


Fiche TD avec le logiciel  : BDFORET

---

Initiation à l'importation, la représentation et l'analyse sous  des données géographiques vectorielles (*shapefile*) du département de l'AIN sur les types de formations végétales (*e.g.* forêt fermée de chênes décidus purs) définis par les *Eaux et Forêts* en FRANCE métropolitaine par le truchement de la BD Forêt<sup>©</sup> version 2

P<sup>r</sup> Jean R. LOBRY

---

Avec un titre aussi long, vous ne pensiez pas sérieusement que j'allais me fendre d'un résumé, non ? Pour faire court, on va jouer aux orpailleurs de données (*data analyst*) en herbe avec des informations sur les types de formations végétales.

## Table des matières

<b>1</b>	<b>Source et importation des données</b>	<b>3</b>
1.1	La BD Forêt <sup>®</sup> version 2 . . . . .	3
1.2	Importation des données . . . . .	3
<b>2</b>	<b>Représentation graphique</b>	<b>5</b>
2.1	Les codes couleurs . . . . .	5
2.2	Limites administratives . . . . .	5
2.3	Carte dynamique . . . . .	6
2.4	Carte statique . . . . .	6
<b>3</b>	<b>Analyse des données</b>	<b>8</b>
3.1	Statistiques spatiales . . . . .	8
3.2	Calcul des surfaces . . . . .	9
3.3	Caractérisation des surfaces . . . . .	10
3.4	Exploitation de la géo-localisation . . . . .	12
<b>4</b>	<b>Annexes</b>	<b>22</b>
4.1	Raboutage de départements . . . . .	22
	<b>Références</b>	<b>31</b>

# 1 Source et importation des données

## 1.1 La BD Forêt<sup>©</sup> version 2

POUR se faire rapidement une idée de la nature des données, le plus simple et le plus édifiant est sans doute d'aller jouer avec la carte interactive proposée par l'observatoire des forêts françaises<sup>1</sup> dont on trouvera une illustration dans la figure 6 page 28. Il s'agit d'un découpage fin du territoire caractérisant les types de formations végétales. Le type de formation végétale est une variable qualitative à 32 modalités dont on trouvera une description détaillée dans l'annexe A du document « Descriptif de contenu » [1]. Par exemple, la modalité FF1G01-01 correspond à une forêt fermée de chênes décidus purs (voir la table 1 page 27).

## 1.2 Importation des données

LES données sont disponibles sous licence libre en téléchargement anonyme sur le site de l'IGN<sup>2</sup> département par département. Une copie des données utilisées ici est disponible dans le dossier BDFORET sur le site du pôle de bio-informatique lyonnais<sup>3</sup>. On utilise ici les données du département de l'AIN dont le dossier pèse de l'ordre de 100 Mio après décompression. Les données sont dans un format vectoriel classique pour des données cartographiques, on peut les importer dans  $\mathbb{R}$  avec la fonction `vect()` du paquet `terra` [7] :

```
chmin <- paste0("BDFORET/",
               "BDFORET_2-0_SHP_LAMB93_D001_2014-04-01/",
               "BDFORET/",
               "1_DONNEES_LIVRAISON/",
               "BDF_2-0_SHP_LAMB93_D001/")
D001 <- terra::vect(paste0(chmin, "FORMATION_VEGETALE.shp"))
D001
class      : SpatVector
geometry   : polygons
dimensions : 13947, 5 (geometries, attributes)
extent     : 833966.4, 943565.9, 6504699, 6603617 (xmin, xmax, ymin, ymax)
source     : FORMATION_VEGETALE.shp
coord. ref.: RGF93 v1 / Lambert-93 (EPSG:2154)
names      : ID CODE_TFV TFV
type       : <chr> <chr> <chr>
values     : FORESTIE00000000001~ FF2-52-52 Forêt fermée de pin~
             FORESTIE00000000001~ FF1-00-00 Forêt fermée à méla~
             FORESTIE00000000001~ FF1-00-00 Forêt fermée à méla~
             TFV_G11 ESSENCE
             <chr> <chr>
Forêt fermée conifères Pin sylvestre
Forêt fermée feuillus Feuillus
Forêt fermée feuillus Feuillus
```

L'OBJET D001 est de la classe `SpatVector`, il définit 13947 polygones en LAMBERT 93 (EPSG:2154). Pour chaque polygone on dispose de sa clef d'identification (ID), de la modalité du type de formation végétale (CODE\_TFV) et de son intitulé (TFV). Les variables TFV\_G11 et ESSENCE sont des recodages agrégatifs de CODE\_TFV en 11 et 17 modalités, respectivement. Sur les 32 modalités possibles (table 1 page 27) seules 28 sont documentées dans le département de l'AIN. Les 4 modalités manquantes correspondent effectivement à des formations végétales que l'on ne s'attend pas à y observer :

```
length(codesobs <- sort(unique(D001$CODE_TFV)))
```

<sup>1</sup><https://foret.ign.fr/carte>

<sup>2</sup><https://geoservices.ign.fr/bdforet>

<sup>3</sup><https://pbil.univ-lyon1.fr/R/donnees/>



FIGURE 1 : Zoom progressif sur le virage du chemin des coupes blanches dans la forêt domaniale de SEILLON, au sud de BOURG-EN-BRESSE, dans le département de l'AIN. Il s'agit de copies d'écran d'une carte interactive générée avec la fonction `terra::plet()` à partir de données de type vectoriel. Notez la finesse du détail des polygones décrivant les zones de type de formation végétale.

[1] 28

```
codesTFV <- readODS::read_ods("BDFORET/codesTFV.ods", as_tibble = FALSE)
subset(codesTFV, !CODE_TFV %in% codesobs)[, 1:2]
  CODE_TFV      TFV
4 FF1G06-06    Forêt fermée de chênes sempervirents purs
11 FF2-51-51    Forêt fermée de pin maritime pur
14 FF2-57-57    Forêt fermée de pin d'Alep pur
15 FF2G58-58 Forêt fermée de pin à crochets ou pin cembro pur
```

## 2 Représentation graphique

### 2.1 Les codes couleurs

POUR les représentations graphiques on a besoin d'associer une couleur à chaque modalité du type de formation végétale. Je n'ai pas trouvé en ligne de fichier définissant les codes RGB utilisés par la carte interactive proposée par l'observatoire des forêts françaises. Je suis donc parti de copies d'écran que j'ai importées dans le logiciel libre GIMP<sup>4</sup> pour utiliser la « pipette à couleurs » afin de déterminer les codes RGB des pixels d'intérêt (voir la figure 7 page 29). La fonction de base `rgb()` permet alors dans **R** de calculer les couleurs correspondantes. Il faut juste faire attention à bien les présenter dans l'ordre des modalités et de censurer celles qui ne sont pas observées.

```
# Import du fichier
codesTFV <- readODS::read_ods("BDFORET/codesTFV.ods", as_tibble = FALSE)
# Calcul des couleurs
codesTFV$col <- with(codesTFV, rgb(R2, G2, B2, maxColorValue = 255))
# Tri dans l'ordre des modalités
codesTFV <- codesTFV[order(codesTFV$CODE_TFV), ]
```

### 2.2 Limites administratives

J'AI récupéré sur le site de l'IGN<sup>5</sup> les données administratives de BD TOPO<sup>®</sup> pour le département de l'AIN, c'est un polygone de 13799 points dont les coordonnées sont en LAMBERT 93.

```
chmin <- paste0("BDFORET/",
               "BDTOPO_3-3_TOUSTHEMES_SHP_LAMB93_D001_2024-03-15/",
               "BDTOPO/",
               "1_DONNEES_LIVRAISON_2024-04-00042/",
               "BDT_3-3_SHP_LAMB93_D001-ED2024-03-15/",
               "ADMINISTRATIF/")
dep <- terra::vect(paste0(chmin, "DEPARTEMENT.shp"))
unique(dep$NOM) # Il y a aussi les départements limitrophes
[1] "Jura"          "Ain"            "Saône-et-Loire" "Rhône"
[5] "Isère"         "Savoie"         "Haute-Savoie"
dep01 <- dep[dep$NOM == "Ain"]
dep01
class      : SpatVector
geometry   : polygons
dimensions : 1, 10 (geometries, attributes)
extent     : 833859.6, 943639.5, 6504695, 6604403 (xmin, xmax, ymin, ymax)
coord.ref. : RGF93 v1 / Lambert-93 (EPSG:2154)
names      : ID INSEE_DEP INSEE_REG NOM DATE_CREAT
type       : <chr> <chr> <chr> <chr> <chr>
values     : DEPARTEME000000~ 01 84 Ain 2018-11-24 00:-
             DATE_MAJ DATE_APP DATE_CONF ID_AUT_ADM CODE_SIREN
             <chr> <chr> <chr> <chr> <chr>
2024-02-09 11:- NA NA SURFACTI000000~ 220100010
nrow(terra::geom(dep01))
[1] 13799
```

<sup>4</sup><https://www.gimp.org/>

<sup>5</sup><https://geoservices.ign.fr/bdtopo#telechargementshpdept>

## 2.3 Carte dynamique

Le code ci-après a été utilisé pour produire la carte interactive dont on trouvera des copies d'écran dans la figure 1 page 4. La fonction `plet()` du paquet `terra` [7] permet de générer un objet pris en charge par le paquet `leaflet` [3]. Sur mon ordinateur il faut attendre environ une minute<sup>6</sup> pour obtenir la carte, mais ensuite la navigation (zooms et déplacements latéraux) est tout à fait fluide.

```
mycol <- subset(codesTFV, CODE_TFV %in% codesobs)$col # Que les modalités observées
mamap <- terra::plet(D001, "CODE_TFV", col = mycol, border = "transparent",
  fill = 0.8, legend = NULL)
library(leaflet)
mamap <- addScaleBar(mamap, "bottomleft", scaleBarOptions(imperial = FALSE))
mamap
```

## 2.4 Carte statique

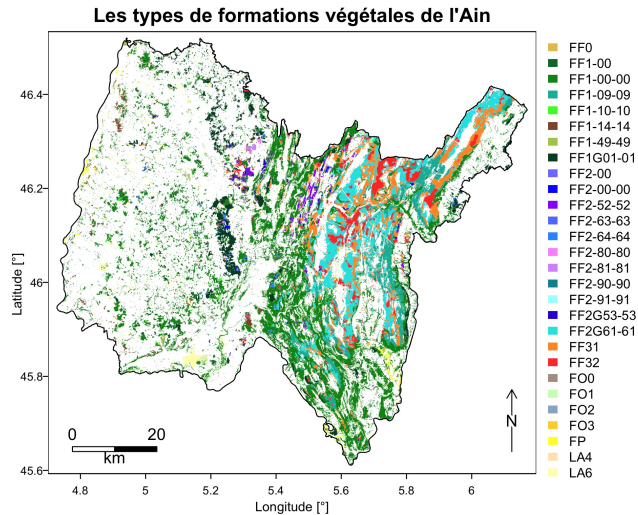
LES cartes dynamiques sont un outil on ne peut plus puissant en analyse exploratoire, mais tôt ou tard, que ce soit pour des besoins de publication ou d'archivage, on aura besoin d'une carte statique. À titre personnel, mes *desiderata* seront ici les suivants : je veux des coordonnées GPS<sup>7</sup>, une légende, les limites administratives, une échelle pour donner une idée de la taille de ce qui est représenté et d'une flèche pointant vers le pôle NORD géographique<sup>8</sup>.

```
GPS <- terra::project(D001, "EPSG:4326")
dep01GPS <- terra::project(dep01, "EPSG:4326")
macarte <- function(SV, codesTFV, ...){
  codesobs <- unique(SV$CODE_TFV)
  codesTFV <- subset(codesTFV, CODE_TFV %in% codesobs) # Que les modalités observées
  terra::plot(SV, "CODE_TFV", col = codesTFV$col, border = "transparent",
    las = 1, xlab = "Longitude [°]",
    ylab = "Latitude [°]", ...)
  terra::lines(dep01GPS)
  terra::sbar(type = "bar", xy = "bottomleft", below = "km")
  terra::north(xy = "bottomright")
}
macarte(GPS, codesTFV, main = "Les types de formations végétales de l'Ain")
```

<sup>6</sup>C'est tout à fait raisonnable quand on pense que l'objet D001 comporte 13947 polygones définissant 5471487 points. On peut accélérer les choses en diminuant la résolution, par exemple avec `terra::simplifyGeom(D001, 10)` pour avoir une distance minimale de 10 m entre les points, il ne reste alors plus que 564464 points en tout, soit 10 fois moins. On peut également « rasteriser » les données, ce que le paquet `terra` [7] fait *très* bien en tant que digne descendant optimisé du paquet `raster` [6].

