Data Science for Actuaries (ACT6100)

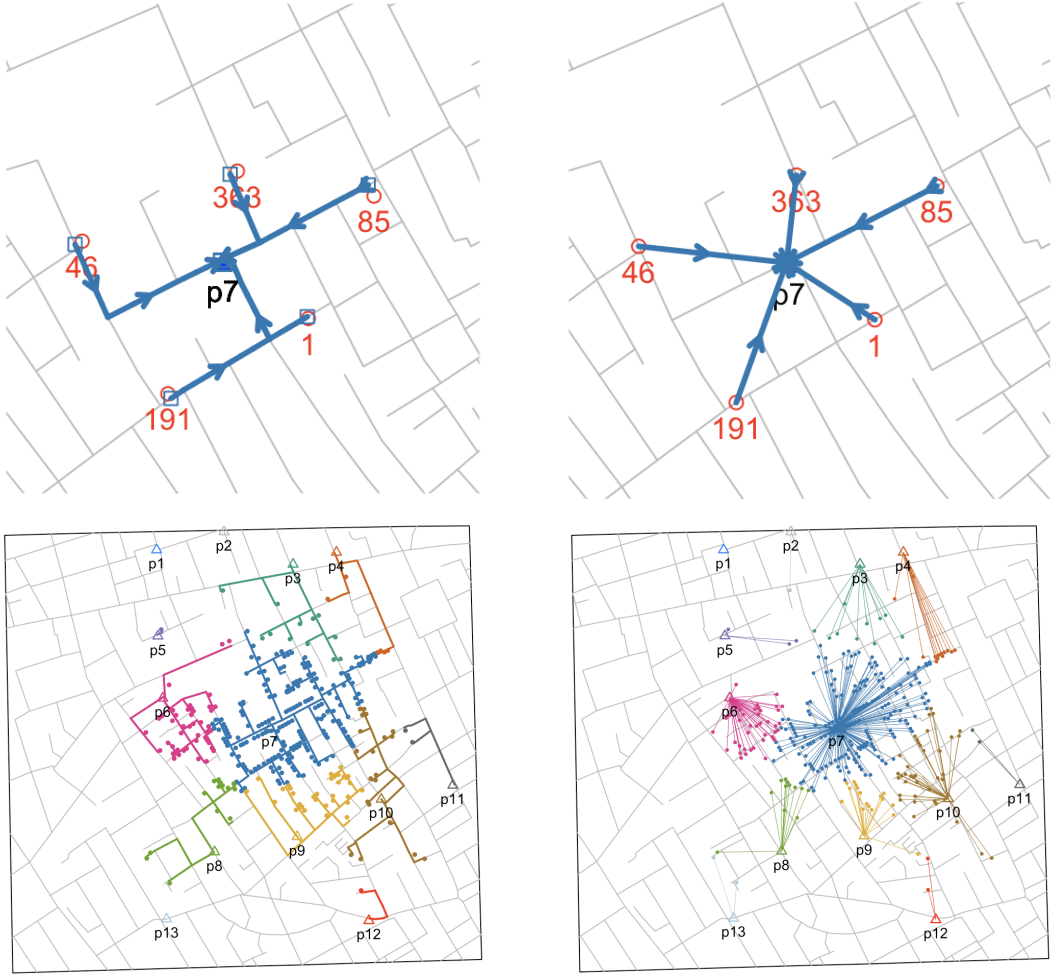
Arthur Charpentier

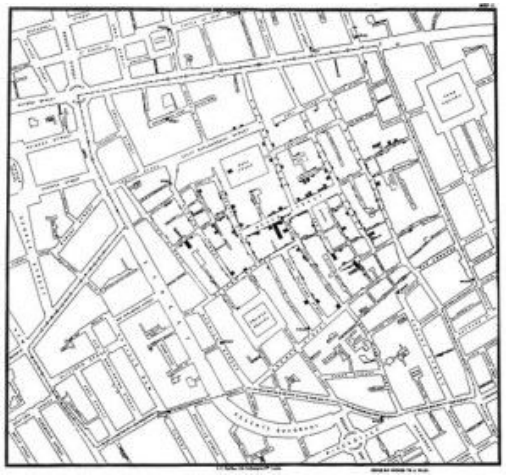
Complément # 2 (Spatial)

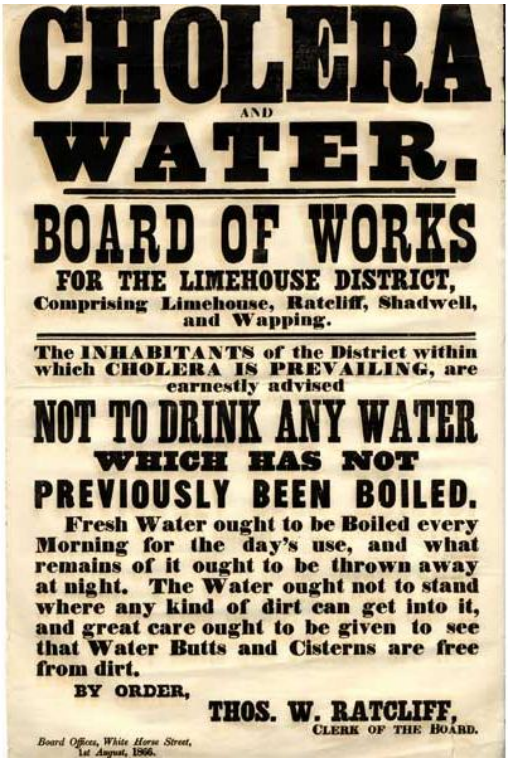
automne 2Q20

Cholera in London (Snow)

The cholera map that changed the world
(inspired by lindbrook's R codes)







CHOLERA
AND
WATER.

BOARD OF WORKS
FOR THE LIMEHOUSE DISTRICT,
Comprising Limehouse, Ratcliff, Shadwell, and Wapping.

The INHABITANTS of the District within which CHOLERA IS PREVAILING, are earnestly advised

NOT TO DRINK ANY WATER
WHICH HAS NOT
PREVIOUSLY BEEN BOILED.

