



# Explorer les modèles spatio-temporels avec R-INLA

Jiayue Liu

16 Janvier 2021

## Contents

<b>Introduction</b>	<b>1</b>
<b>I. Fondements mathématiques</b>	<b>2</b>
1. La probabilité bayésienne . . . . .	2
2. Modélisation et inférence bayésienne . . . . .	3
<b>II. Modèles spatiaux et spatio-temporels avec R-INLA</b>	<b>4</b>
1. Importation et préparation des données . . . . .	4
2. Modélisation en R-INLA . . . . .	8
<b>Bibliographie</b>	<b>13</b>

## Introduction

Ce rapport vise à explorer les modèles spatio-temporels, utiles dans un bon nombre de domaines d'études (santé publique, écologie etc.), sous un angle mathématique. Plus précisément, nous allons étudier la méthode INLA (*integrated nested Laplace approximation*) ainsi que son application dans l'environnement R. Enfin, nous livrons un cas d'études à l'aide des packages `leaflet` et `shiny` pour produire des résultats sur une carte interactive.

Le modèle qui fait l'objet de notre étude a été réalisé par Pr. Paula Moraga dans son ouvrage *Geospatial Health Data : Modeling and Visualization with R-INLA and Shiny*.

# I. Fondements mathématiques

## 1. La probabilité bayésienne

Avant de se plonger dans l'univers de l'INLA, il convient d'abord de connaître la philosophie bayésienne et ses implications dans notre travail de modélisation.

En effet le concept de l'inférence bayésienne est fondamentale. Elle stipule qu'à partir des événements connus (issus des observations sur le monde réel), on peut calculer la probabilité des différentes causes hypothétiques. À la différence de la probabilité classique (*Frequentist probability*), la pensée bayésienne se base sur la tolérance de la subjectivité. Cette philosophie peut être illustrée avec le théorème de Bayes qui peut s'écrire comme suit :

$$p(B | A) = \frac{p(A | B) * p(B)}{p(A)}$$

La probabilité de la cause hypothétique  $B$  étant donné que l'événement  $A$  est survenu peut être obtenue à partir de la probabilité dite *a priori* ( $p(B)$ ), la probabilité marginale de l'événement  $A$  ( $p(A)$ ), ainsi que le résultat statistique  $p(A | B)$  (c'est à dire la probabilité de survenue de l'événement  $A$  étant donné que l'hypothèse  $B$  est vrai).

Notez bien que la probabilité *a priori* peut provenir de l'expérience ou de l'intuition, mais aussi des résultats des études précédentes. Quoi qu'il en soit, elle s'appuie sur une croyance subjective de l'individu, tandis que la probabilité *a posteriori* ( $p(B | A)$ ) vient l'ajuster. (SANDERSON 2019)

Voyons à travers un exemple comment la méthode bayésienne permet d'ajuster certains "a priori" de notre vision du monde. Nous savons qu'en France, 0,3% de la population adulte est porteur du VIH. Supposons maintenant qu'avec un test sérologique, 95% des porteurs ont un résultat positif, tandis que 98% des non-porteurs ont un résultat négatif. La question est de savoir la probabilité qu'un adulte soit réellement porteur du VIH lorsque son test sérologique est positif. (BLANGIARDO 2015)

La raisonnement bayésien se déroule ainsi : que ce soit positif soit négatif, le résultat d'un test est un événement connu, tandis que la cause (c'est-à-dire l'infection par le virus) est incertaine. En appliquant le théorème de Bayes, on peut dénommer le fait d'être infecté comme un événement  $B$ , et celui de ne pas l'être comme un contre-événement  $B^C$ . L'événement  $A$  signifierait le fait d'avoir un résultat positif. On sait en revanche que le taux d'infection chez la population adulte, c'est-à-dire la probabilité *a priori*  $p(B)$  est de 0,003. Nous savons également que lorsqu'une personne est porteur du virus, le taux de positivité est de 95% ( $p(A | B) = 0,95$ ). Par ailleurs, la probabilité marginale de l'événement  $p(A)$  est la somme des probabilités d'être réellement infecté en tenant compte des différentes possibilités du résultat :  $p(A) = p(A | B) * p(B) + p(A | B^C) * p(B^C)$