<sup>7</sup>Pour être précis et non ambigu je veux de l'EPSG:4326 parce que ce sont ces coordonnées qui sont en passe d'être le mieux comprises par le commun des mortels.

<sup>8</sup>Je n'ai pas cherché comme dans la figure 8 page 30 à faire un cartouche pour situer la carte dans un contexte plus général, mais pour ceux ou celles qui seraient intéressé(e)s la fonction `terra::inset()` est votre amie.

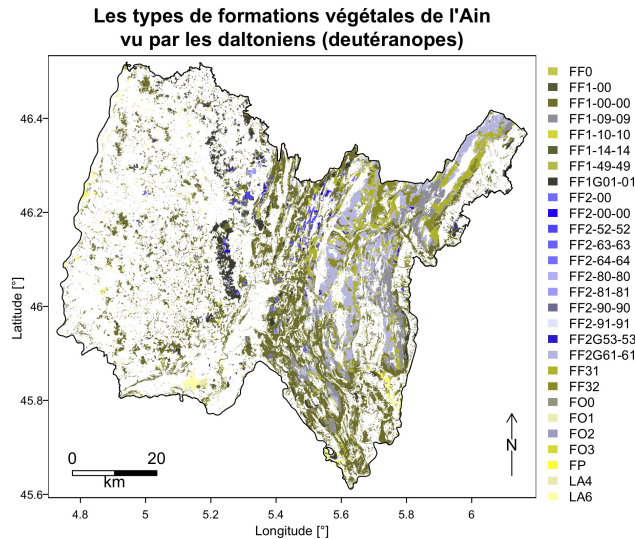


Le rendu semble être cohérent avec la carte du document de synthèse livré avec les données (figure 8 page 30). Le paquet `dichromat` [10], basé sur les travaux de Françoise VIÉNOT et collaborateurs [14] permet de simuler facilement ce que voient les daltoniens<sup>9</sup>.

```
library(dichromat)
codesTFVdalt <- codesTFV
codesTFVdalt$col <- dichromat(codesTFVdalt$col, type = "deutan")
main <- paste0("Les types de formations végétales de l'Ain\n",
              "vu par les daltoniens (deutéranopes)")
macarte(GPS, codesTFVdalt, main = main)
```

<sup>9</sup>Au sens originel ici : John DALTON était atteint de deutéranopie.





La perte d'information n'est pas si dramatique que ça puisque dans l'ensemble on discrimine assez bien les feuillus FF1\* des conifères FF2\*. Il y a quand même la modalité FF32 (mélange de conifères *prépondérants* et feuillus) qui est un peu trompeuse avec un rendu plus voisin des feuillus que des conifères<sup>10</sup>.

### 3 Analyse des données

#### 3.1 Statistiques spatiales

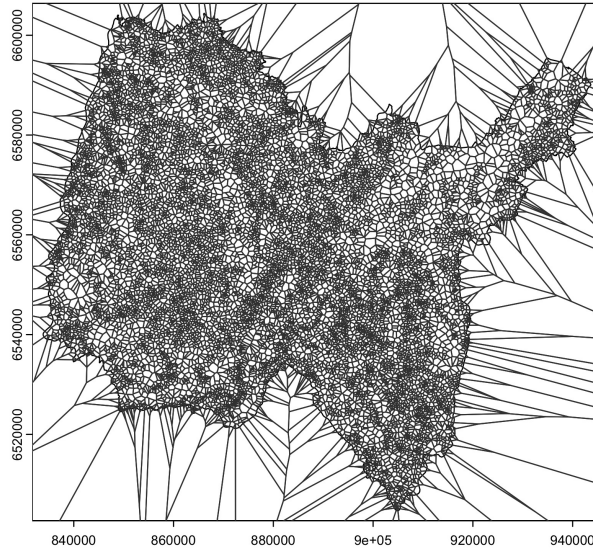
Les statistiques spatiales, c'est à dire quand on s'intéresse à des variables géo-localisées, forment un pan entier des statistiques. Pour une initiation je recommande d'aller explorer le site<sup>11</sup> *Spatial Data Science with R and "terra"*. Une présentation générale déborderait du cadre de cette fiche, mais je ne résiste pas au plaisir de représenter le diagramme de VORONOÏ [16, 15] généré par les centres des polygones.

```
centr <- terra::centroids(D001)
v <- terra::voronoi(centr)
terra::plot(dep01)
terra::lines(v, col = grey(0.25))
```

<sup>10</sup>L'utilisation des couleurs n'est pas un problème simple, pour une mise en garde on pourra consulter la fiche « (Ne pas) utiliser des couleurs » à <https://pbil.univ-lyon1.fr/R/pdf/tdr18.pdf>

<sup>11</sup><https://rspatial.org/>





### 3.2 Calcul des surfaces

La fonction `terra::expanse()` permet de calculer la surface des polygones d'un objet de type `SpatVector`. Par exemple, dans le cas de l'objet `dep01` qui contient un seul polygone définissant les contours du département de l'AIN on obtient :

```
terra::expanse(dep01, unit = "ha")
[1] 578483.8
```

CETTE valeur est proche des 578523 ha donnés dans la fiche de synthèse<sup>12</sup> accompagnant la distribution des données, mais pas parfaitement égale. Je ne sais pas pourquoi j'ai un déficit de 39.2 ha (soit une différence relative de -0.007 %) ici. Avec la fonction `tapply()` il est facile de calculer des statistiques agrégées en fonction des modalités d'une variable qualitative. Par exemple, pour calculer les surfaces occupées par les différents types de formations végétales :

```
ShaTFV <- tapply(D001, D001$CODE_TFV, \(x) sum(terra::expanse(x, unit = "ha")))
ShaTFV
      FFO      FF1-00      FF1-00-00      FF1-09-09      FF1-10-10      FF1-14-14
6.112773e+02 4.873242e+03 9.006168e+04 1.895435e+04 3.352416e+02 2.799644e+03
      FF1-49-49      FF1G01-01      FF2-00      FF2-00-00      FF2-52-52      FF2-63-63
7.510461e+02 1.687166e+04 1.672275e+02 9.850877e+02 2.489540e+03 9.485672e+00
      FF2-64-64      FF2-80-80      FF2-81-81      FF2-90-90      FF2-91-91      FF2G53-53
1.043119e+03 1.351442e+02 3.463079e+02 5.849434e+01 1.121699e+01 6.903074e+02
      FF2G61-61      FF31      FF32      F00      F01      F02
2.703223e+04 2.028125e+04 1.249947e+04 6.763241e-01 4.767659e+03 2.364205e+02
      F03      FP      LA4      LA6
5.107267e+02 3.113097e+03 1.929183e+03 1.923546e+03
sprintf("%.0f", ShaTFV)
```

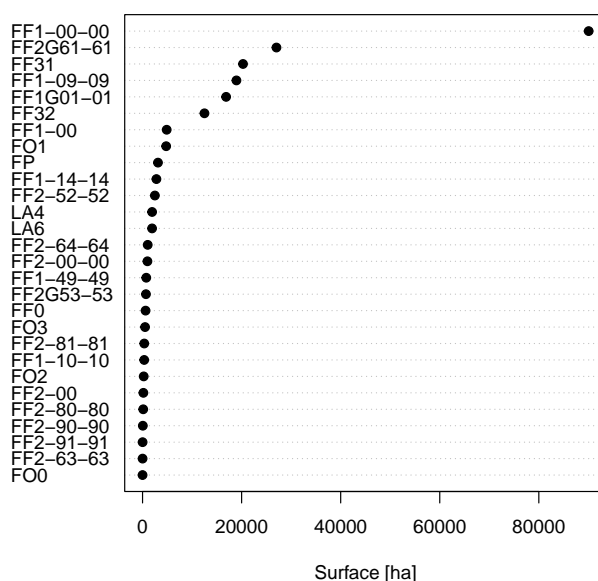
<sup>12</sup>BDF\_2\_Synthese\_01.pdf

```
[1] "611" "4873" "90062" "18954" "335" "2800" "751" "16872" "167" "985"
[11] "2490" "9" "1043" "135" "346" "58" "11" "690" "27032" "20281"
[21] "12499" "1" "4768" "236" "511" "3113" "1929" "1924"
```

ICI encore je trouve des résultats similaires, mais pas parfaitement identiques, à ceux donnés dans la fiche de synthèse. La représentation avec un diagramme de CLEVELAND [4] de cette information est la suivante :

```
dotchart(sort(c(ShaTFV)), pch = 19, xlab = "Surface [ha]",
main = "Surface des types de formations végétales dans l'Ain")
```

Surface des types de formations végétales dans l'Ain



LES 6 types de formations végétales dominants sont tous du type « forêt fermée », c'est à dire avec plus de 40 % de taux de couvert arboré. Dominent de très loin les mélanges de feuillus (hors chênes) suivis par les sapins ou épicéa, les mélange de feuillus prépondérants et conifères, les hêtres purs, les chênes décidus purs, les mélanges de conifères prépondérants et feuillus. Ces 6 types représentent à eux seuls 87 % des surfaces documentées.

