Fiche TD avec le logiciel R : geolocRNSA

La géolocalisation des sites du RNSA

Pr Jean R. Lobry

La représentation graphique de la localisation de 127 sites RNSA en France métropolitaine pose le problème de la gestion du chevauchement des textes sur une carte.

1 Source des données

Les données sont publiques[1], il s'agit ici de la version du 29 septembre 2026. C'est un fichier de type tableur qui pour 130 sites RNSA donne sa latitude, longitude, altitude et la hauteur par rapport au sol du capteur de pollen.

```
library(readxl)
rnsa <- read_xlsx("geolocRNSA/liste-des-stations-de-mesures-pollen-coordonnees-gps.xlsx")
rnsa <- as.data.frame(rnsa)
comment(rnsa) <- paste("Généré le :", Sys.time())
save(rnsa, file = "geolocRNSA/rnsa.Rda")
```

```
chmin <- "https://esb.univ-lyon1.fr/donnees/geolocRNSA/"
load(url(paste0(chmin, "rnsa.Rda")))
```

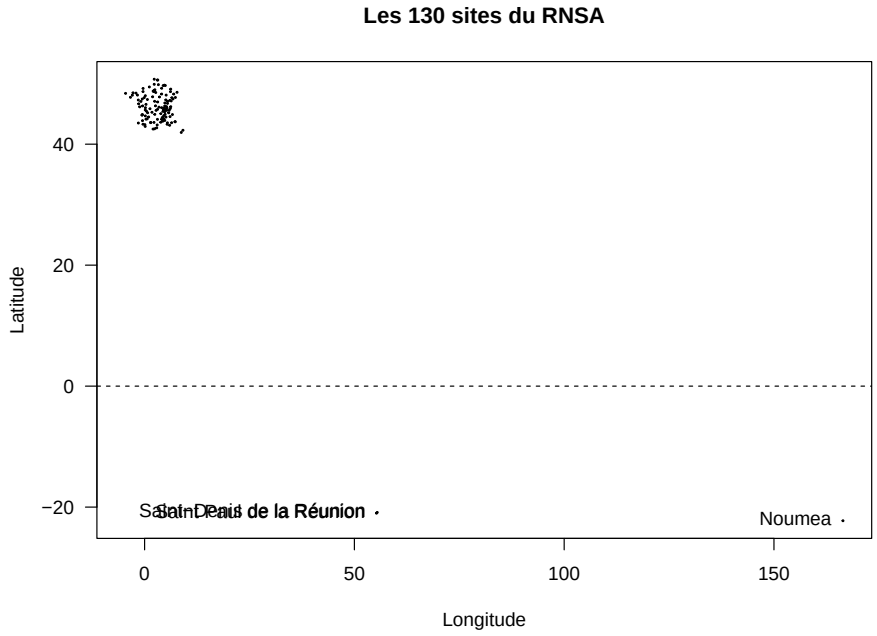
Une représentation brute des données nous permet de repérer trois sites que nous allons exclure ici car situés en outre-mer (Noumea, Saint-Denis de la Réunion, Saint-Paul de la Réunion). Il nous restera donc 127 sites dans la table `fm`. La superposition des noms de Saint-Denis de la Réunion et de Saint-Paul de la Réunion préfigure le problème que nous allons rencontrer dans la suite pour représenter de façon lisible le nom des sites.

```
with(rnsa, {
  plot(Longitude, Latitude, pch = 19, cex = 0.2, las = 1,
       main = "Les 130 sites du RNSA")
  abline(h = 0, lty = 2)
  rdom <- which(Latitude < 0) # hémisphère Sud
  text(Longitude[rdom], Latitude[rdom], Stations[rdom], pos = 2)
})
subset(rnsa, Latitude < 0)[ , 1:3]
```

```
                    Stations Latitude Longitude
88                    Noumea -22.2553  166.4506
107 Saint-Denis de la Réunion -20.8789   55.4481
115  Saint Paul de la Réunion -21.0097   55.2697
```

```
fm <- subset(rnsa, Latitude > 0) # France métropolitaine
```

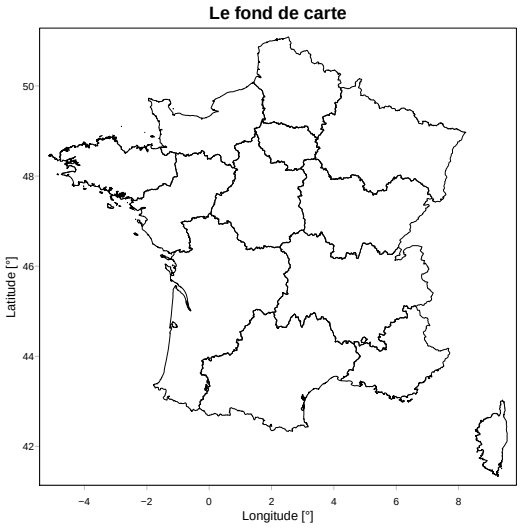
[1] https://www.data.gouv.fr/datasets/donnees-historiques-de-surveillance-des-pollens-et-des-moisissures