Ainsi, la fameuse probabilité *a posteriori* peut se calculer avec l'équation suivante :

$$\begin{aligned} p(B | A) &= \frac{p(A | B) * p(B)}{p(A)} \\ &= \frac{p(A | B) * p(B)}{p(A | B) * p(B) + p(A | B^C) * p(B^C)} \\ &= \frac{0,95 * 0,003}{0,95 * 0,003 + 0,02 * 0,997} = 0,125 \end{aligned}$$

Nous pouvons donc conclure qu'en France, en vue des statistiques sur la propagation du sida et pour toute la population adulte confondue, lorsque le résultat d'un test VIH est positif, la probabilité que la personne est réellement infectée est de 12,5%.

## 2. Modélisation et inférence bayésienne

Toute (ou presque toute) la difficulté d'appliquer la méthode bayésienne dans la modélisation statistique réside dans la capacité à calculer (ou estimer) les valeurs incertaines. Dans notre exemple du test VIH, il n'y a que 2 possibles résultats : positif ou négatif, tandis que dans de nombreux phénomènes qui font l'objet des études statistiques, les résultats peuvent s'avérer multiples voire infinis. En plus, ladite probabilité *a priori* n'est pas toujours une valeur fixe et connue comme c'est le cas pour le taux de positivité d'un test sérologique. C'est pour toutes ces raisons - entre autres - qu'il nous faut, pour ainsi dire, calculer les "probabilités des probabilités".

On remplace maintenant les notations dans le théorème de Bayes par celles qui sont plus couramment utilisées dans l'inférence bayésienne :

$$\pi(\theta | \mathbf{y}) = \frac{\pi(\mathbf{y} | \theta)\pi(\theta)}{\pi(\mathbf{y})} \quad (1)$$

**N.B.**  $\pi$  remplace  $p$  car on n'est plus dans la logique d'une probabilité fixée, mais bel et bien dans celle d'une "loi de distribution".  $\pi(\theta | \mathbf{y})$  est ainsi appelé la "loi *a posteriori* conjointe" (*joint posterior distribution*).  $\pi(\theta)$  est quant à lui la loi des paramètres  $\theta$ , lequel prend sa valeur dans un espace  $\Theta$ , étant donnée les observations  $\mathbf{y}$ . Par construction,  $\pi(\mathbf{y} | \theta)$  est la vraisemblance (*likelihood* en anglais) des données observées  $\mathbf{y}$  étant donné les paramètres  $\theta$ . Enfin,  $\pi(\mathbf{y})$  est la vraisemblance marginale des données. (BLANGIARDO 2015)

Ce dernier variable est particulièrement épineux, car souvent il faut avoir recours au calcul intégral :

$$\pi(\mathbf{y}) = \int_{\theta \in \Theta} \pi(\mathbf{y} | \theta)\pi(\theta)d\theta \quad (2)$$

C'est la raison pour laquelle le théorème de Bayes s'exprime souvent comme suit :

$$\pi(\theta | \mathbf{y}) \propto \pi(\mathbf{y} | \theta)\pi(\theta) \quad (3)$$

Autrement dit, la loi *a posteriori* conjointe est proportionnelle au produit de la vraisemblance  $\pi(\mathbf{y} | \theta)$  et la loi *a priori*  $\pi(\theta)$ .

### Loi *a priori* conjuguée

Dans certains cas, la forme fonctionnelle de la loi *a priori* pourrait coïncider avec celle de la vraisemblance. Par exemple, si la vraisemblance est une loi gaussienne avec une précision connue, la loi *a priori* conjuguée à la moyenne est elle aussi une loi gaussienne. Cela veut dire que la loi *a posteriori* de la moyenne est également une loi gaussienne. D'autres modèles dits "conjugués" sont le modèle bêta-binomial, le modèle Poisson-Gamma etc. Toutefois, la plupart des modélisations sur les phénomènes du monde réel n'ont pas de loi *a priori* sous forme conjuguée – il faudrait donc confronter le problème de calcul intégral évoqué plus haut autrement.

