Lecture 03: Spatial Data

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

Bruno Conte

28/Feb/2023

Spatial data in economics: schedule

1. ~~Introduction to (spatial) data and programming in R~~ [14.Feb.2023]

2. ~~Spatial data basics: vector data + assignment~~ [21.Feb.2023]

3. Basic operations with vector data ~~+ assignment~~ [28.Feb.2023]

    - Attribute operations with vector data (slicing, filtering, aggregating)
    - Spatial operations of vetor data (e.g. intersections, touching, etc.)
    - Spatial merging/joining (based on overlaps and/or distances)

4. Geometry operations and miscelanea + follow-up + assignment [07.Mar.2023]

5. Raster data and operations + assignment [14.Mar.2023]

6. Take-home exam [12.Apr.2023]

Main references for this class

1. Lovelace, R., Nowosad, J. and Muenchow, J., 2019. **Geocomputation with R.** Chapman and Hall/CRC.

    - Chapter 3.2 (attribute data operations)
    - Chapter 4 (spatial data operations)

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.".

Vector data operations: attribute and spatial

- Data operations: manipulation of vector data (in terms of geometry and attribute structure). Basic operations are:
  - **Selecting:** restricting the fields (i.e. variables) of a `sf`
  - **Slicing:** restricting the features (i.e. observations) of a `sf`
  - **Filtering:** restricting based on data attributes
  - **Joining/merging:** linking attributes (i.e. data) between different `sf` (or data sets)
  - **Aggregating:** processing attributes (i.e. data) within a `sf` based on some fields
- Operations can be either attribute- or spatial-based

Attribute data operations: selecting (choose fields)

```
world
```

```
## Simple feature collection with 177 features and 10 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 × 11
##    iso_a2 name_long       continent region_un subregion type  area_km2
##  * <chr>  <chr>           <chr>     <chr>     <chr>     <chr>    <dbl>
##  1 FJ     Fiji            Oceania   Oceania   Melanesia Sove…   1.93e4
##  2 TZ     Tanzania        Africa    Africa    Eastern … Sove…   9.33e5
##  3 EH     Western Sahara  Africa    Africa    Northern… Inde…   9.63e4
##  4 CA     Canada          North Am… Americas  Northern… Sove…   1.00e7
##  5 US     United States   North Am… Americas  Northern… Coun…   9.51e6
##  6 KZ     Kazakhstan      Asia      Asia      Central … Sove…   2.73e6
##  7 UZ     Uzbekistan      Asia      Asia      Central … Sove…   4.61e5
##  8 PG     Papua New Guin… Oceania   Oceania   Melanesia Sove…   4.65e5
##  9 ID     Indonesia       Asia      Asia      South-Ea… Sove…   1.82e6
```

Attribute data operations: selecting (choose fields)

```
world %>% select(name_long, continent)
```

```
## Simple feature collection with 177 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 177 × 3
##    name_long        continent                                      geom
##    <chr>            <chr>                                <MULTIPOLYGON [°]>
##  1 Fiji             Oceania       (((-180 -16.55522, -179.9174 -16.50…
##  2 Tanzania         Africa        (((33.90371 -0.95, 31.86617 -1.0273…
##  3 Western Sahara   Africa        (((-8.66559 27.65643, -8.817828 27.…
##  4 Canada           North America (((-132.71 54.04001, -133.18 54.169…
##  5 United States    North America (((-171.7317 63.78252, -171.7911 63…
##  6 Kazakhstan       Asia          (((87.35997 49.21498, 86.82936 49.8…
##  7 Uzbekistan       Asia          (((55.96819 41.30864, 57.09639 41.3…
##  8 Papua New Guinea Oceania       (((141.0002 -2.600151, 141.0171 -5.…
##  9 Indonesia        Asia          (((104.37 -1.084843, 104.0108 -1.05…
```

