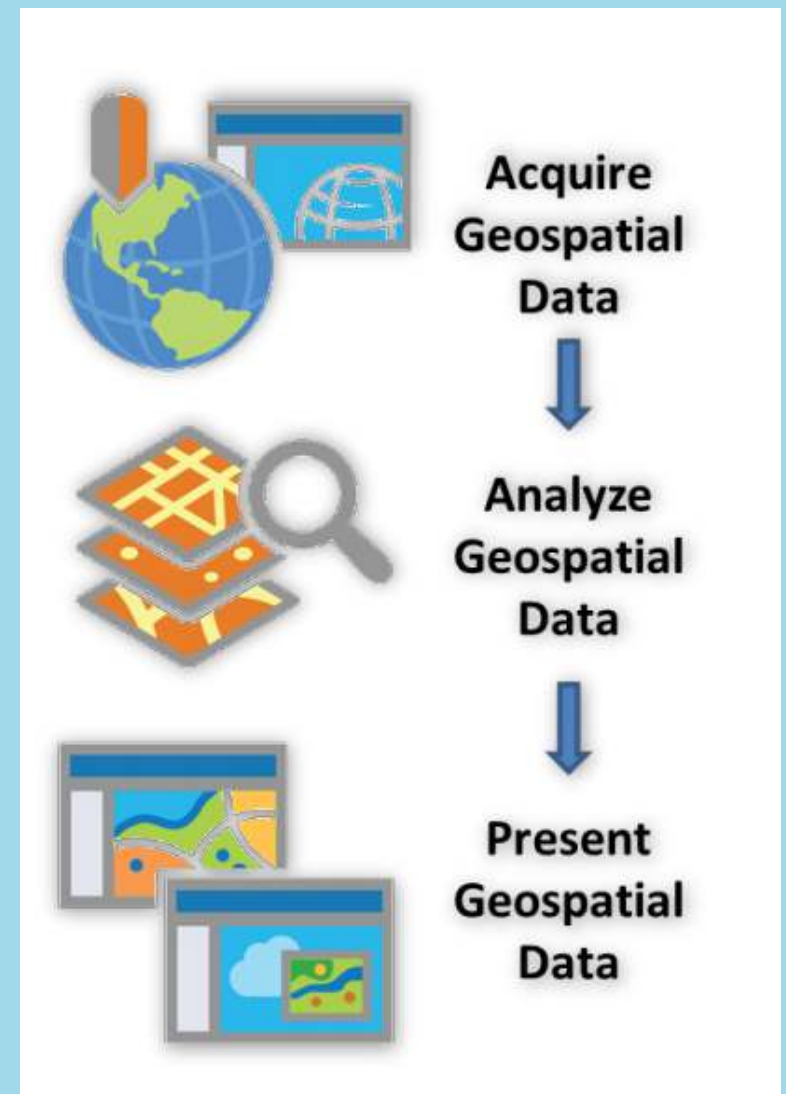
- **Locational Fallacy:** Error due to the spatial characterization chosen for elements of study
- **Atomic Fallacy:** Applying conclusions from individuals to entire spatial units
- **Ecological Fallacy:** Applying conclusions from aggregated information to individuals

Spatial analysis is an inherently complex endeavor and one that is advancing rapidly. So-called “best practices” for addressing many of these issues are still being developed and debated. This doesn’t mean you shouldn’t do spatial analysis, but you should keep these things in mind as you design, implement, and interpret your analyses

Workflows for spatial analysis

Workflows for spatial analysis

- Acquisition (not really a focus, but see **Resources**)
- Geoprocessing
- Analysis
- Visualization



courtesy of **University of Illinois**

Geoprocessing

Manipulation of data for subsequent use

- Alignment
- Data cleaning and transformation
- Combination of multiple datasets
- Selection and subsetting