### 3.3 Caractérisation des surfaces

L'AVANTAGE d'être dans R est que l'on bénéficie d'un environnement de programmation complet pour faire des statistiques, on n'est pas prisonnier des méthodes prédéfinie dans un environnement de type « clickodrome ». Pour illustrer ceci, je vais faire des statistiques un peu exotiques. Je vais m'intéresser à décrire la variabilité des surface des polygones en fonction du type de formation végétale. Je veux utiliser le très recommandable [9] coefficient de variation de KVALSETH [8]. Ce coefficient, <sup>K</sup>CV, n'existe pas en standard<sup>13</sup>, donc on le définit :

<sup>13</sup>Pas plus que le classique coefficient de variation, <sup>P</sup>CV, de PEARSON [11] d'ailleurs.

```
# Écart-type de l'échantillon (en n)
sdn <- function(x, i, ...){
  n <- sum(!is.na(x)) # number of non missing values
  return(sqrt((n - 1)/n)*sd(x[i], ...))
}
# CV de Pearson
PCV <- function(x, i, ...){
  barx <- mean(x[i], ...)
  if(isTRUE(all.equal(barx, 0))) return(0)
  return(sdn(x[i], ...)/barx)
}
# CV de Kvaalseth
KCV <- function(x, i, ...){
  PCV2 <- PCV(x, i, ...)^2
  return(sqrt(PCV2/(1 + PCV2)))
}
```

NOTEZ que toutes ces fonctions sont définies pour pouvoir les utiliser directement pour calculer la valeur de la statistique ou bien pour les passer en argument à la fonction `boot()` pour calculer les intervalles de confiance. On peut s'étonner de ce que l'appel à une de ces fonctions marche même quand l'argument `i` est manquant alors qu'il n'a pas de valeur par défaut. C'est parce que dans **R** si `i` est manquant alors `x[ ]` est quand même défini et vaut `x`. On définit maintenant la fonction `getKCV()` pour calculer la valeur de <sup>K</sup>CV et son intervalle de confiance. On utilise des intervalles de confiance de type `bca`, type recommandé par le « pape » du *bootstrap* [5].

```
library(boot)
getKCV <- function(x){
  boot.out <- boot(x, KCV, R = 9999, na.rm = TRUE)
  est <- boot.out$t0
  ci <- boot.ci(boot.out, type = "bca")$bca[4:5]
  inf <- ci[1] ; sup <- ci[2]
  return(c(est, inf, sup))
}
```

ON peut maintenant faire rouler notre fonction `getKCV()` sur le jeu de données en stratifiant par type de formation végétale. Au préalable, on censure<sup>14</sup> les polygones de moins de 0.5 ha qui ne correspondent pas à la définition internationale d'une forêt [2]. On caviarde également les types de formations végétales représentés par moins de 30 polygones pour n'avoir que des « grands » échantillons<sup>15</sup>.

```
D001$surf_ha <- terra::expanses(D001, unit = "ha")
D001foret <- D001[D001$surf_ha > 0.5, ]
npoly <- tapply(D001foret$surf_ha, D001foret$CODE_TFV, length)
npoly <- npoly[npoly > 30]
D001foret <- D001foret[D001foret$CODE_TFV %in% names(npoly)]
tabres <- tapply(D001foret$surf_ha, D001foret$CODE_TFV, getKCV)
tabres <- as.data.frame(matrix(unlist(tabres), nrow = length(npoly), ncol = 3, byrow = TRUE))
colnames(tabres) <- c("est", "inf", "sup")
tabres$CODE_TFV <- names(npoly)
tabres$n <- npoly
save(tabres, file = "BDFORET/tabres.Rda")
terra::writeVector(D001foret, "BDFORET/D001foret.shp", overwrite = TRUE)
```

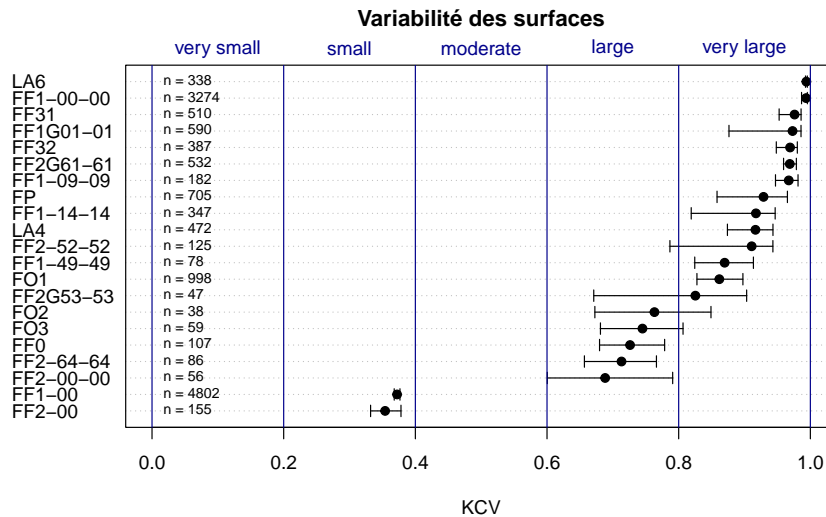
LES temps de calcul pouvant être assez longs, j'ai sauvegardé les résultats intermédiaires au format binaire compatible multi-plateformes XDR [12]. Attention, ce format n'est pas utilisable pour les objets de la classe `SpatVector`

<sup>14</sup>Voir à ce sujet la note en bas de page 25

<sup>15</sup>Si cette valeur seuil *magique* de 30 pour considérer que l'on a un « grand » échantillon vous intrigue, je vous renvoie vers la fiche « [étude empirique de l'approximation de la loi de STUDENT par la loi de LAPLACE-GAUSS] » à <http://pbil.univ-lyon1.fr/R/pdf/bs31.pdf>

parce qu'ils contiennent des pointeurs vers des objets externes (sur disque). C'est le prix à payer pour pouvoir jouer avec des données qui seraient trop lourdes à charger intégralement en mémoire (*e.g.* des images satellites à haute résolution). On peut proposer la représentation graphique suivante :

```
load("BDFORET/tabres.Rda")
tabres <- tabres[order(tabres$est), ]
dotchart(tabres$est, pch = 19, labels = tabres$CODE_TFV, xlab = "KCV",
         main = "Variabilité des surfaces", xlim = c(0, 1))
arrows(tabres$inf, 1:nrow(tabres), tabres$sup, 1:nrow(tabres), code = 3, angle = 90, le = 0.05)
KvalsethBreaks <- seq(0, 1, by = 0.2)
abline(v = KvalsethBreaks, col = "darkblue")
KvalsethVerb <- c("very small", "small", "moderate",
                 "large", "very large")
text(KvalsethBreaks[-1] - 0.1, 23, KvalsethVerb, col = "darkblue", xpd = NA)
text(rep(0, nrow(tabres)), 1:nrow(tabres), paste("n =", tabres$n), pos = 4, cex = 0.75)
```



POUR la majorité des types de formations végétales la variabilité des surfaces des polygones, selon l'échelle verbale de KVÅLSETH [8], est grande ou très grande. Deux types (*i.e.* FF1-00 et FF2-00) font exception avec une variabilité faible. Ces deux modalités correspondent à des forêts fermées de feuillus ou de conifères, respectivement, en îlots isolés d'une superficie comprise entre 0.5 et 2 ha, ce qui limite la variabilité de leurs surfaces. À l'autre extrémité du spectre nous avons les formations herbacées (LA6) et les forêts fermées à mélange de feuillus (FF1-00-00) dont le niveau de variabilité est très élevé. Dans le cas des forêts fermées à mélange de feuillus la surface des polygones varie de plus de 4 ordres de grandeur entre 0.5 et 10000 ha. Le polygone ayant le record de la plus grande surface est représenté dans la figure 2 page 13.

### 3.4 Exploitation de la géo-localisation

NOUS n'avons pas réellement fait de statistiques spatiales jusqu'à présent, même si nous avons eu besoin des coordonnées des polygones pour calculer leurs surfaces. Nous allons maintenant illustrer ce point avec un exemple concret. On s'intéresse au taux de couvert en chênes décidus dans un rayon autour d'un

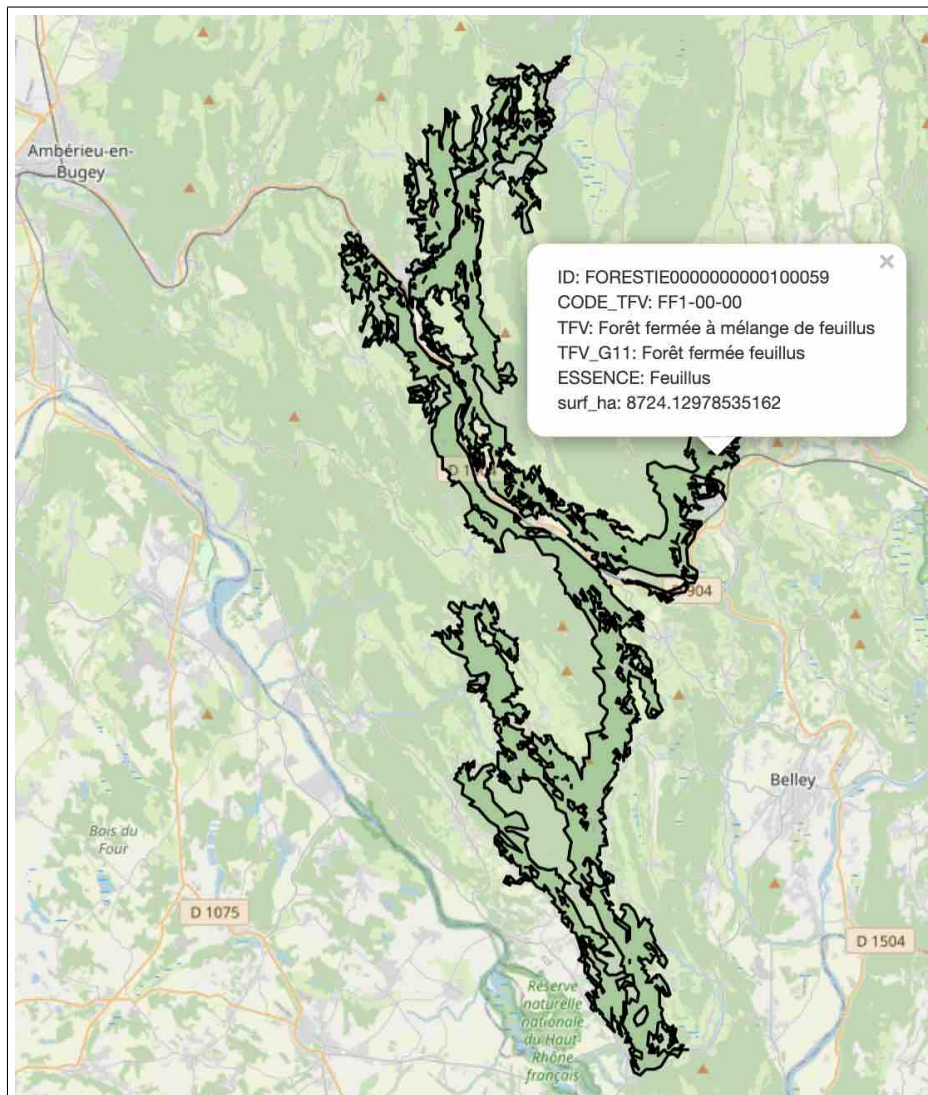
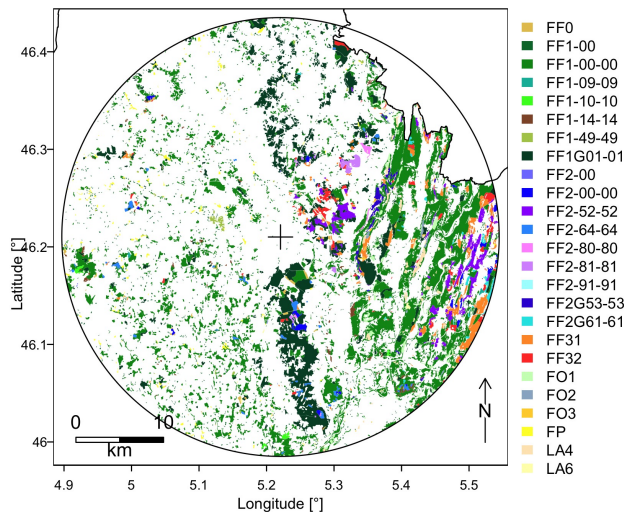