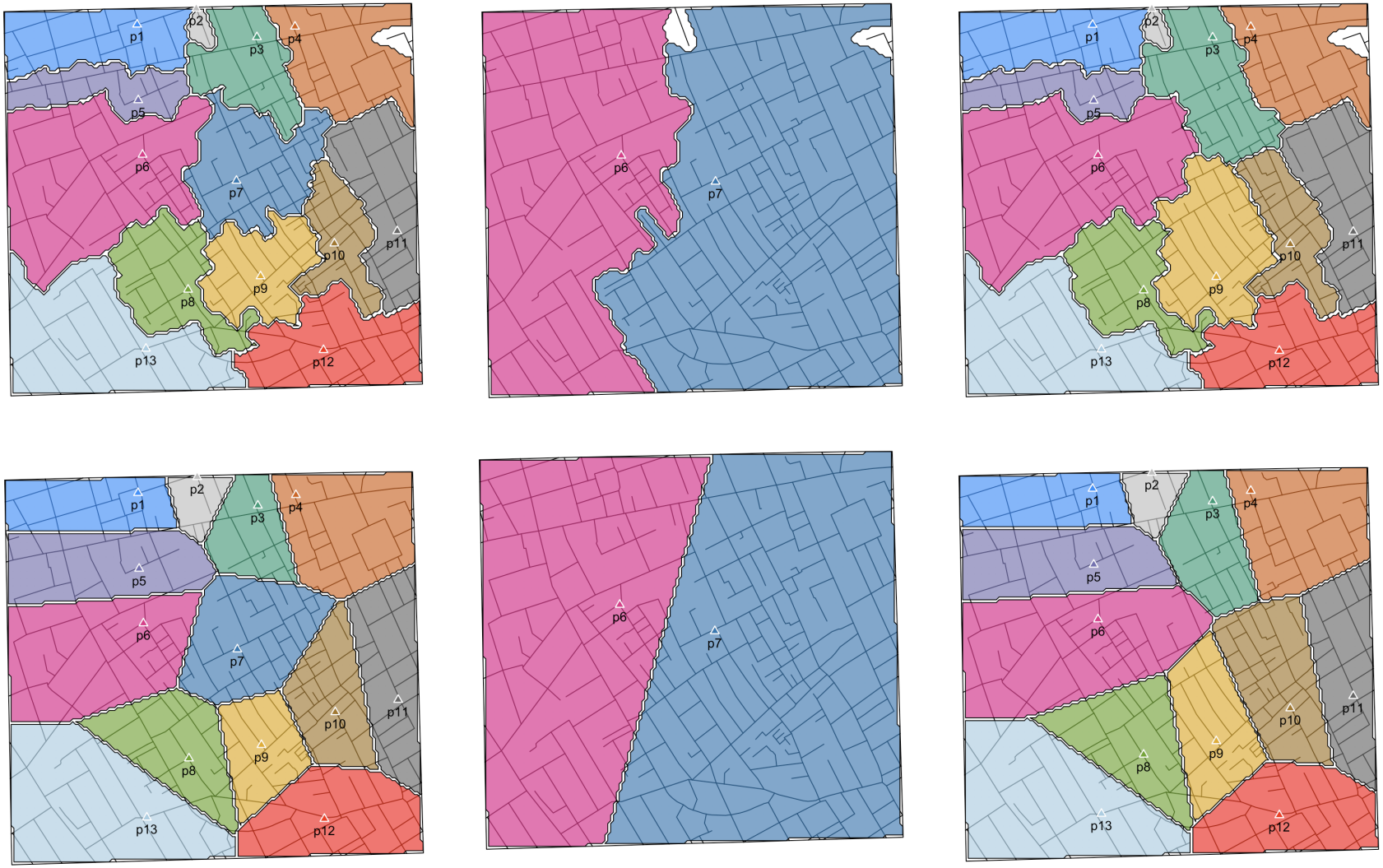
Fresh Water ought to be Boiled every Morning for the day's use, and what remains of it ought to be thrown away at night. The Water ought not to stand where any kind of dirt can get into it, and great care ought to be given to see that Water Butts and Cisterns are free from dirt.

BY ORDER,

THOS. W. RATCLIFF,
CLERK OF THE BOARD.

Board Offices, White Horse Street, 1st August, 1866.

Cholera in London (Snow)



See also Optimal transport on large networks: a practitioner guide

Angles

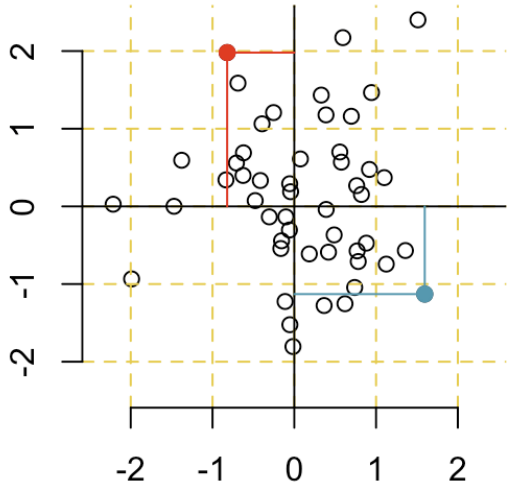
Coordinates on the earth (sphere), or some projection
Classical reprensentation,

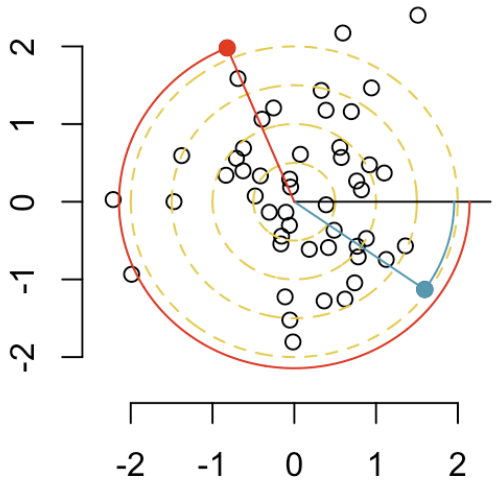
$$\boldsymbol{z} = \begin{pmatrix} x \\ y \end{pmatrix}$$ in the cartesian coordinate system

Alternatively, use polar coordinates

$$\boldsymbol{z} = \begin{pmatrix} r\cos(\theta) \\ r\sin(\theta) \end{pmatrix}$$