Databases and attributes

Databases and attributes

| | A | B | C | D | E | F |
|----|----------------|-------------|----------------|---------|-----------------|---------|
| | AREA | PERIMETER | APM | LANDUSE | LOT_SIZE | HERRING |
| 1 | 6474154.36275 | 13146.88872 | 00100400000000 | HFAJAG | 8790300.000000 | 40000 |
| 2 | 7070794.10172 | 15644.47836 | 00100400015000 | HFAJAG | 8969000.000000 | 40000 |
| 3 | 12903007.20884 | 18307.82117 | 00100200000000 | HFAJAG | 12229172.000000 | 40000 |
| 4 | 2843043.70203 | 7886.52182 | 00100400030000 | HFAJAG | 2741200.000000 | 40000 |
| 5 | 102725.06962 | 5215.30257 | 00100200000000 | WGACCA | 187300.000000 | 40000 |
| 6 | 38655.06531 | 3710.30011 | 00100200000000 | WGACCA | 26355.800000 | 40000 |
| 7 | 238716.26834 | 3578.41230 | 00100200000000 | WGACCA | 259106.800000 | 40000 |
| 8 | 1271388.36556 | 13888.48818 | 00100200000000 | HFAJAG | 13061100.000000 | 40000 |
| 9 | 9530649.19776 | 11620.31722 | 00100200000000 | HFAJAG | 8015157.600000 | 40000 |
| 10 | 2534654.40019 | 7726.17855 | 00100200000000 | HFAJAG | 2800472.400000 | 40000 |
| 11 | 2403652.50813 | 7081.64811 | 00100200000000 | HFAJAG | 2080000.900000 | 40000 |
| 12 | 4080660.54001 | 2997.10462 | 00100200000000 | WGACCA | 4420488.800000 | 40000 |
| 13 | 385170.08143 | 13027.80186 | 00100100400000 | WGACCA | 432930.000000 | 40000 |
| 14 | 8702821.66378 | 13367.67718 | 00100100000000 | WGACCA | 8887982.400000 | 40000 |
| 15 | 1408916.88818 | 2488.00862 | 00100100000000 | WGACCA | 1484100.000000 | 40000 |
| 16 | 228970.51558 | 4888.26427 | 00100100150000 | WGACCA | 217364.400000 | 40000 |
| 17 | 1368014.23168 | 5011.54836 | 00100100110000 | WGACCA | 1152488.800000 | 40000 |
| 18 | 1815128.08861 | 713.48518 | 00100100140000 | WGACCA | 1504296.000000 | 40000 |
| 19 | 32486.36388 | 3274.84752 | 00100510010000 | ATBDBA | 35142.000000 | 40000 |
| 20 | 895458.08888 | 28027.24830 | 00101000000000 | WGACCA | 630092.400000 | 40000 |
| 21 | 3458710.31762 | 2465.20572 | 00100500010000 | WGACCA | 4154392.400000 | 40000 |
| 22 | 210706.26381 | 11026.88528 | 00101000000000 | WGACCA | 230295.200000 | 40000 |
| 23 | 798957.83178 | 8774.17183 | 00100530050000 | WGACCA | 20337.800000 | 40000 |
| 24 | 281178.57538 | 808.85834 | 00100100130000 | WGACCA | 235224.000000 | 40000 |
| 25 | 37128.21791 | 3205.72915 | 00100530080000 | ATBDBA | 36808.000000 | 40000 |
| 26 | 198741.88422 | | | | 181172.000000 | 40000 |

- Previous focus has been largely on *location*
- Geographic data often also includes non-spatial data
- Attributes: Non-spatial information that further describes a spatial feature
- Typically stored in tables where each row represents a spatial feature
 - Wide vs. long format

courtesy of [Giscommons](#)

Common attribute operations

- `sf` designed to work with `tidyverse`
- Allows use of `dplyr` data manipulation verbs
- Can use `scales` package for units
- Also allows `%>%` to chain together multiple steps
- geometries are “sticky”
- Pay attention to masking!!

Revisiting the tidyverse

Revisiting the tidyverse

- Features refer to the individual observations in the dataset
- Selecting features

```
1 head(world)[1:3, 1:3] %>%  
2   st_drop_geometry()
```

```
# A tibble: 3 × 3  
  iso_a2 name_long      continent  
* <chr>  <chr>          <chr>  
1 FJ     Fiji         Oceania  
2 TZ     Tanzania       Africa  
3 EH     Western Sahara Africa
```

```
1 world %>%  
2   filter(continent == "Asia") %>%  
3     dplyr::select(name_long, conti  
4     st_drop_geometry() %>%  
5     head(.)
```

```
# A tibble: 6 × 2  
  name_long      continent  
  <chr>          <chr>  
1 Kazakhstan Asia  
2 Uzbekistan Asia  
3 Indonesia Asia  
4 Timor-Leste Asia  
5 Israel Asia  
6 Lebanon Asia
```