FIGURE 2 : Le plus grand polygone des données du département de l'AIN qui se situe dans le massif du BUGEY a une surface de 8724 ha et 153 enclaves (des trous en clair). On pourra trouver curieux qu'il soit d'un seul tenant sans être scindé par la D1504 et la ligne SNCF LYON-GENÈVE dans la vallée de l'ALBARINE. La jonction se fait au nord du lac des HÔPITAUX.



point d'intérêt, par exemple le site du capteur RNSA<sup>16</sup> de BOURG-EN-BRESSE. Les données sur les concentrations polliniques dans l'air « sont jugées représentatives dans un rayon de 20 à 30, voire 40 ou 50 km, autour du capteur, au moins dans les régions topographiquement et biogéographiquement homogènes [13]. » Voyons comment se présentent les données dans un rayon de 25 km autour du capteur de BOURG-EN-BRESSE. La fonction `terra::buffer()` permet facilement de générer un cercle de centre et de rayon donné, et la fonction `terra::intersect()` de sélectionner les données qui tombent dans ce cercle.

```
xlon <- +5.220700 ; ylat <- +46.21010
FRBOUB <- terra::vect(cbind(xlon, ylat), crs = "EPSG:4326")
cercle <- terra::buffer(FRBOUB, width = 25*103, quadsegs = 100)
inter <- terra::intersect(GPS, cercle)
macarte(inter, codesTFV)
terra::points(FRBOUB, pch = 3, cex = 2)
terra::lines(cercle)
```



Le problème est quasiment résolu, on veut simplement rapporter la surface des polygones de type FF1G01-01 à la surface du cercle qui, en km<sup>2</sup>, vaut :

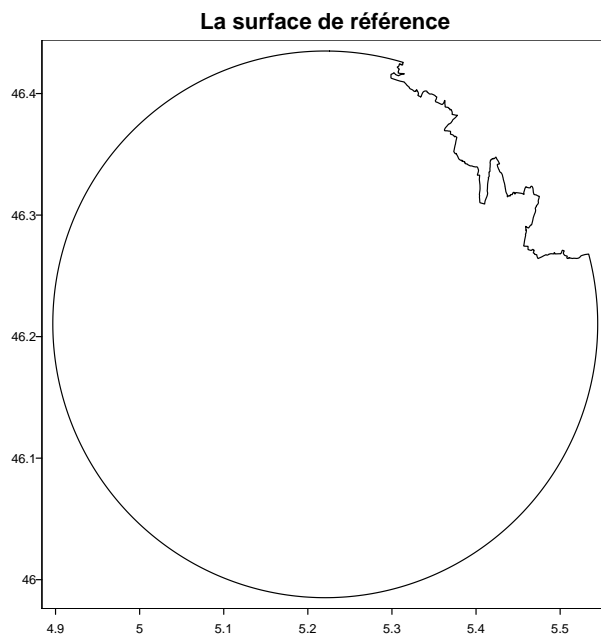
```
terra::expance(cercle, unit = "km")
[1] 1963.412
pi*252
[1] 1963.495
```

La petite différence vient ici de ce que l'objet `cercle` n'est pas un véritable cercle mais un polygone à 400 cotés (c'est l'argument `quadseg` de la fonction `terra::buffer()` qui permet de contrôler le nombre de cotés désirés). Mais il y a plus grave : nous avons des données manquantes dans le nord-est parce que

<sup>16</sup>Acronyme pour « Réseau National de Surveillance Aérobiologique » (<https://www.pollens.fr/>), les coordonnées GPS du site de BOURG-EN-BRESSE sont +5.220700 et +46.21010.

l'on déborde sur le département du JURA. Il convient donc de retrancher cette partie non-documentée ici du cercle.

```
ccercle <- terra::intersect(cercle, dep01GPS)
terra::plot(ccercle, main = "La surface de référence", las = 1)
```



Il suffit maintenant de calculer la surface relative occupée par les essences qui nous intéressent, ici les chênes décidus, codés par la modalité FF1G01-01.

```
(surfRef_km2 <- terra::expanse(terra::intersect(cercle, dep01GPS), unit = "km"))
[1] 1876.378
(surfCHS_km2 <- sum(terra::expanse(inter[inter$CODE_TFV == "FF1G01-01"] , unit = "km"))
[1] 105.5193
100*surfCHS_km2/surfRef_km2
[1] 5.623563
```

DANS un rayon de 25 km autour du capteur RNSA, les chênes décidus couvrent donc approximativement 5.6 % de la surface. Dans le cas du département de l'AIN il s'agit essentiellement<sup>17</sup> de chênes sessiles (*Quercus petraea*) et pédonculés (*Q. robur*), plus marginalement de chênes pubescents (*Q. pubescens*) et exceptionnellement de chênes rouges (*Q. rubra*).

NOUS avons donc calculé ici une *statistique* (le taux de couvert en chênes décidus) *spatialisée* (dans un rayon de 25 km autour du capteur RNSA de BOURG-EN-BRESSE). Cette valeur de 25 km est un peu arbitraire, mais rien ne nous empêche de la faire varier pour voir comment évolue notre statistique avec la valeur du rayon du cercle.

<sup>17</sup>Pour plus de détails voir la fiche « [g]éolocalisation de 11 espèces de chênes observées en France métropolitaine entre 2005 et 2021 » à <https://pbil.univ-lyon1.fr/R/pdf/dataIFN.pdf>

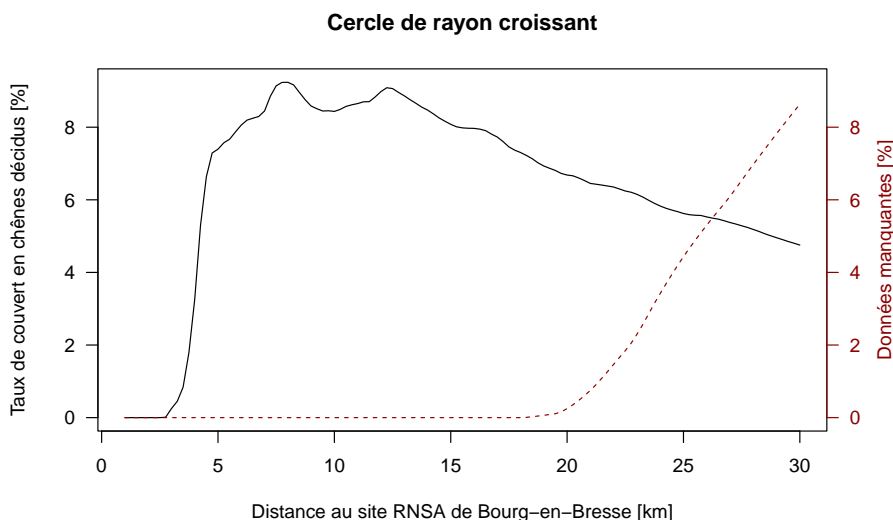


```

radseq_km <- seq(1, 30, by = 0.25) ; nr <- length(radseq_km)
cnames <- c("radius_km", "surfRef_km2", "surfCHS_km2", "CHS_pct", "pct_NA")
tabrad <- as.data.frame(matrix(NA, nrow = nr, ncol = length(cnames)))
colnames(tabrad) <- cnames
for(i in seq_len(length(radseq_km))){
  the_radius <- radseq_km[i]*10^3
  cercle <- terra::buffer(FRBOUB, width = the_radius, quadsegs = 100)
  inter <- terra::intersect(GPS, cercle)
  tabrad[i, "radius_km"] <- radseq_km[i]
  tabrad[i, "surfRef_km2"] <- terra::expanse(terra::intersect(cercle, dep01GPS), unit = "km")
  tabrad[i, "surfCHS_km2"] <- sum(terra::expanse(inter[inter$CODE_TFV == "FF1G01-01"], unit = "km"))
  tabrad[i, "CHS_pct"] <- 100*tabrad[i, "surfCHS_km2"]/tabrad[i, "surfRef_km2"]
  sc_km2 <- terra::expanse(cercle, unit = "km")
  tabrad[i, "pct_NA"] <- 100*(sc_km2 - tabrad[i, "surfRef_km2"])/sc_km2
}
save(tabrad, file = "BDFORET/tabrad.Rda")

load("BDFORET/tabrad.Rda")
par(mar = c(5, 4, 4, 4) + 0.1)
plot(tabrad$radius_km, tabrad$CHS_pct, type = "l", las = 1,
      xlab = "Distance au site RNSA de Bourg-en-Bresse [km]",
      ylab = "Taux de couvert en chênes décidus [%]",
      main = "Cercle de rayon croissant")
lines(tabrad$radius_km, tabrad$pct_NA, col = "red4", lty = 2)
axis(4, col = "red4", las = 1, col.axis = "red4")
mtext("Données manquantes [%]", side = 4, col = "red4", line = 2)

```



J'USQU'À environ 3 km du capteur RNSA le taux de couvert est nul, ce qui est normal, nous sommes en zone urbaine et donc sans forêt. Le taux de couvert augmente ensuite rapidement pour atteindre un maximum de 9.3 % pour un rayon d'environ 8 km. À partir de 12 km, le taux de couvert commence à diminuer lentement, on retrouve au passage la valeur de 5.6 % pour un rayon de 25 km obtenue précédemment. Les données manquantes commencent à augmenter à partir de 20 km, quand on commence à sortir des limites du département de l'AIN<sup>18</sup>.