### Méthodes de calcul approximatif et INLA

En général, quand il ne s'agit pas d'un modèle conjugué, il conviendrait d'appliquer d'autres méthodes qui visent à simuler les valeurs de la distribution *a posteriori*, telles que Monte Carlo (MC) ou sa dérivée Markov chain Monte Carlo (MCMC). Les simulations statistiques génèrent des valeurs aléatoires à partir d'une fonction de densité donnée par un ordinateur et peuvent devenir une procédure chronophage et coûteuse en termes de calcul lorsque les loi de distribution concernées sont complexes ou que la quantité de données est importante.

Par conséquent, une autre approche alternative est aujourd'hui fréquemment utilisée dans la modélisation statistique, en particulier pour les modèles ayant des lois de distribution gaussiennes latentes : INLA, ou *integrated nested Laplace approximation*.

A la différence des méthodes MC ou MCMC qui cherchent l'approximation par simulation, l'INLA adopte une approche plutôt analytique dans sa philosophie : l'approximation Laplace.

Pour plus de détails sur les lois gaussiennes latentes ainsi que la méthode de l'approximation Laplace, vous pouvez consulter l'ouvrage de Marta BLANGIARDO & Michela CAMELETTI (BLANGIARDO 2015).

## II. Modèles spatiaux et spatio-temporels avec R-INLA

Dans cette partie, nous allons découvrir comment construire un modèle spatio-temporel avec le package R-INLA. Cet exercice consiste à évaluer l'évolution des risques du cancer de poumon dans l'État d'Ohio aux Etats-Unis. **Le jeu de données, les étapes de modélisation, ainsi que le package SpatialEpiApp sont issus de l'ouvrage de Pr. Paula Moraga.** (MORAGA 2020)

Attention, le package R-INLA n'est pas déposé sur CRAN, il faut donc l'installer manuellement avec les codes suivants :

```
install.packages("INLA", repos=c(getOption("repos"), INLA="https://inla.r-inla-download.org/R/stable"), d
```

### 1. Importation et préparation des données

On importe les données sous forme csv via les lignes de commandes suivantes :

```
library(SpatialEpiApp)
namecsv <- "SpatialEpiApp/Data/Ohio/dataohiocomplete.csv"
dohio <- read.csv(system.file(namecsv, package = "SpatialEpiApp"))
head(dohio)
```

```
##   county gender race year y      n NAME
## 1      1      1    1 1968 6  8912 Adams
## 2      1      1    1 1969 5  9139 Adams
## 3      1      1    1 1970 8  9455 Adams
## 4      1      1    1 1971 5  9876 Adams
## 5      1      1    1 1972 8 10281 Adams
## 6      1      1    1 1973 5 10876 Adams
```

Ensuite, afin de pouvoir traiter la cartographie, on a également besoin d'un fichier shapefile également contenu dans le package SpatialEpiApp.

```
library(rgdal)
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.5-19, (SVN revision 1092)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.1.4, released 2020/10/20
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rgdal/gdal
## GDAL binary built with GEOS: TRUE
```

```
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ_VERSION: 631]
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rgdal/proj
## Linking to sp version:1.4-4
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading rgdal.
```

```
library(sf)
```

```
## Linking to GEOS 3.8.1, GDAL 3.1.4, PROJ 6.3.1
```

```
nameshp <- system.file("SpatialEpiApp/data/Ohio/fe_2007_39_county/fe_2007_39_county.shp",
                        package = "SpatialEpiApp")
map <- readOGR(nameshp, verbose = FALSE)
plot(map)
```



Le jeu de données contient des informations sur le nombre de cas de cancer de poumon, la population locale catégorisée par le sexe et le groupe racial pour chaque commune (county) dans l'Etat d'Ohio entre 1968 et 1988. Pour modéliser les risques non observables liés à la géographie et au temps (ainsi on parle de champs gaussien latent), on devrait d'abord calculer et structurer les variables suivantes :

- **county** : numéro d'identification des communes ;
- **year** : année ;
- **Y** : nombres de cas de cancer de poumon observés par commune par an ;
- **E** : nombres de cas de cancer de poumon modélisés par commune par an ;

- SIR : le SIR (taux d'incidence standardisé, ou *Standardized Incident Rate*) par commune par an.