2 Le fond de carte

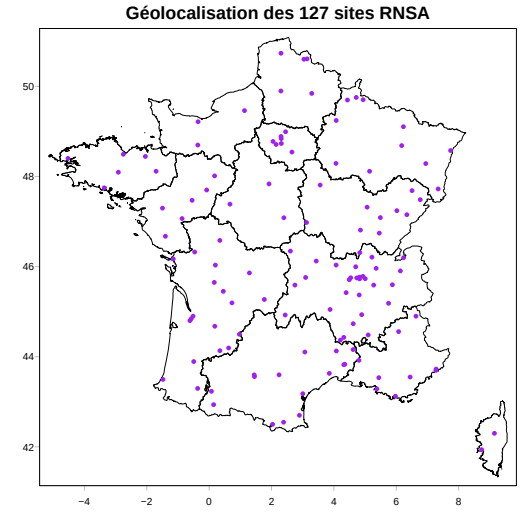
On utilise la fonction `gadm()` du paquet `geodata` [4] pour récupérer le fond de carte, en basse résolution, de la France métropolitaine avec le découpage en régions et on le représente avec le paquet `terra` [3].

```
library(geodata)
fra <- gadm(country = "FRA", level = 1, path = ".", resolution = 2)
library(terra)
plot(fra, las = 1, main = "Le fond de carte", xlab = "Longitude [°]",
     ylab = "Latitude [°]")
```



3 Géolocalisation des sites

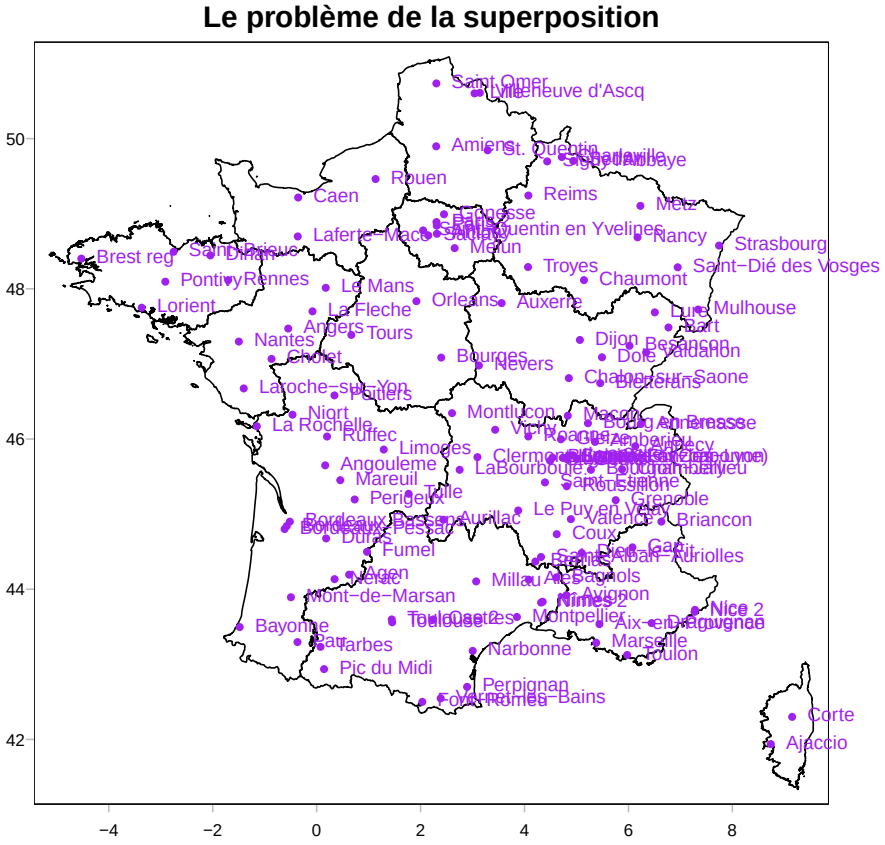
```
pts <- with(fm, vect(cbind(Longitude, Latitude), crs = "EPSG:4326"))
pts$nom <- fm$Stations
plot(fra, las = 1, main = "Géolocalisation des 127 sites RNSA")
points(pts, col = "purple")
```



4 Le problème de la superposition

Le problème auquel nous sommes confrontés semble suffisamment évident au vu la carte suivante pour ne pas avoir à le décrire en long, en large et en travers.

```
noms <- fm$Stations
plot(fra, las = 1, main = "Le problème de la superposition")
points(pts, col = "purple")
text(pts, noms, col = "purple", cex = 0.75, pos = 4, xpd = NA)
```

