1. Descriptive Spatial Analysis

SOPHIE AUDRIC, MARIE-PIERRE DE BELLEFON, ERIC DURIEUX
INSEE

Abstract

The objective of spatial analysis is to understand and explore the entanglement between the spatial positioning of objects and phenomena and their characteristics. The literature traditionally distinguishes three types of spatial data – point data, continuous data and areal data. To each type of data correspond specific analytical methods. However, whatever the nature of the spatial data, the first step consists in manipulating them and aggregating them at a geographical scale appropriate to the underlying spatial process. Mapping the data can offer a synthesised view of a situation, make it understandable to a broader audience and give insight into which statistical tools would be best-suited to continue the study. This first descriptive analysis can also, when taking place as part of a study, bring to light specific problems in the data (collection, missing data, outliers, etc.) or lead to invalidate certain hypotheses necessary for the development of econometric methods. We have incorporated into this chapter the concepts in cartographic semiology needed to produce a quality map. [1]

1. These concepts in semiology were taken from the INSEE publication "Guide to Cartographic Semiology" (2017) and to which a large number of people contributed, whom we thank.

This chapter describes how spatial data can be manipulated using the R software and how the first descriptive maps be created. Studies carried out at various European statistical institutes are used to illustrate these concepts.

1.1 Different types of spatial data

Spatial data is an observation we know the value and the location of. The support of observations, defined as the set of spatial coordinates of the objects to be processed, offers a potentially rich source of information for the process analysis.
Some properties of spatial data contradict the assumptions necessary to the use of standard statistical methods. For instance, the hypothesis that observations are independent, a requirement in most econometric models, is not verified when *spatial dependence exists* – when the value of observation i influences the value of the neighbouring j observation. Another possible characteristic of spatial data is *spatial heterogeneity*: the influence of explanatory variables on the dependent variable depends on the location in space. A variable may prove influential within one neighbourhood, but not in another. Many methods have thus been specifically developed to analyse spatial data.
The methods and their objectives depend on the nature of the spatial data involved. According to the classification suggested by Cressie 1993b, three types of spatial data can be identified:

— point data;
— continuous data;
— areal data.

The fundamental difference between these data is not the size of the geographical unit in question, but the process that generated the data.

1.1.1 Point data

Point spatial data are characterised by the **spatial distribution** of the observations. The data generating process generates the geographic coordinates associated with the emergence of an observation. The value associated with the observation is not studied; only the location counts. The latter can be, for example, the location where a disease emerges during an epidemic, or how certain tree species are distributed in space. Spatial analysis of point data is aimed at **quantifying the gap between the spatial distribution of observations and a completely random distribution in space**. If the data are more aggregated than if they had been randomly distributed across the territory, clusters can be identified and their significance measured.

The main methods for analysing point data are described in Chapter 4: "Spatial distributions of points"

■ **Example 1.1 — Cluster Detection.** Fotheringham et al. 1996 have focused on detecting significant clusters of uncomfortable houses. They compare the spatial distribution of uncomfortable houses as identified on the ground with the distribution that would have emerged if they were distributed randomly among all houses. The hypotheses on random distribution in space make it possible to assess the significance of house groupings (Figure 1.1).

■

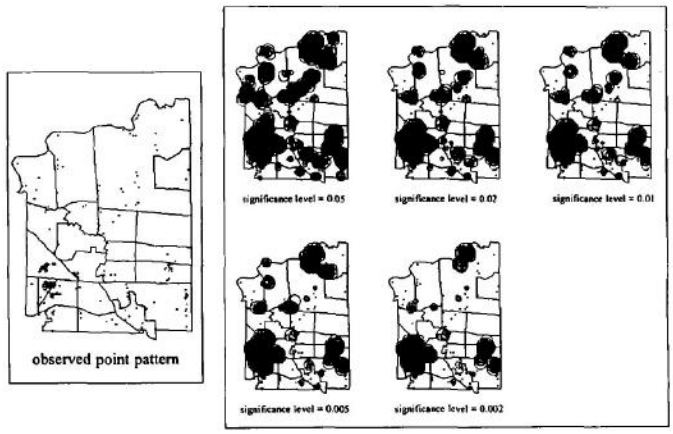


Figure 1.1 – Detecting significant clusters
Source: *Fotheringham et al. 1996*

1.1.2 Continuous data

With continuous data, there is a value for the variable of interest at any point across the territory studied. Data are generated on a continuous basis, across a subset of $\mathbf{R}^2$. However, these data are measured only in a discrete number of points. These include, for example, the chemical composition of the soil (data beneficial to the mining industry), water or air quality (for studies on pollution), or various meteorological variables. Spatial analysis of continuous data, also referred to as geostatistics, is aimed at predicting the value of a variable at a point where it has not been sampled, as well as the reliability of this prediction. Geostatistics also helps optimise the data sampling plan.

The main methods for analysing continuous data are described in Chapter 5: "Geostatistics".

■ Example 1.2 — Predicting pollution. *Chiles et al. 2005*
The researchers at the GeoSiPol (Pratices of geostatistics in polluted sites) working group take into account the spatial dependency structure between data using the *kriging* technique. They predict the quantity of pollutant found in places where the soil has not been sampled and quantify the estimation uncertainty (Figure 1.2). ■

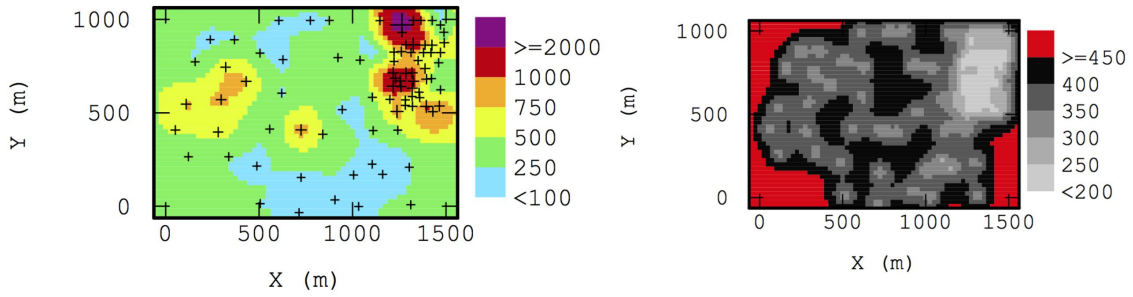


Figure 1.2 – Predicting the pollutant content in soil (mg/kg/m2) (left) and the standard deviation of this prediction (right)
Source: *GéoSiPol manual - Mines de Paris* Chiles et al. 2005

1.1.3 Areal data

Where areal data are concerned, while the location of the observations is assumed to be fixed, the associated values are generated according to a statistical process. These data are most often based upon a partition of the territory into contiguous zones, but they can also be fixed points on the territory. This includes, for example, GDP by region, or the number of marriages per town hall. The term 'areal' is therefore misleading, as these data are not necessarily represented on a surface. The focus here is on the relationship between **values of neighbouring observations**. The spatial analysis of areal data begins with **defining the neighbourhood structure of the observations**, then proceeds to **quantify the influence that observations have on their neighbours, and lastly assesses the significance of this influence.**

R The main techniques used for analysing areal data are described in chapter 2: "Codifying neighbourhood structure" and chapter 3: "Spatial Autocorrelation Indices", as well as in Part 3.

■ **Example 1.3 — Local spatial dependency.** Givord et al. 2016 have aimed to answer the question: "Are privileged lower secondary schools always located in a privileged environment?" For this purpose, the authors use *local spatial autocorrelation indices* [2]. These indices compare the similarity between a lower secondary school's social level and that of its environment with the similarity they would have if the same various social levels of lower secondary schools were randomly distributed among the schools. Local spatial autocorrelation indices make it possible to identify the lower secondary schools for which the influence of the surrounding social environment is significant (Figure 1.3).

