Lecture 04: Spatial Data

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

Bruno Conte

07/Mar/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]

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

   - Follow-up: double-check course's pace and missing concepts
   - Unary geometry operations: simplifying, centroids, buffers, etc
   - Binary geometry operations: clipping, unions, etc.

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 5 (spatial geometry 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.".

Follow-up: any feedback/issues?

- Course pace, contents, (lack of) complexity?
- Timetable, assignments, examination/grading?
- Course format (exposition + practice)?
- The instructor? ;)
- Anything else (email if not comfortable)?

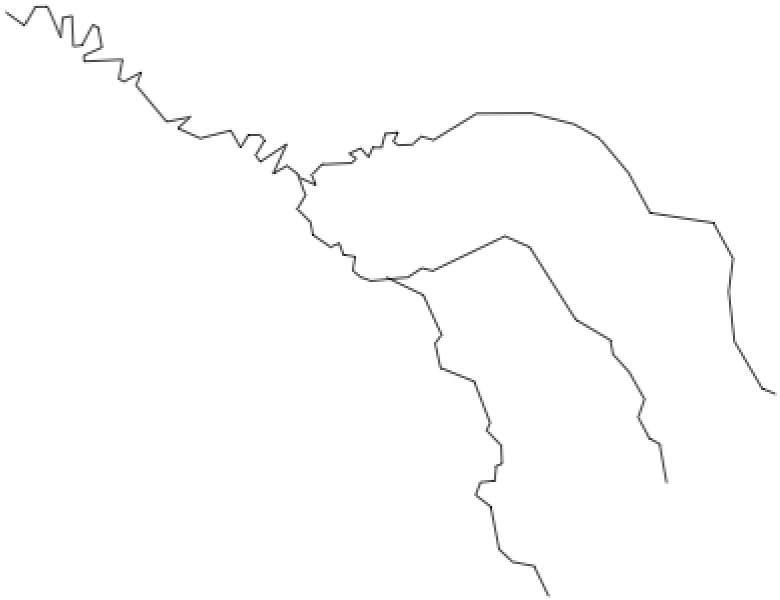
Geometry operations

- Geometry operations: manipulation of vector data that uses/manipulate its geometry. Operations can be both:
  - **Unary:** geometry operations that require (and manipulate) a single feature
    1. Simplification
    2. Centroids
    3. Buffers
    4. Casting
  - **Binary:** operations that interact two features (e.g. distance)
    1. Clipping/subsetting
    2. Distances

Unary geometry operations: simplify

Raw geometry:

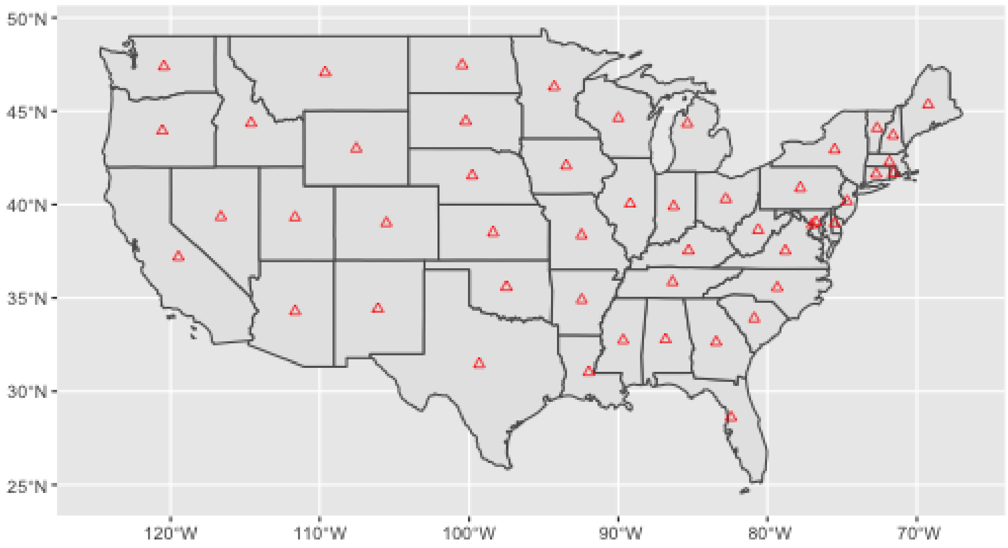
Simplified geometry:



Unary geometry operations: centroids

Assigns the center of mass (**a point**) of a geometry (line or polygon). For disjoint centroids: use `st_point_on_surface()`!