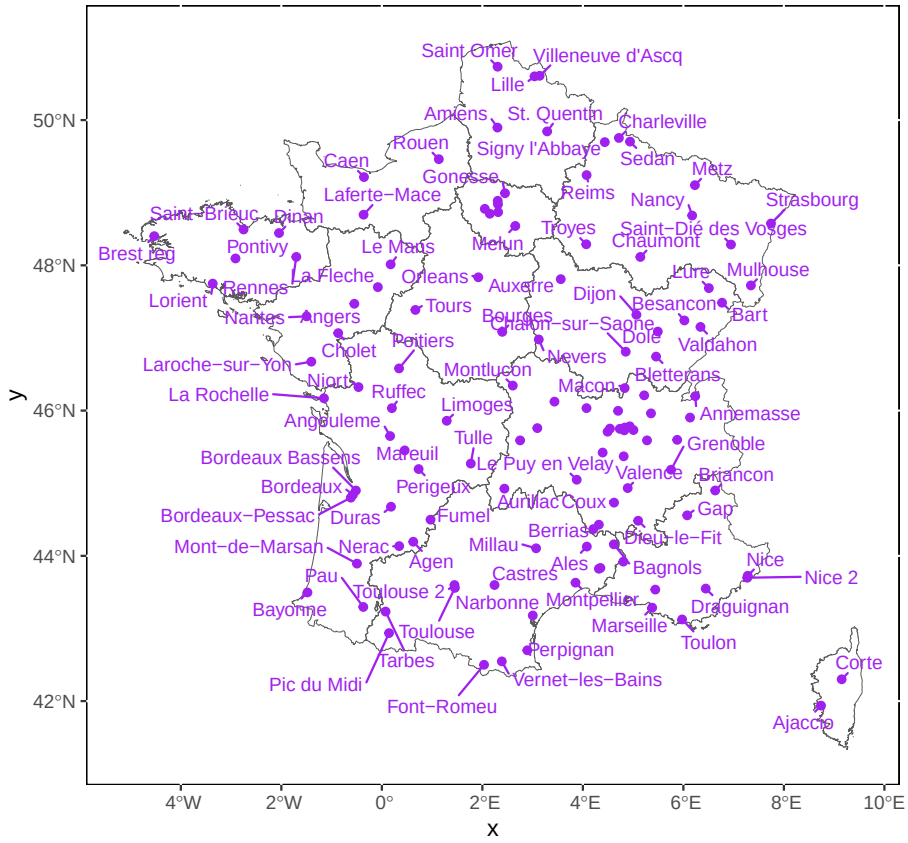


5 Le déplacement des étiquettes

Nous allons utiliser la fonction `ggrepel()` du paquet éponyme [6] pour repousser automatiquement les textes afin d'éviter tout chevauchement. Le problème est que `ggrepel()` fait parti de l'écosystème de `ggplot2` [7] qui utilise une toute autre logique que les graphiques de base. Heureusement, le paquet `tidyterra` [2] nous permet d'intégrer facilement des graphiques générés avec `terra` dans un contexte `ggplot2`.

```
library(ggplot2)
library(tidyterra)
library(ggrepel)
ptsdf <- as.data.frame(geom(pts)[, c("x", "y")])
ggplot() +
  geom_spatvector(data = fra, bg = "transparent") +
  geom_point(aes(x = x, y = y), data = ptsdf, col = "purple") +
  geom_text_repel(
    data = ptsdf,
    aes(x = x, y = y, label = noms),
```

```
  size = 3, col = "purple",
  min.segment.length = 0
) +
theme(panel.background = element_rect(fill = "white", color = "black"))
```