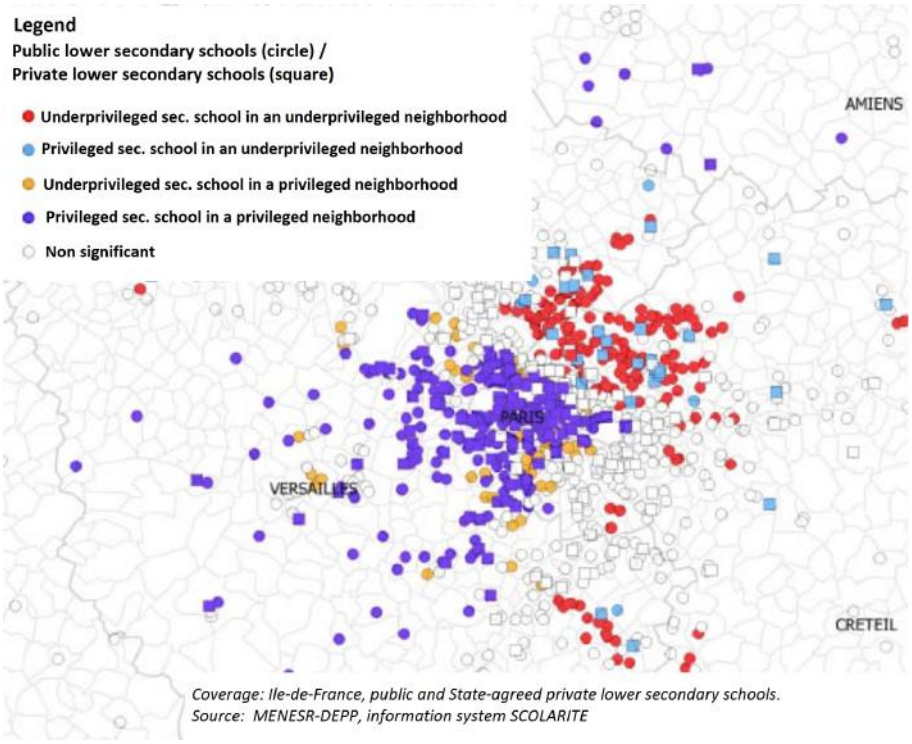


Figure 1.3 – Influence of the social level of the neighbourhood of a lower secondary school on the social level of the school itself
Source: *Givord et al. 2016*

2. Refer to chapter 3 "Spatial autocorrelation indices" for more details

■

Box 1.1.1 — Spatial data may fall into multiple categories. Dividing spatial data into three categories allows the analyst to choose the most appropriate method. However, it should be kept in mind that these categories are permeable and that the decision to analyse phenomena from one point of view stems from the scale of analysis and the very purpose of the study. For example, a house is considered to be a point object when studying significant groupings across space, but can also be seen as areal data when the aim is to identify the spatial correlation between the ages of the houses' inhabitants.

1.2 Concepts in cartographic semiology

1.2.1 What is cartographic semiology?

Cartographic semiology is the set of rules that make it possible to convey information as clearly as possible thanks to a cartographic image. It is good to have these rules in mind before moving on to designing a map using the R software. Cartographic semiology is a full-fledged language developed to facilitate communication, using graphic tools referred to as visual variables. When properly used, these variables reinforce the message while also making it clearer.

Visual variables include the shape, texture and size of the object to be depicted, its orientation and its colour. The latter may be connected with transparency effects or display a gradient of colours, according to a given scale of values. Dynamics have appeared more recently as a visual variable, with the creation of such outputs as animated maps.

Visual variables are distinctive for their ability to highlight:

— quantities, often represented as proportional circles;
— a hierarchy, representing an ordered set of relative values, for example population densities;
— differences between entities represented, for example industry and tourism;
— similarities, by grouping into a single set various objects reflecting the same theme.

Moreover, when well-combined, multiple visual variables can stress the point.

1.2.2 The objectives of a map

Graphs make it possible to directly and comprehensively grasp information and are an advantageous alternative to lengthy tables. This is even truer for maps. Their main interest lies in how they can integrate the spatial dimension, especially when the number of territories is relatively high. For instance, a map makes it possible to pick up on information at a single glance. The spatial dimension embraces all at once the geographical location, proximity to the coast, the mountains, large cities, neighbouring countries, etc., hence the importance of adding geographical benchmarks – neighbouring regions and countries, city names, rivers, thoroughfare, etc. Moreover, maps serve as good communications tools. They are easy to understand because their territory can generally be recognised and they offer a pleasant illustration. Technological advances in mapping tools, which are free of charge and easy to access, now make it easy to produce attractive maps. However, it is important that aesthetics not take precedence over relevance, and even more that they do not distort the information provided by the map.

1.2.3 To each type of data, its visual variable

The first question to be settled is that of the knowledge the map is intended to convey. To represent a variable in volume terms or an absolute number, proportional circles are used. For ratios, densities, trends, shares and typology, solid colour maps are recommended. Bilocated data or flows are best illustrated by radial flow maps, proportional arrows or vector resultants. Lastly, the location, for example of equipment, is shown using maps with symbols.

With solid colour maps (also referred to as class analysis or choropleth maps), the positive values are shown in warm tones (red, orange) while negative values generally appear in cool tones (blue, green). Moreover, a hierarchy in values can be reflected using a colour gradient, with the darkest (or brightest) colours reflecting the extreme values.

There are also rules for discretising data, *i.e.*, how observations are grouped into classes. The number of classes is calculated according to the number of observations. There are several theories for determining the optimal number. According to the Sturges rule, for example, it is equal to $1+3,3*log_{10}(N)$, where N is the number of observations.

In practice:
— for fewer than 50 observations: 3 classes;
— for 50 to 150 observations: 4 classes;
— for more than 150 observations: 5 classes.

The form of data distribution can also facilitate this choice. For instance, a class is added where both negative and positive values are found. Once the number of classes has been determined, a grouping method has to be chosen. There are several methods for doing so, each with advantages and drawbacks.

— **The quantile method** consists in using the same number of values per class. It results in a harmonious and easy-to-read map, on which the colours in the key are distributed equally. However, it is not always suited to the distribution of data.
— **The "classes of the same amplitude" method**: in this method, the interval between values is divided into ranges of the same length. This method is simple to understand but is very rarely suited to the distribution. Some classes may contain no value at all.
— **The Jenks and Kmeans methods** are designed to create homogeneous classes by maximising the variance between classes and minimising the variance within them. Unlike both previous methods, these methods are perfectly suited to data as they eliminate threshold effects. However, the Jenks method can entail a very protracted calculation time, if a large number of observations needs to be processed. In the latter case, the Kmeans method, which enables quicker calculation time even with a high number of observations, can be used. However, it can be unstable, resulting in different classes for a single dataset. This problem can be managed by repeating the Kmeans multiple times in order to keep the best limits in the end.
— **The manual arrangement method**: in this method, the mapper defines the limits of the classes. It is useful when aiming to highlight significant values (zero or near-zero boundary, average...) or to marginally improve the positioning of certain thresholds in accordance with local distribution. It also makes it possible to make different maps comparable with one another, by setting identical class boundaries. This method requires that the data distribution be analysed in advance, first using the Jenks or Kmeans method to develop homogeneous classes and then manually adjusting the class boundaries to prevent any threshold effects.

1.2.4 Some advice

— **One simple message per map**. Maps are often difficult to understand when they contain too much information. For example, because it is overly complicated, no message emerges from the map shown in picture 1.4. For an effective map, the rule "less is more" applies. Concretely, this means limiting the number of variables to be depicted on the same map.
— **Show the basic information**. A map must contain an informative title (most often connected with a descriptive sub-heading), a reference to the zoning shown, a key, a source and a copyright. The scale, logo or North arrow may also be included.
— **Not depicting the territory as an island**. It is best when readers are also provided with environmental information, as this enables them to locate the territory shown – for example,

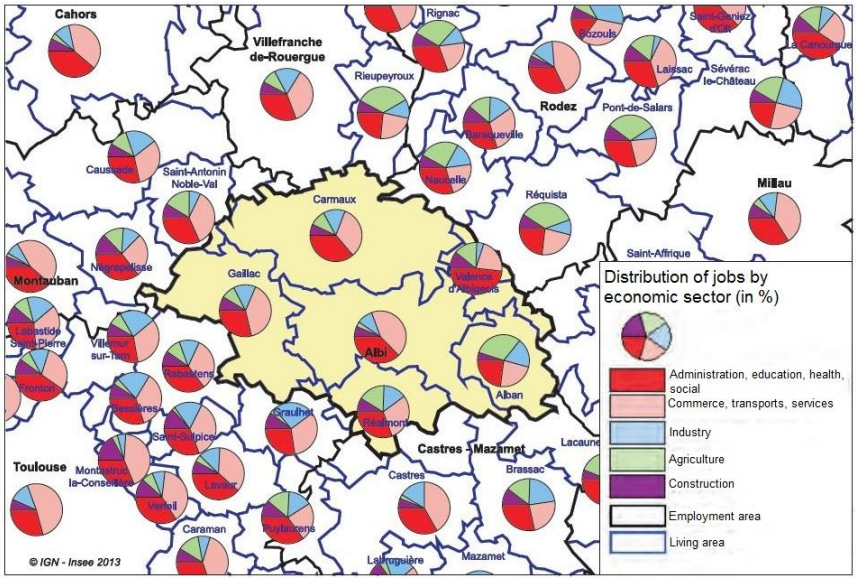


Figure 1.4 – Breakdown of jobs by business sector in living areas

bordering departments or regions, topographic elements such as the sea or the road system. In Figure 1.5, it would have been wise to represent the municipalities of the surrounding departments, in particular Dijon in the north or Lyon in the south, so as to illustrate the title, which is not very explicit.

