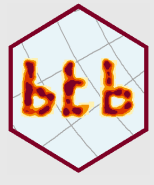
Beyond The Border

Spatial smoothing with R

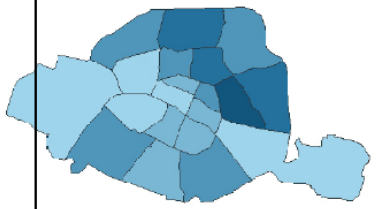


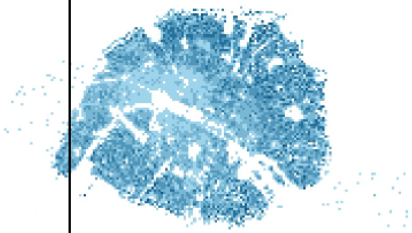
Insee

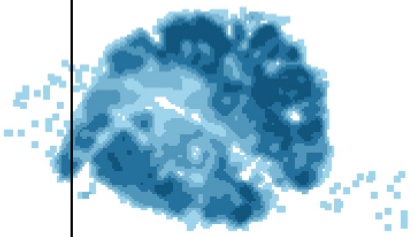
Mesurer pour comprendre

01 Spatial smoothing with R

3 ways of mapping geographical data







1. **The territory**: a non-regular division of space. Several difficulties: mainly the Modifiable areal unit problem (MAUP) effect ;

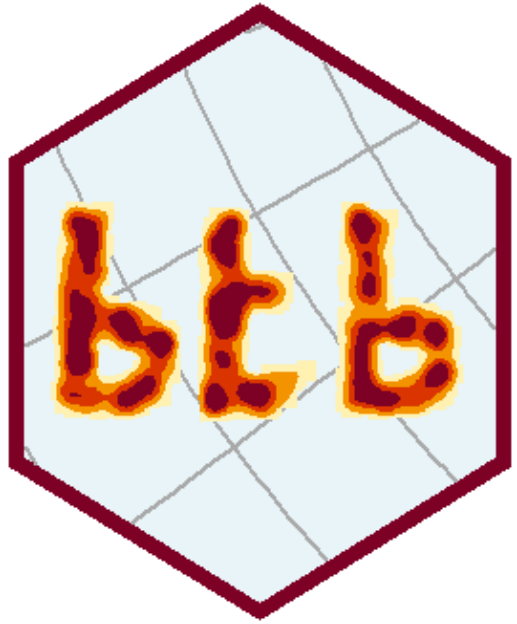
2. **The grid**: a regular division of space in the form of a grid of square cells. By construction, the gridded data can be very erratic ;

3. **Spatial smoothing**: an extension of gridding consisting in describing the population environment within a given radius.

Existing packages

- `KernSmooth`
- `spatstat`...

→ Often, **Fast Fourier Transform**.
Not suitable for **edge effects**



btb

- developed in 2018 (Insee, France)
- deals with edge effects
- allows quantile smoothing (less sensitive to extreme values)
- developed in C++ (`Rcpp`)

Choice of parameters

- The **kernel** describes how the neighborhood is approached ;
- The **bandwidth** quantifies the "size" of this neighborhood (to be chosen according to a bias/variance trade-off) ;
- The **geographical level** from which the smoothed values are estimated ;
- The **treatment of edge effects** makes explicit how geographical boundaries and the limits of observation territory are taken into account in the analysis.

In btb...

- ***quadratic kernel*** *estimation method*
- *a* ***variable bandwidth***
- ***square*** *whose size can be chosen*
- *taken into account.* ***conservative method***

02 Example: the housing prices in Paris in 2021

« Demandes de Valeurs Foncières »,

- Real-estate transactions database (houses and apartments)
- In 2021
- Paris region
- Geolocalized (1 transaction = 1 geographic point)

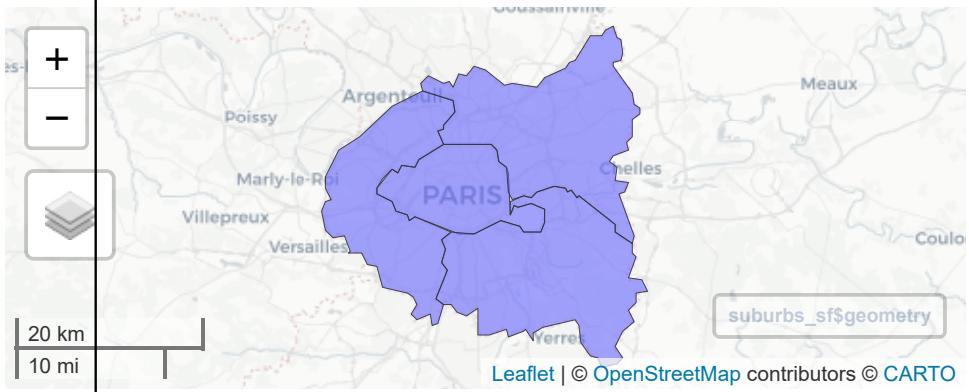
Variables used in the following example :