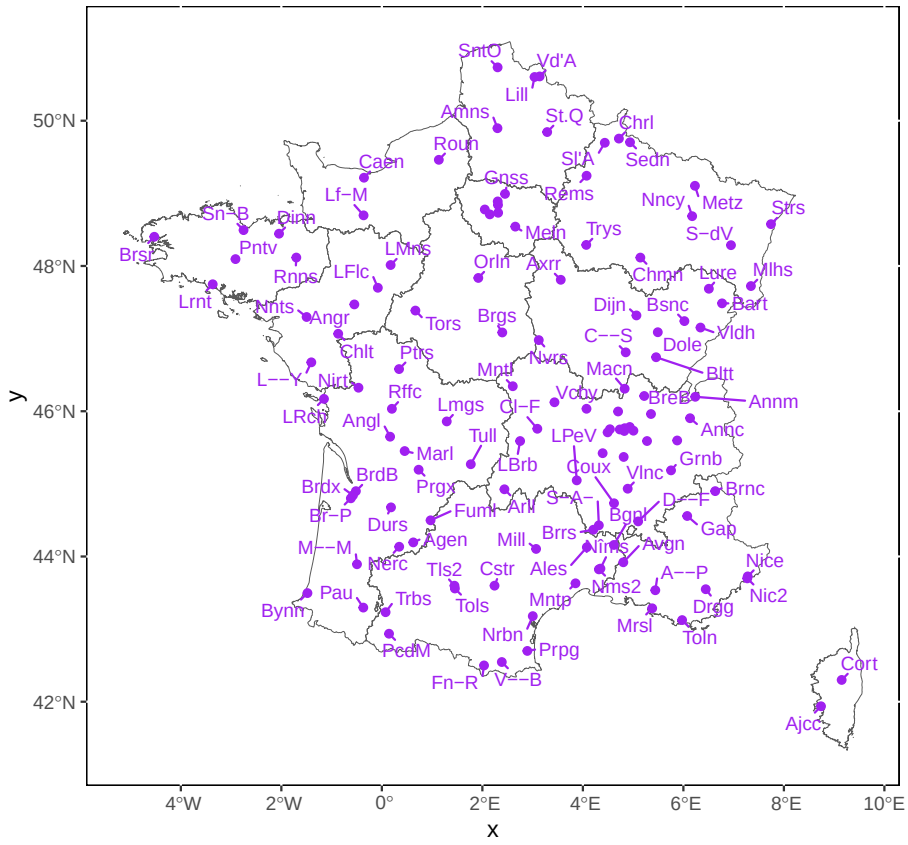


Les textes sont bien lisibles, mais tous n'ont pas été représentés, en particulier au niveau des grandes agglomérations où les sites RNSA sont concentrés pour pouvoir arriver à un résultat décent.

6 Abbréviation des étiquettes

La fonction R de base `abbreviate()` permet de condenser de façon non ambiguë des chaînes de caractères. Le résultat n'est pas très convainquant ici.

```
ggplot() +
  geom_spatvector(data = fra, bg = "transparent") +
  geom_point(aes(x = x, y = y), data = ptsdf, col = "purple") +
  geom_text_repel(
    data = ptsdf,
    aes(x = x, y = y, label = abbreviate(noms)),
    size = 3, col = "purple",
    min.segment.length = 0
  ) +
  theme(panel.background = element_rect(fill = "white", color = "black"))
```



7 Utilisation des rangs

Je me suis inspiré ici de la représentation graphique utilisée par Éliane Schermer dans son rapport de master (figure 1 page 7). L'idée est de remplacer le nom des sites par leur rang de façon à n'avoir qu'un minimum de caractères à poser sur la carte. La lecture est aisée dans le sens carte → légende, mais plus fastidieux dans le sens légende → carte.