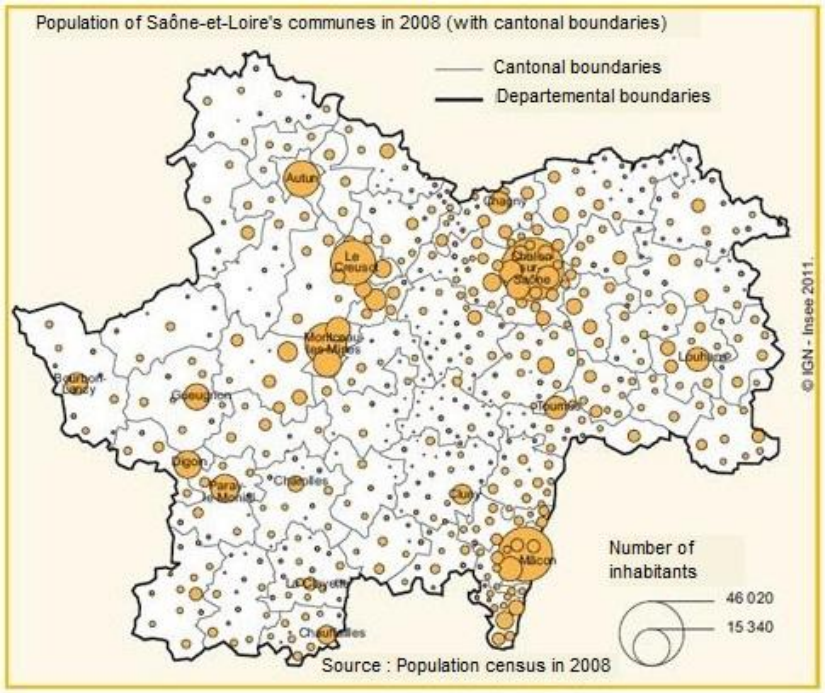


Figure 1.5 – Multiple medium-sized cities

Furthermore, it may be interesting to extend the analyses carried out on the territory to the surrounding environment, on the condition that the territory of interest emerges clearly, as in Figure 1.6 (dark green contour and light green line). The expanded analysis here makes it

possible to identify the geography of Toulouse's demographic dynamism compared to that of Bordeaux and to better understand the importance of the Languedoc urban system, in line with that of the Rhône corridor.

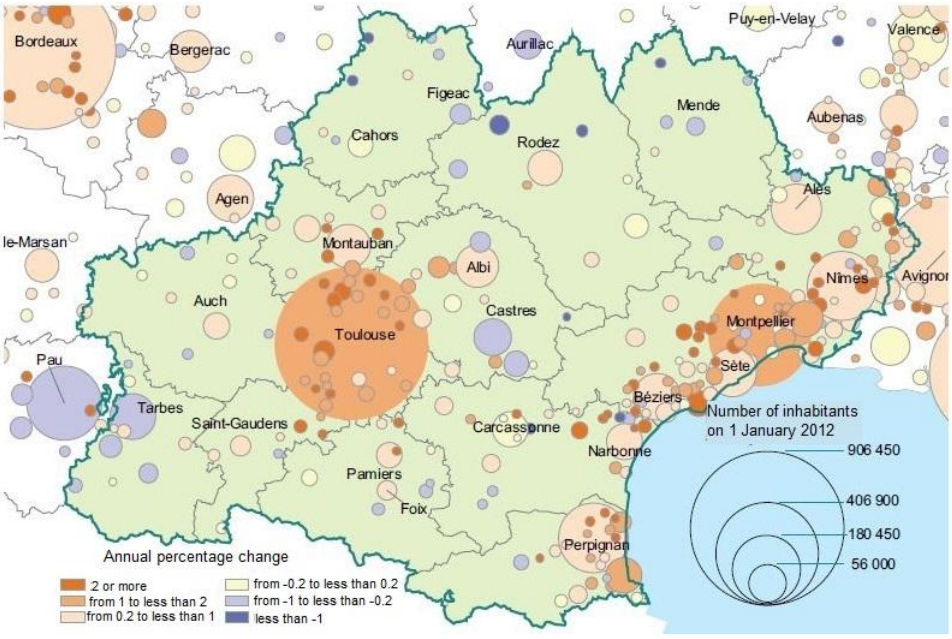


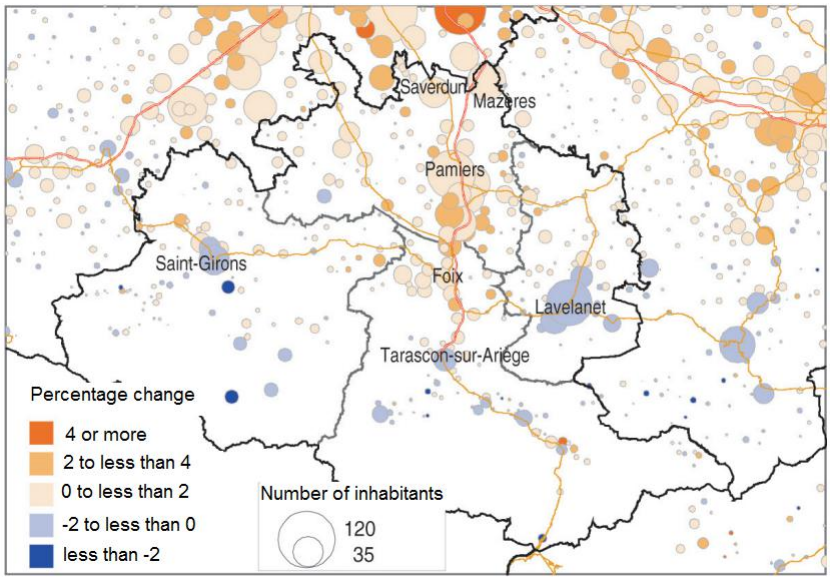
Figure 1.6 – A monocentric urban system around Toulouse and polycentric on the coast

— **Comparable maps**. When two maps showing the same territory with the same visual variables are placed side by side or below one another, it is an incentive for the user to make comparisons. To facilitate that process, both maps should have a harmonised key (same classes, circles or arrows) and the same scale with identical zoom. In the maps on Figure 1.7, the harmonised legends make it possible to compare the annual trend in population over the two periods from 1982-2011 and 2006-2011.

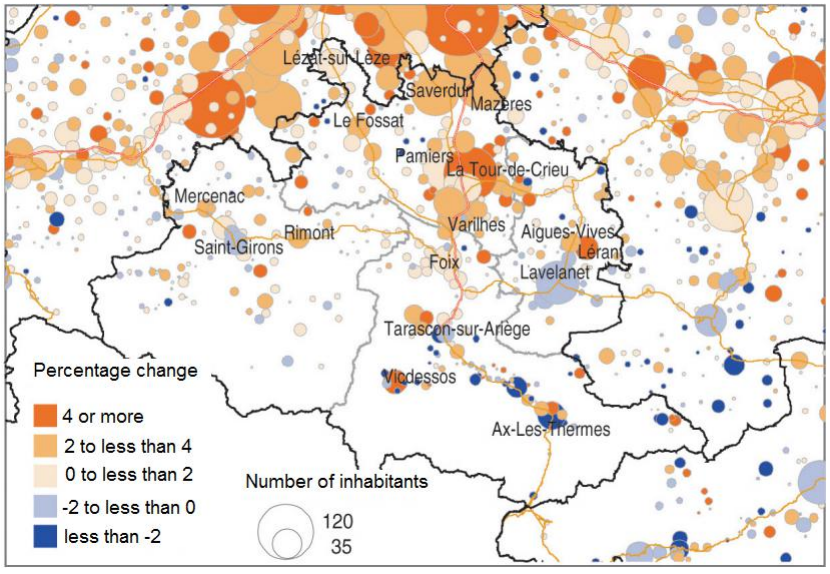
— **Choosing an indicator: proportion or volume?** Class-based analysis is used to represent a sub-population in relative value (or proportion) or a trend. It is to be proscribed when representing numbers of individuals or volumes, as it could lead the reader to misinterpret the map. The eye would establish a correspondence between the volume represented and the surface of the coloured territory. For instance, a class analysis of the number of inhabitants per municipality would lead to a visual overestimation of the population of Arles, the largest municipality in France. Furthermore, class-only analysis can sometimes be misleading as high percentages can sometimes apply to small numbers. This is why it is sometimes necessary to combine this type of analysis with a proportional circle analysis covering the numbers of individuals. Depending on the message we want to convey, we will choose to colour circles with a class analysis (Figure 1.7) or superimpose circles on a class analysis (Figure 1.8). When using coloured circles, the eye is more attracted to the size of the circles, while with the superimposed circles, the eye will first be drawn to the darkest colours in the class analysis.