LA critique que l'on pourrait faire de l'utilisation d'un cercle est que l'on induit automatiquement une diminution asymptotique de la statistique :

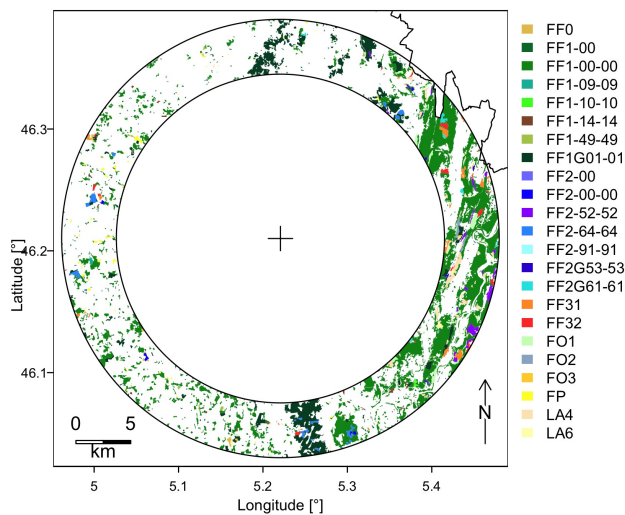
<sup>18</sup>On trouvera dans la section 4.1 page 22 quelques indications pour rabouter les données de plusieurs départements

si par la pensée on imagine un cercle de 20 000 km de rayon on voit que l'on couvre l'intégralité du globe terrestre et que l'on aura un taux de couvert quasi-nul. Pour pallier ce problème on peut utiliser un anneau plutôt qu'un cercle. Techniquement, on va manipuler un polygone avec un trou, obtenu facilement avec la fonction `terra::symdif()`. Faisons une représentation dans le cas d'un anneau de 15 km de rayon et de 5 km d'épaisseur.

```

cercle_ext <- terra::buffer(FRBOUB, width = 20*10^3, quadsegs = 100)
cercle_int <- terra::buffer(FRBOUB, width = 15*10^3, quadsegs = 100)
ring <- terra::symdif(cercle_ext, cercle_int)
inter <- terra::intersect(GPS, ring)
macarte(inter, codesTFV)
terra::points(FRBOUB, pch = 3, cex = 2)
terra::lines(ring)

```



COMME précédemment, on peut s'amuser à faire varier le rayon de l'anneau pour voir comment évolue la statistique. Ici nous faisons varier son rayon de 0.5 à 29.5 km avec un pas de 250 m. Son épaisseur est constante, toujours égale à 1 km.

```

radseq_km <- seq(1, 30, by = 0.25) ; nr <- length(radseq_km)
cnames <- c("radius_km", "surfRef_km2", "surfCHS_km2", "CHS_pct", "pct_NA")
tabring <- as.data.frame(matrix(NA, nrow = nr, ncol = length(cnames)))
colnames(tabring) <- cnames
for(i in seq_len(length(radseq_km))){
  the_radius <- radseq_km[i]*10^3
  cercle_ext <- terra::buffer(FRBOUB, width = the_radius + 500, quadsegs = 100)
  cercle_int <- terra::buffer(FRBOUB, width = the_radius - 500, quadsegs = 100)
  ring <- terra::symdif(cercle_ext, cercle_int)
  inter <- terra::intersect(GPS, ring)
  tabring[i, "radius_km"] <- radseq_km[i]
  tabring[i, "surfRef_km2"] <- terra::expanse(terra::intersect(ring, dep01GPS), unit = "km")
  tabring[i, "surfCHS_km2"] <- sum(terra::expanse(inter[inter$CODE_TFV == "FF1G01-01"], unit = "km"))
  tabring[i, "CHS_pct"] <- 100*tabring[i, "surfCHS_km2"]/tabring[i, "surfRef_km2"]
  sr_km2 <- terra::expanse(ring, unit = "km")
  tabring[i, "pct_NA"] <- 100*(sr_km2 - tabring[i, "surfRef_km2"])/sr_km2
}

```

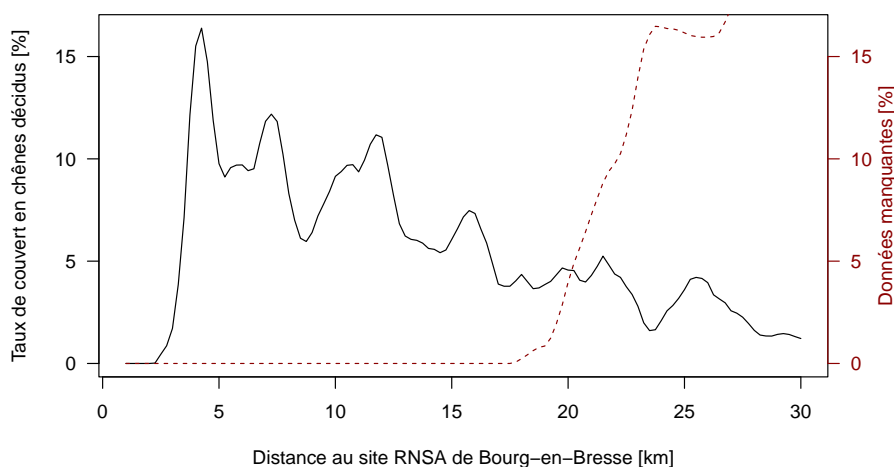
```

}
save(tabring, file = "BDFORET/tabring.Rda")

load("BDFORET/tabring.Rda")
par(mar = c(5, 4, 4, 4) + 0.1)
plot(tabring$radius_km, tabring$CHS_pct, type = "l", las = 1,
      xlab = "Distance au site RNSA de Bourg-en-Bresse [km]",
      ylab = "Taux de couvert en chênes décidus [%]",
      main = "Anneau de rayon croissant de 1 km d'épaisseur")
lines(tabring$radius_km, tabring$pct_NA, col = "red4", lty = 2)
axis(4, col = "red4", las = 1, col.axis = "red4")
mtext("Données manquantes [%]", side = 4, col = "red4", line = 2)

```

Anneau de rayon croissant de 1 km d'épaisseur



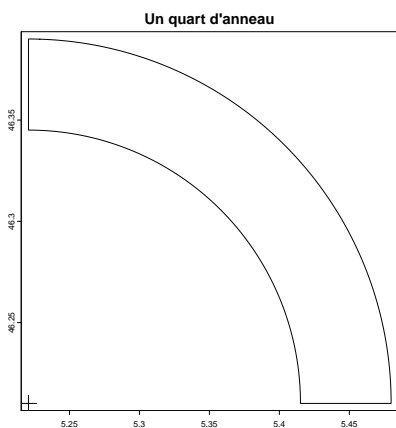
ON visualise ici un pic de 16.3 % du taux de couvert pour un rayon de 4.25 km. Il correspond à la forêt domaniale de SEILLON, c'est un cas assez exceptionnel : il est rare qu'une forêt de chênes décidus soit aussi proche d'une zone urbaine.

POUR raffiner encore un peu les choses, on peut décider de stratifier les données selon les quatre points cardinaux. Je commence par prélever un quart de mon anneau :

```

ringext <- as.vector(terra::ext(ring))
xmax <- ringext["xmax"]; ymax <- ringext["ymax"]
xmin <- xlon; ymin <- ylat
car <- terra::vect(cbind(c(xmin, xmin, xmax, xmax), c(ymin, ymax, ymax, ymin)),
                  type = "polygon")
sector <- terra::intersect(ring, car)
terra::plot(sector, main = "Un quart d'anneau")
terra::points(FRBOUB, pch = 3, cex = 2)

```

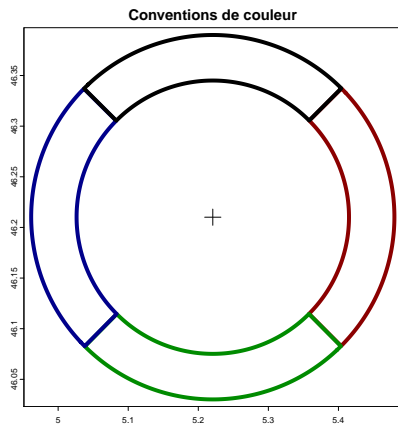


NOUS allons maintenant faire tourner ce quart d'anneau pour définir les 4 secteurs géographiques qui nous intéressent. La fonction `terra::spin()` ne fonctionnant qu'avec des géométries projetées, on définit une petite fonction utilitaire pour faire cette opération en LAMBERT 93 dans les coulisses.

```
spinEPSG4326 <- function(x, angle, x0, y0){
  center <- terra::vect(cbind(x0, y0), crs = "EPSG:4326") # GPS
  centerL93 <- terra::project(center, "EPSG:2154") # Lambert 93
  xlonL93 <- terra::geom(centerL93)[1, "x"]
  ylatL93 <- terra::geom(centerL93)[1, "y"]
  xL93 <- terra::project(x, "EPSG:2154")
  rotated <- terra::spin(xL93, angle, xlonL93, ylatL93)
  back <- terra::project(rotated, "EPSG:4326")
  return(back)
}
```

COMME convention de couleurs, on utilise le rouge pour l'est (bloc de l'Est, rouge), le vert pour le sud (chênes verts du sud, vert), le bleu pour l'ouest (océan ATLANTIQUE à l'ouest, bleu) et le noir pour le nord (mines de charbon du nord, noir) :

```
terra::plot(ring, main = "Conventions de couleur")
terra::points(FRBOUB, pch = 3, cex = 2)
mycols <- c("red4", "green4", "blue4", "black")
sectorE <- spinEPSG4326(sector, 45, xlon, ylat)
terra::lines(sectorE, col = mycols[1], lwd = 5)
sectorS <- spinEPSG4326(sector, 45 + 90, xlon, ylat)
terra::lines(sectorS, col = mycols[2], lwd = 5)
sectorW <- spinEPSG4326(sector, 45 + 2*90, xlon, ylat)
terra::lines(sectorW, col = mycols[3], lwd = 5)
sectorN <- spinEPSG4326(sector, 45 + 3*90, xlon, ylat)
terra::lines(sectorN, col = mycols[4], lwd = 5)
```



Il ne nous reste plus qu'à faire varier le rayon comme précédemment pour voir comment évolue notre statistique. Notez que ci-après nous avons factorisé le code en faisant une boucle sur les quatre points cardinaux.