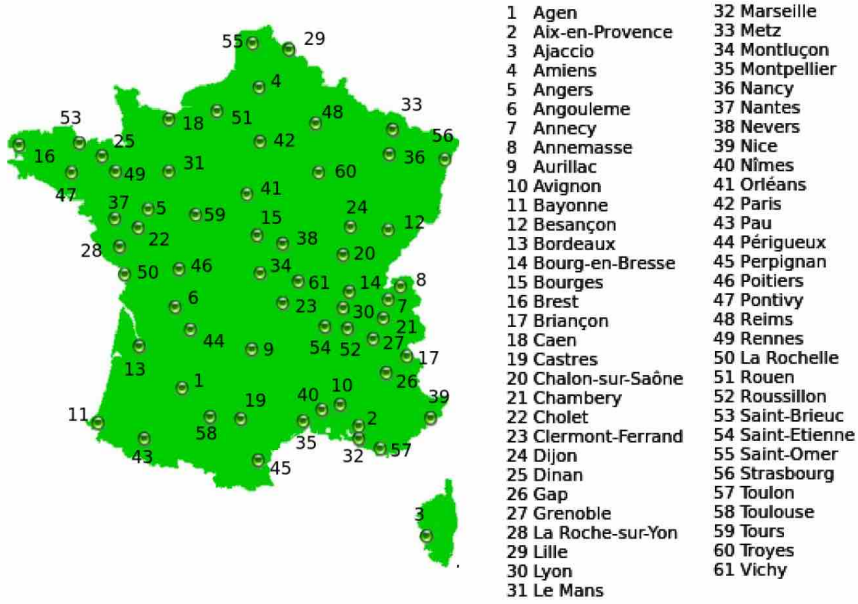
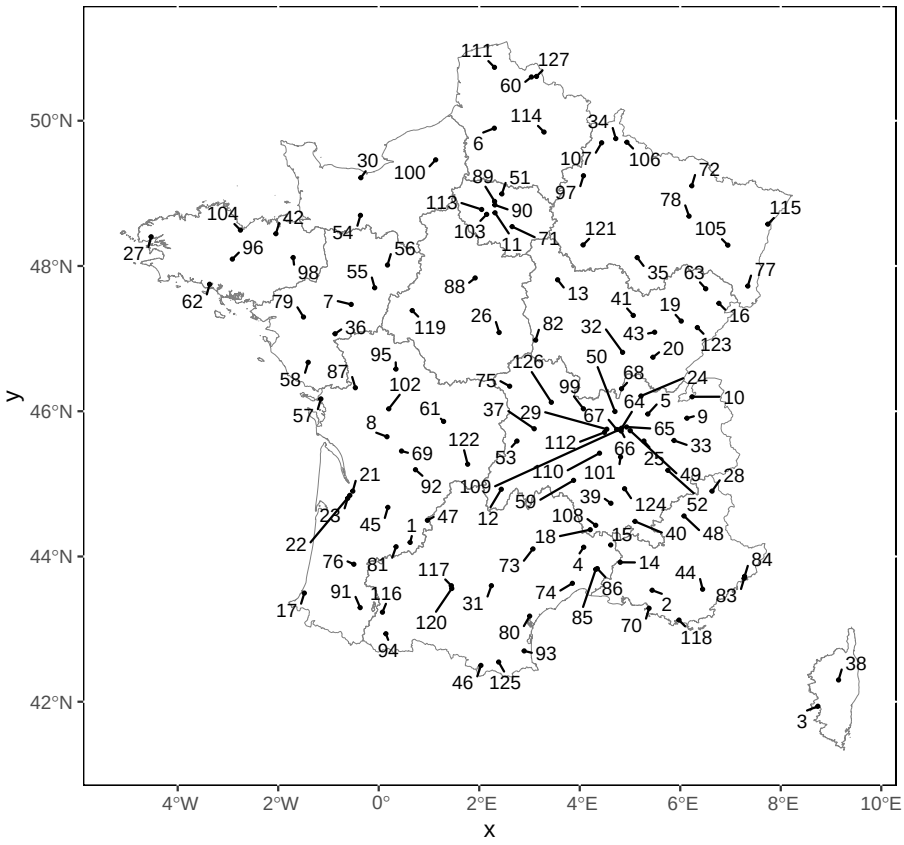


Figure 1 : Copie d'écran de [5, Fig. 2] : « Carte de France représentant la répartition des sites sélectionnés parmi les nombreux sites suivis par le RNSA (Réseau National de Surveillance Aérobiologique). »



1 Agen	33 Chambery	65 Lyon site3	97 Reims
2 Aix−en−Provence	34 Charleville	66 Lyon	98 Rennes
3 Ajaccio	35 Chaumont	67 Lyon West (Craponne)	99 Roanne
4 Ales	36 Cholet	68 Macon	100 Rouen
5 Amberieu	37 Clermont−Ferrand	69 Mareuil	101 Roussillon
6 Amiens	38 Corte	70 Marseille	102 Ruffec
7 Angers	39 Coux	71 Melun	103 Saclay
8 Angouleme	40 Dieu−le−Fit	72 Metz	104 Saint−Brieuc
9 Annecy	41 Dijon	73 Millau	105 Saint−Dié des Vosges
10 Annemasse	42 Dinan	74 Montpellier	106 Sedan
11 Antony	43 Dole	75 Montlucon	107 Signy l'Abbaye
12 Aurillac	44 Draguignan	76 Mont−de−Marsan	108 Saint−Alban−Auriolles
13 Auxerre	45 Duras	77 Mulhouse	109 Sainte−Foy−lès−Lyon
14 Avignon	46 Font−Romeu	78 Nancy	110 Saint−Etienne
15 Bagnols	47 Fumel	79 Nantes	111 Saint Omer
16 Bart	48 Gap	80 Narbonne	112 St. Genis
17 Bayonne	49 Genas	81 Nerac	113 Saint−Quentin en Yvelines
18 Berrias	50 Gleize	82 Nevers	114 St. Quentin
19 Besancon	51 Gonesse	83 Nice 2	115 Strasbourg
20 Bletterans	52 Grenoble	84 Nice	116 Tarbes
21 Bordeaux Bassens	53 LaBourboule	85 Nîmes 2	117 Toulouse 2
22 Bordeaux	54 Laferte−Mace	86 Nîmes	118 Toulon
23 Bordeaux−Pessac	55 La Fleche	87 Niort	119 Tours
24 Bourg en Bresse	56 Le Mans	88 Orleans	120 Toulouse
25 Bourgoin−Jallieu	57 La Rochelle	89 Paris 2	121 Troyes
26 Bourges	58 Laroche−sur−Yon	90 Paris	122 Tulle
27 Brest reg	59 Le Puy en Velay	91 Pau	123 Valdahon
28 Briancon	60 Lille	92 Perigeux	124 Valence
29 Brussieu Lanzoni	61 Limoges	93 Perpignan	125 Vernet−les−Bains
30 Caen	62 Lorient	94 Pic du Midi	126 Vichy
31 Castres	63 Lure	95 Poitiers	127 Villeneuve d'Ascq
32 Chalon−sur−Saone	64 Lyon2	96 Pontivy	