1.3 Mapping data with R

Geolocated data can be aggregated on a more or less large geographical scale. They can then be mapped in different ways. In this section, we will describe how to simply start out in mapping



(a) Average annual trend in the population of the municipalities between 2006 and 2011



(b) Average annual trend in the population of the municipalities between 1982 and 2011

Figure 1.7 – Average annual trend in the population of the municipalities of Basse-Ariège
Source: *INSEE, Population census 1982, 2006, 2011*

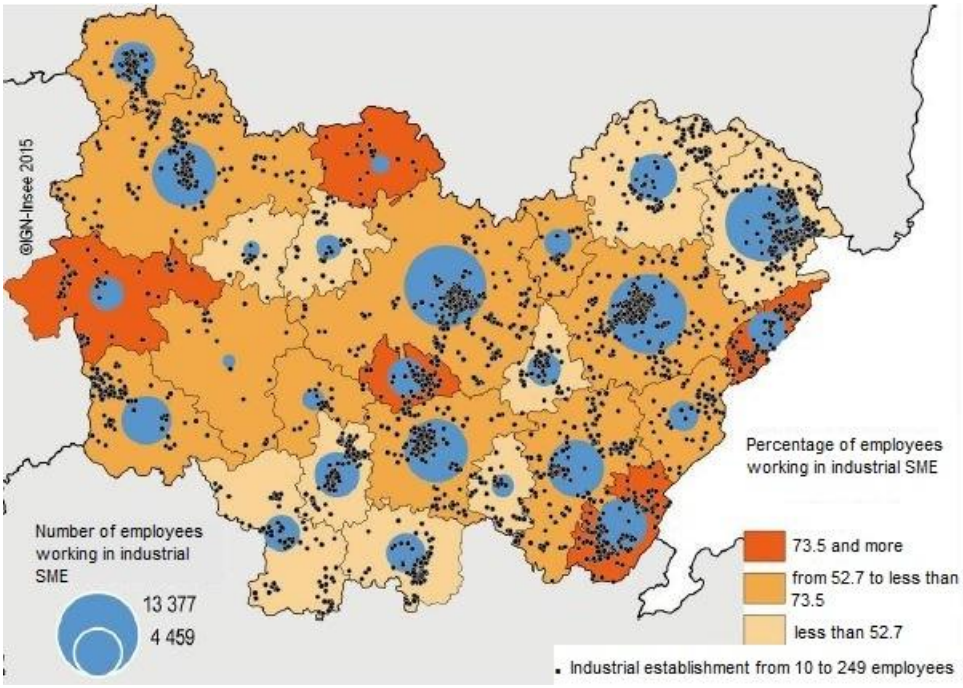


Figure 1.8 – Breakdown of employees working in an SME in the industrial sector
Source : *INSEE, Local Knowledge of Productive Resources - CLAP 2012*

with R, and present some appropriate packages. Many packages can be used to represent spatial data. The ones we will implement in this manual are:

— *sp*: basic package defining spatial objects;
— *rgdal*: import/export of spatial objects;
— *rgeos*: geometric manipulation;
— *cartography*: producing analysis maps.

We will also present the *sf* package that groups together all the functions of packages *sp*, *rgdal* and *rgeos*.

1.3.1 Manipulating spatial objects

Points, polygons, lines

The *sp* package makes it possible to create or convert various geometries into an `sp` object such as points, lines, polygons or grids, for instance. In general, each `sp object` is made up of different parts known as slots. Each slot contains specific information (geographical coordinates, table of attributes, system of coordinates, spatial scope, etc.)

Access to a slot for an `sp` object will take place *via* the operator `@` (`objet@slot`).

Spatial objects can be addressed in different forms. The first corresponds to **points**, *i.e.* a set of georeferenced points.

```
library(sp)

# contents of a communal table containing the coordinates of the town halls
# in WGS84 (latitude/longitude)

head(infoCom)
##         nom_commune latitude longitude préfecture
```

```
##                <chr>    <dbl>     <dbl>      <chr>
## 1  Faches-Thumesnil 50.58333  3.066667      Lille
## 2              Lille 50.63333  3.066667      Lille
## 3           Lezennes 50.61667  3.116667      Lille
## 4              Lille 50.63333  3.066667      Lille
## 5            Ronchin 50.60000  3.100000      Lille
## 6 Villeneuve-d'Ascq 50.68333  3.141667      Lille

# Transforming into a spatial object

municipalities<-  SpatialPoints(coords=infoCom[,c(2,3)])

#Viewing available slots

slotNames(municipalities)
##[1] "coords"      "bbox"        "proj4string"

# Understanding the spatial scope

municipalities@bbox  # ou bbox(municipalities)
##                 min       max
##latitude  50.000000 51.083333
##longitude  2.108333  4.183333
```

This object can also be represented graphically *via* the standard graphic instruction `plot` (illustration in Figure 1.9).

```
plot(municipalities)
```

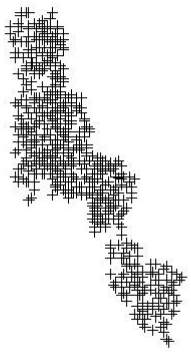


Figure 1.9 – Municipalities of Northern France
Source: *INSEE*

Our spatial object can also have a table of attributes describing the geographical objects it contains. The object then belongs to the `SpatialPointsDataFrame` class:

```
#Adding the attribute table

nord <- SpatialPointsDataFrame(coords=infoCom[,c(2,3)], data=infoCom[, c
    (1,4)])
```

This table of attributes is accessed *via* the new slot created `@data`:

```
nord@data
##          nom_commune préfecture
##                <chr>      <chr>
## 1   Faches-Thumesnil      Lille
## 2              Lille      Lille
## 3           Lezennes      Lille
## 4              Lille      Lille
## 5            Ronchin      Lille
## 6  Villeneuve-d'Ascq      Lille
## 7       La Madeleine      Lille
## 8              Lille      Lille
## 9            Comines      Lille
## 10         Deulemont      Lille
## # ... with 611 more rows
```

The creation of **georeferenced polygons**, although slightly more complex, follows the same logic.

First of all, we will create simple polygons using the coordinates of the vertices:

```
# Creating the sets of coordinates
x1 <- c(439518.5, 433091.8, 455774.1, 476566.1, 476944.2, 459554.4,
    439518.5)
y1 <- c(8045280, 8031293, 8018439, 8026756, 8044902, 8054731, 8045280)
        c1 <- data.frame(x1, y1)
x2 <- c(444929.2, 417667.9, 501837.1, 499792.5, 444929.2)
y2 <- c(8121306, 8078029, 8067465, 8109039, 8121306)
        c2 <- data.frame(x2, y2)
x3 <- c(456530.1, 450481.5, 472785.8, 476566.1, 456530.1)
y3 <- c(8101608, 8089510, 8087620, 8099717, 8101608)
        c3 <- data.frame(x3, y3)

# Creating the polygons
p1 <- Polygon(coords = c1, hole = F)
p2 <- Polygon(coords = c2, hole = F)
p3 <- Polygon(coords = c3, hole = T)
```

The parameter `hole` is used to identify polygons representing holes inside other polygons.

These objects have 5 slots, including:
- `@labpt`, which provides the coordinates of the centre;
- `@hole`, which establishes whether it is a hole;
- `@coords`, which allows the coordinates of the vertices to be retrieved.

They can then be assembled into multiple polygons:

```
P1 <- Polygons(srl = list(p1), ID = "PolygA")
P2 <- Polygons(srl = list(p2, p3), ID = "PolygB")
```

Consequently, polygon *P1* will be composed of *p1* and *P2* will be *p2* with a hole in the centre defined by *p3*.

They still have 5 different slots, including:
— `@Polygons` which lists the polygons used for its creation;
— `@ID` which provides the identifiers given to the polygon.

We then spatialise this set of polygons to make it a single spatial object:

```
SP <- SpatialPolygons(Srl = list(P1, P2))
```

Our spatial object is structured as follows: the `SpatialPolygons` contains a list of two `polygons` (multiple polygons), each containing a list of `Polygons` (single polygons), which contain the coordinates that delineate them. Thus, to access the coordinates of the first simple polygon contained in the second multiple polygon, we have to write:

```
SP@polygons[[2]]@Polygons[[1]]@coords
```

```
##          x2      y2
## [1,] 444929 8121306
## [2,] 499793 8109039
## [3,] 501837 8067465
## [4,] 417668 8078029
## [5,] 444929 8121306
```

To add an attribute table to our geographical object, simply create a `dataframe` containing as many lines as there are multiple polygons in our object. The lines must be sorted in the same order as the polygons and each line identified by the same identifier.

```
Info <- c("Simple", "Hole")
Value <- c(342, 123)
mat <- data.frame(Info, Value)
rownames(mat) <- c("PolygA", "PolygB")

SPDF <- SpatialPolygonsDataFrame(Sr = SP, data = mat)
```

A new `@data` slot is added to retrieve the table of attributes. This object can be represented graphically, resulting in Figure1.10.

```
plot(SPDF, col=c("lightgrey", "black"))
```

Objects such as **georeferenced lines** can be constructed in the same way as that shown previously for the polygons. This time, the functions `SpatialLines` and `SpatialLinesDataFrame` will be used.

Thus introduced, our municipal, departmental, etc. background maps will be objects of `SpatialPolygons (DataFrame)` type, our road or waterway background maps will be of `SpatialLines (DataFrame)` type and our airport or townhall background maps will be of `SpatialPoints (DataFrame)` type.