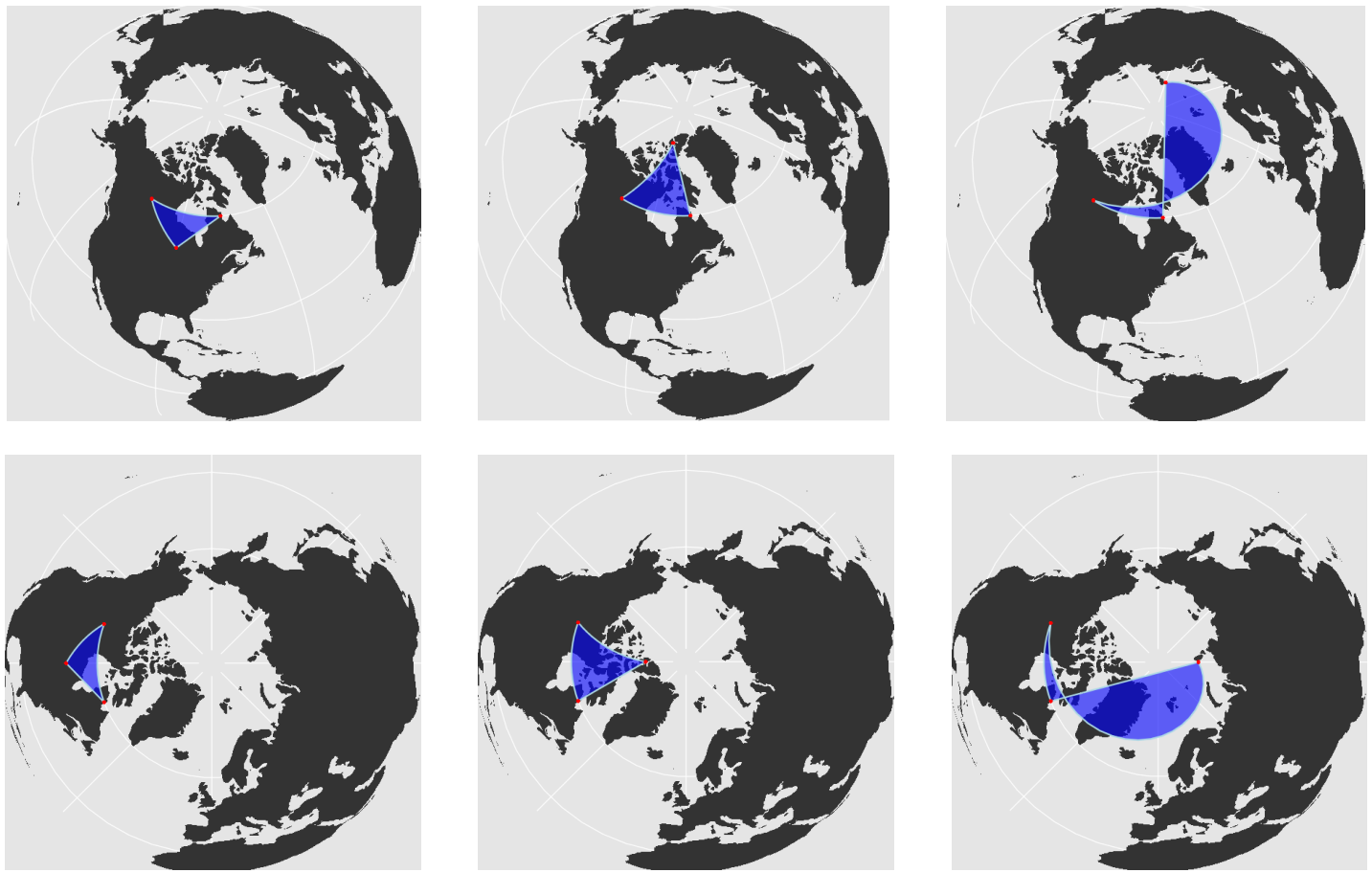
see circular data





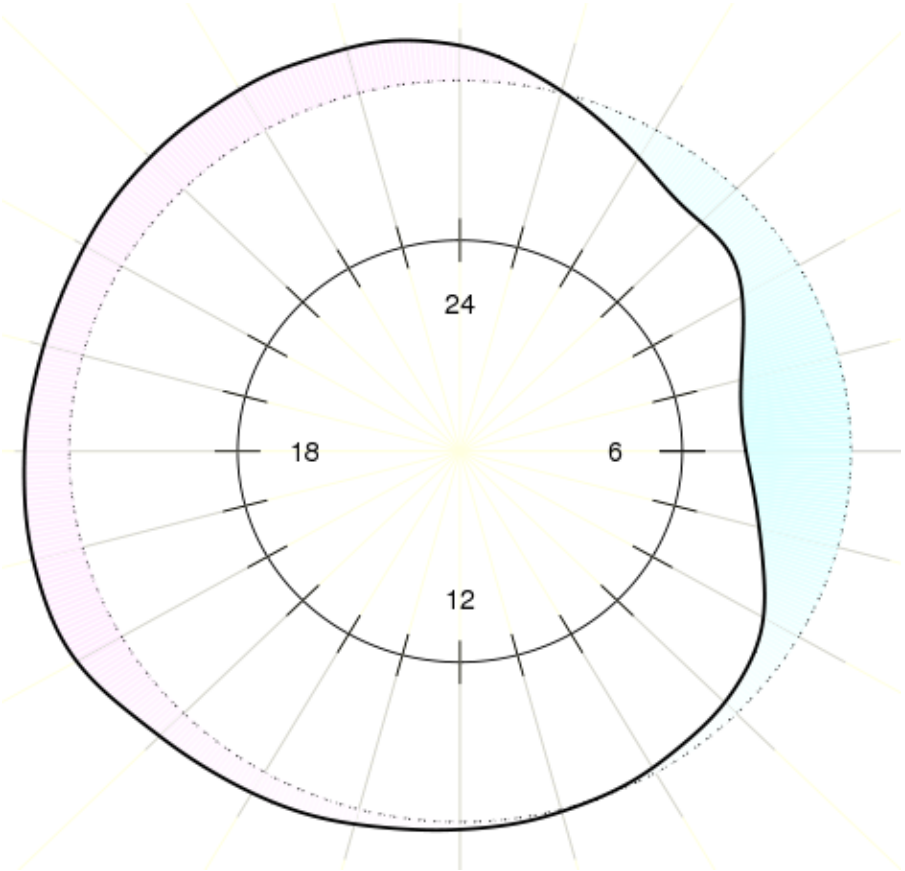
Spatial Data

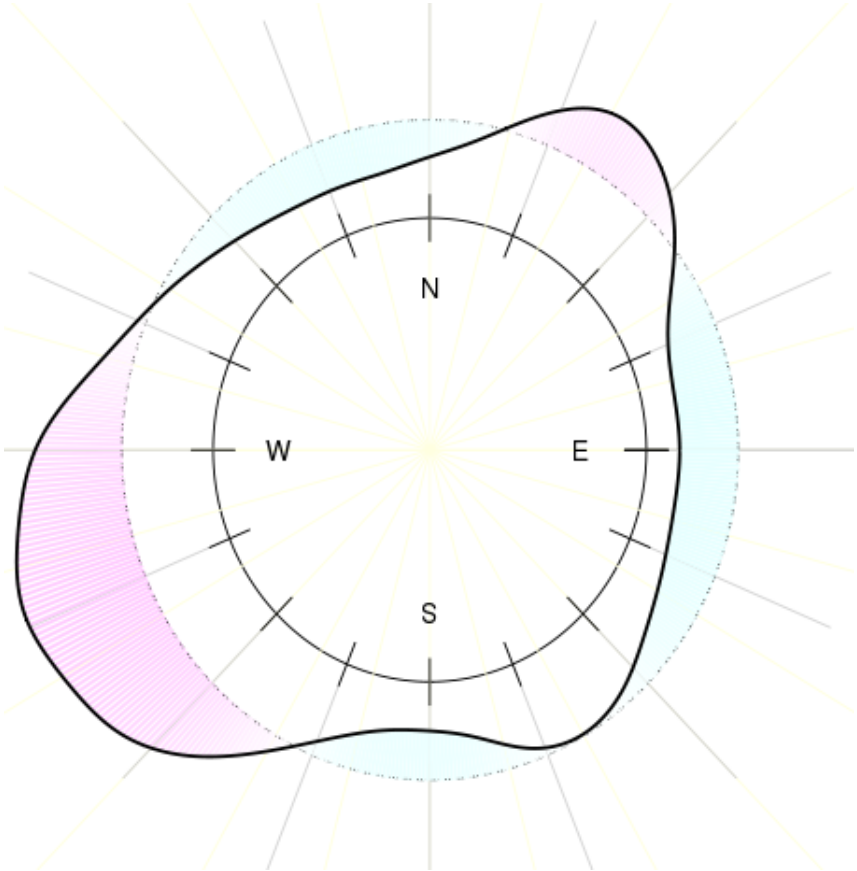
Problem of circular/spherical data expressed in cartesian coordinates (latitude, longitude): singularity of the poles.



Circular Data

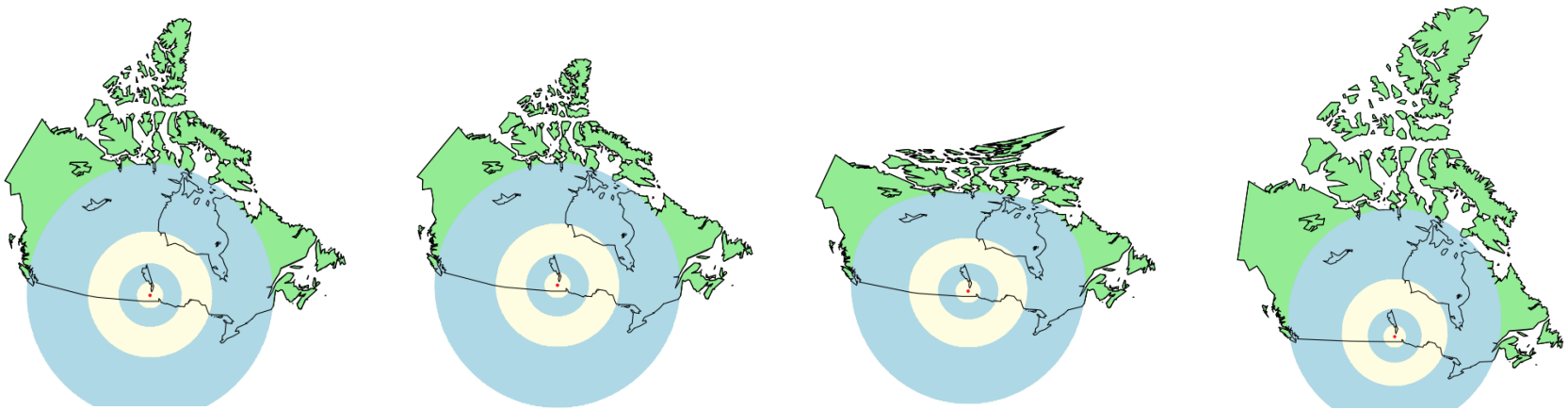