- `id_mutation` : id for each transaction
- `valeur_fonciere` : price in euros
- `surface_reelle_bati` : surface in square meters
- `x` : longitude (**Lambert 93 projection**)
- `y` : latitude (**Lambert 93 projection**)

Load map layer

```
url_suburbs <-"https://minio.lab.sspcloud.fr/proj
suburbs_sf <- sf::st_read(url_suburbs)
```

```
suburbs_sf <- suburbs_sf %>%
  rename(geometry=geom)
```



Load database

```
url_file <- url("https://minio.lab.sspcloud.fr/pr
dfBase <- readRDS(url_file)
dfBase <- dfBase[,c("id_mutation",
                    "valeur_fonciere",
                    "surface_reelle_bati",
                    "x","y")]
```

id_mutation	valeur_fonciere	surface_
2021-447023	480000	
2021-447024	345000	
2021-447025	384000	
2021-447027	261900	
2021-447029	407200	

1. Transform observations into geometric points

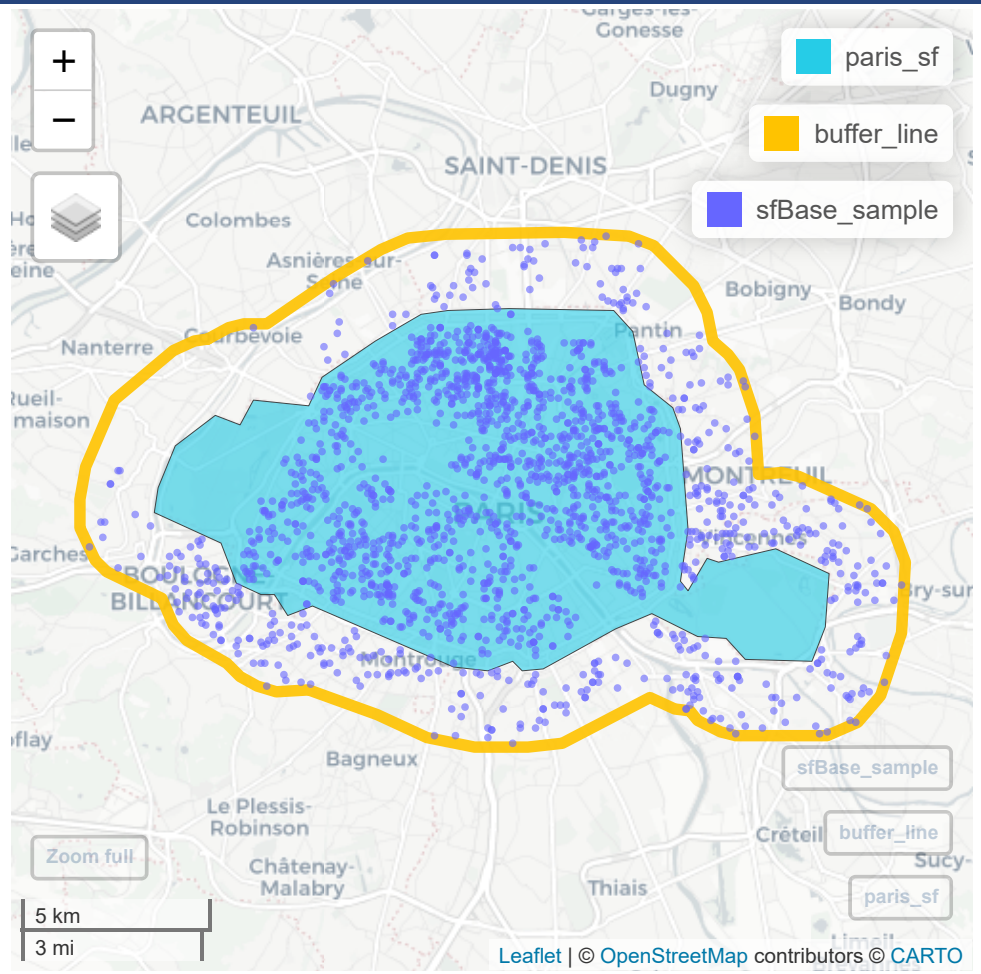
```
sfBase <- sf::st_as_sf(dfBase,
                       coords = c("x", "y"),
                       crs = 2154)
```