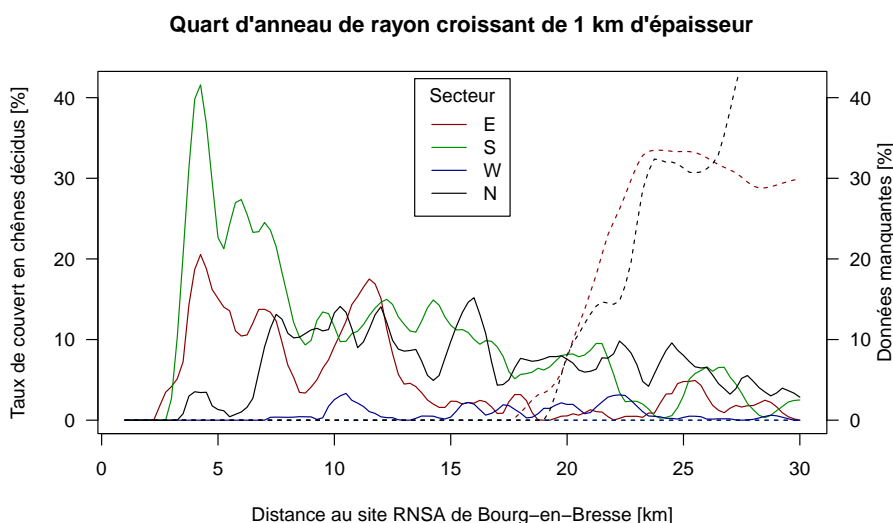
```
# Préparation de la table des résultats
radseq_km <- seq(1, 30, by = 0.25) ; nr <- length(radseq_km)
cardinaux <- c("E", "S", "W", "N")
varcard <- c("surfRef_km2", "surfCHS_km2", "CHS_pct", "pct_NA")
lastcolnames <- paste(rep(cardinaux, each = length(varcard)), varcard, sep = "_")
cnames <- c("radius_km", lastcolnames)
tabsector <- as.data.frame(matrix(NA, nrow = nr, ncol = length(cnames)))
colnames(tabsector) <- cnames
# Boucle sur les rayons
for(i in seq_len(length(radseq_km))){
  tabsector[i, "radius_km"] <- radseq_km[i]
  the_radius <- radseq_km[i]*10^3
  cercle_ext <- terra::buffer(FRBOUB, width = the_radius + 500, quadsegs = 100)
  cercle_int <- terra::buffer(FRBOUB, width = the_radius - 500, quadsegs = 100)
  ring <- terra::symdif(cercle_ext, cercle_int)
  # Quart d'anneau
  ringext <- as.vector(terra::ext(ring))
  xmax <- ringext["xmax"] ; ymax <- ringext["ymax"]
  xmin <- xlon ; ymin <- ylat
  car <- terra::vect(cbind(c(xmin, xmin, xmax, xmax), c(ymin, ymax, ymax, ymin)),
    type = "polygon")
  sector <- terra::intersect(ring, car)
  # Boucle sur les cardinaux
  for(j in seq_len(length(cardinaux))){
    mknm <- function(x) paste(cardinaux[j], x, sep = "_")
    the_sector <- spinEPSG4326(sector, 45 + (j - 1)*90, xlon, ylat)
    inter <- terra::intersect(GPS, the_sector)
    tabsector[i, mknm("surfRef_km2")] <-
      terra::expance(terra::intersect(the_sector, dep01GPS), unit = "km")
    tabsector[i, mknm("surfCHS_km2")] <-
      sum(terra::expance(inter[inter$CODE_TFV == "FF1G01-01"], unit = "km"))
    tabsector[i, mknm("CHS_pct")] <-
      100*tabsector[i, mknm("surfCHS_km2")]/tabsector[i, mknm("surfRef_km2")]
    sr_km2 <- terra::expance(the_sector, unit = "km")
    if(sr_km2 <= 0){
      tabsector[i, mknm("pct_NA")] <- 100
    } else {
      tabsector[i, mknm("pct_NA")] <- 100*(sr_km2 - tabsector[i, mknm("surfRef_km2")])/sr_km2
    }
  }
}
save(tabsector, file = "BDFORET/tabsector.Rda")

load("BDFORET/tabsector.Rda")
par(mar = c(5, 4, 4, 4) + 0.1)
plot.new() ; plot.window(xlim = range(tabsector$radius_km), ylim = c(0, max(tabsector[,4*(1:4)]))
axis(1) ; axis(2, las = 1) ; box()
```

```

title(xlab = "Distance au site RNSA de Bourg-en-Bresse [km]",
      ylab = "Taux de couvert en chênes décidus [%]",
      main = "Quart d'anneau de rayon croissant de 1 km d'épaisseur")
cardinaux <- c("E", "S", "W", "N")
x <- tabsector$radius_km
for(i in seq_len(length(cardinaux))){
  mknm <- fonction(x) paste(cardinaux[i], x, sep = "_")
  lines(x, tabsector[ , mknm("CHS_pct")], col = mycols[i])
  lines(x, tabsector[ , mknm("pct_NA")], col = mycols[i], lty = 2)
}
legend("top", inset = 0.02, legend = cardinaux, lty = 1, col = mycols, title = "Secteur")
axis(4, las = 1)
mtext("Données manquantes [%]", side = 4, line = 2)

```



La forêt domaniale de SEILLON est bien mise en évidence avec le pic de 41.6 % de taux de couvert en chênes décidus à une distance de 4.25 km dans le secteur sud. Dans le secteur est on note un autre pic de 20.5 % également à 4.25 km correspondant aux forêts dans le secteur de l'aérodrome de CEYZÉRIAT et un pic de 17.7 % à 11.5 km correspondant aux forêts du REVERMONT dans l'étage collinéen du sud du massif du JURA. Le secteur ouest est presque toujours à zéro, nous sommes dans le secteur des étangs de la DOMBES avec très peu d'emprise forestière. Les données manquantes commencent à augmenter dans les secteurs nord et est quand on déborde sur le département du JURA, non documenté ici.

NOUS avons donc exhibé ici un exemple de statistique descriptive spatialisée. Ce qu'elle permet de prédire c'est que pendant la période de diffusion pollinique, si les conditions météo s'y prêtent, les capteurs de pollen du site RNSA de BOURG-EN-BRESSE devraient voir leurs comptes augmenter pour les chênes décidus quand les vents sont issus des secteurs sud et est.

## 4 Annexes

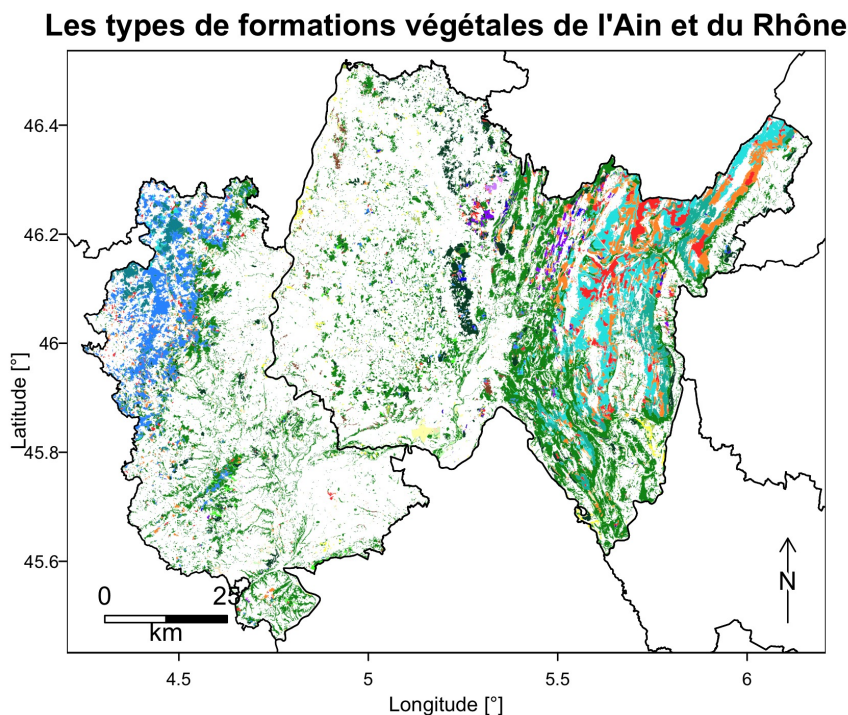
### 4.1 Raboutage de départements

POUR illustrer cette partie nous allons récupérer les données du département du RHÔNE, limitrophe du département de l'AIN.

```
chmin <- paste0("BDFORET/",
               "BDFORET_2-0__SHP_LAMB93_D069_2014-04-01/",
               "BDFORET/",
               "1_DONNEES_LIVRAISON/BDF_2-0_SHP_LAMB93_D069/")
D069 <- terra::vect(paste0(chmin, "FORMATION_VEGETALE.shp"))
```

LA fonction de base `rbind()` permet de concaténer deux objets de type `SpatVector`.

```
D001069 <- rbind(D001, D069)
mycol <- subset(codesTFV, CODE_TFV %in% unique(D001069$CODE_TFV))$col
terra::plot(terra::project(D001069, "EPSG:4326"), "CODE_TFV",
            border = "transparent", col = mycol,
            main = "Les types de formations végétales de l'Ain et du Rhône",
            xlab = "Longitude [°]", ylab = "Latitude [°]", las = 1,
            legend = NULL)
terra::lines(terra::project(dep, "EPSG:4326"))
terra::sbar(type = "bar", xy = "bottomleft", below = "km")
terra::north(xy = "bottomright")
```





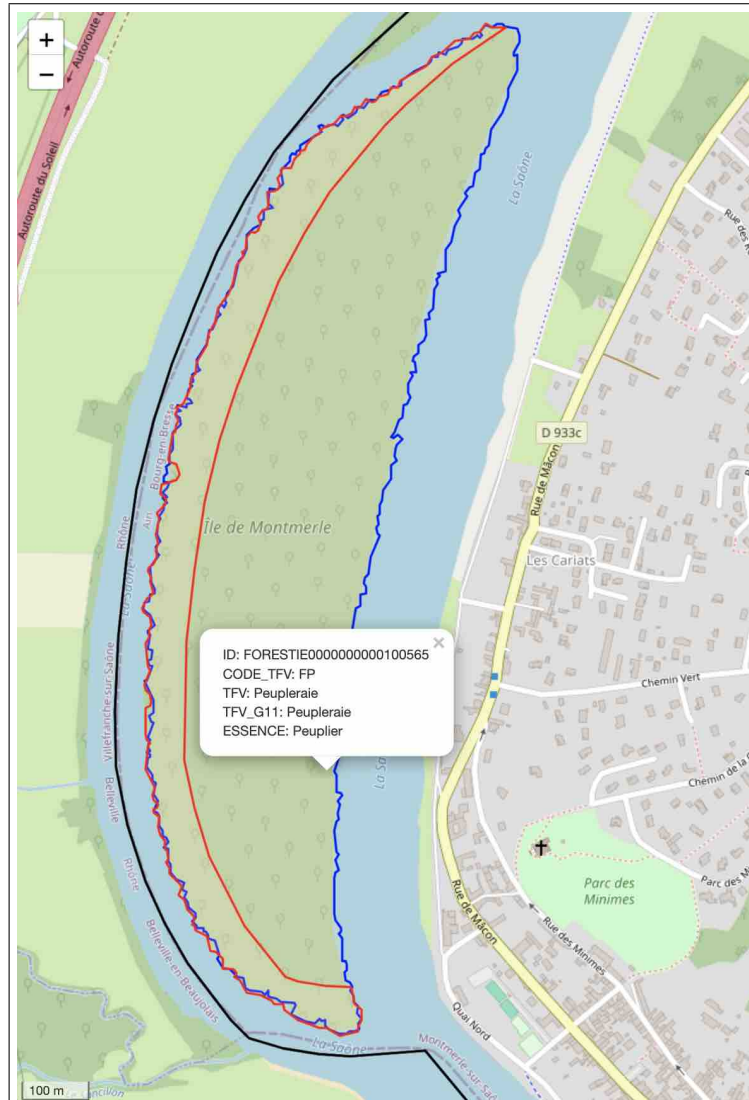


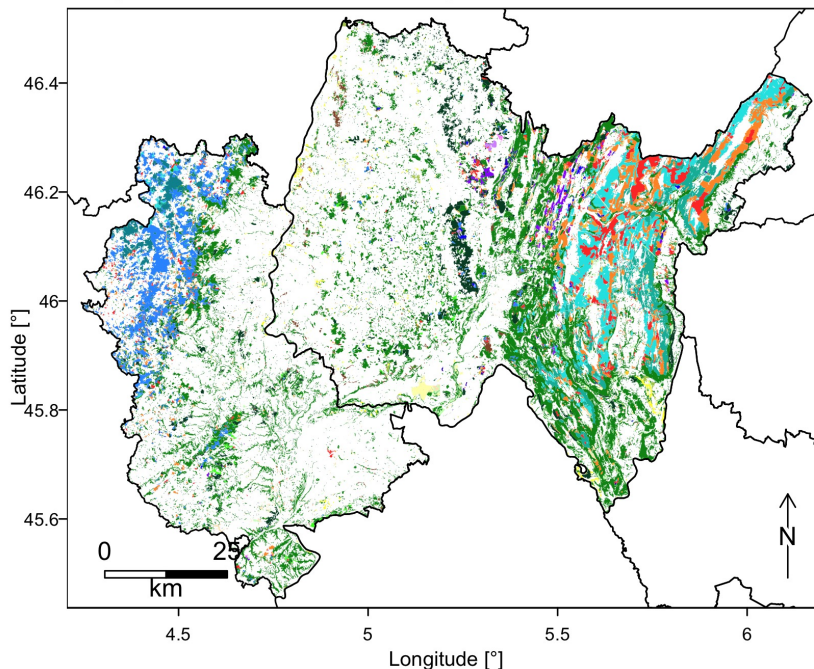
FIGURE 3 : La peupleraie de l'île de MONTMERLE, en face de MONTMERLE-SUR-SAÔNE, est intégralement dans le département de l'AIN puisqu'à l'est de la ligne noire de démarcation du département du RHÔNE et de l'AIN. Le polygone de contour bleu est celui des données du département de l'AIN, il couvre l'intégralité de l'île de MONTMERLE. Le polygone de contour rouge est celui des données du département du RHÔNE, il illustre la zone du « buffer » de 100 m : il ne couvre que la partie de l'île qui est à moins de 100 m de la limite inter-départementale.