E.g. phone calls at 911, time of the day (left) and wind direction in Montréal (right)





Projections

CRS (Coordinate Representation Systems) can get complicated

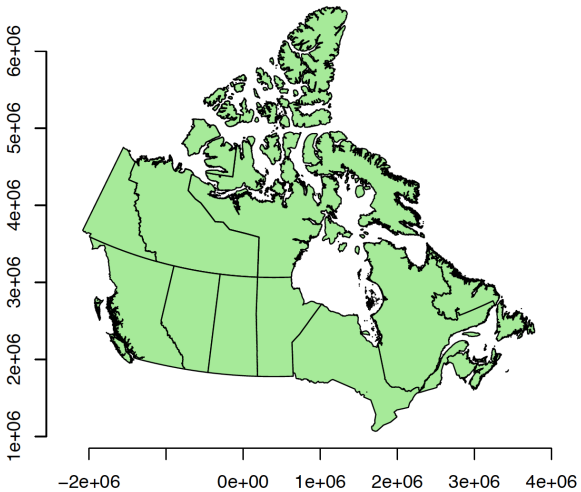


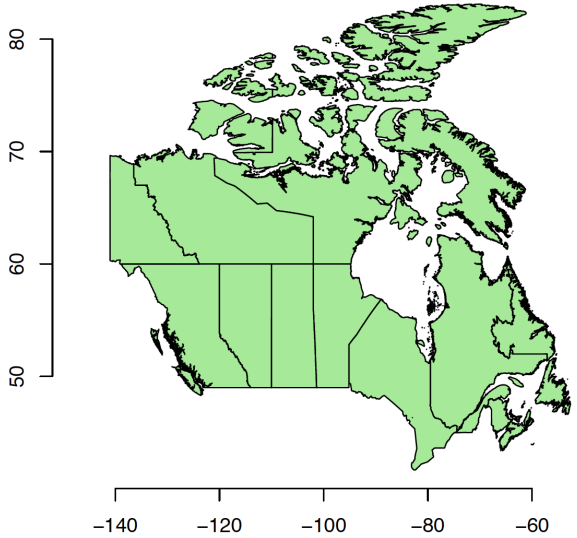
E.g. EPSG:4326 latitude, longitude in WGS-84 coordinate system
EPSG:900913 and EPSG:3857: Google spherical Mercator
ESRI:102718: and NAD 1983 State Plane NY Long Island