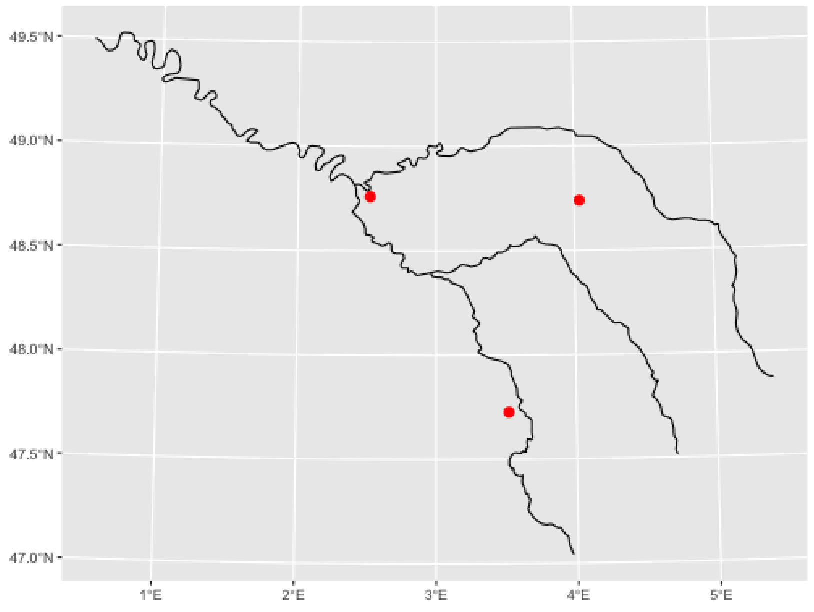
```
sf.us <- us_states
sf.cent <- st_centroid(sf.us)
ggplot() +
  geom_sf(data = sf.us) +
  geom_sf(data = sf.cent, shape = 3)
```



Unary geometry operations: centroids

Assigns the center of mass (**a point**) of a geometry (line or polygon). For disjoint centroids: use `st_point_on_surface()`!

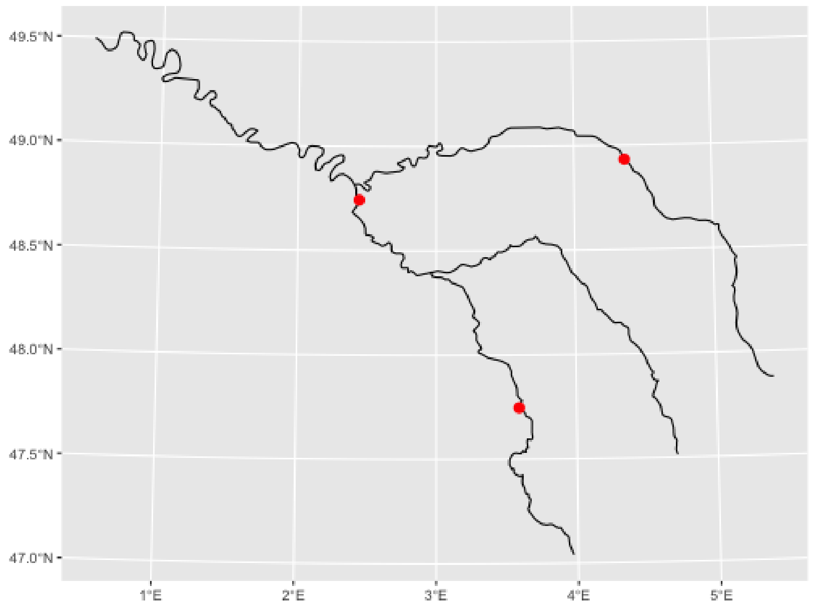
```
sf.river <- seine
sf.cent <- st_centroid(sf.river)
ggplot() +
  geom_sf(data = sf.river) +
  geom_sf(data = sf.cent,
          shape = 16,
          color = 'red'
          )
```



Unary geometry operations: centroids

Assigns the center of mass (**a point**) of a geometry (line or polygon). For disjoint centroids: use `st_point_on_surface()`!