Pour le variable Y, il faut agréger les données brutes avec la fonction `aggregate()`.

```
d <- aggregate(
  x = dohio$y,
  by = list(county = dohio$NAME, year = dohio$year),
  FUN = sum
)
names(d) <- c("county", "year", "Y")
head(d)
```

```
##      county year  Y
## 1     Adams 1968   6
## 2     Allen 1968  32
## 3   Ashland 1968  15
## 4 Ashtabula 1968  27
## 5    Athens 1968  12
## 6 Auglaize 1968   7
```

Pour le variable E, on peut faire appel au package `SpatialEpi` qui permet d'obtenir facilement les valeurs attendues. Mais avant de procéder à cette étape, il nous faut classer les données par commune par an, puis par sexe et groupe racial.

```
dohio <- dohio[order(
  dohio$county,
  dohio$year,
  dohio$gender,
  dohio$race
), ]
dohio[1:20, ]
```

```
##      county gender race year y      n NAME
## 1         1      1    1 1968 6    8912 Adams
## 22        1      1    2 1968 0     24 Adams
## 43        1      2    1 1968 0    8994 Adams
## 64        1      2    2 1968 0     22 Adams
## 2         1      1    1 1969 5    9139 Adams
## 23        1      1    2 1969 0     20 Adams
## 44        1      2    1 1969 0    9289 Adams
## 65        1      2    2 1969 0     24 Adams
## 3         1      1    1 1970 8    9455 Adams
## 24        1      1    2 1970 0     18 Adams
## 45        1      2    1 1970 1    9550 Adams
## 66        1      2    2 1970 0     24 Adams
## 4         1      1    1 1971 5    9876 Adams
## 25        1      1    2 1971 0     20 Adams
## 46        1      2    1 1971 1    9991 Adams
## 67        1      2    2 1971 0     27 Adams
## 5         1      1    1 1972 8   10281 Adams
## 26        1      1    2 1972 0     23 Adams
## 47        1      2    1 1972 2   10379 Adams
## 68        1      2    2 1972 0     31 Adams
```

Puisque la population est catégorisée par deux dimensions (sexe et groupe racial) qui sont elles-mêmes bidimensionnelles, nous spécifions que `n.strata = 4`. Ainsi, pour calculer la valeur attendue d'un échantillon vis-à-vis de la population, nous utilisons la fonction `expected()` :

```
library(SpatialEpi)
n.strata <- 4
E <- expected(
  population = dohio$n,
  cases = dohio$y,
  n.strata = n.strata
)
```

Maintenant il faut stocker les valeur attendues dans une dataframe dénommée `dE`, avec trois colonnes : `county`, `year`, et `E`. Ensuite, on fusionne les deux dataframes `dE` et `d` pour obtenir une dataframe bien préparée. Dernièrement, on fait un simple calcul du SIR.

```
#Obtenir le nombre d'années et le nombre de communes
nyears <- length(unique(dohio$year))
ncounties <- length(unique(dohio$NAME))
#Répéter le processus d'estimation pour chaque année
countiesE <- rep(unique(dohio$NAME),
  each = nyears)
#Répéter le processus d'estimation pour chaque commune
yearsE <- rep(unique(dohio$year),
  times = ncounties)
#Stocker les valeurs E dans la dataframe
dE <- data.frame(county = countiesE, year = yearsE, E = E)

d <- merge(d, dE, by = c("county", "year"))

d$SIR <- d$Y / d$E

head(d)
```

```
##   county year  Y      E      SIR
## 1  Adams 1968  6 8.278660 0.7247549
## 2  Adams 1969  5 8.501767 0.5881130
## 3  Adams 1970  9 8.779313 1.0251372
## 4  Adams 1971  6 9.175276 0.6539313
## 5  Adams 1972 10 9.548736 1.0472591
## 6  Adams 1973  7 10.099777 0.6930846
```

En résumé, nous avons maintenant un tableau avec, pour chaque commune et chaque année, le nombre de cas de cancer de poumon observé (`Y`) et `E` représentant la valeur théorique dans le modèle construit. Le `SIR` qualifie si le nombre de cas de cancer dans la commune  $i$  en l'an  $j$  est plus élevé que la valeur théorique ( $SIR_{ij} > 1$ ).