Projections

Usually spatial data are in one speccific CRS but we can convert to another one https://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/

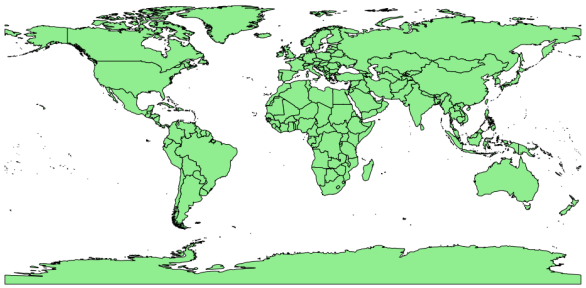
```
library(rgdal)
library(maptools)
Proj = CRS("+proj=longlat +datum=WGS84")
CAN_shp = readShapePoly("CAN_adm1.shp", verbose=TRUE, proj4string=Proj)
plot(CAN_shp)
new_CAN_shp = spTransform(CAN_shp, CRS("+init=epsg:26978"))
plot(new_CAN_shp)
```





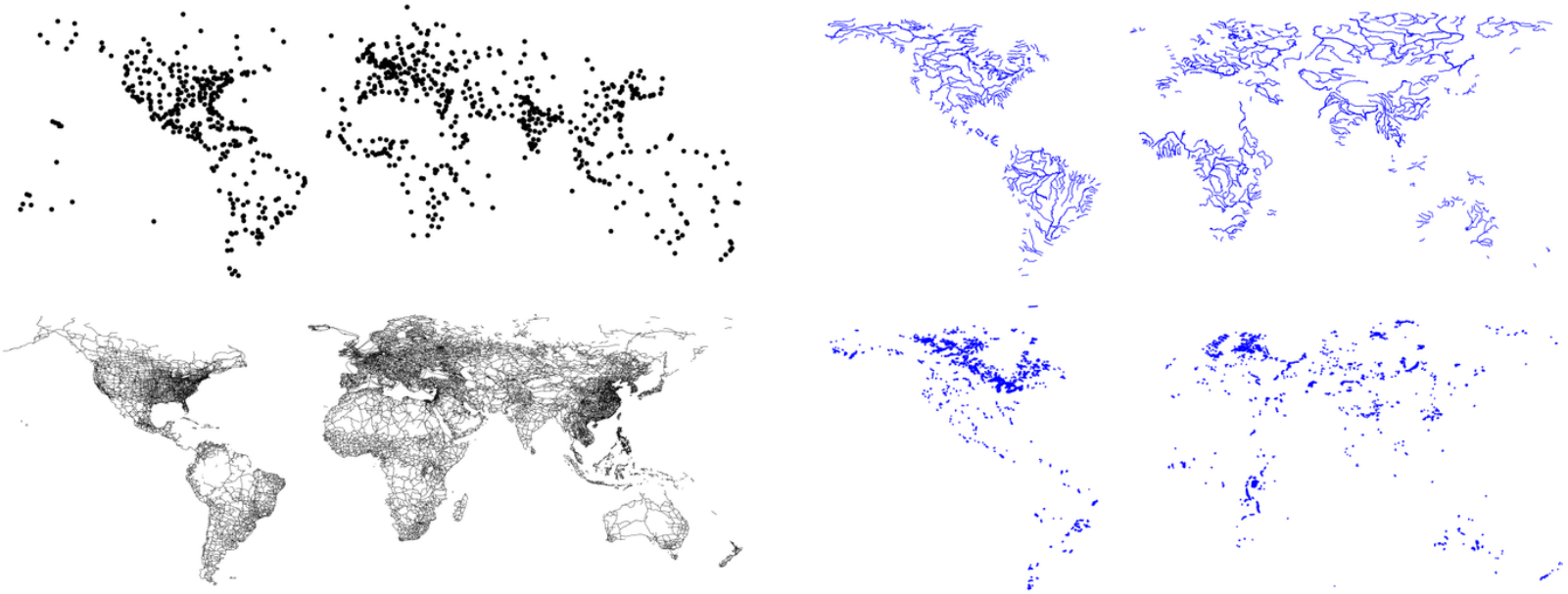
Projections

```
library(sp)
library(rworldmap)
data(countriesLow)
crs.WGS84 = CRS("+proj=longlat +datum=WGS84")
countries.WGS84 = spTransform(countriesLow, crs.WGS84)
plot(countries.WGS84,col="light green")
crs.laea = CRS("+proj=laea +lat_0=52 +lon_0=10 +x_0=4321000 +y_0=3210000 +ellps=GRS80 +units=m +no_defs")
countries.laea = spTransform(countriesLow, crs.laea)
plot(countries.laea,col="light green")
```



Maps: Dots, Lines and Polygons

See minimalist maps (with R codes)
with dots (cities), lines (rivers, roads) and polygons (lakes)



Polygons & Regions

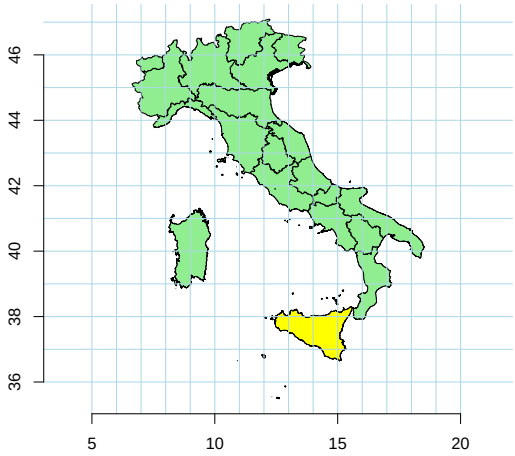
Shapefile from https://gadm.org/

```
> library(rgdal)
> ita1=readOGR(dsn="/Italy",layer="
    ITA_adm1",verbose=FALSE)
> plot(ita1,col="light green")
```

```
> ita1@data[,"NAME_1"][1:6]
[1] Abruzzo   Apulia    Basilicata
[4] Calabria Campania Emilia-Romagna
> pos=which(ita1@data[,"NAME_1"] == "
    Sicily")
> Poly_Sicily=ita1[pos,]
> plot(ita1,col="light green")
> plot(Poly_Sicily,col="yellow", add=
    TRUE)
```