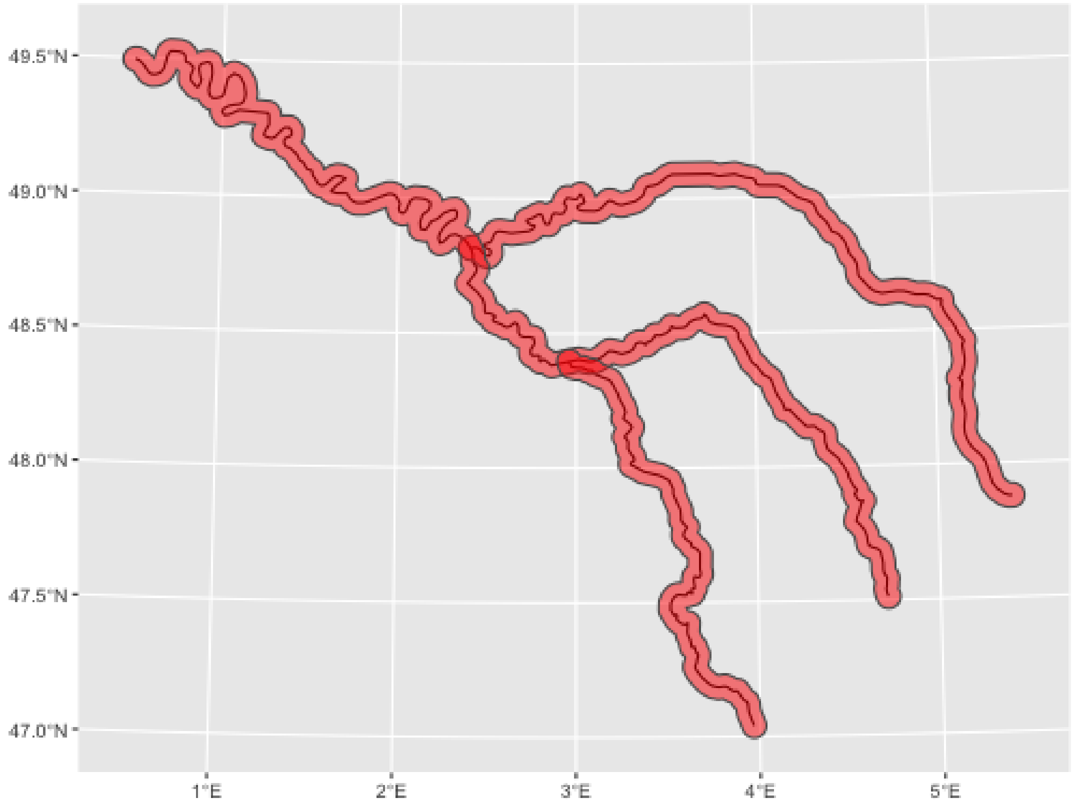
```
sf.river <- seine
sf.cent <- st_point_on_surface(sf.river
ggplot() +
  geom_sf(data = sf.river) +
  geom_sf(data = sf.cent,
          shape = 16,
          color = 'red'
          )
```



Unary geometry operations: buffers

- **Buffer zone:** area around a feature containing locations/space within a certain distance.

```
sf.river <- seine
sf.buff <- st_buffer(sf.river,
                     dist = 5000
                     )
ggplot() +
  geom_sf(data = sf.river) +
  geom_sf(data = sf.buff,
          fill = 'red',
          alpha=.5
          )
```



Unary geometry operations: type transformations (casting)

- **Casting:** transform a feature geometry's type into another based on its vertices

```
sf.river <- seine
sf.river
```

```
## Simple feature collection with 3 features and 1 field
## Geometry type: MULTILINESTRING
## Dimension:     XY
## Bounding box:  xmin: 518344.7 ymin: 6660431 xmax: 879955.3 ymax: 6938864
## Projected CRS: RGF93 / Lambert-93
##    name                       geometry
## 1 Marne MULTILINESTRING ((879955.3 ...
## 2 Seine MULTILINESTRING ((828893.6 ...
## 3 Yonne MULTILINESTRING ((773482.1 ...
```