2. Buffer zone

```
paris_sf <- suburbs_sf %>% filter(code=="75")
buffer_sf <- st_buffer(paris_sf,dist = 2000)
```

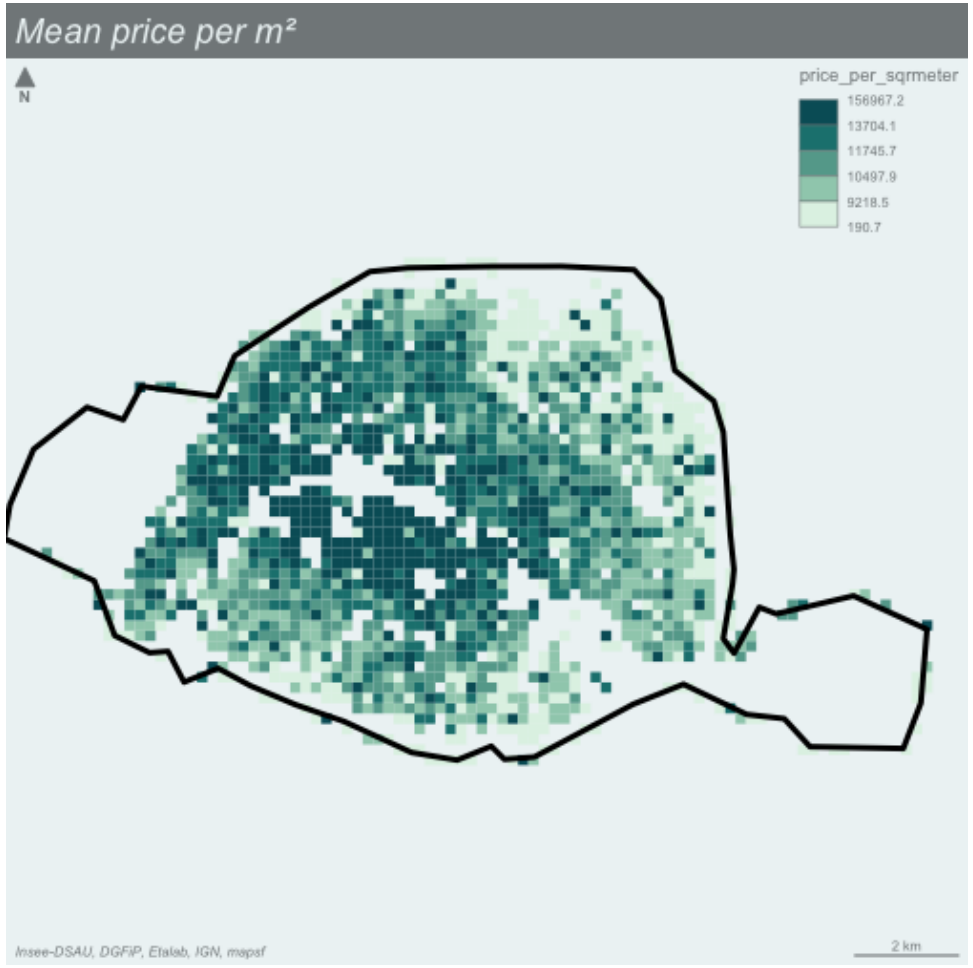
3. Geographical intersection

```
sfBase_buffer <- st_join(sfBase, buffer_sf,
                         left=FALSE)
```



- Better understand your data before smoothing it : **aggregation** inside grid cells !
- It's really easy using `btb` functions such as :
  - `btb_add_centroids`
  - `btb_ptsToGrid`

➡ Let's calculate the mean price per square meters (among 2021 transactions)



Spatial smoothing with `btb::btb_smooth`

- `pts` : table of points with only numeric variables (geometric object or not) ;
- `iCellSize` : cell size in meters (granularity of the underlying grid) ;
- `iBandwidth` : smoothing bandwidth in meters.

```
pts <- sfBase_buffer[,c("valeur_fonciere","surface_reelle_bati")]
smooth_result <- btb::btb_smooth(pts = pts,
                                 iCellSize = 50,
                                 iBandwidth = 800)
```

x	y	valeur_fonciere	surface_reelle_bati	geometry
651075	6855175	1631.677	0.3346249	POLYGON ((651050 6855200, 6...
651125	6855175	1609.976	0.3315480	POLYGON ((651100 6855200, 6...
651175	6855175	1584.218	0.3279684	POLYGON ((651150 6855200, 6...
651225	6855175	1556.653	0.3240574	POLYGON ((651200 6855200, 6...
651275	6855175	1536.788	0.3216315	POLYGON ((651250 6855200, 6...
651325	6855175	1544.332	0.3254294	POLYGON ((651300 6855200, 6...