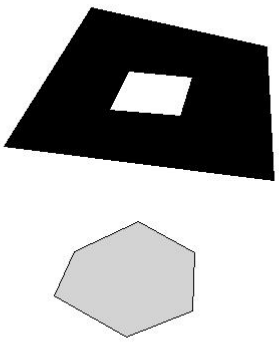


Figure 1.10 – Polygons generated

Working on a vector layer

Most of the time, we do not create geographical objects from scratch, and instead manipulate objects that already exist. Multiple packages can be used to import or export geographical objects. The simplest and most complete remains *rgdal* which makes it possible to read and manipulate a very large number of formats. The most common vector format is the "ESRI ShapeFile", which provides a base of maps through 5 files to be shown side-by-side in the same folder (.shp, .shx, .dbf, .prj, .cpg). All these files have the same name; only the extension will differ.

To import the background map, the function `readOGR` is used:

```
library(rgdal)
comr59<-  readOGR(dsn = "My_path\\", layer = "comr59", verbose = FALSE)
```

The parameters of function `readOGR` are:
- `dsn`: the pathway to the folder containing the files;
- `layer`: file name (without extension).

The result is then an object of R-type `SpatialPolygonsDataFrame` (example in Figure 1.11).

```
class(comr59)
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
plot(comr59)
```

`readOGR` makes it possible to import a wide range of cartographic formats. When using a background map from MapInfo, the syntax changes little:

```
comr59MI<- readOGR(dsn= "My_path\\comr59.tab", layer="comr59", verbose=
    FALSE)
```

As with the ShapeFiles, the MapInfo format is composed of a number of files that must all be found in the same folder, with the same name but different extensions. In this case, the `dsn` points to the .tab file, and the `layer` takes the name of the files in the background map.

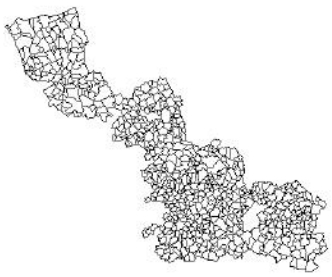


Figure 1.11 – North municipal background map
Source: *INSEE*

To select a subset of our map, refer to the associated dataframe *via* slot `@data`. For instance, to select municipalities with a surface area exceeding 200 km^2:

```
comr59_extended <- comr59[comr59@data$surf_m2>20000000, ]
```

To view this selection, the 2 objects are superimposed by colouring the selection in grey (the result can be seen in Figure1.12):

```
plot(comr59)
plot(comr59_extended,col ="darkgrey",add =TRUE)
```

Parameter `add=TRUE` makes it possible to superimpose the 2 funds.

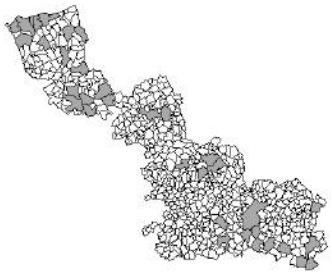


Figure 1.12 – Extended North municipal background map
Source: *INSEE*

To save our new cartographic background map, we use the `writeOGR` function which has as parameters:

— `obj`: R object to be exported;
— `dsn`: backup folder path;
— `layer`: common name of files (without extension);
— `driver`: object export format.

All possible formats are provided by the `ogrDrivers()` function.
For instance, to export our selection in the ShapeFile format:

```
writeOGR(comr59_extended,dsn="My_path",layer="comr59_extended",
driver="ESRI Shapefile")
```

In MapInfo format:

```
writeOGR(comr59_extended, dsn="My_path\\comr59_extended_MI.tab",
layer="comr59_extended_MI", driver="MapInfo File")
```

1.3.2 Producing statistical maps

Projection system

Spatial data are always associated with a projection system. The latter is identifiable by the slot `@proj4string`.

```
comr59@proj4string
## CRS arguments:
## +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
```

A projection system can be assigned to a created object. Let's use again previous presented polygons. They do not by default have any related projection system. To signify that they are in WGS84:

```
SPDF@proj4string <- CRS("+proj=longlat +datum=WGS84 +ellps=WGS84")
```

The EPSG standard now makes it possible to identify projection systems with a single code. It is 4326 for WGS84. In this context, the previous assignment could be coded:

```
SPDF@proj4string <- CRS("+init=epsg:4326")
```

All EPSG code matches can be obtained by executing `make_EPSG()`. Thus for Lambert 93 (used inter alia by IGN) the EPSG code is 2154.

To change the projection of a geographic object into a new coordinate system, the `spTranform()` function is used

```
comr59_l93 <- spTransform(comr59, CRSobj=CRS("+init=epsg:2154"))
```

This projection anew is necessary in particular to be able to superimpose two background maps that do not have the same coordinate system.

To produce maps very simply, we present the package *cartography* which in addition to being easy to handle is relatively comprehensive in its possibilities.

Maps in proportional symbols:
Mapping of stock data (such as population, number of equipment...) is done using symbols proportional to the size represented. The most common is the circle, but any other symbol can also be used. The code below makes it possible to call up Figure1.13.

```
library(rgdal)
library(cartography)
metr_nice <- readOGR(dsn="My_path",layer="metr_nice",verbose=F)

# Population data table
head(donnees_communales)
```

```
##   CODGEO                    LIBGEO  REG DEP P13_POP
## 1  01001  L'Abergement-Clémenciat   84  01     767
## 2  01002    L'Abergement-de-Varey   84  01     236
## 3  01004        Ambérieu-en-Bugey   84  01   14359
## 4  01005      Ambérieux-en-Dombes   84  01    1635
## 5  01006                  Ambléon   84  01     108
## 6  01007                 Ambronay   84  01    2503

#Background map plot
plot(metr_nice)

#addition of analysis
propSymbolsLayer(spdf=metr_nice,df =donnees_communales, spdfid = "Codgeo",
                         dfid = "CODGEO",var = "P13_POP",col ="salmon",
symbols="circle",legend .pos="right")

#map presentation
layoutLayer(title=  "Population of Nice Cote d'Azur metropolis",
            author = "INSEE", sources = "Census 2013",
            scale = NULL, north = TRUE)
```

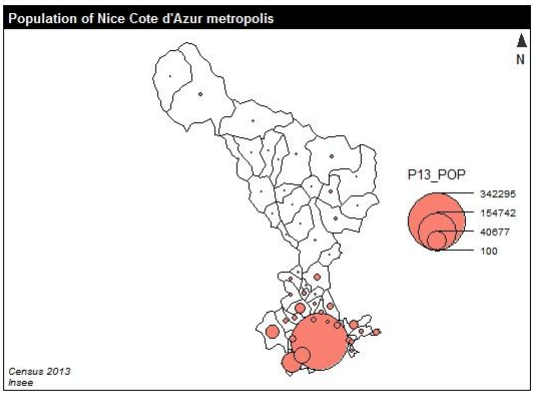


Figure 1.13 – Proportional symbols
Source: *INSEE, 2013 census*

The parameters of the function are:
- `spdf`: The SpatialPolygonsDataFrame;
- `df`: the dataframe containing the data to be analysed;
- `spdfid`: the map mesh identifier (in the slot `@data`);
- `dfid`: the line identifier in the dataframe. Must match the former;
- `var`: the dataframe variable to be analysed.

Other parameters exist and can be listed using the function.

Choropleth maps:
To depict the rates, solid colour or choropleth maps are used. The variable is divided into classes

and a colour gradient reflects the increase of values.

```
plot(metr_nice)

choroLayer(spdf=metr_nice,df =donnes_communales4, spdfid = "Codgeo",
dfid = "CODGEO",var = "TCHOM", nclass=4, method="fisher-jenks",
legend.pos="right")

layoutLayer(title=  "Unemployment rate at municipality level in the
    metropolis of Nice Cote d'Azur",
             author = "INSEE", sources = "Census 2013",
             scale = NULL, north = TRUE)
```

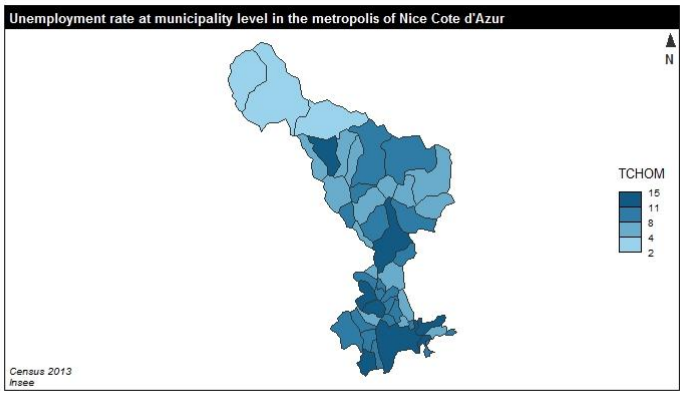


Figure 1.14 – Choropleth maps
Source: *INSEE, 2013 census*

The classification is done either by specifying the number of classes (`nclass`) and the grouping method (`method` that makes it possible to choose from among the methods shown in section 1.2) or by providing a threshold vector (`breaks`).