Polygons & Regions

```
> plot(ita1,col="light green")
> plot(Poly_Sicily,col="yellow",add=TRUE
  )
> abline(v=5:20,col="light blue")
> abline(h=35:50,col="light blue")
> axis(1)
> axis(2)
```



Polygons & Regions

```
> pos=which(ita1@data[,"NAME_1"] %in% c("Abruzzo","
    Apulia","Basilicata","Calabria","Campania","Lazio
    ","Molise","Sardegna","Sicily"))
> ita_south = ita1[pos,]
> ita_north = ita1[-pos,]
> plot(ita1)
> plot(ita_south,col="red",add=TRUE)
> plot(ita_north,col="blue",add=TRUE)
```

Polygons & Regions

```
> library(rgeos)
> ita_s=gUnionCascaded(ita_south)
> ita_n=gUnionCascaded(ita_north)
> plot(ita1)
> plot(ita_s,col="red",add=TRUE)
> plot(ita_n,col="blue",add=TRUE)
```

Polygons & Regions

```
> plot(ita1,col="light green")
> plot(Poly_Sicily,col="yellow",add=TRUE
    )
> gCentroid(Poly_Sicily,byid=TRUE)
SpatialPoints:
          x        y
14 14.14668 37.58842
Coordinate Reference System (CRS)
    arguments:
+proj=longlat +ellps=WGS84
+datum=WGS84 +no_defs +towgs84=0,0,0
> points(gCentroid(Poly_Sicily,byid=TRUE
    ),pch=19,col="red")
```

```
> G = as.data.frame(gCentroid(ita1,byid=
    TRUE))
> plot(ita1,col="light green")
> text(G$x,G$y,1:20)
```

Polygons & Regions

Consider two trajectories, caracterized by a series of knots (from the centroïds list)

```
> c1 =G[c(17,20,6,16,8,12,2),]
> c2 = G[c(13,11,1,5,3,4),]
```

We can convert segments into SpatialLines

```
L1=SpatialLines(list(Lines(list(Line(c1)
    ),"Line1")))
L2=SpatialLines(list(Lines(list(Line(c2)
    ),"Line2")))
```

then

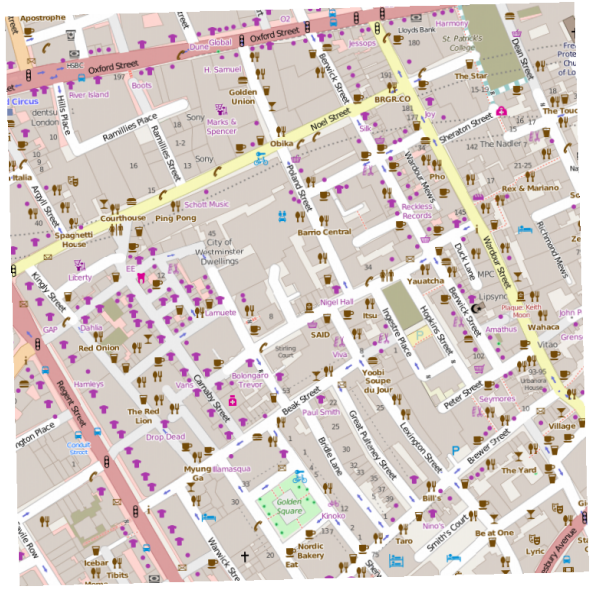
```
L1@proj4string=ita1@proj4string
L2@proj4string=ita1@proj4string
> cross_road =gIntersection(L1,L2)
```

Polygons & Regions

```
> cross_road@coords
SpatialPoints:
          x        y
1 11.06947 44.03287
1 14.20034 41.74879
Coordinate Reference System (CRS)
    arguments:
+proj=longlat +ellps=WGS84
+datum=WGS84 +no_defs +towgs84=0,0,0
> plot(ita1,col="light green")
> plot(L1,col="red",lwd=2,add=TRUE)
> plot(L2,col="blue",lwd=2,add=TRUE)
> plot(cross_road,pch=19,cex=1.5,add=
    TRUE)
```

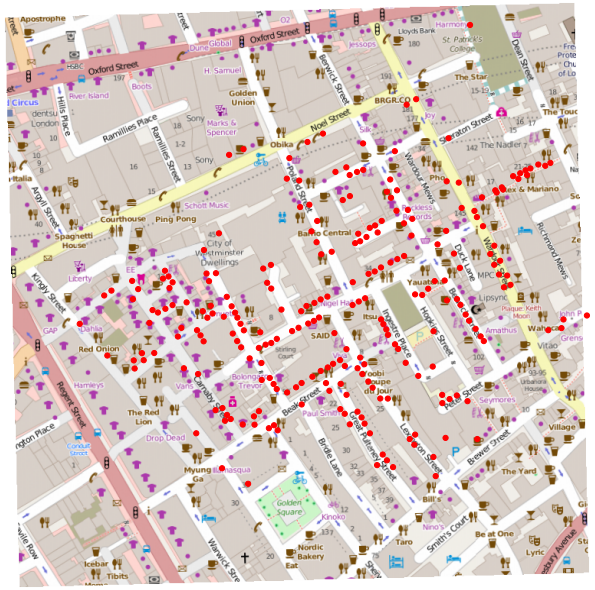
Points & HeatMap

```
> library(OpenStreetMap)
> map = openmap(c(lat= 51.516,   lon
    = -.141),
  c(lat= 51.511,   lon= -.133))
map=openproj(map, projection = "+
    init=epsg:27700")
> plot(map)
```



It is a standard R plot, then add points

```
> library(maptools)
> deaths=readShapePoints("
    Cholera_Deaths")
> head(deaths@coords)
  coords.x1 coords.x2
0  529308.7  181031.4
1  529312.2  181025.2
2  529314.4  181020.3
3  529317.4  181014.3
> points(deaths@coords,col="red",
    pch=19, cex=.7 )
```