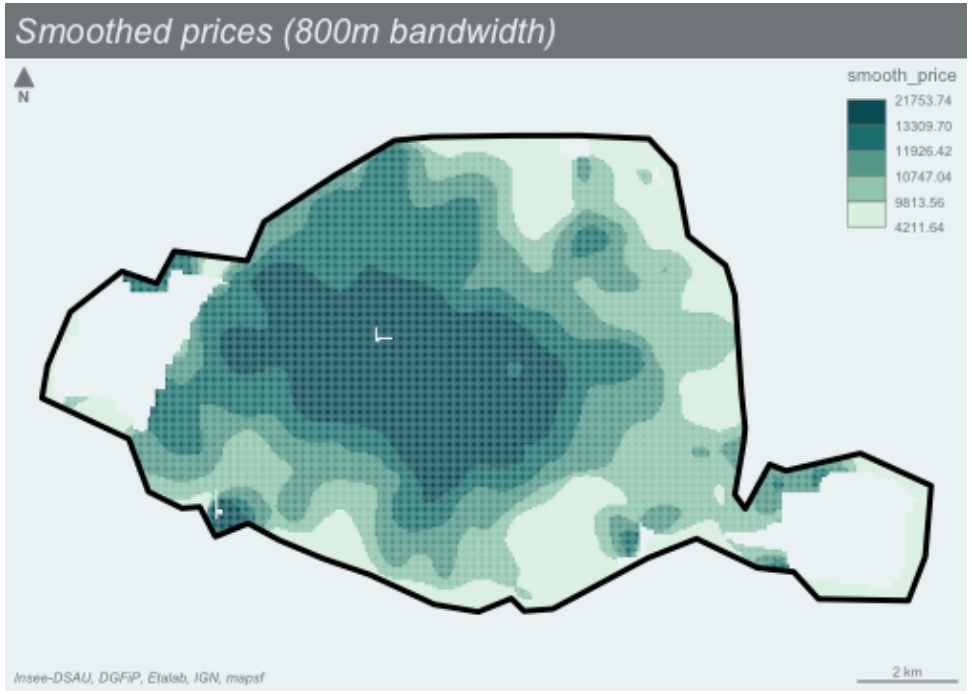
Get the smoothed prices per m²

```
smooth_result <- smooth_result %>%
  mutate(smooth_price=
    valeur_fonciere / surface_reelle_bati)
```