Other mapping functions:

— *propSymbolsChoroLayer*: this is a combination of proportional symbols and choropleth maps (to simultaneously represent a number of unemployed workers and an unemployment rate, for example);
— *typoLayer*: to represent a typology by specifying a qualitative variable and a colour vector of the same length as the number of modalities;
— *gradLinkLayer*: to represent flows or links.

Other packages make it possible to produce statistical maps using R. One example is:
— *RgoogleMaps*: To produce maps using road rasters or satellite GoogleMaps;
— *leaflet*: To produce interactive maps with OpenStreetMap Raster that can be inserted into web pages or even RShiny.

1.3.3 *sf*: the future of spatial data processing under R

As we have seen previously, cartographic data has been processed up to now using three main R packages:

— *sp* to implement spatial type classes;
— *rgdal* for input/output libraries;
— *rgeos* for operations on geometric objects.

Most recently, there has been a single package, named *sf*, which brings together all the functionalities of these 3 packages combined together. It provides users with a unique class for handling all spatial objects. In this chapter, we quickly present the main features of package *sf*. To better understand this package, handling rich geometry or managing inputs/outputs, the reader may refer to the various vignettes made available with the package on the CRAN website. This package is not compatible yet with all the spatial analysis packages presented in this manual, which are most often constructed using *sp*, *rgdal* and *rgeos*.

It is notable that inputs/outputs are much faster with *sf* than with *rgdal*.

Since objects of class *sf* are defined as `data.frame` augmented with geometric attributes, the manipulation of geographical objects is simplified, and is natively made, just like what is done in R for any table. Concretely, the package defines three classes of different objects:
— `sf`: a `data.frame` with spatial attributes;
— `sfc`: the column of the `data.frame` storing geometric data;
— `sfg`: the geometry of each recording.
A spatial object will thus be represented as shown in Figure1.15.

```
Simple feature collection with 96 features and 4 fields
geometry type:  MULTIPOLYGON
dimension:      XYZ
bbox:           xmin: 99225.97 ymin: 6049647 xmax: 1242375 ymax: 7110480
epsg (SRID):    NA
proj4string:    +proj=lcc +lat_1=44 +lat_2=49.00000000001 +lat_0=46.5 +lon_0=3
+x_0=700000 +y_0=6600000 +datum=NAD83 +units=m +no_defs
First 10 features:
   CODGEO                    LIBGEO INTREG REG                          geometry
1      01                       Ain     ES  82 MULTIPOLYGON Z (((943513 65...
2      02                     Aisne     NE  22 MULTIPOLYGON Z (((790281 69...
3      03                    Allier     ES  83 MULTIPOLYGON Z (((777281 65...
4      04 Alpes-de-Haute-Provence     SE  93 MULTIPOLYGON Z (((1016633 6...
5      05             Hautes-Alpes     SE  93 MULTIPOLYGON Z (((1022838 6...
6      06          Alpes-Maritimes     SE  93 MULTIPOLYGON Z (((1077507 6...
7      07                  Ardèche     ES  82 MULTIPOLYGON Z (((848816 64...
8      08                Ardennes      NE  21 MULTIPOLYGON Z (((873032.1 ...
9      09                  Ariège      SO  73 MULTIPOLYGON Z (((632344 61...
10     10                    Aube      NE  21 MULTIPOLYGON Z (((838365 67...
```


Figure 1.15 – Representation of a spatial object with the *sf* package

Importing existing map background is simplified under *sf* and is formatted as follows:

```
library(sf)
depf<- st _read("J:/CARTES/METRO/An15/Shape/Depf_region.shp")
```

It should be noted that there is no need to specify the import driver. `st_read` automatically adapts to the input file format. The function is compatible with the vast majority of common cartographic formats (ESRI-Shapefile, MapInfo, PostGIS, etc.). Spatial data can easily be mapped with the `plot` function (Figures 1.16 and 1.17).

Background map export is just as simple:

```
st_write(depf, "U:/fond_dep.shp")
```

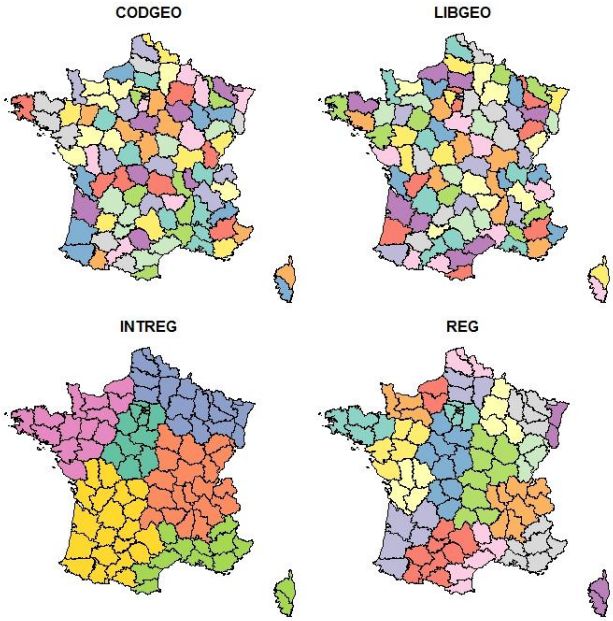


Figure 1.16 – Map obtained with code: `plot(depf)`

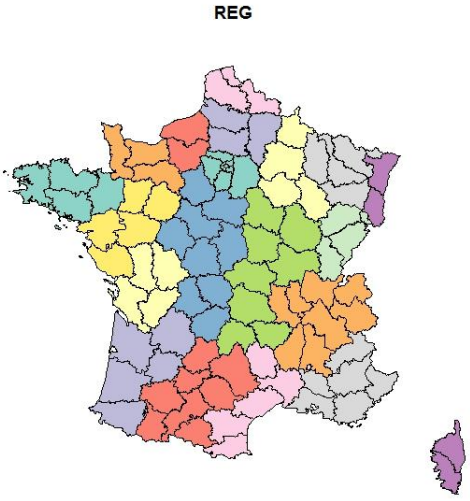


Figure 1.17 – Map obtained with code : `plot(depf["REG"])`

More generally, package *sf* offers a set of spatial data operators, all bearing the `st_` prefix and presented in Figure 1.18. The *sf* package is also fully integrated into the `tidyverse` environment

```
##  [1] "st_agr<-.sf"          "st_agr.sf"            "st_as_sf.sf"
##  [4] "st_bbox.sf"           "st_boundary.sf"       "st_buffer.sf"
##  [7] "st_cast.sf"           "st_centroid.sf"       "st_convex_hull.sf"
## [10] "st_coordinates.sf"    "st_crs<-.sf"          "st_crs.sf"
## [13] "st_difference.sf"     "st_geometry<-.sf"     "st_geometry.sf"
## [16] "st_intersection.sf"   "st_is.sf"             "st_line_merge.sf"
## [19] "st_make_valid.sf"     "st_polygonize.sf"     "st_precision.sf"
## [22] "st_segmentize.sf"     "st_set_precision.sf"  "st_simplify.sf"
## [25] "st_split.sf"          "st_sym_difference.sf" "st_transform.sf"
## [28] "st_triangulate.sf"    "st_union.sf"          "st_voronoi.sf"
## [31] "st_zm.sf"
```

Figure 1.18 – All the operators found in package *sf*

and thus perfectly matches the functionality of package *dplyr*:

```
library(dplyr)
depf<- left _join(depf,pop_dep,by="CODGEO")
```

or also the code below, illustrated in Figure1.19.

```
library(dplyr)
depf %>%
mutate(
area = st_area(.), # creating the new variable across the surface area
) %>%
group_by(REG) %>%
summarise(mean_area = mean(area)) %>%
plot
```

Most packages related to the processing of geographical data have adapted to this new category of objects. Some, like *spdep*, are in the test phase of their adaptation and therefore still require the use of the *sp* package.

Regarding *cartography*, the adjustment has been effective since version 2.0 and the syntax has changed:

```
choroLayer(x, spdf, spdfid,df , dfid,var , ...)
```

where `x` is an object of the `sf` type. If filled in, the objects `spdf`, `spdfid`, `df` and`dfid` are ignored because all the related information is included in the `x` object.

1.3.4 From the surface to the point, and *vice versa*

One special feature of areal data is that it may consist in a partition of the whole territory or a set of reference points with distinct geographical coordinates. However, it is easy to move from one representation to another:

— Voronoï polygons create a partition of the territory based on the reference points;
— using the centroid of an area makes it possible to move from a partition of the territory to a set of points.