Attribute data operations: slicing (choose observations)

```
world %>% select(name_long, continent) %>% slice(1:2)
```

```
## Simple feature collection with 2 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -18.28799 xmax: 180 ymax: -0.95
## Geodetic CRS:  WGS 84
## # A tibble: 2 × 3
##   name_long continent                                             geom
##   <chr>     <chr>                                       <MULTIPOLYGON [°]>
## 1 Fiji      Oceania   (((-180 -16.55522, -179.9174 -16.50178, -179.79…
## 2 Tanzania  Africa    (((33.90371 -0.95, 31.86617 -1.02736, 30.76986 …
```

Attribute data operations: filtering (based on data)

```
world %>% select(name_long, continent) %>% filter(continent=='South America')
```

```
## Simple feature collection with 13 features and 2 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -81.41094 ymin: -55.61183 xmax: -34.72999 ymax: 12.4373
## Geodetic CRS:  WGS 84
## # A tibble: 13 × 3
##    name_long        continent                                        geom
##  * <chr>            <chr>                                  <MULTIPOLYGON [°]>
##  1 Argentina        South America (((-68.63401 -52.63637, -68.63335 -…
##  2 Chile            South America (((-69.59042 -17.58001, -69.85844 -…
##  3 Falkland Islands South America (((-61.2 -51.85, -60.7 -52.3, -59.8…
##  4 Uruguay          South America (((-57.62513 -30.21629, -57.87494 -…
##  5 Brazil           South America (((-53.37366 -33.76838, -52.7121 -3…
##  6 Bolivia          South America (((-69.52968 -10.95173, -68.66508 -…
##  7 Peru             South America (((-69.89364 -4.298187, -70.39404 -…
##  8 Colombia         South America (((-66.87633 1.253361, -67.18129 2.…
##  9 Venezuela        South America (((-60.73357 5.200277, -61.4103 5.9…
```

Attribute data operations: joining (merging data)

```
world %>% select(name_long, continent) %>% filter(continent=='South America') %>%
  left_join(coffee_data) # data of coffee production by country (name_long)
```

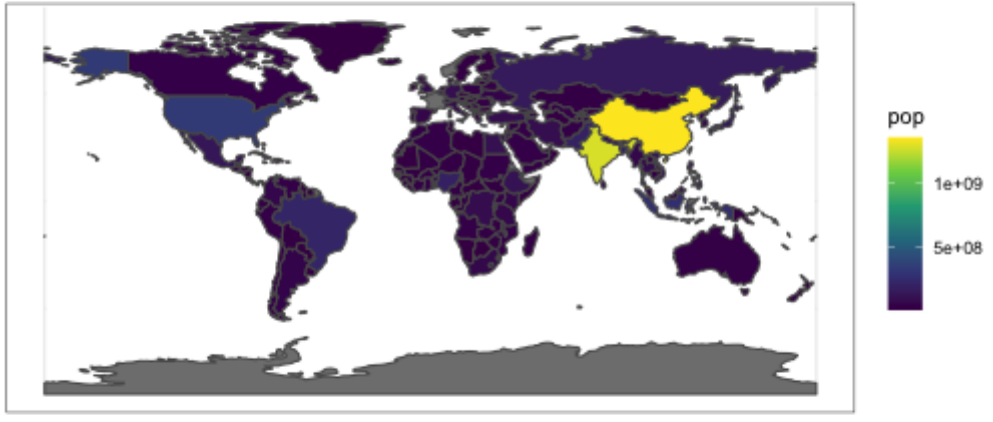
```
## Simple feature collection with 13 features and 4 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -81.41094 ymin: -55.61183 xmax: -34.72999 ymax: 12.4373
## Geodetic CRS:  WGS 84
## # A tibble: 13 × 5
##    name_long      continent                       geom coffee_producti…
##    <chr>          <chr>                <MULTIPOLYGON [°]>          <int>
##  1 Argentina      South Am… (((-68.63401 -52.63637, …                NA
##  2 Chile          South Am… (((-69.59042 -17.58001, …                NA
##  3 Falkland Isla… South Am… (((-61.2 -51.85, -60.7 -…                NA
##  4 Uruguay        South Am… (((-57.62513 -30.21629, …                NA
##  5 Brazil         South Am… (((-53.37366 -33.76838, …              3277
##  6 Bolivia        South Am… (((-69.52968 -10.95173, …                 3
##  7 Peru           South Am… (((-69.89364 -4.298187, …               585
##  8 Colombia       South Am… (((-66.87633 1.253361, -…              1330
```