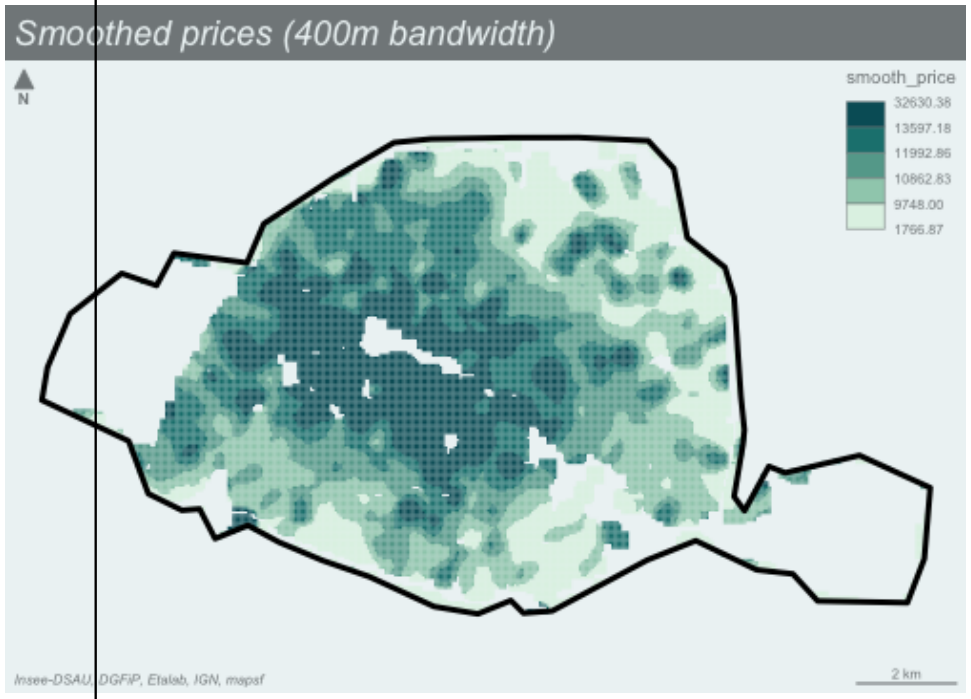
Only keep the cells inside Paris

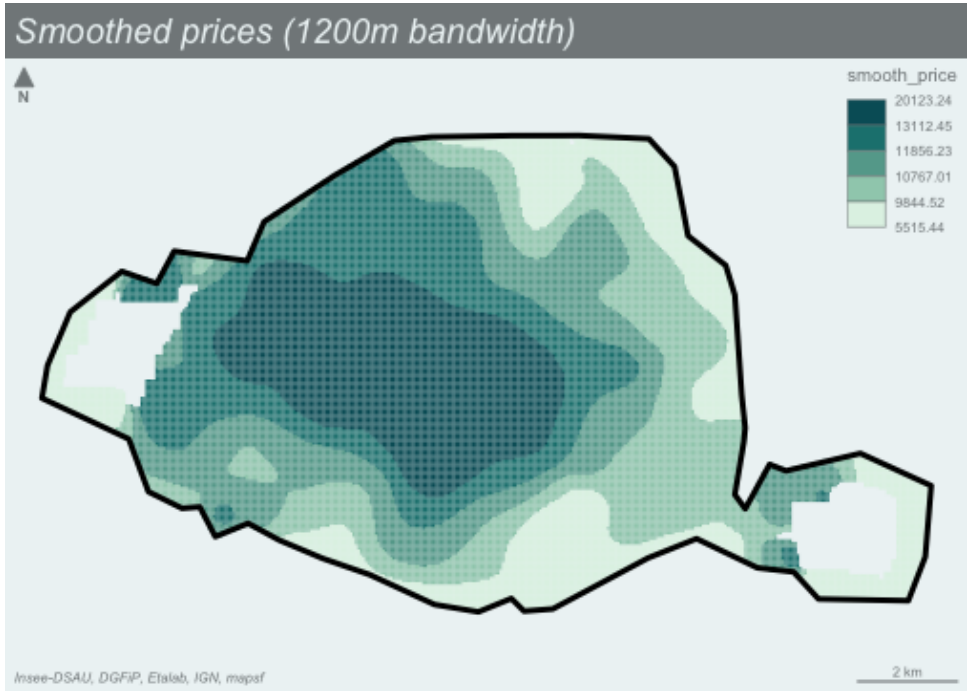
```
smooth_result <- smooth_result %>%
  st_join(paris_sf[,"geometry"],left=F)
```

Map the result



Possible with different bandwidths





btb available on CRAN and Github

```
install.packages("btb")
```

Find all the documentation you need

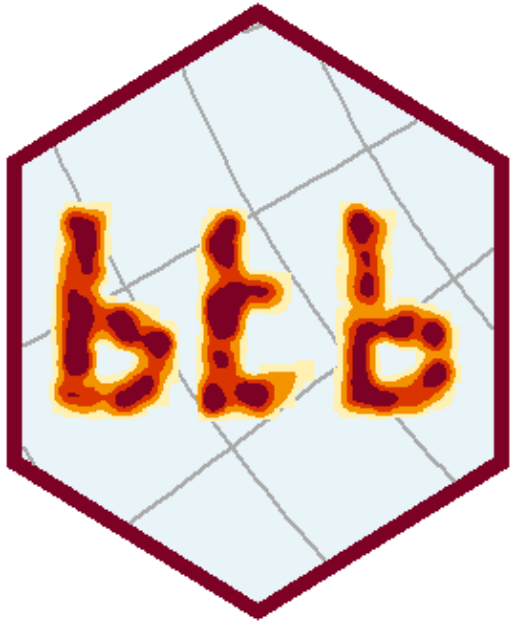
Please visit btb website :

→ https://inseefr.github.io/btb/

And do not hesitate to contact us !

→ analyse-urbaine@insee.fr

Thank you for your attention !



Find us on:

insee.fr

Kim Antunez
Julien Pramil
Insee, France
analyse-urbaine@insee.fr