MAIS, attention, cette opération se contente de collationner les polygones des deux objets. Or, la documentation [1] précise que « [l]es données BD Forêt<sup>®</sup> version 2 sont livrées selon un découpage départemental (limite des départements de la BDTOPO<sup>®</sup> augmentée d'un buffer de 100 m) ou selon une emprise particulière ». Si on n'y prend pas garde, certaines zones vont être comptées en double (voir par exemple la figure 3 page 23). Pour pallier ce problème, il suffit de restreindre les données au strict périmètre départemental *avant* de les concaténer.

```
D001c <- terra::intersect(D001, dep01)
dep69 <- dep[dep$NOM == "Rhône", ]
D069c <- terra::intersect(D069, dep69)
D001069c <- rbind(D001c, D069c)
terra::writeVector(D001069c, "BDFORET/D001069c.shp")

D001069c <- terra::vect("BDFORET/D001069c.shp")
mycol <- subset(codesTFV, CODE_TFV %in% unique(D001069c$CODE_TFV))$col
terra::plot(terra::project(D001069c, "EPSG:4326"), "CODE_TFV",
  border = "transparent", col = mycol,
  main = "Les types de formations végétales de l'Ain et du Rhône",
  xlab = "Longitude [°]", ylab = "Latitude [°]", las = 1,
  legend = NULL)
terra::lines(terra::project(dep, "EPSG:4326"))
terra::sbar(type = "bar", xy = "bottomleft", below = "km")
terra::north(xy = "bottomright")
```

### Les types de formations végétales de l'Ain et du Rhône



ÉVIDEMMENT, à cette échelle on ne voit pas bien la différence avec la carte précédente, il y en a pourtant une puisque l'on a perdu environ 1200 ha dans l'opération. C'est certes peu (0.4 %) par rapport aux 302064 ha caractérisés dans les deux départements, mais si on ne le fait pas, les rivières étant souvent une frontière naturelle entre les départements, on risque d'introduire un biais systématique en faveur des essences les plus hygrophiles. La fusion de polygones de part et d'autre de la frontière n'est pas triviale parce que les types de formations végétales ne sont pas forcément identiques (exemple dans la figure 4 page 26). Une limite administrative rasant une forêt pouvant induire un polygone de très faible surface, on prendra soin de censurer les polygones de moins de 0.5 ha (surface minimale d'une forêt selon les normes internationales) si on souhaite faire des statistiques sur les surfaces<sup>19</sup>.

```
id01 <- terra::intersect(D001, D069)$ID
id69 <- terra::intersect(D069, D001)$ID
idall <- c(id01, id69)
inter0169 <- D001069[D001069$ID %in% idall, ]
mycol <- subset(codesTFV, CODE_TFV %in% unique(inter0169$CODE_TFV))$col
terra::plot(terra::project(inter0169, "EPSG:4326"), "CODE_TFV",
  border = "transparent", col = mycol,
  main = "Les types de formations végétales de l'Ain et du Rhône",
  xlab = "Longitude [°]", ylab = "Latitude [°]", las = 1)
terra::lines(terra::project(dep, "EPSG:4326"))
terra::sbar(type = "bar", xy = "bottomleft", below = "km")
terra::north(xy = "bottomright")

tabrel <- terra::relate(D069, D001, "intersects", pairs = TRUE)
join <- cbind(as.data.frame(D069[tabrel[, 1], c("ID", "CODE_TFV")]),
  as.data.frame(D001[tabrel[, 2], c("ID", "CODE_TFV")]))
colnames(join) <- c("ID69", "TFV_69", "ID01", "TFV_01")
join

#D001$surf_m2 <- round(terra::expanse(D001))
#D069$surf_m2 <- round(terra::expanse(D069))
#piti01 <- which(D001$surf_m2 < 5000)
piti01 <- which(D001$surf_m2 < 5000)
D001piti <- D001[piti01, ]
mycol <- subset(codesTFV, CODE_TFV %in% unique(D001piti$CODE_TFV))$col
piti69 <- which(D069$surf_m2 < 5000)
D069piti <- D069[piti69, ]
mycol <- subset(codesTFV, CODE_TFV %in% unique(D069piti$CODE_TFV))$col
mamap <- terra::plet(D069piti, col = mycol, fill = 0.8)
#mamap <- terra::points(mamap, terra::centroids(D069piti))
mamap <- terra::lines(mamap, dep69, col = "red")
library(leaflet)
mamap <- addScaleBar(mamap, "bottomleft", scaleBarOptions(imperial = FALSE))
mamap

mycol <- subset(codesTFV, CODE_TFV %in% unique(D001069c$CODE_TFV))$col
myborder <- ifelse(D001069c$ID_1 %in% D069$ID, "red", "blue")
mamap <- terra::plet(D001069c, "CODE_TFV", col = mycol, border = myborder, fill = 0.8, lwd = 0.5)
mamap <- terra::lines(mamap, dep)
library(leaflet)
mamap <- addScaleBar(mamap, "bottomleft", scaleBarOptions(imperial = FALSE))
mamap
```

<sup>19</sup>Ceci dit, j'ai compté 11 polygones de moins de 0.5 ha dans les données du département de l'AIN et 56 dans celles du département du RHÔNE. La figure 5 page 26 exhibe un exemple pour chaque département. Ils sont tous situés en bordure de la limite du département, c'est donc un artefact de coupure administrative. Il y a bien un « buffer » de 100 m pour le département du RHÔNE, mais cela ne semble pas être le cas pour le département de l'AIN.

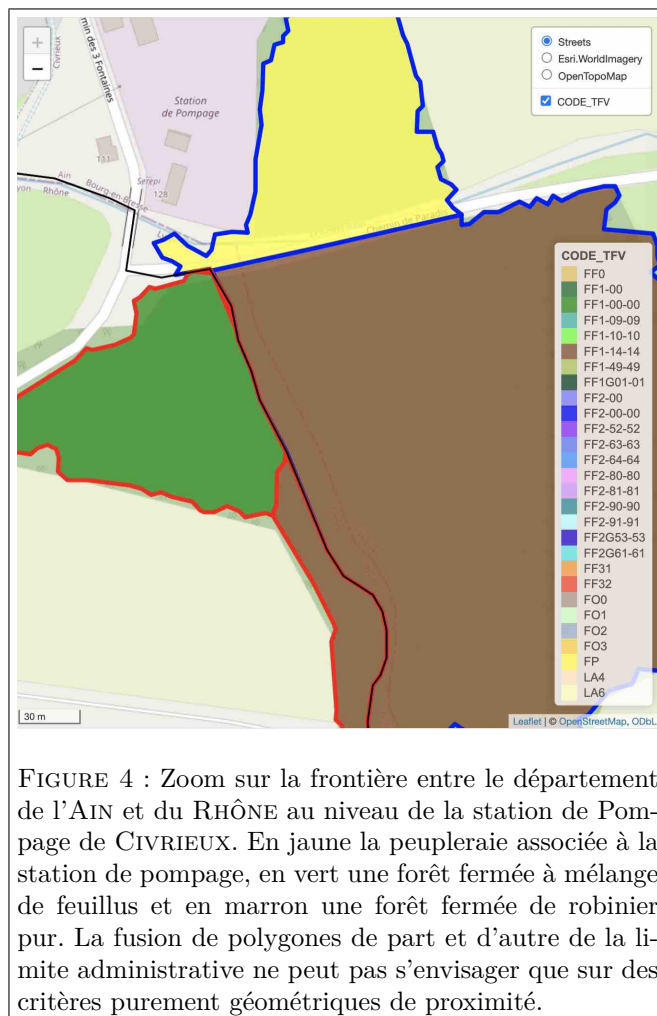


FIGURE 5 : Les plus petits polygones de BD Forêt<sup>®</sup> version 2 : à gauche pour le département de l'AIN et à droite pour celui du RHÔNE.

Modalité	Intitulé
FF0	Forêt fermée sans couvert arboré
FF1-00	Forêt fermée de feuillus purs en îlots
FF1G01-01	Forêt fermée de chênes décidus purs
FF1G06-06	Forêt fermée de chênes sempervirents purs
FF1-09-09	Forêt fermée de hêtre pur
FF1-10-10	Forêt fermée de châtaignier pur
FF1-14-14	Forêt fermée de robinier pur
FF1-49-49	Forêt fermée d'un autre feuillu pur
FF1-00-00	Forêt fermée à mélange de feuillus
FF2-00	Forêt fermée de conifères purs en îlots
FF2-51-51	Forêt fermée de pin maritime pur
FF2-52-52	Forêt fermée de pin sylvestre pur
FF2G53-53	Forêt fermée de pin laricio ou pin noir pur
FF2-57-57	Forêt fermée de pin d'Alep pur
FF2G58-58	Forêt fermée de pin à crochets ou pin cembro pur
FF2-81-81	Forêt fermée d'un autre pin pur
FF2-80-80	Forêt fermée à mélange de pins purs
FF2G61-61	Forêt fermée de sapin ou épicéa
FF2-63-63	Forêt fermée de mélèze pur
FF2-64-64	Forêt fermée de douglas pur
FF2-91-91	Forêt fermée d'un autre conifère pur autre que pin
FF2-90-90	Forêt fermée à mélange d'autres conifères
FF2-00-00	Forêt fermée à mélange de conifères
FF31	Forêt fermée à mélange de feuillus prépondérants et conifères
FF32	Forêt fermée à mélange de conifères prépondérants et feuillus
F00	Forêt ouverte sans couvert arboré
F01	Forêt ouverte de feuillus purs
F02	Forêt ouverte de conifères purs
F03	Forêt ouverte à mélange de feuillus et conifères
FP	Peupleraie
LA4	Lande
LA6	Formation herbacée

TABLE 1 : Les 32 modalités utilisées pour caractériser les types de formations végétales dans BD Forêt<sup>®</sup> version 2.