La dernière étape de cette préparation des données consiste à fusionner les données géographiques avec la dataframe.

```
library(tidyr)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2      v dplyr  1.0.2
## v tibble  3.0.4      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

dw <- reshape(d,
  timevar = "year",
  idvar = "county",
  direction = "wide"
)
map_sf <- st_as_sf(map)

map_sf <- merge(
  map_sf, dw,
  by.x = "NAME",
  by.y = "county"
)

map_sf <- gather(map_sf, year, SIR, paste0("SIR.", 1968:1988))
map_sf$year <- as.integer(substring(map_sf$year, 5, 8))

m <- map_sf %>% select(NAME, year, SIR, geometry)
```

## 2. Modélisation en R-INLA

En utilisant le modèle Bernardinelli (celui-ci constituant justement nos connaissances *a priori*), nous allons maintenant estimer le risque relatif du cancer de poumon pour chaque commune et chaque année avec l'équation suivante :

$$Y_{ij} \sim Po(E_{ij}\theta_{ij}) \log(\theta_{ij}) = \alpha + u_i + v_i + (\beta + \delta_i) * t_j.$$

où pour la commune  $i$  en l'an  $j$ ,  $Y_{ij}$  est le nombre de cas de cancer observé  $E_{ij}$  est le nombre de cas de cancer théorique, et  $\theta_{ij}$  représente donc le risque relatif (RR). Pour ce dernier, nous avons l'intersection qui est dénotée par  $\alpha$ ,  $(u_i + v_i)$  qui capte l'effet aléatoire d'une commune,  $\beta$  qui est l'effet de tendance linéaire, tandis que  $\delta_i$  représente le décalage entre cette tendance générale et la tendance spécifique à une commune. Dernièrement, bien entendu, nous avons  $t_j$  qui capte l'aspect temporel.

### Matrice de voisinage

L'effet aléatoire spatial est décrit par une matrice dite de voisinage (*Neighborhood Matrix*) dénommée **g** avec les fonctions **poly2nb** et **nb2INLA** dans le package **spdep**. Cette étape permet notamment de mesurer l'effet aléatoire entre les communes voisines puisque dans la modélisation spatiale, nous devrions prendre en compte la continuité de l'espace.

```
library(INLA)
```

```
## Loading required package: Matrix
```



```
##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##   expand, pack, unpack

## Loading required package: parallel

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##   accumulate, when

## This is INLA_20.03.17 built 2021-01-10 16:42:21 UTC.
## See www.r-inla.org/contact-us for how to get help.
## To enable PARDISO sparse library; see inla.pardiso()

library(spdep)

## Loading required package: spData

## To access larger datasets in this package, install the spDataLarge
## package with: 'install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')'

nb <- poly2nb(map)

head(nb)

## [[1]]
## [1] 26 55 71 72 80 85
##
## [[2]]
## [1] 19 35 42 70 82 86
##
## [[3]]
## [1] 5 8 16 25 30 59 69
##
## [[4]]
## [1] 14 27 28 34 49 51
##
## [[5]]
## [1] 3 16 29 30 44
##
## [[6]]
## [1] 11 12 83 84
```

```
nb2INLA("map.adj", nb)
g <- inla.read.graph(filename = "map.adj")
```

## Inférence avec R-INLA et interprétation des résultats

Nous créons deux vecteurs pour stocker les indices des communes et du temps.

```
d$idarea <- as.numeric(as.factor(d$county))
d$idarea1 <- d$idarea
d$idtime <- 1 + d$year - min(d$year)
```

La formule Bernardinelli citée plus haut peut s'écrire avec les lignes de commandes :

```
formula <- Y ~ f(idarea, model = "bym", graph = g) +
  f(idarea1, idtime, model = "iid") + idtime
```

où  $f(\text{idarea}, \text{model} = \text{"bym"}, \text{graph} = g)$  correspond aux variables  $u_i + v_i$ , tandis que  $f(\text{idarea1}, \text{idtime}, \text{model} = \text{"iid"})$  dénote  $\delta * t_j$  et  $\text{idtime}$  égale à  $\beta * t_j$