Revisiting the tidyverse

- Creating new fields

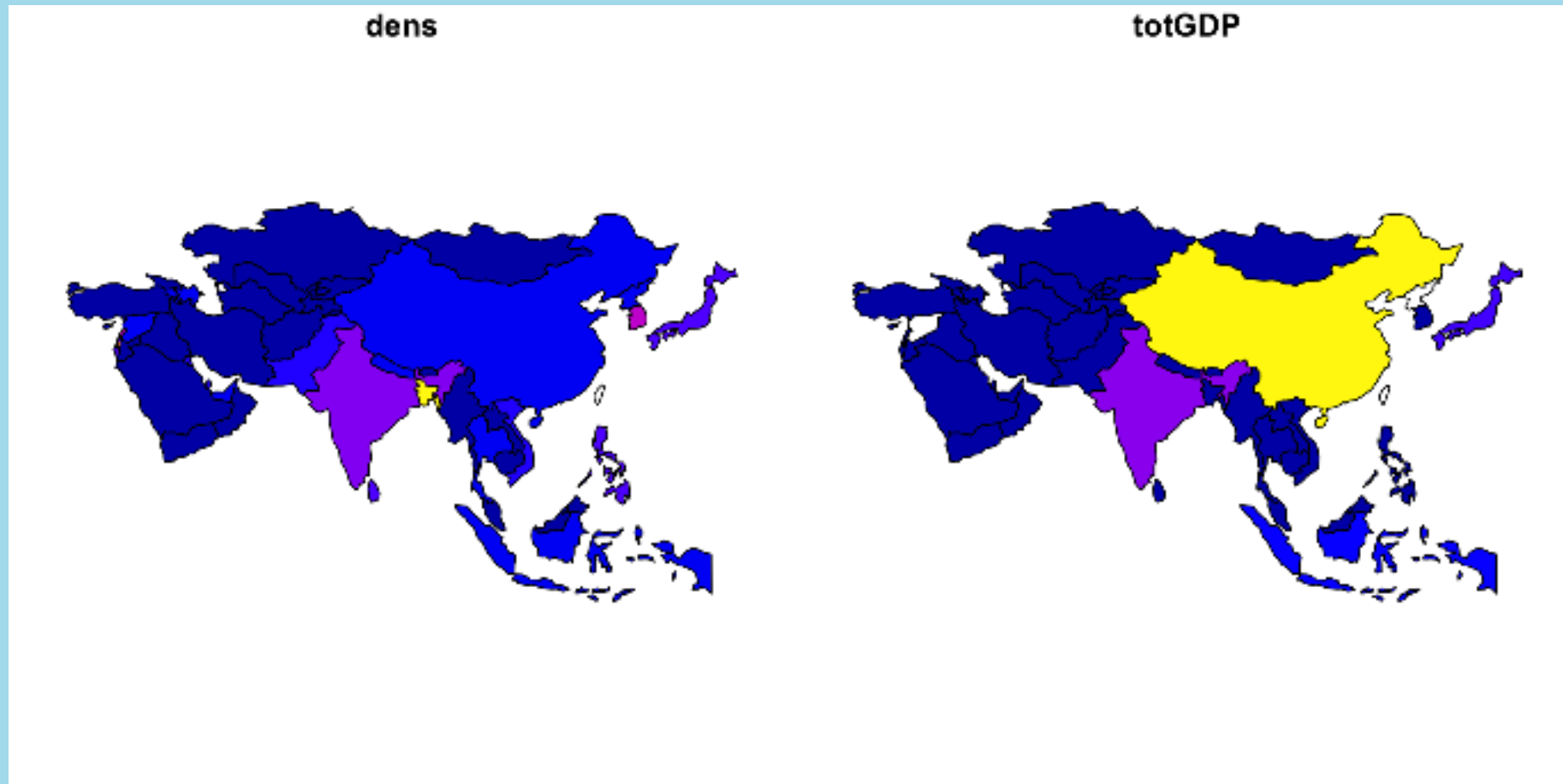
```
1 world %>%
2   filter(continent == "Asia") %>%
3   dplyr::select(name_long, continent, pop, gdpPercap ,area_km2) %>%
4   mutate(., dens = pop/area_km2,
5           totGDP = gdpPercap * pop) %>%
6   st_drop_geometry() %>%
7   head(.)
```