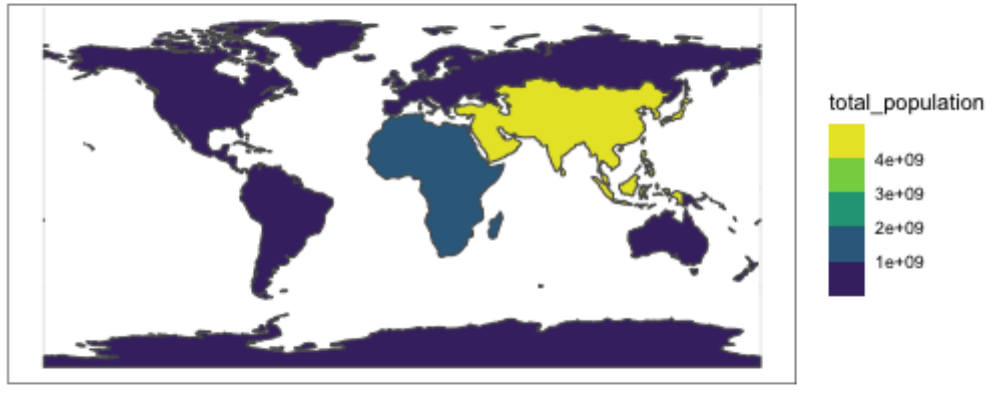
Attribute data operations: aggregating (based on attributes)

```
world %>% select(name_long, continent, pop) %>%
  group_by(continent) %>%
  summarise(total_population = sum(pop, na.rm = T))
```

```
## Simple feature collection with 8 features and 2 fields
## Geometry type: GEOMETRY
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.64513
## Geodetic CRS:  WGS 84
## # A tibble: 8 × 3
##   continent                     total_population                                  geom
##   <chr>                                    <dbl>                      <GEOMETRY [°]>
## 1 Africa                                1154946633 MULTIPOLYGON (((43.1453 11…
## 2 Antarctica                                  0 MULTIPOLYGON (((-180 -89.9…
## 3 Asia                                  4311408059 MULTIPOLYGON (((104.37 -1.…
## 4 Europe                                 669036256 MULTIPOLYGON (((-180 64.97…
## 5 North America                          565028684 MULTIPOLYGON (((-132.71 54…
## 6 Oceania                                 37757833 MULTIPOLYGON (((-180 -16.5…
## 7 Seven seas (open ocean)                     0 POLYGON ((68.935 -48.625, …
```