Le code pour générer la carte est le suivant :

```
ggplot() +
  geom_spatvector(data = fra, bg = "transparent", col = grey(0.5)) +
  geom_point(aes(x = x, y = y), data = ptsdf, col = "black", cex = 0.5) +
  geom_text_repel(
    data = ptsdf,
    aes(x = x, y = y, label = 1:length(noms)),
    size = 3, col = "black",
    min.segment.length = 0,
    max.overlaps = 1000
  ) +
  theme(panel.background = element_rect(fill = "white", color = "black"))
```

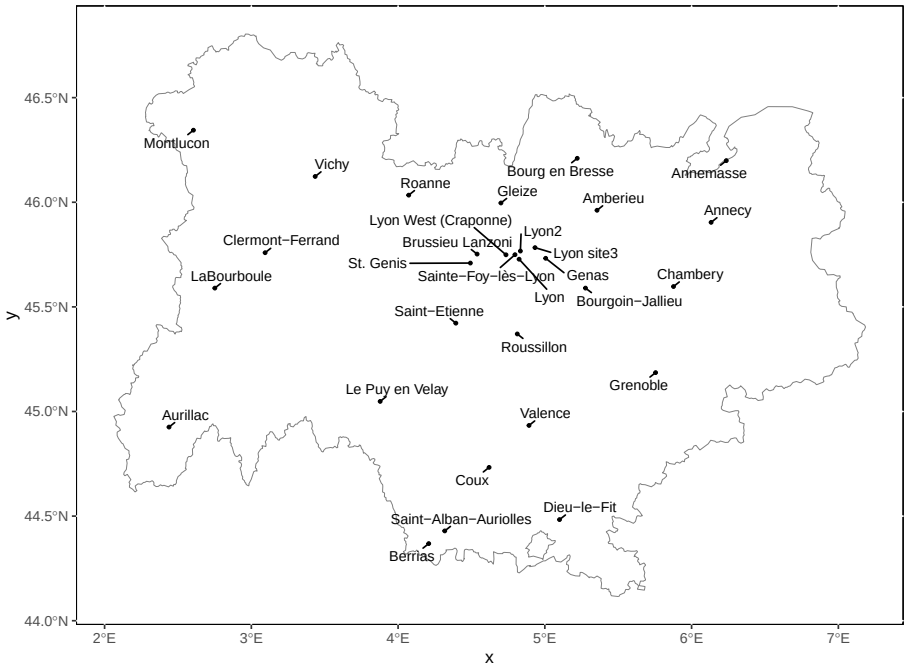
Le code pour générer la légende est le suivant :

```
par(mar = c(0,0,0,0) + 0.1)
plot.new() ; plot.window(0:1, 0:1)
legend("topleft", inset = 0,
       legend = paste(1:length(noms), noms), ncol = 4, cex = 0.8)
```

8 Zoomer

Une autre solution consiste à zoomer sur les zones de forte densité en points. On perd la vue d'ensemble, mais on gagne en lisibilité. Par exemple, pour la région Auvergne-Rhône-Alpes, on a la carte suivante :

```
ara <- fra[fra$NAME_1 == "Auvergne-Rhône-Alpes", ]
ptsara <- intersect(pts, ara)
ptsdfara <- as.data.frame(geom(ptsara)[, c("x", "y")])
nomsara <- ptsara$nom
ggplot() +
  geom_spatvector(data = ara, bg = "transparent", col = grey(0.5)) +
  geom_point(aes(x = x, y = y), data = ptsdfara, col = "black", cex = 0.75) +
  geom_text_repel(
    data = ptsdfara,
    aes(x = x, y = y, label = nomsara),
    size = 3, col = "black",
    min.segment.length = 0,
    max.overlaps = 1000
  ) +
  theme(panel.background = element_rect(fill = "white", color = "black"))
```



On peut également faire une carte dynamique pour zoomer à la demande avec le paquet `leaflet` [1] avec le code suivant. La figure 2 page 11 donne une idée du résultat obtenu.

```
mamap <- plet(pts, "nom", cex = 3, col = "red", legend = NULL)
library(leaflet)
mamap <- addScaleBar(mamap, "bottomleft", scaleBarOptions(imperial = FALSE))
mamap
```

J'ai vérifié à l'aide de cette interactive qu'il n'y avait d'erreur de localisation grossière des sites RNSA en France métropolitaine : on tombe toujours sur la commune attendue. La localisation est parfois assez précise pour qu l'on puisse voir sur le toit de quel bâtiment on se situe, mais ce n'est pas toujours le cas[2].