A tibble: 6 × 7

| | name_long | continent | pop | gdpPercap | area_km2 | dens | totGDP |
|---|-------------|-----------|-----------|-----------|----------|-------|---------|
| | <chr> | <chr> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
| 1 | Kazakhstan | Asia | 17288285 | 23587. | 2729811. | 6.33 | 4.08e11 |
| 2 | Uzbekistan | Asia | 30757700 | 5371. | 461410. | 66.7 | 1.65e11 |
| 3 | Indonesia | Asia | 255131116 | 10003. | 1819251. | 140. | 2.55e12 |
| 4 | Timor-Leste | Asia | 1212814 | 6263. | 14715. | 82.4 | 7.60e 9 |
| 5 | Israel | Asia | 8215700 | 31702. | 22991. | 357. | 2.60e11 |
| 6 | Lebanon | Asia | 5603279 | 13831. | 10099. | 555. | 7.75e10 |

Revisiting the tidyverse

- Creating new fields



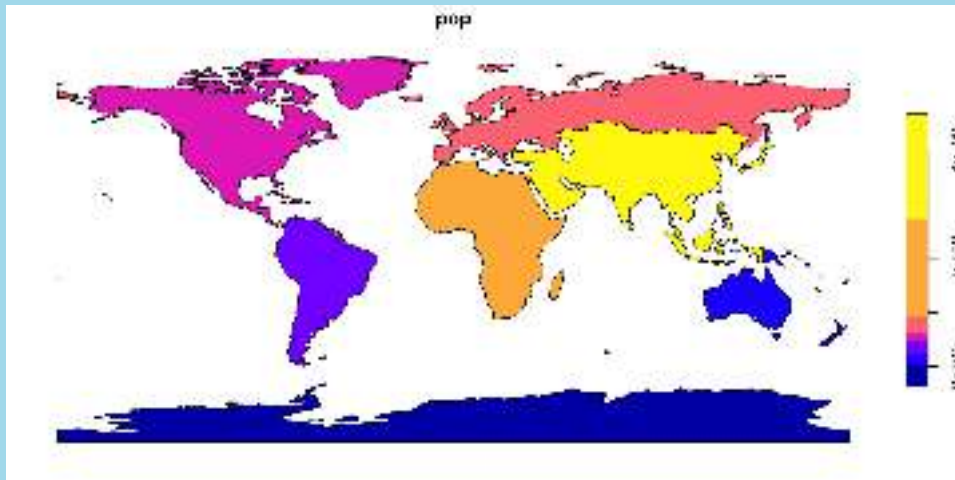
Revisiting the tidyverse

- Aggregating data

```
1 world %>%  
2   st_drop_geometry(.) %>%  
3   group_by(continent) %>%  
4   summarize(pop = sum(pop, na.rm =
```

```
# A tibble: 8 × 2
```

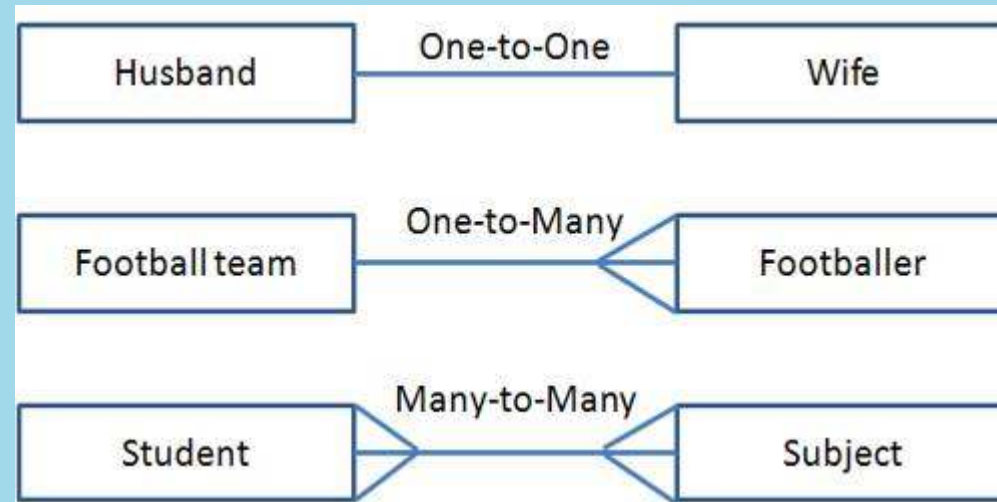
| | continent | pop |
|---|-------------------------|------------|
| | <chr> | <dbl> |
| 1 | Africa | 1154946633 |
| 2 | Antarctica | 0 |
| 3 | Asia | 4311408059 |
| 4 | Europe | 669036256 |
| 5 | North America | 565028684 |
| 6 | Oceania | 37757833 |
| 7 | Seven seas (open ocean) | 0 |
| 8 | South America | 412060811 |