Unary geometry operations: type transformations (casting)

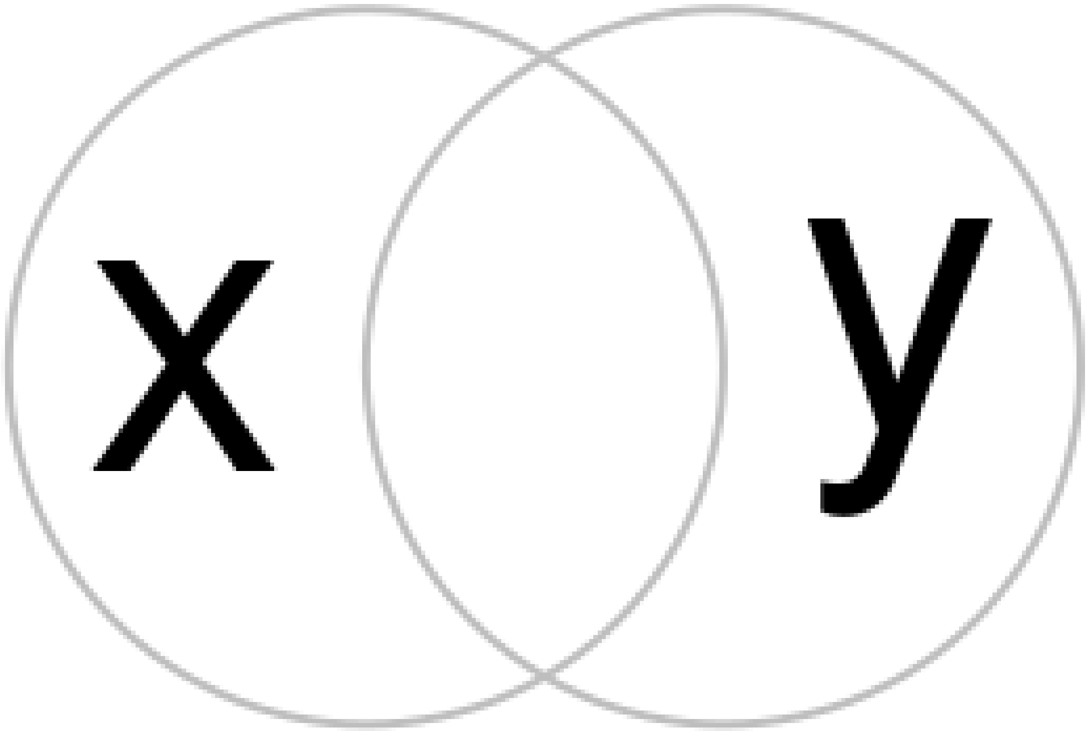
- **Casting:** transform a feature geometry's type into another based on its vertices

```
sf.river <- seine
sf.cast <- st_cast(sf.river,
                   to = 'MULTIPOINT'
                   )
sf.cast
```

```
## Simple feature collection with 3 features and 1 field
## Geometry type: MULTIPOINT
## Dimension:     XY
## Bounding box:  xmin: 518344.7 ymin: 6660431 xmax: 879955.3 ymax: 6938864
## Projected CRS: RGF93 / Lambert-93
##    name                       geometry
## 1 Marne MULTIPOINT ((879955.3 67557...
## 2 Seine MULTIPOINT ((828893.6 67138...
## 3 Yonne MULTIPOINT ((773482.1 66604...
```