9 Carte animée

On veut faire une carte animée, année par année, pour avoir une idée de la disponibilité des données. La nomenclature des sites n'est pas la même dans le fichier des données GPS et dans ceux des données bi-horaires (voir la fiche[3] « [l]es données historiques du RNSA »). La correspondance entre les deux nomenclatures est donné dans la table `corresp`.

```
library(readxl)
corresp <- read_xlsx("geolocRNSA/corespondance-GPS-bihoraire.xlsx")
corresp <- as.data.frame(corresp)
```

[2]Les cas où la géolocalisation me semble très imprécise sont ceux de Dieulefit (on tombe en pleine forêt à 6 km au Sud-Est), La Ferté-Macé (dans la commune de Briouze, rue des Acacias, à 10 km au Nord) et Saint-Omer (commune d'Arques, résidence les mouettes, à 5 km au Sud-Est).

[3]`https://esb.univ-lyon1.fr/pdf/histoRNSA.pdf`

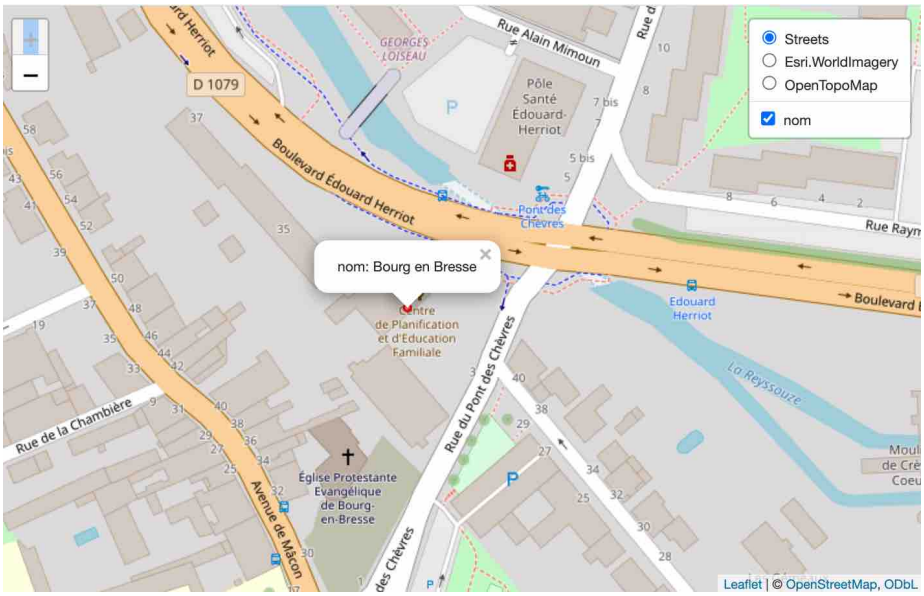


Figure 2 : Exemple de carte dynamique obtenue avec le paquet `leaflet` [1]. On a zoomé ici au maximum sur le site de Bourg-en-Bresse.

```
corresp <- corresp[ , c("Stations", "NomBiHo")]
comment(corresp) <- paste("Généré le ", Sys.time())
save(corresp, file = "geolocRNSA/corresp.Rda")
```

```
chmin <- "https://esb.univ-lyon1.fr/donnees/geolocRNSA/"
load(url(paste0(chmin, "corresp.Rda")))
corresp[is.na(corresp$NomBiHo), ]
```

```
         Stations NomBiHo
55       La Fleche    <NA>
95     Pic du Midi    <NA>
109 Signy l'Abbaye    <NA>
```

Il y a trois sites pour lesquels on a la géolocalisation mais pas de données sur la concentration pollinique : La Flèche, Pic du Midi et Signy-l'Abbaye.

```
chmin <- "https://esb.univ-lyon1.fr/donnees/histoRNSA/"
load(url(paste0(chmin, "pol2h.Rda")))
allBiHo <- names(pol2h)
allBiHo[-(which(allBiHo %in% corresp$NomBiHo))]
```

```
[1] "ANDORRA"          "BRUSSIEU BURKARD" "CAMBO-LES-BAINS"  "TAVAUX"
[5] "THIZY"            "TORCY"
```

Dans l'autre sens, il y a 6 sites RNSA pour lesquels on a des données sur les concentrations polliniques mais pas la géolocalisation des capteurs :

`ANDORRA` : 13 années disponibles de 2012 à 2024, mais ce n'est pas en France métropolitaine.

`BRUSSIEU BURKARD` : seule une année de disponible (2011). Il y a également un `BRUSSIEU` avec 14 années disponibles de 2011 à 2024, c'est lui que l'on a associé dans la table `corresp` avec `Brussieu Lanzoni`.