Joining (a)spatial data

Joining (a)spatial data

- Requires a “key” field
- Multiple outcomes possible
- Think about your final data form

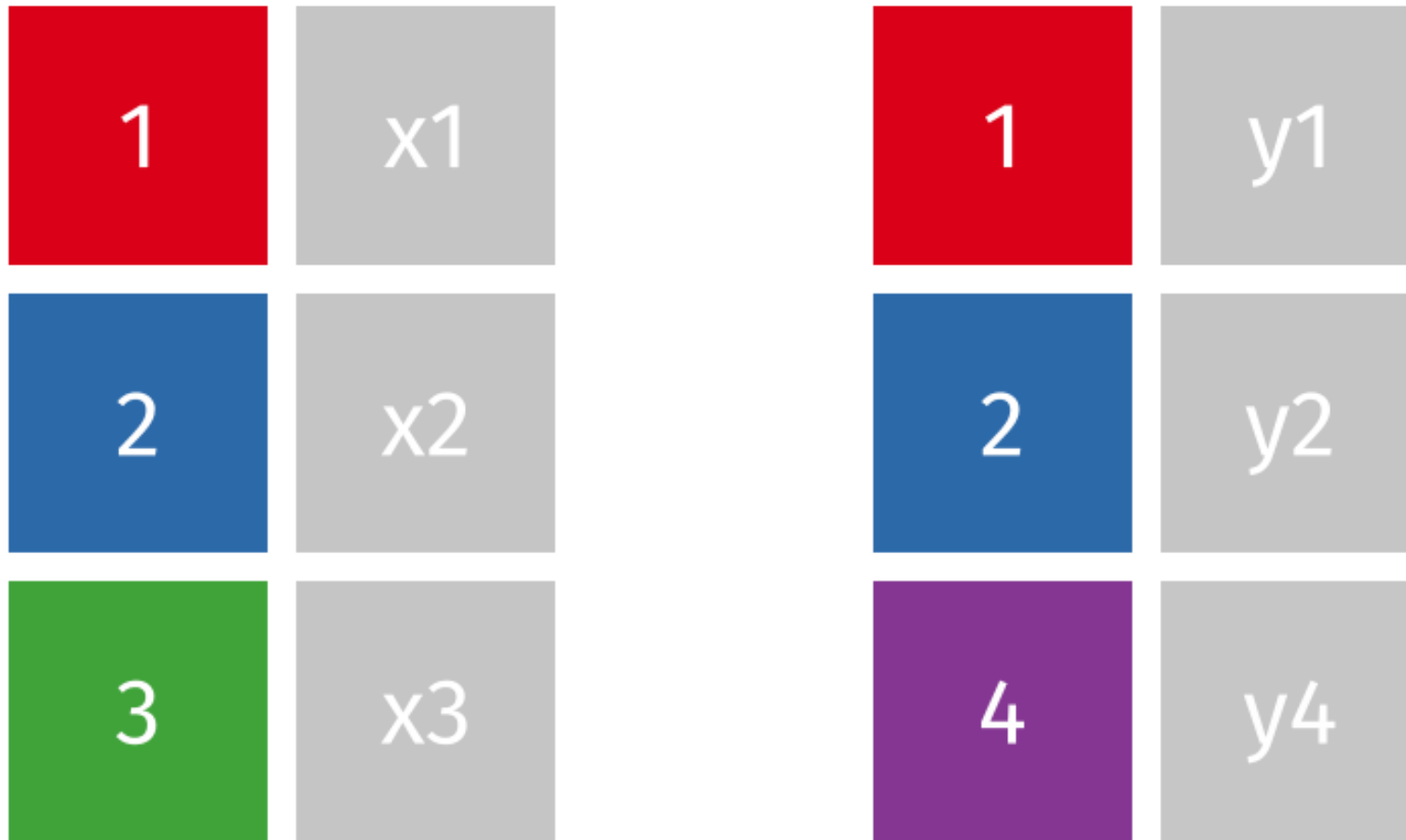


Left Join

- Useful for adding other attributes not in your spatial data
- Returns all of the records in **x** attributed with **y**
- Pay attention to the number of rows!