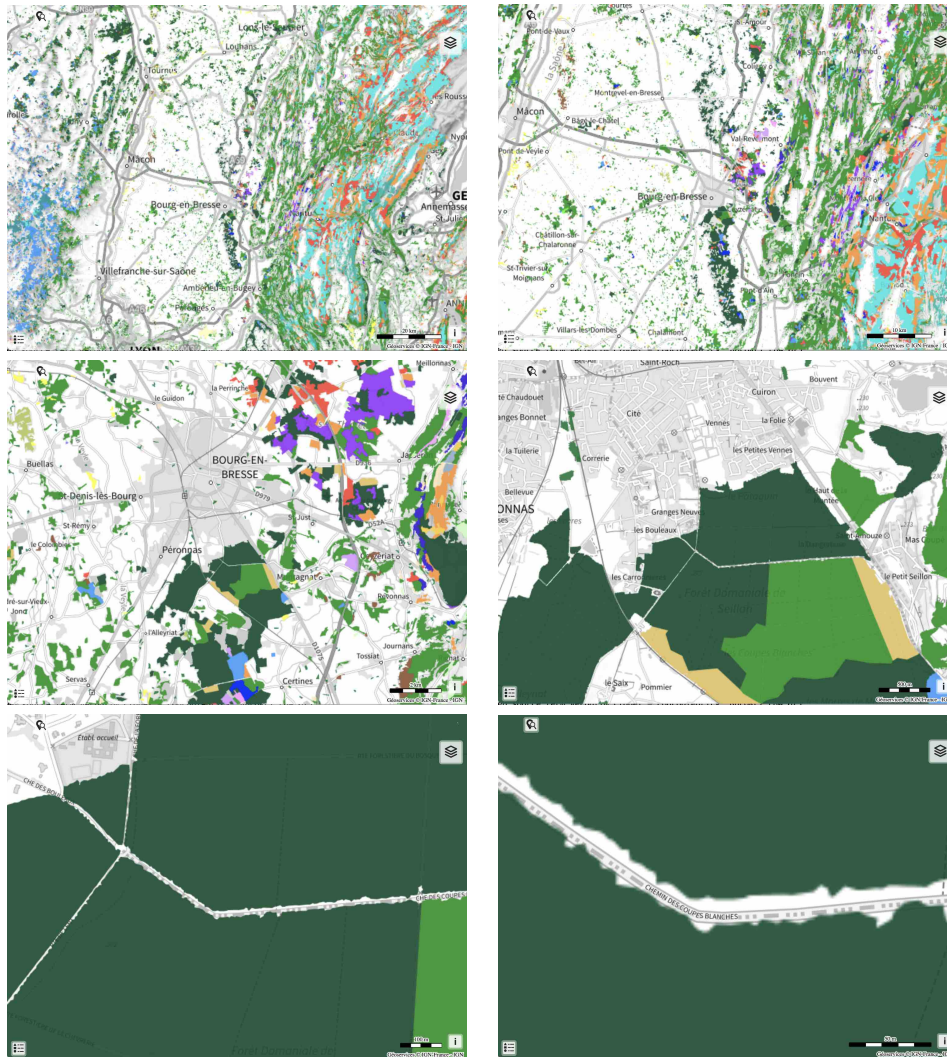
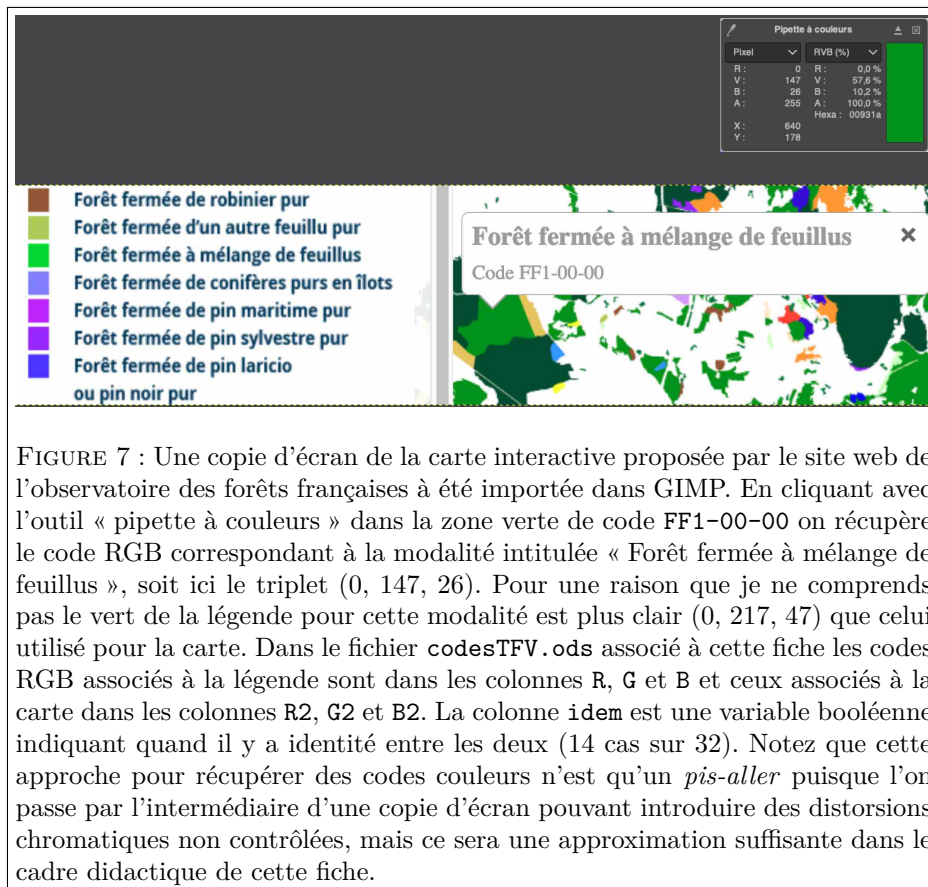


FIGURE 6 : Zoom progressif sur le virage du chemin des coupes blanches dans la forêt domaniale de SEILLON, au sud de BOURG-EN-BRESSE, dans le département de l'AIN. Il s'agit de copies d'écran de la carte interactive disponible sur le site de l'observatoire des forêts françaises (<https://foret.ign.fr/carte>). Au plus forts grossissements on peut voir que les données ont été « rasterisées » pour accélérer la vitesse d'affichage. Le fait que les houppiers des arbres semblent surplomber le chemin est dû à la largeur exagérée de ce dernier (comparer avec la figure 1 page 4).





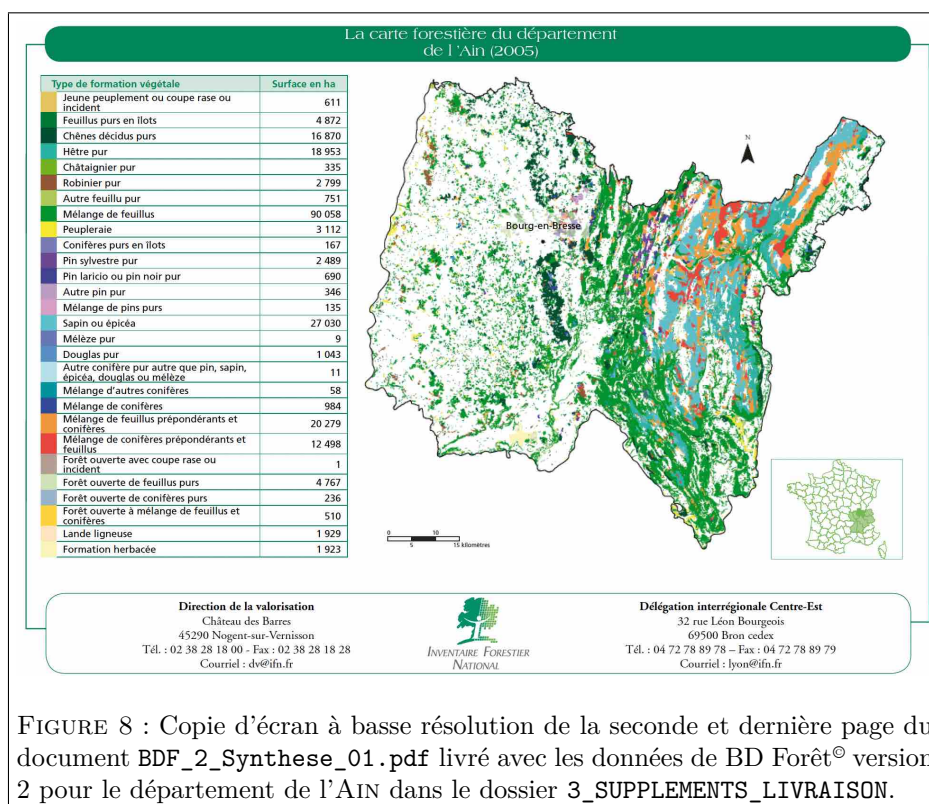


FIGURE 8 : Copie d'écran à basse résolution de la seconde et dernière page du document BDF\_2\_Synthese\_01.pdf livré avec les données de BD Forêt® version 2 pour le département de l'Ain dans le dossier 3\_SUPPLEMENTS\_LIVRAISON.

## Références

- [1] Anonymous. BD Forêt<sup>©</sup> Version 2 – Descriptif de contenu. Technical report, Institut National de l'Information Géographique et Forestière, 2014. Révision septembre 2014.
- [2] Anonymous. Évaluation des ressources forestières mondiales 2020. Termes et définitions. Technical report, Organisation des Nations unies pour l'alimentation et l'agriculture, 2018. Document de travail n° 188 d'évaluation des ressources forestières mondiales.
- [3] 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.
- [4] W.S. Cleveland. *The elements of graphing data*. Wadsworth, Monterey, CA, USA, 1985.
- [5] B. Efron. Better bootstrap confidence intervals. *Journal of the American statistical Association*, 82(397) :171–185, 1987.
- [6] R.J. Hijmans. *raster : Geographic Data Analysis and Modeling*, 2023. R package version 3.6-26.
- [7] R.J. Hijmans. *terra : Spatial Data Analysis*, 2024. R package version 1.7-71.
- [8] T.O. Kvålseth. Coefficient of variation : the second-order alternative. *Journal of Applied Statistics*, 44 :402–415, 2017.
- [9] J.R. Lobry, M.-C. Bel-Venner, M. Bogdziewicz, A. Hacket-Pain, and S. Venner. The CV is dead, long live the CV! *Methods in Ecology and Evolution*, 14 :2780–2786, 2023.
- [10] T. Lumley. *dichromat : Color Schemes for Dichromats*, 2022. R package version 2.0-0.1.
- [11] K. Pearson. VII. mathematical contributions to the theory of evolution.—III. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A*, (187) :253–318, 1896.
- [12] Sun Microsystems. XDR : external data representation standard. RFC 1014. Technical report, Network Working Group, 1987.
- [13] M. Thibaudon, D. Caillaud, and J.-P. Besancenot. Méthodes d'étude des pollens atmosphériques et calendriers polliniques. *Revue des maladies respiratoires*, 30(6) :463–479, 2013.
- [14] F. Viénot, H. Brettel, and J.D. Mollon. Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research and Application*, 24 :243–252, 1999.
- [15] G. Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélogrammes primitifs. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(134) :198–287, 1908.

- [16] G. Voronoï. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1908(133) :97–102, 1908.