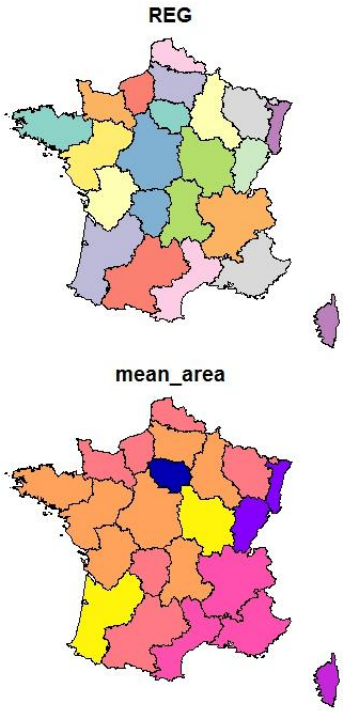


Figure 1.19 – Map produced using packages *sf* et *dplyr*

Definition 1.3.1 — Voronoï polygon associated with point x_i. This is the area of space that is closer to x_i than to any other point in the set being studied $\mathbf{x}$:

$$C(x_i|\mathbf{x}) = \left\{ u \in \mathbb{R}^2 : ||u - x_i|| = \min_j ||u - x_j|| \right\} \tag{1.1}$$

Box 1.3.1 — Very frequently-used polygons. Voronoï polygons are among the most widely used geometric structures in the scientific community. According to Aurenhammer 1991, there are three main reasons explaining this interest. The first is that Voronoï polygons are directly observable in nature (in crystalline arrangements, for example). Secondly, they are one of the most fundamental structures defined by a discrete set of points: they show a very large number of mathematical properties and are connected to multiple other fundamental geometric structures. Lastly, Voronoï polygons make it possible to simplify a large number of algorithmic problems. The Voronoï polygon associated with a point is often considered as its "area of influence".

Historically, Gauss (in 1840) and later Dirichlet (in 1850) used Voronoï polygons in their study on quadratic forms. Voronoï took their work one step further to higher dimensions in 1908. A few years later, in 1934, Delaunay built a triangulation associated with the Voronoï polygons and demonstrated the richness of its mathematical properties.

The R package *deldir* makes it possible to calculate the Voronoï polygons associated with a set of points. The `deldir` function returns an object that can be represented with the `plot` function. The package also calculates multiple statistics associated with polygons, such as the surface of each polygon, or the number of its vertices (see detailed documentation).

Many algorithms can be used to build Voronoï polygons (the most effective is the Fortune algorithm) (Fortune 1987). The algorithm implemented by the `deldir` function begins by building a Delaunay triangulation from reference points. This triangulation maximises the triangles' minimal

angle. The vertices of the Voronoï diagram are the centres of the circles circumscribed in the triangles from the Delaunay triangulation. The edges of the Voronoï diagram are on the mediators of the edges of the Delaunay triangulation (the algorithm is detailed in Lee et al. 1980).

Application with R

```
#Packages required
library(deldir)
library(sp)

# Generating random points
x <- rnorm(20, 0, 1.5)
y <- rnorm(20, 0, 1)

#"Deldir" function used to calculate the Voronoï polygons
#based on two sets of geographic coordinates
vtess <- deldir(x, y)

#creates a working window
plot(x, y, type="n", asp=1)

#represents the points
points(x, y, pch=20, col="red", cex=1)

#represents the associated Voronoï polygons
plot(vtess, wlines="tess", wpoints="none", number=FALSE, add=TRUE, lty=1)
```

To move from a partition of the territory to a set of points, we can calculate the centroids of the surfaces (Figure1.20).

Definition 1.3.2 — Centroid of an S surface. Point that minimises the average quadratic distance to all S points:

$$\min_c \frac{1}{a(S)} \int_S ||x-c||^2 dx$$

$$c = \frac{1}{a(S)} \int_S x dx$$

Coordinates of c: average of coordinates of **all S points**

Application with R

```
#Calculating polygon centroids
#From a "Spatial Polygon Data Frame" file

library(GISTools)
centroids <- getSpPPolygonsLabptSlots(polygon)

plot(polygon)
points(centroids, pch = 20, col = "Green", cex=0.5)
```

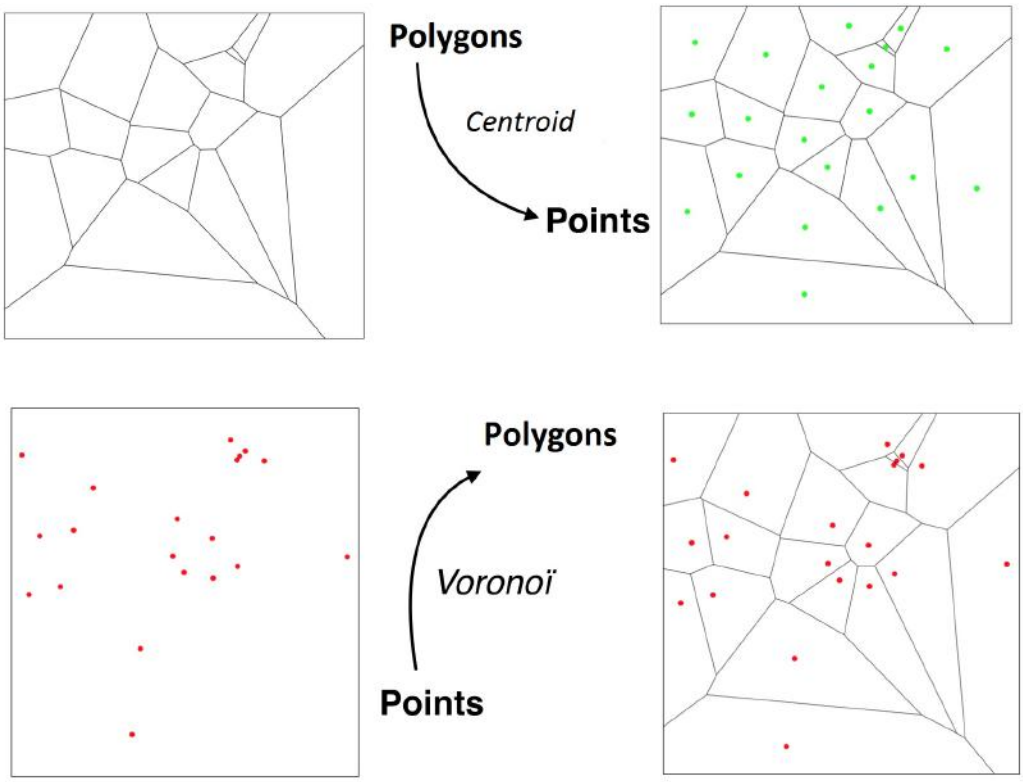


Figure 1.20 – Converting points to polygons and polygons to centroids

1.4 Examples of studies using aggregated spatial data

The European Data Integration Group [3] emphasises that representing data on a map with good spatial and temporal resolution makes it possible to detect phenomena that are otherwise invisible. An appropriate representation makes it possible to properly understand the economic, social or environmental situation and to implement relevant public policies. Through the work carried out by three European Statistical Institutes, this section illustrates the variety of descriptive analyses using spatial data – a European project to study regional poverty rates; the analysis of distance to green spaces by the Swedish Statistical Institute; the analysis of optimal location of wind turbines by the British Cartographic Society.

1.4.1 Access to green spaces - Statistics Sweden

Increasing access to public green spaces is one of the environmental objectives of the Swedish public policy. In many Swedish municipalities, debate opposes those in favour of increasing the concentration of living spaces with those in favour of preserving green spaces.

The combination of satellite mapping data and localised statistical information from the census makes it possible to better understand the situation on the ground and thus adjust public policies. This study is part of the United Nations' 11th Sustainable Development Goal: "Make cities and human settlements inclusive, safe, resilient and sustainable".

In 2013, the Swedish Statistical Institute drew upon the joint analysis of satellite images and administrative data to characterise the green spaces in Sweden, according to their ownership status and the quality of their vegetation. In most Swedish urban areas, more than 50% of the land is covered by green spaces. On average, three quarters of these spaces are public. Lidingö is the

3. UN-GGIM: United Nations Commitee of Experts on Global Geospatial Information Management - Working Group B - Europe

Swedish city covered by the highest proportion of green areas, since they represent approximately 72 % of its total area (figure 1.21) The second part of the study focuses on the accessibility of these green spaces. Using population census data, the Swedish Institute studied the proportion of adults and children living less than a certain distance from a public green space. It found, for instance, that **in 26 Swedish urban areas, less than one percent of the population lives more than 300 meters away from an accessible green space**. In some cities, however, such as Malmö, 15% of children under the age of 6 do not have access to a green space within less than 200 meters from their home (Figure 1.21).