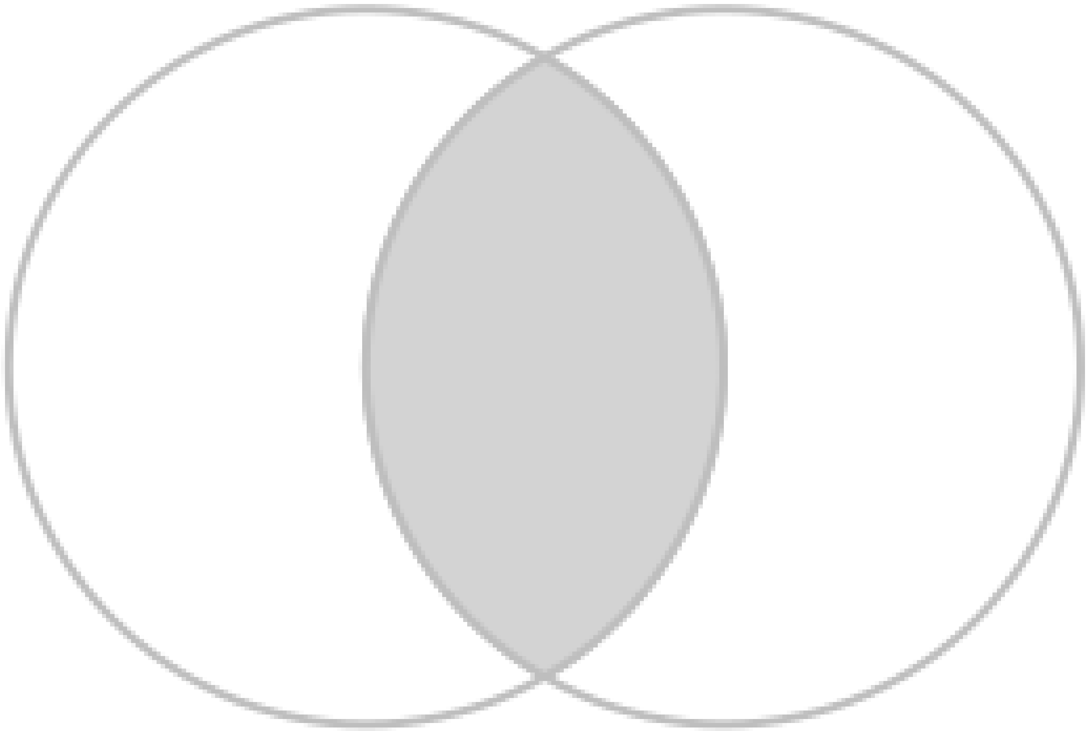
Binary geometry operations: clipping

Clipping: restricting geometry space within the topological relationship between two features.
Example - intersection:

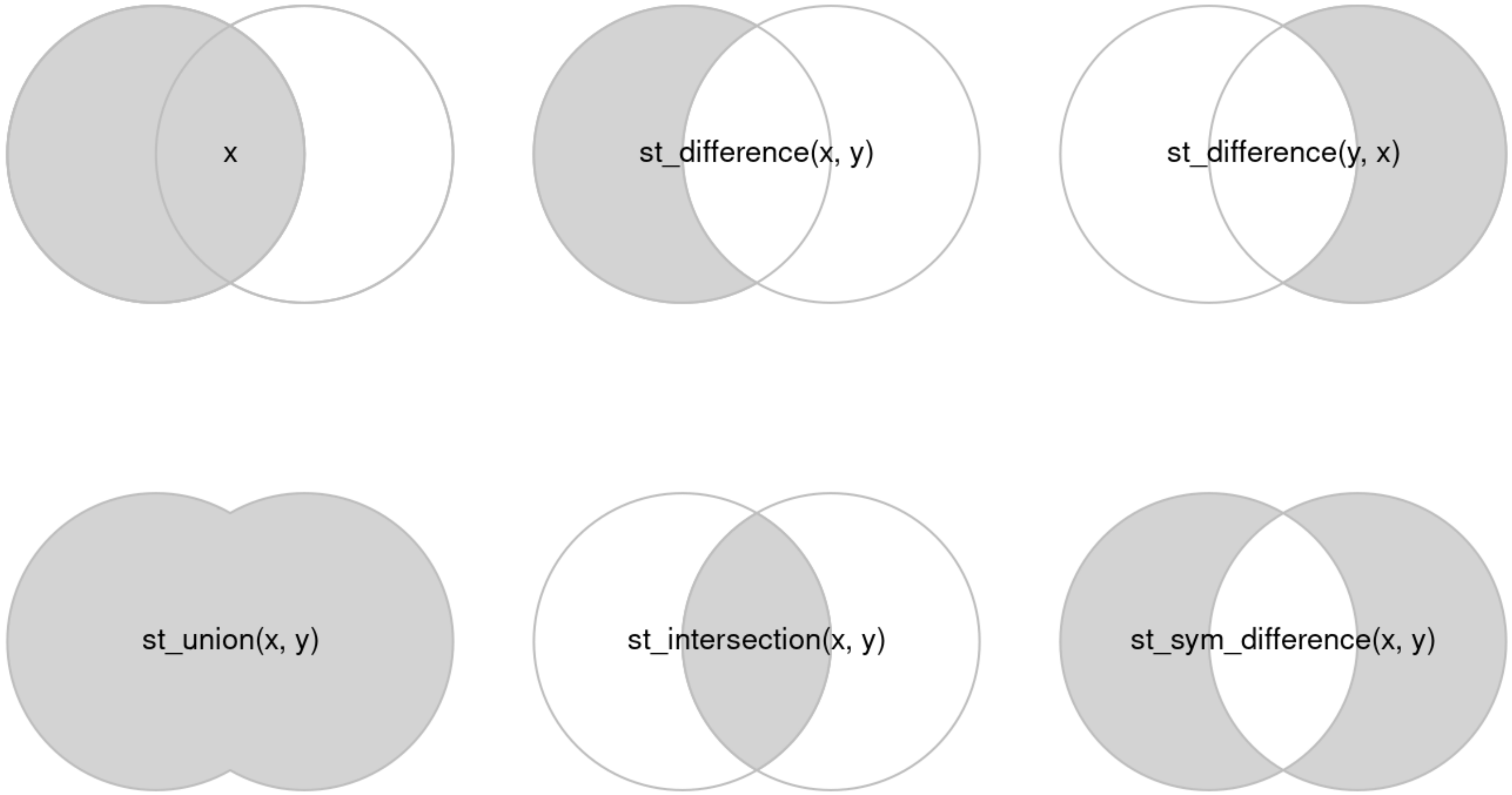


Binary geometry operations: clipping

Clipping: restricting geometry space within the topological relationship between two features.
Example - intersection:



Binary geometry operations: clipping (other relationships)



Binary geometry operations: distances

One of the **most used GIS tools** for economics, with `st_distance()`. Works both with a single or a pair of geometries!

```
library(spData)
library(sf)
sf.centr <- st_centroid(seine)
st_distance(sf.centr)
```

```
## Units: [m]
##           1         2         3
## 1       0.0 111458.4 119082.6
## 2 111458.4       0.0 136560.9
## 3 119082.6 136560.9       0.0
```

Binary geometry operations: distances

One of the **most used GIS tools** for economics, with `st_distance()`. Works both with a single or a pair of geometries!

```
library(spData)
library(sf)
sf.centr <- st_centroid(seine)
st_distance(sf.centr, by_element = T)
```

```
## Units: [m]
## [1] 0 0 0
```

Binary geometry operations: distances

One of the **most used GIS tools** for economics, with `st_distance()`. Works both with a single or a pair of geometries!

```
library(spData)
library(sf)
sf.centr <- st_centroid(seine)
st_distance(sf.centr, seine)
```

```
## Units: [m]
##            [,1]     [,2]      [,3]
## [1,]  32317.409 27135.47 77735.006
## [2,]   3262.824  6126.58 51637.691
## [3,] 129403.368 72121.86  5876.571
```

Your turn: Take-home Assignment

Take-home assignment (1/2)

Combine the world shape `world` with:

- Population (point) data (do not use rasters!)
- Ports, airports, etc.
- All data is available at Natural Earth!

Produce:

- Map of total population by country
- Histogram of country population distribution by continent
- Histogram of (country-level) average distances between locations and ports **or** airports by continent.

Take-home assignment (2/2)

Briefly read

- Porteous, O., 2019. High trade costs and their consequences: An estimated dynamic model of African agricultural storage and trade. *American Economic Journal: Applied Economics*, *11(4)*, pp.327-66.

Then, download the paper's data (here), combine with `world` and transportation (e.g. road, railroad) data to:

1. Generate an `sf` `POINT` feature of market prices across Africa; plot them

2. Calculate minimum distance of each market to the (i) coast, (ii) nearest road, and (iii) nearest airport

3. Produce 3 scatter plots relating average prices with the minimum distances

Note: the paper appendix contains a lot of useful information about the market price data!