Points & HeatMap

Cholera outbreak, in London, 1854,
dataset collected by John (not Jon) Snow

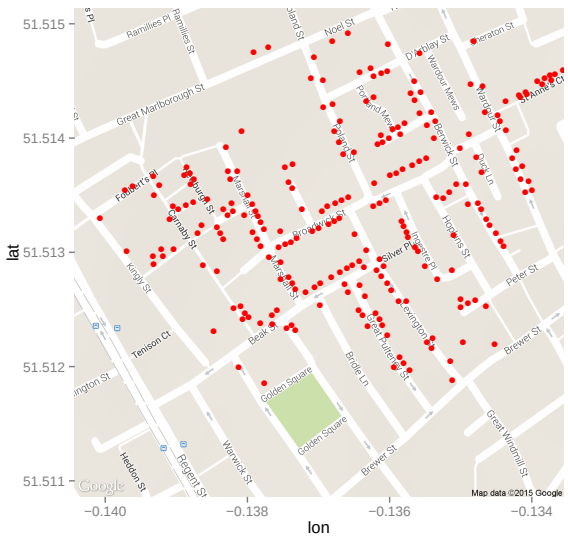
```
> library(ggmap)
> get_london = get_map(c(-.137,51.513),
    zoom=17)
> london = ggmap(get_london)
> london
```

Let us add points

```
> df_deaths = data.frame(X)
```

Points & HeatMap

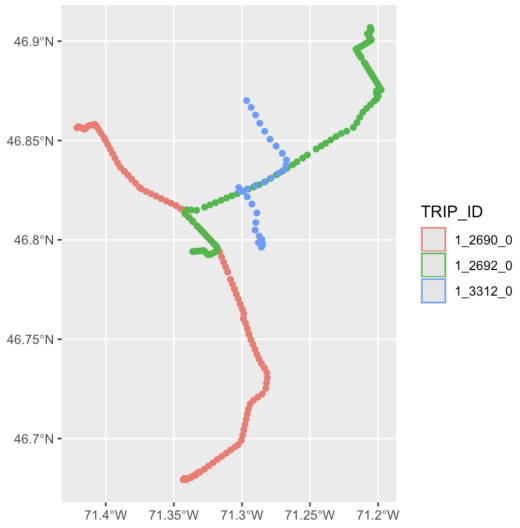
```
> library(sp)
> library(rgdal)
> coordinates(df_deaths)=~coords.x1+
    coords.x2
> proj4string(df_deaths)=CRS("+init=
    epsg:27700")
> df_deaths = spTransform(df_deaths,CRS
    ("+proj=longlat +datum=WGS84"))
> london + geom_point(aes(x=coords.x1,y
    =coords.x2),data=data.frame(
    df_deathscoords),col="red")
```

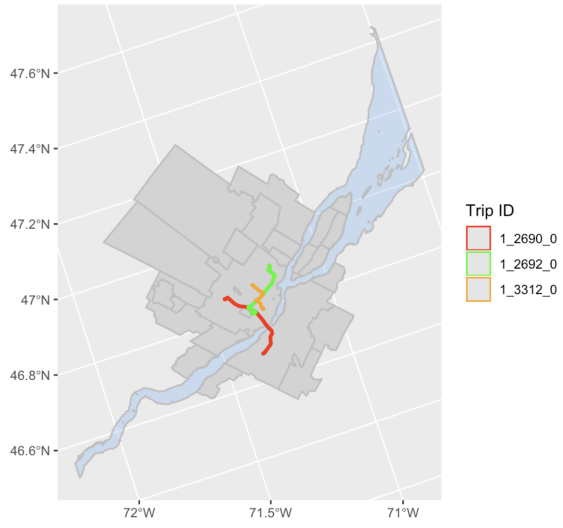


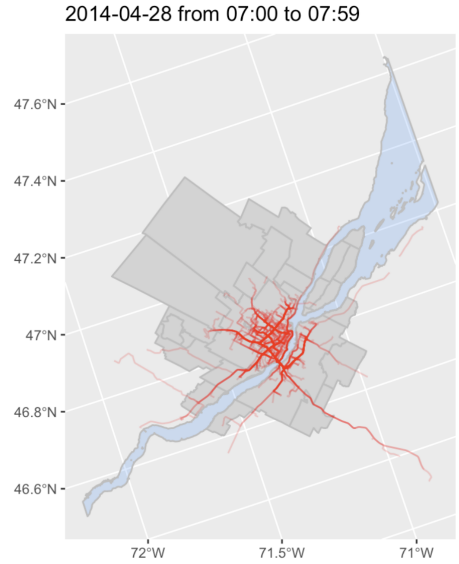
and we can add some heat map, too, see https://freakonometrics.hypotheses.org/19213