Left Join

`left_join(x, y)`



Left Join

```
1 head(coffee_data)
```

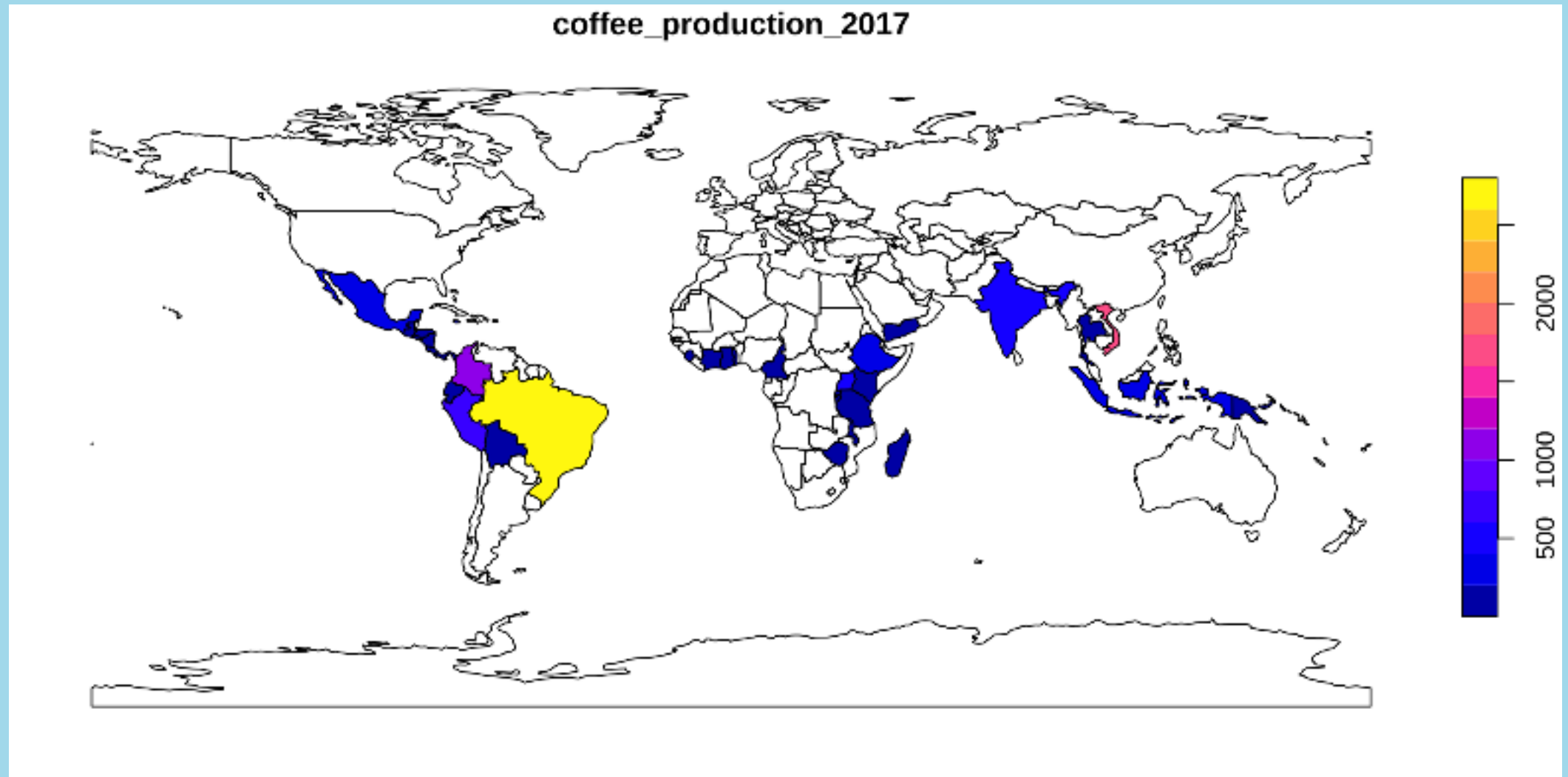
```
# A tibble: 6 × 3
```

```
  name_long  
coffee_production_2016  
coffee_production_2017  
  <chr>  
<int>          <int>  
1 Angola  
NA          NA  
2 Bolivia  
3          4  
3 Brazil  
3277        2786  
4 Burundi  
37          38  
5 Cameroon  
~          ~
```

```
1 world_coffee = left_join(world, coffee_data)  
2 nrow(world_coffee)
```

```
[1] 177
```

Left Join



Inner Join

- Useful for subsetting to “complete” records
- Returns all of the records in **x** with matching **y**
- Pay attention to the number of rows!