CAMBO-LES-BAINS : 12 années disponibles de 1989 à 2000.

TAVAUX : une seule année disponible (2022).

THIZY : 3 années disponibles de 2010 à 2012.

TORCY : une seule année disponible (2022).

Il y aurait donc les séries de Cambo-les-Bains et d'Andorre-la-Vieille qui vaudraient le coup de géolocaliser[4]. On utilise la boucle suivante pour générer une carte par année, en ne représentant que les sites pour lesquels il y a des données polliniques.

```
library(lubridate)
years <- vector("list", length(noms))
names(years) <- noms
for(i in seq_len(length(years))){
  ii <- which(noms[i] == corresp$Stations)
  the_NomBiHo <- corresp[ii, "NomBiHo"]
  if(is.na(the_NomBiHo)){
    years[[i]] <- 0
  } else {
    years[[i]] <- unique(year(pol2h[[the_NomBiHo]]$Date))
  }
}
tmp <- unlist(years) ; tmp <- tmp[tmp>0]
yearseq <- min(tmp):max(tmp)
for(i in seq_len(length(yearseq))){
  print(i)
  ichar <- sprintf("%06d", i)
  the_year <- yearseq[i]
  mycol <- ifelse(sapply(years, \(x) the_year %in% x), "black", "transparent")
  ggplot() +
  ggtitle(paste("Sites RNSA en", the_year)) +
  geom_spatvector(data = fra, bg = "transparent", col = grey(0.5)) +
  geom_point(aes(x = x, y = y), data = ptsdf, col = mycol, cex = 0.5) +
  geom_text_repel(
    data = ptsdf,
    aes(x = x, y = y, label = 1:length(noms)),
    size = 3, col = mycol,
    min.segment.length = 0,
    max.overlaps = 1000,
    max.time = 10,
    seed = 1
  ) +
  theme(panel.background = element_rect(fill = "white", color = "black"))

  ggsave(paste0("png/image", ichar, ".png"))
}
```

Le code R utilisé pour faire la petite animation, à faible résolution, sur le site[5] montrant les sites RNSA au cours du temps est le suivant :

```
library(gifski)
png_files <- list.files(path = "png/", pattern = ".png", full.names = TRUE)
gifski::gifski(png_files, gif_file = "geolocRNSAsmall.gif",
  delay = 1, width = 800/4, height = 600/4)
```

[4] D'après ce que j'ai trouvé sur internet ça ne devrait pas être trop difficile. Pour Cambo-les-Bains : « ce capteur était historiquement hébergé et géré en collaboration avec le Centre de Pneumologie *Les Terrasses*, un établissement de soins de suite et de réadaptation respiratoire très réputé de la commune, situé au Square Albeniz (ou 18 rue du Docteur Constant Colbert). » Pour Andorre-la-Vieille : « le capteur se trouve à Andorre-la-Vieille (*Andorra la Vella*). Plus précisément, il est installé et géré au niveau de la *Unitat de Medi Atmosfèric* (l'Unité du Milieu Atmosphérique) du gouvernement d'Andorre, rattachée au ministère chargé de l'Environnement. Les bureaux et les installations techniques se situent au *c/ Prat de la Creu*, 62-64. »

[5] `http://esb.univ-lyon1.fr/enseignement_div.php?contents=html/quercus`

Et celui pour faire l'animation à haute résolution (`https://esb.univ-lyon1.fr/donnees/geolocRNSA/geolocRNSA.gif`) est le suivant :

```
library(gifski)
png_files <- list.files(path = "png/", pattern = ".png", full.names = TRUE)
gifski::gifski(png_files, gif_file = "geolocRNSA.gif",
  delay = 1)
```

Références

[1] J. Cheng, B. Schloerke, B. Karambelkar, and Y. Xie. *leaflet : Create Interactive Web Maps with the JavaScript 'Leaflet' Library*, 2024. R package version 2.2.2.

[2] D. Hernangómez. Using the tidyverse with terra objects : the tidyterra package. *Journal of Open Source Software*, 8(91) :5751, 2023.

[3] R.J. Hijmans. *terra : Spatial Data Analysis*, 2024. R package version 1.7-71.

[4] R.J. Hijmans, M. Barbosa, A. Ghosh, and A. Mandel. *geodata : Download Geographic Data*, 2023. R package version 0.5-9.

[5] E. Schermer. Vers une théorie unifiée du masting : analyse de l'impact des conditions météorologiques sur la pollinisation combinée à la révision des modèles mécanistes du masting. Master's thesis, Université de Rennes 1, 2015.

[6] K. Slowikowski. *ggrepel : Automatically Position Non-Overlapping Text Labels with 'ggplot2'*, 2026. R package version 0.9.8.

[7] H. Wickham. *`ggplot2` : Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.