Lines & Telematic Data

See Ewen Gallic's tutorial





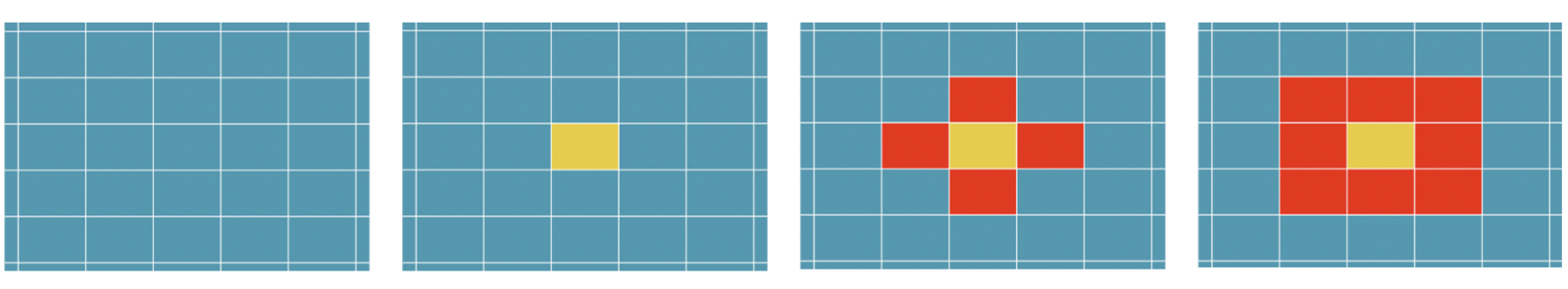


or Manuel d'analyse spatiale

Spatial Region

Need to define some concepts of contiguity,

- von Neumann neighborhoods, or rook
- Moore neighborhoods, or queen



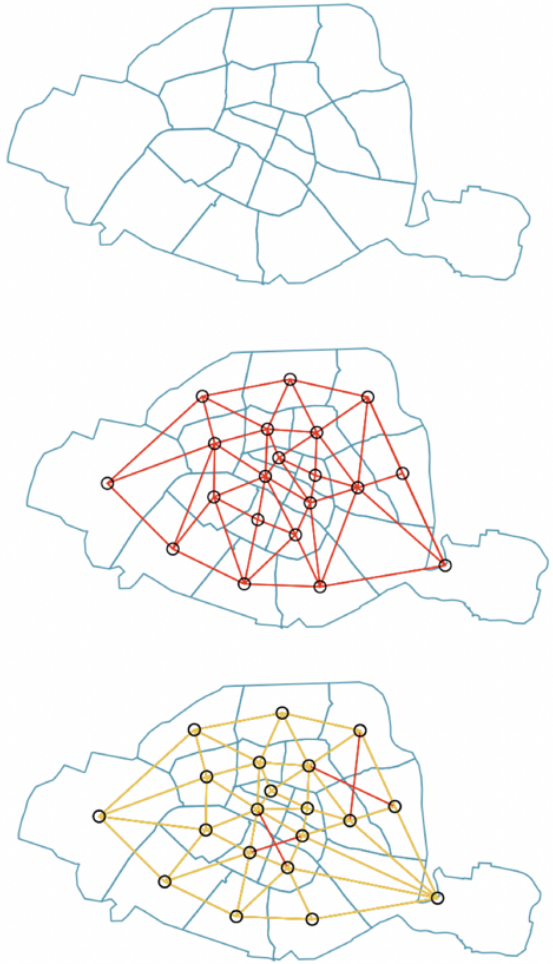
Given a collection of polygons, use

```
> library(spdep)
> contiguity_rook = poly2nb(shp, queen=FALSE)
> contiguity_queen = poly2nb(shp, queen=TRUE)
```

Spatial Region

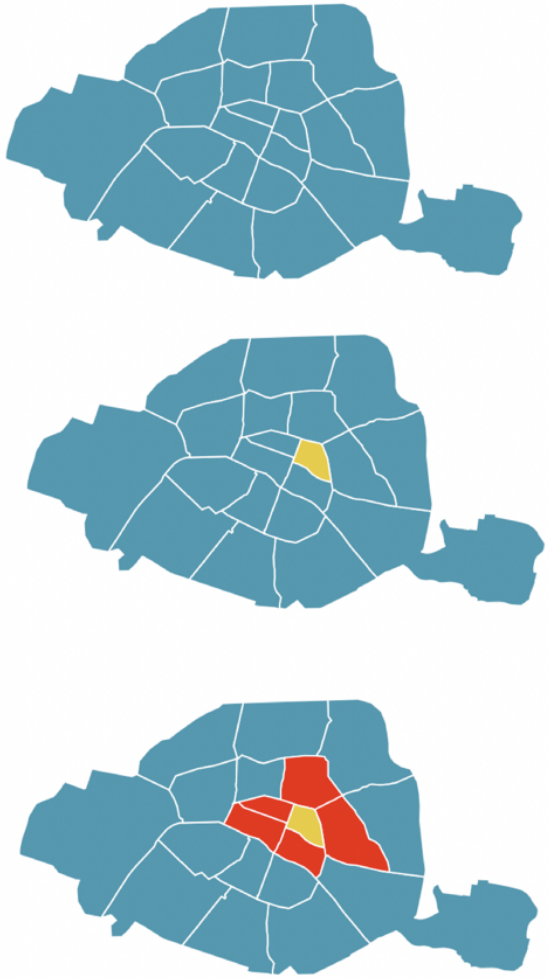
Shapefile from https://www.data.gouv.fr/fr/datasets

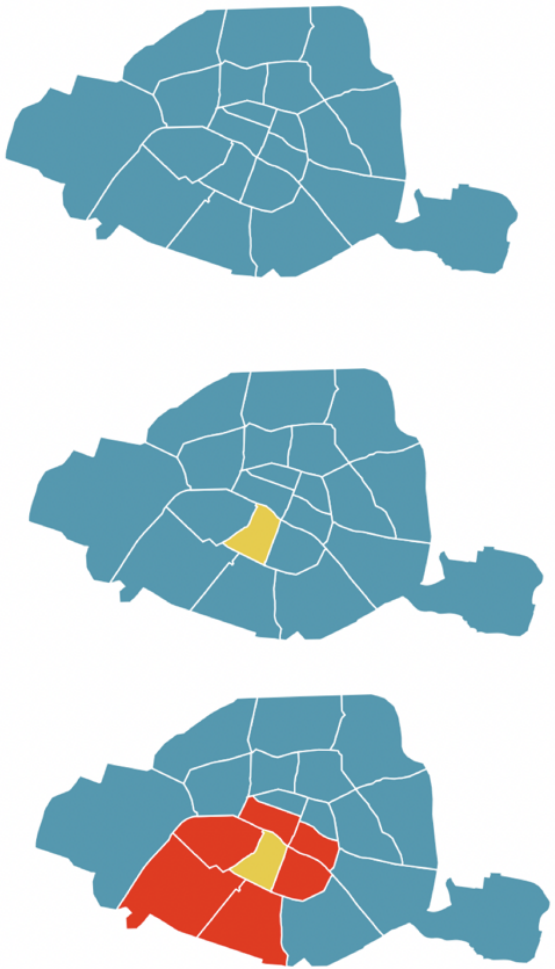
```
library(rgdal)
library(maptools)
library(tripack)
arr75 = readOGR("/arrondissements/")
plot(arr75)
coords = coordinates(arr75)
arr_nb = graph2nb(relativeneigh(coords))
plot(Sy7_nb,coordinates(arr75),add=TRUE,col=colr[2])
arr75_nb_queen = poly2nb(arr75)
plot(arr75)
plot(arr75_queen, coordinates(arr75), add=TRUE,col="red")
plot(arr75_nb_rook, coordinates(arr75), add=TRUE,col="yellow")
```



Spatial Region

```
> arr75_nb = poly2nb(arr75)
> plot(arr75, border="white", col=colr
    [3])
> COLR8 = rep("blue",21)
> COLR8[7] = "yellow"
> COLR8[arr75_nb[[7]]] = "red"
> plot(arr75, border="white", col=COLR8
    )
> arr75_nb[[7]]
[1]  5 10 16 17 18
```



Spatial Region

```
> arr75_nb = poly2nb(arr75)
> plot(arr75, border="white", col=colr
    [3])
> COLR12 = rep("blue",21)
> COLR12[11] = "yellow"
> COLR12[arr75_nb[[11]]] = "red"
> plot(arr75, border="white", col=COLR8
    )
> arr75_nb[[11]]
[1]  3  5  6  8  9 17
```