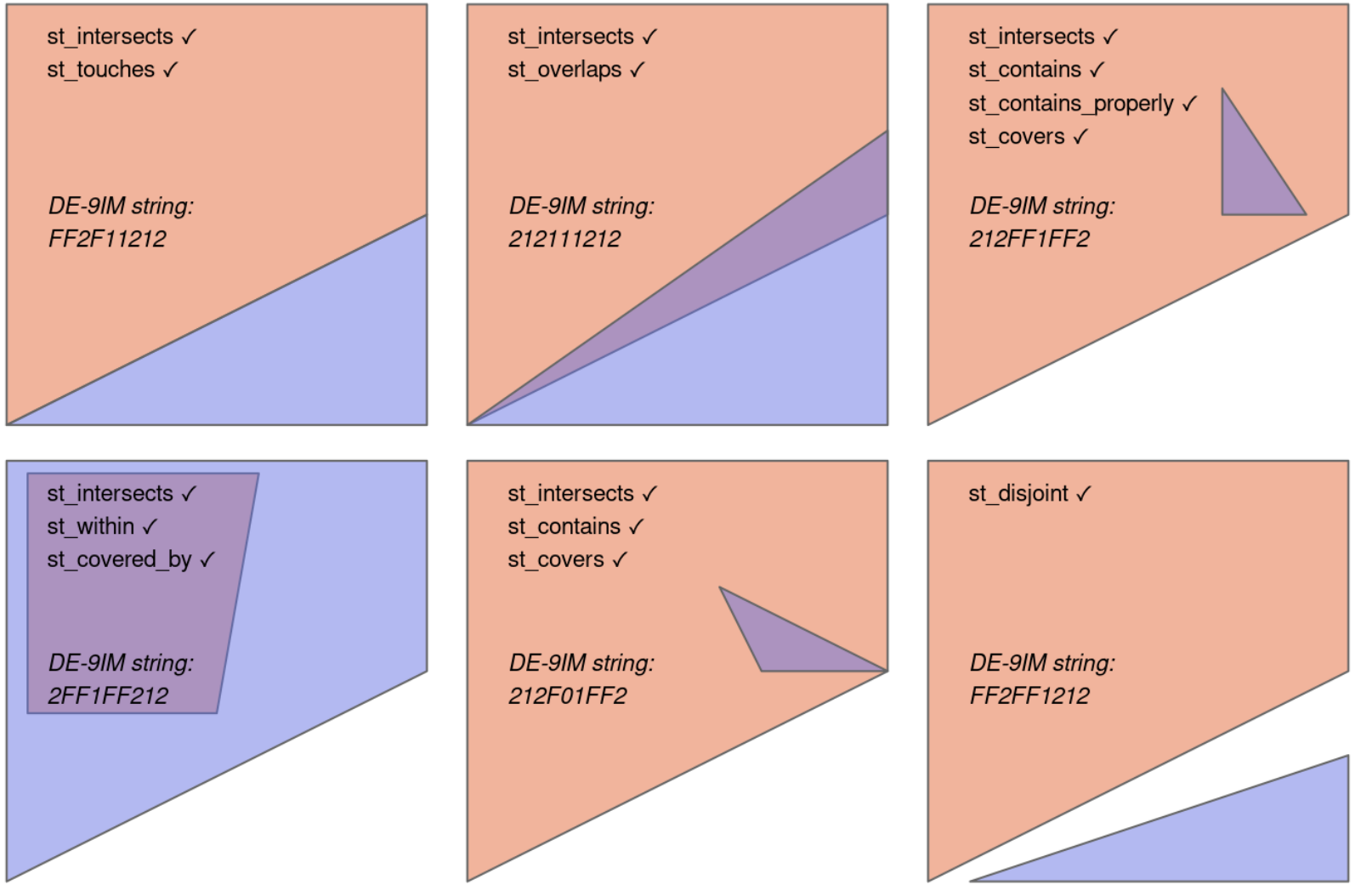
Attribute data operations: aggregating (based on attributes)