Enfin, on appelle la fonction `inla` pour calculer les lois *a posteriori* sur les paramètres prédicteurs (`fitted.values`), dont la moyenne est le risque relatif.

```
res <- inla(formula,
  family = "poisson", data = d, E = E,
  control.predictor = list(compute = TRUE))
summary(res)
```

```
##
## Call:
##   c("inla(formula = formula, family = \"poisson\", data = d, E = E, \", \"
##   control.predictor = list(compute = TRUE))")
## Time used:
##   Pre = 2.71, Running = 3.64, Post = 0.303, Total = 6.66
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) -0.530 0.024    -0.578   -0.530    -0.483 -0.530   0
## idtime       0.037 0.001     0.035    0.037     0.039  0.037   0
##
## Random effects:
##   Name      Model
##   idarea BYM model
##   idarea1 IID model
##
## Model hyperparameters:
##
##           mean      sd 0.025quant 0.5quant
## Precision for idarea (iid component)    24.63    4.22    17.14    24.37
## Precision for idarea (spatial component) 1882.66 1850.66    129.62 1337.72
## Precision for idarea1                    49497.23 17423.81  24244.32 46521.75
##
##           0.975quant   mode
## Precision for idarea (iid component)    33.67    23.94
## Precision for idarea (spatial component) 6757.57   355.86
## Precision for idarea1                    91686.88 41195.61
```

```
##
## Expected number of effective parameters(stdev): 111.13(4.70)
## Number of equivalent replicates : 16.63
##
## Marginal log-Likelihood: -5940.88
## Posterior marginals for the linear predictor and
## the fitted values are computed
```

```
head(res$summary.fitted.values)
```

```
##              mean          sd 0.025quant  0.5quant 0.975quant
## fitted.Predictor.0001 0.6778382 0.05302825  0.5790022 0.6760372 0.7869924
## fitted.Predictor.0002 0.7028935 0.05326590  0.6033340 0.7011830 0.8122300
## fitted.Predictor.0003 0.7289127 0.05357837  0.6284895 0.7272884 0.8385653
## fitted.Predictor.0004 0.7558780 0.05398361  0.6544266 0.7543334 0.8661332
## fitted.Predictor.0005 0.7838906 0.05451133  0.6811989 0.7824180 0.8949614
## fitted.Predictor.0006 0.8129216 0.05518460  0.7087327 0.8115121 0.9250979
##              mode
## fitted.Predictor.0001 0.6724978
## fitted.Predictor.0002 0.6978100
## fitted.Predictor.0003 0.7240749
## fitted.Predictor.0004 0.7512690
## fitted.Predictor.0005 0.7794900
## fitted.Predictor.0006 0.8087048
```

Enfin, nous devons également fusionner les données issues de l'inférence et les données dites cartographiques dans `map_sf`.

```
d$RR <- res$summary.fitted.values[, "mean"]
d$LL <- res$summary.fitted.values[, "0.025quant"]
d$UL <- res$summary.fitted.values[, "0.975quant"]

m <- merge(
  m, d,
  by.x = c("NAME", "year", "SIR"),
  by.y = c("county", "year", "SIR")
)
```

## Cartographier le risque relatif

A l'aide des packages `leaflet` et `shiny`, nous allons maintenant visualiser ces résultats.

```
##
## Listening on http://127.0.0.1:7986

## Warning: sf layer has inconsistent datum (+proj=longlat +datum=NAD83 +no_defs).
## Need '+proj=longlat +datum=WGS84'
```