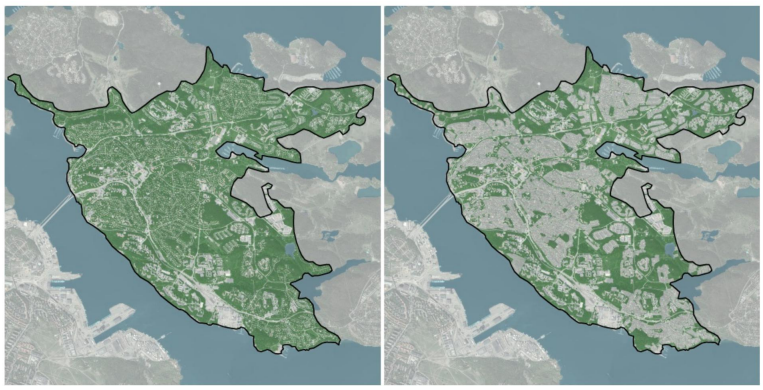


Figure 1.21 – City of Lidingö. Left: all green spaces; Right: green spaces accessible to the public (non-private)
Source: *Swedish National Institute of Statistics*

1.4.2 Regional poverty rate - European ESPON programme

The European EPSON project aims to promote the harmonisation of European public policies by making available regional statistics relevant to decision-makers. Differences in wealth between regions can exacerbate feelings of exclusion and tensions at the national level. Mapping the population's regional poverty rate makes it possible to distinguish the most fragile areas and thus to better target development aid policies.

The poverty threshold (*At-Risk-of-Poverty (ARoP) threshold*) is defined as 60% of the median national standard of living. The poverty threshold therefore varies by country (from €20,362 in Switzerland to €5,520 in Greece). The poverty rate (*ARoP rate*) is defined as the share of individuals whose standard of living is below the national poverty threshold. Figure 1.22 represents the ratio between this indicator calculated at the infra-national level (NUTS3) and the national poverty rate. This makes it possible to **identify the countries with the largest regional disparities**, and to **view the most extreme areas within each country**. The greatest inter-regional disparities in at-risk populations are observed in Turkey, Albania, Hungary, Germany, Croatia, Italy and Spain. The Scandinavian countries, the Netherlands, the Baltic States, Portugal and Greece have a more uniform distribution of *ARoP rates*. Within countries, there are low levels of poverty on the outskirts of capitals and cities, but not necessarily in the cities themselves. The poverty rate is higher in the least accessible regions, such as southern Italy, central Spain or eastern Hungary.

The mapping of poverty rates thus defined helps public decision-making at both national and European levels. To this end, the ESPON programme has published numerous mapping analyses of demographic and social data – a map of male-female ratios by region; various innovation profiles; variations in employment rates or the potential impact of climate change (`https://www.espon.eu/tools-maps`).

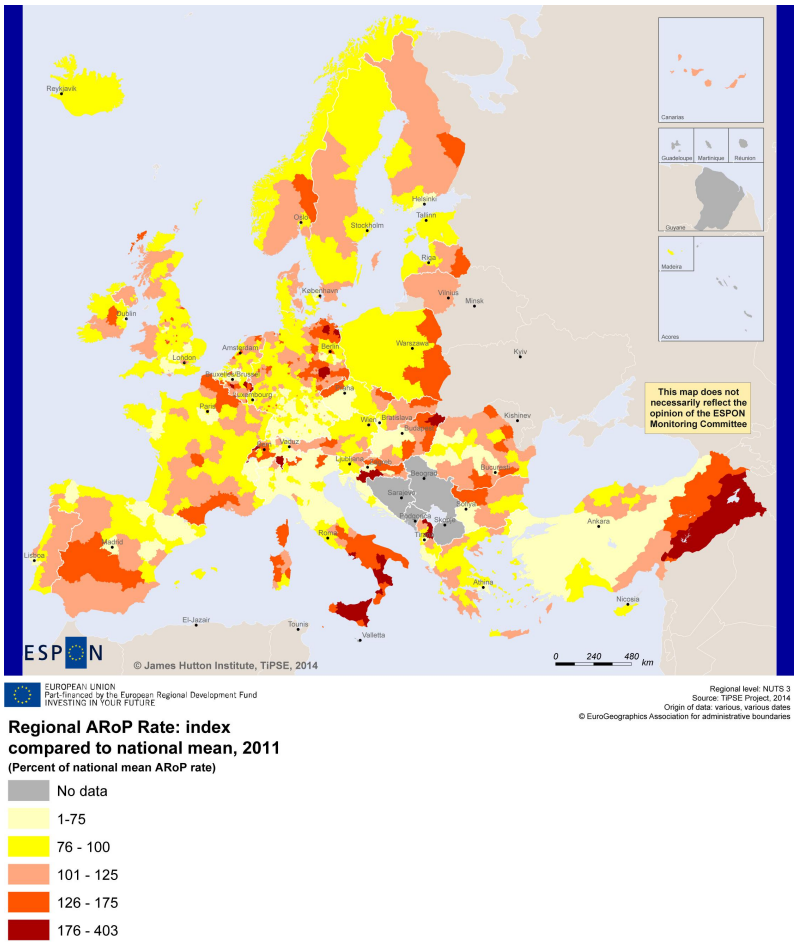


Figure 1.22 – High At Risk of Poverty Indicator
Source: *ESPON Project*

1.4.3 Optimal location of wind turbines - British Cartographic Society

The Scottish Government aims to increase renewable energy production by 2020. The Regional Council plays a key role in preserving the local balance, seeking to develop wind farms, while preserving the inhabitants' quality of living. The spatial data provided by the British Cartographic Society (*Ordnance Survey*) have proved very valuable to local decision making.

The objectives of the study are to provide clear and practical guidelines for the location of wind farms. It is planned that the study will take into account many environmental and social factors, such as landscape characteristics and the extent to which views are scenic. The mapping data must be sufficiently detailed to be used by local planners, while remaining easy enough to read and to be understood quickly by all stakeholders (Figure 1.23).

To achieve these objectives, the *Ordnance Survey* worked with many local experts and used many geolocated bases. The various players could monitor the progress of the study using an interactive map. Kevin Belton, GIS Officer, member of the Regional General Council, emphasises the study's added value: "Communicating complex planning information through spatial data has allowed the Council to engage with a wide range of stakeholders, from interested members of the public to commercial developers. This guidance is the end result and it helps ensure developers do not waste resources on applications that are contrary to policy – safeguarding protected areas, the environment and local communities."

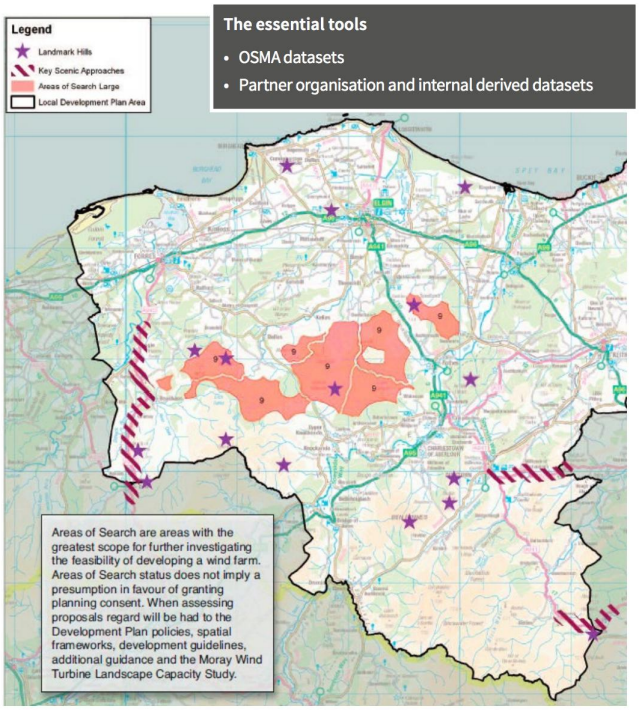


Figure 1.23 – Wind plant implantation study
Source: *British Ordnance Survey*

References - Chapter 1

Aurenhammer, Franz (1991). « Voronoi diagrams: a survey of a fundamental geometric data structure ». *ACM Computing Surveys (CSUR)* 23.3, pp. 345–405.

Bivand, Roger S et al. (2008). *Applied spatial data analysis with R*. Vol. 747248717.

Chiles, Jean-Paul et al. (2005). *Les pratiques de la géostatistique dans le domaine des sites et sols pollués*. GeoSiPol.

Cressie, Noel A.C. (1993b). « Statistics for spatial data: Wiley series in probability and statistics ». *Wiley-Interscience, New York* 15, pp. 105–209.

Fortune, Steven (1987). « A sweepline algorithm for Voronoi diagrams ». *Algorithmica* 2.1-4, p. 153.

Fotheringham, A. Stewart and F. Benjamin Zhan (1996). « A comparison of three exploratory methods for cluster detection in spatial point patterns ». *Geographical analysis* 28.3, pp. 200–218.

Givord, Pauline et al. (2016). « Quels outils pour mesurer la ségrégation dans le système éducatif ? Une application à la composition sociale des collèges français ». *Education et formation*.

Lee, Der-Tsai and Bruce J Schachter (1980). « Two algorithms for constructing a Delaunay triangulation ». *International Journal of Computer & Information Sciences* 9.3, pp. 219–242.