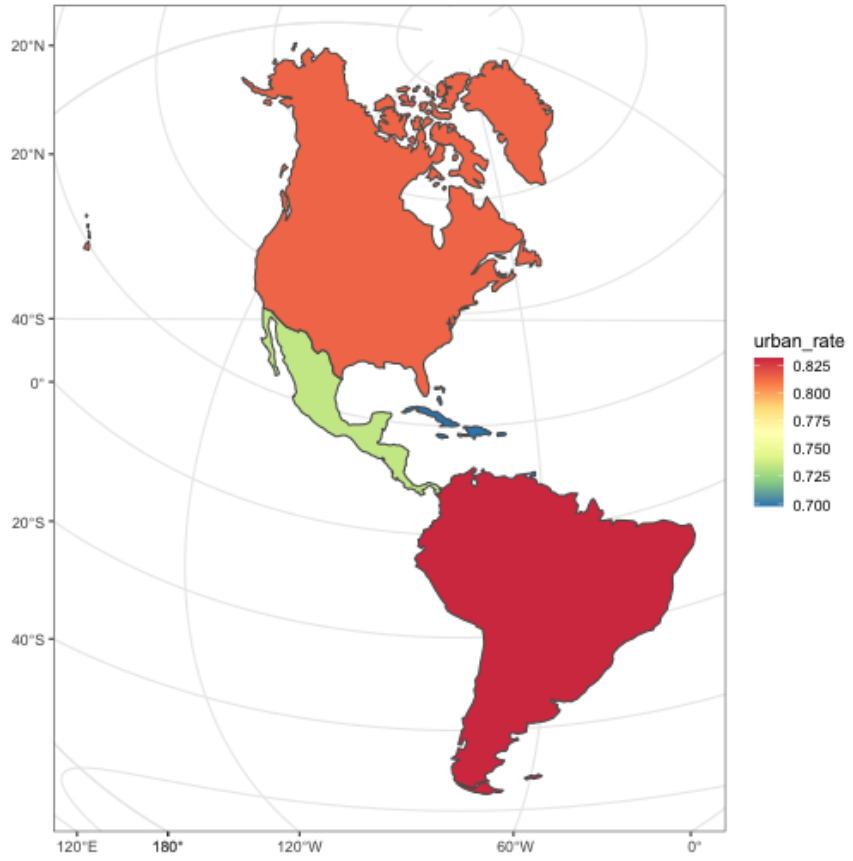
Spatial data operations

- Same intution, but now spatial aspects determine the operations
  - Before: based on the underlying attributes
- Spatial relationship of `sf` objects: determined by different **topological relations**
  - Examples: intersection, containing, touching, etc.
- Intuition (and workflow with data): the same as with attribute data
- **Detailed exposition:** on class material `01_class03.R`
- Next: types of topological relationships


st_intersects ✓
st_touches ✓
DE-9IM string:
FF2F11212
st_intersects ✓
st_overlaps ✓
DE-9IM string:
212111212
st_intersects ✓
st_contains ✓
st_contains_properly ✓
st_covers ✓
DE-9IM string:
212FF1FF2
st_intersects ✓
st_within ✓
st_covered_by ✓
DE-9IM string:
2FF1FF212
st_intersects ✓
st_contains ✓
st_covers ✓
DE-9IM string:
212F01FF2
st_disjoint ✓
DE-9IM string:
FF2FF1212

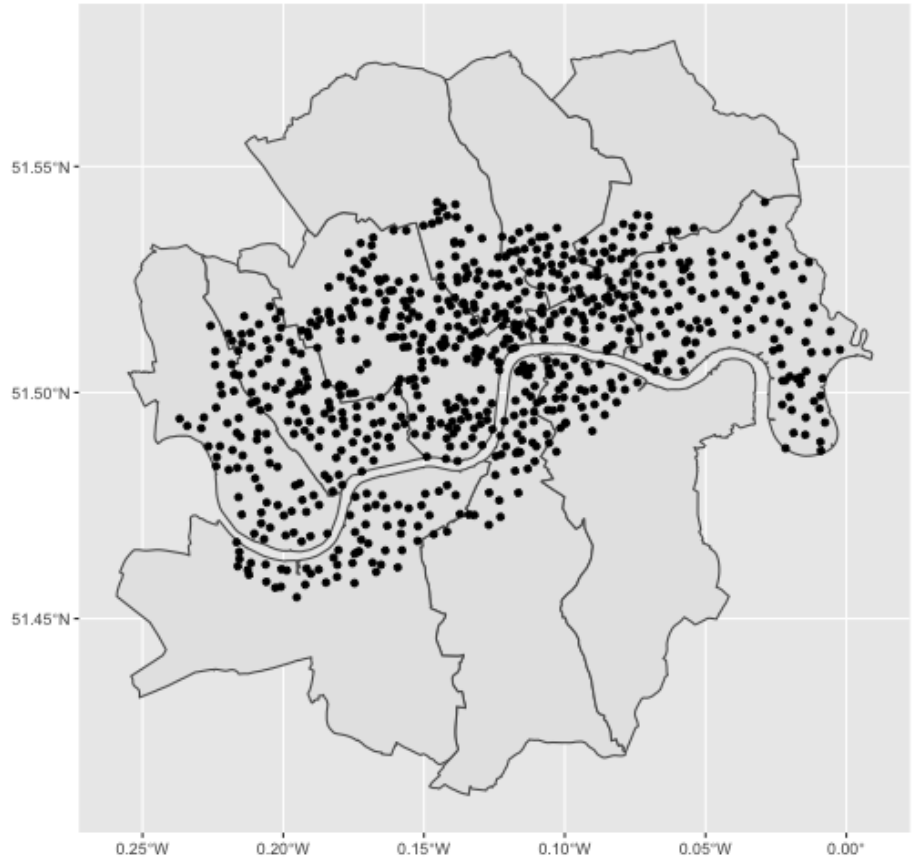
Hands-in: your turn! (1/2)