Inner Join

`inner_join(x, y)`

| | | | |
|---|----|---|----|
| 1 | x1 | 1 | y1 |
| 2 | x2 | 2 | y2 |
| 3 | x3 | 4 | y4 |

Inner Join

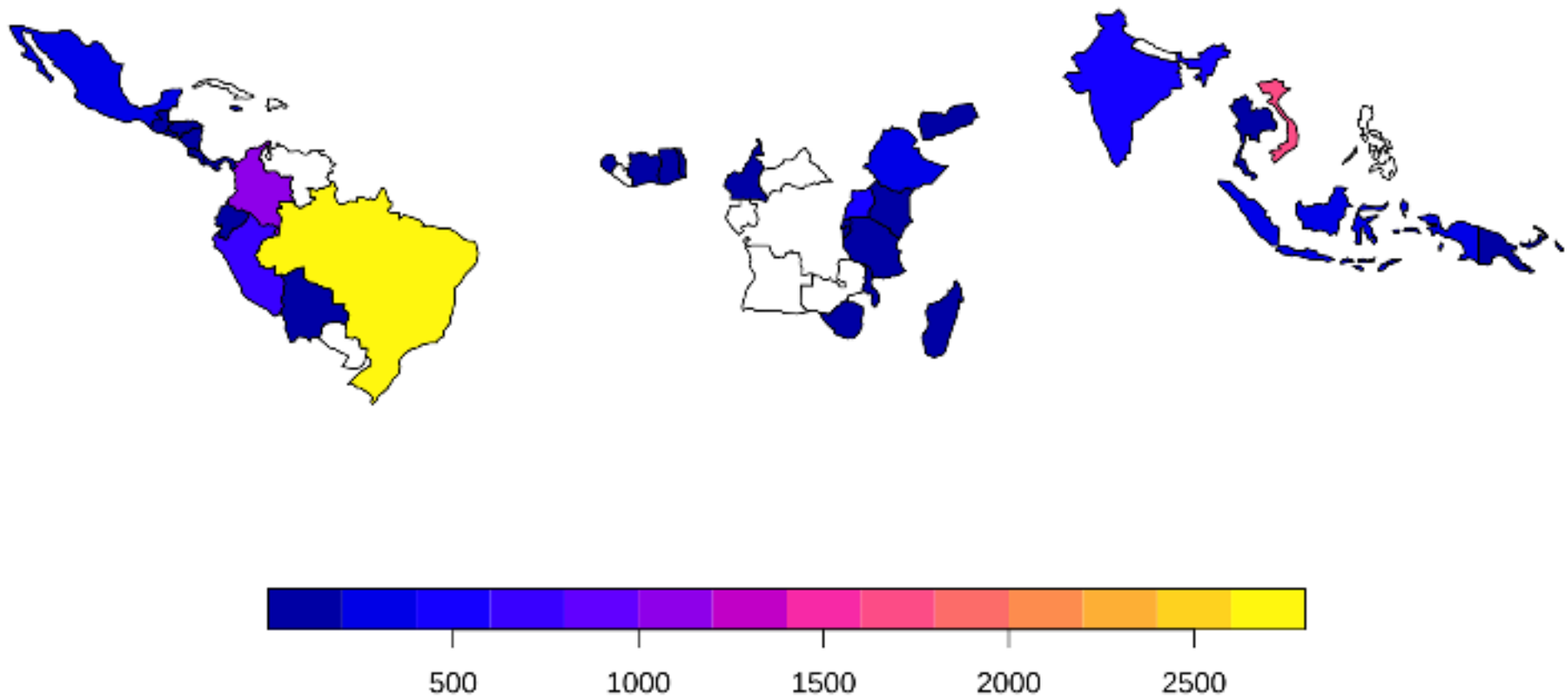
```
1 world_coffee_inner = inner_join(wc  
2 nrow(world_coffee_inner)
```

```
[1] 45
```

```
1 setdiff(coffee_data$name_long, wor  
[1] "Congo, Dem. Rep. of" "Others"
```

Inner Join

coffee_production_2017



Other Joins

- `right_`, `outer_`, and `anti_`
- Spatial Joins (next week)