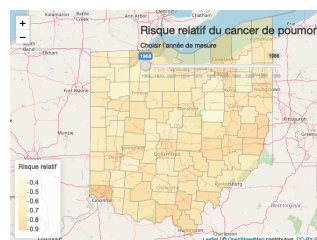
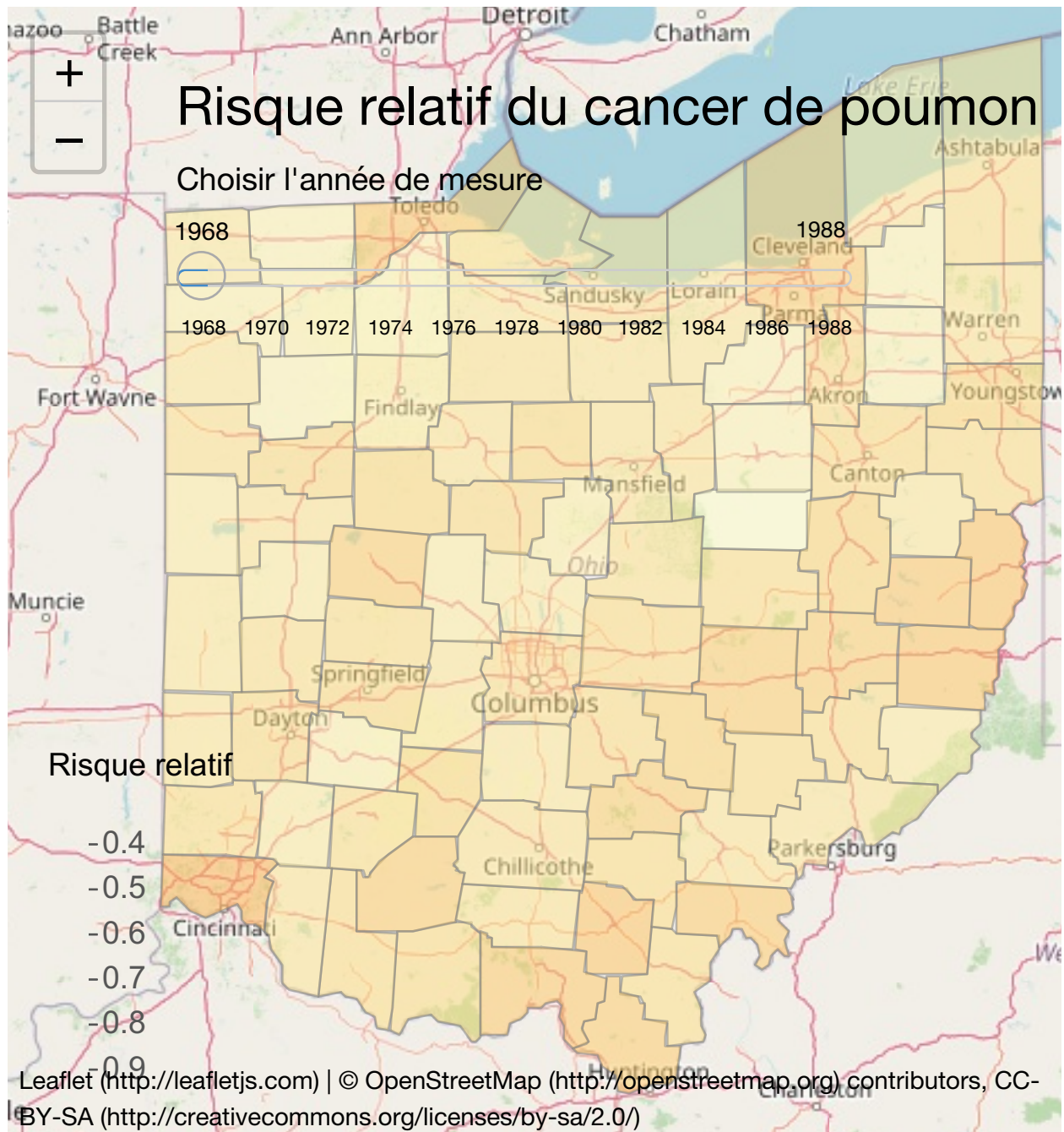


Figure 1: "Carte interactive du risque relatif du cancer de poumon en Ohio, 1968-1988"

## Bibliographie

BLANGIARDO, Michela, Marta & CAMELETTI. 2015. *Spatial and Spatio-Temporal Bayesian Models with R-Inla*. John Wiley & Sons, Ltd.

MORAGA, Laura. 2020. *Geospatial Health Data*. CRC Press. <https://www.paulamoraga.com/book-geospatial/>.

SANDERSON, Grant. 2019. *Bayes Theorem (Vidéo)*. 3blue1brown. <https://www.3blue1brown.com>.