- Combine `world` (`sf`) and `worldbank_df` (`data.frame`)
- Filter only countries in America
- Calculate urban rate by `subregion`
- urban rate = urban population/total population
- Plot of Americas by subregions' urban rates:



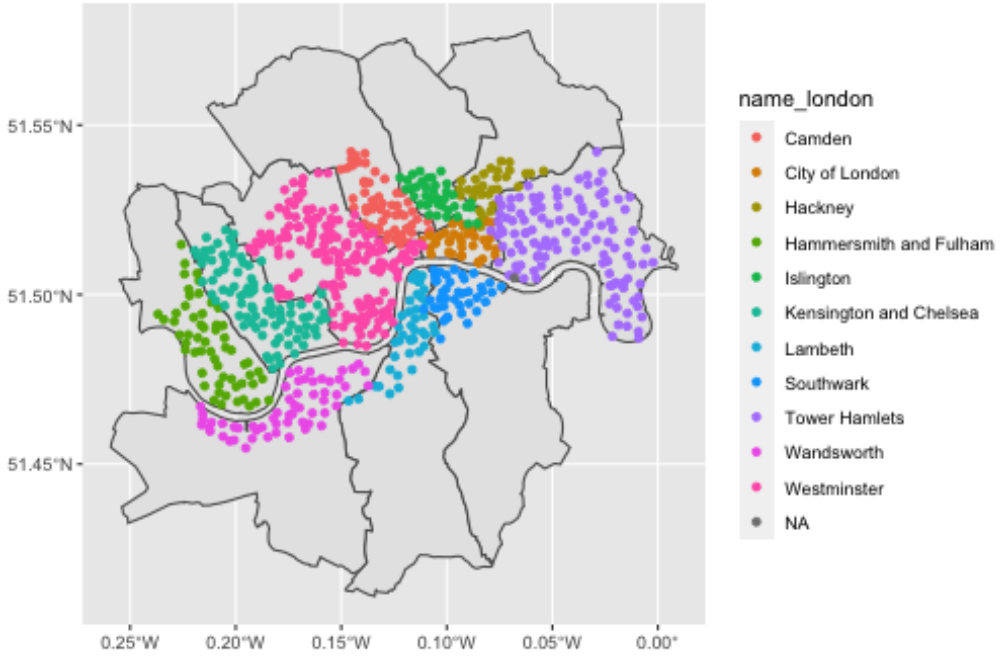
Hands-in: your turn! (2/2)

- Combine `lnd` (Great London) and `cycle_hire` (location of bike stations)
- Filter London regions with bike stations, **plot the two together**
- Join both datasets, plot bike stations by London neighborhood
- Aggregate datasets, plot London neighborhoods by number of bikes



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

- Combine `lnd` (Great London) and `cycle_hire` (location of bike stations)
- Filter London regions with bike stations, plot the two together
- Join both datasets, **plot bike stations by neighborhood**
- Aggregate datasets, plot London neighborhoods by number of bikes



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

- Combine `lnd` (Great London) and `cycle_hire` (location of bike stations)
- Filter London regions with bike stations, plot the two together
- Join both datasets, plot bike stations by London neighborhood
- Aggregate datasets, plot **London neighborhoods by number of bikes**