Fiche TD avec le logiciel R : histoRNSA

Les données historiques du RNSA

Pr Jean R. Lobry

Importation dans R des données historiques du feu RNSA maintenant disponibles en licence libre.

Table des matières

1 Nomenclature des sites

L'importation des données dans R ne pose pas de problèmes particuliers et le code correspondant est donné dans la section 2 page 5. Les données horaires et journalières sont dans les listes de tables `pol2h` et `polj`, respectivement.

```
chmin <- "https://esb.univ-lyon1.fr/donnees/histoRNSA/"
load(url(paste0(chmin, "pol2h.Rda"))) ; comment(pol2h)
[1] "Généré le : 2026-06-16 15:46:00.099511"
load(url(paste0(chmin, "polj.Rda"))) ; comment(polj)
[1] "Généré le :  2026-06-17 15:06:14.702467"
```

Les sites sont désignés avec trois nomenclatures différentes dont aucune ne ressemble à celle du feu RNSA, par exemple `FRBOUB` pour le capteur de Bourg-en-Bresse.

1.1 Nomenclature des dossiers

Il y a le nom des 133 dossiers des sites dans les données horaires (dossier `BDD`). Ils sont sont en majuscules non diacritées avec un tiret ou une espace comme séparateur si besoin est :

```
length(pol2h)
[1] 133
head(names(pol2h))
[1] "AGEN"              "AIX-EN-PROVENCE"   "AJACCIO"           "ALES"
[5] "AMBERIEU-EN-BUGEY" "AMIENS"
```

C'est ce qui se rapproche le plus d'une nomenclature lisible par un être humain, mais l'absence de diacritiques et l'usage non standard des majuscules fait que cela n'est pas exploitable directement pour une typographie de qualité : comparez `MONTLUCON` à Montluçon.

1.2 Nomenclature dans les fichiers `*.csv`

Dans chaque dossier pour les sites il y a un fichier `*.csv` pour chaque année portant en deuxième colonne une autre nomenclature, toujours en majuscules non diacritées, mais plus compacte, c'est sans doute une notation pouvant être utilisée comme nom de variable. Ça ne pose pas de problème dans R à condition de protéger, hors contexte où les chaînes de caractères peuvent être utilisées, les noms avec des accents graves, par exemple :

```
`Trifouilli-lès-Oies` <- pi
`Trifouilli-lès-Oies`
[1] 3.141593
```

On range dans le vecteur `allVillesCSV` la nomenclature des fichiers `*.csv` en lisant la première ligne de la deuxième colonne de toutes les tables.

```
allVillesCSV <- unlist(lapply(pol2h, \(x) x[1, 2]))
head(allVillesCSV)
            AGEN   AIX-EN-PROVENCE          AJACCIO             ALES
          "AGEN"        "AIXENPRO"        "AJACCIO"           "ALES"
AMBERIEU-EN-BUGEY           AMIENS
       "AMBERIEU"          "AMIENS"
```

Il y a trois sites pour lesquels le nom n'est pas renseigné dans le fichier `*.csv`, ce sont en fait des sites pour lesquels il n'y a pas de données : les fichiers `*.csv` sont vides.

```
(NAville <- names(allVillesCSV)[is.na(allVillesCSV)])
```

```
[1] "CORTE"       "DIEU-LE-FIT" "SEDAN"
```

```
pol2h[NAville]
```

```
$CORTE
 [1] Date    Site    Species X0100   X0300   X0500   X0700   X0900   X1100   X1300
[11] X1500   X1700   X1900   X2100   X2300   Total
<0 rows> (or 0-length row.names)
$`DIEU-LE-FIT`
 [1] Date    Site    Species X0100   X0300   X0500   X0700   X0900   X1100   X1300
[11] X1500   X1700   X1900   X2100   X2300   Total
<0 rows> (or 0-length row.names)

$SEDAN
 [1] Date    Site    Species X0100   X0300   X0500   X0700   X0900   X1100   X1300
[11] X1500   X1700   X1900   X2100   X2300   Total
<0 rows> (or 0-length row.names)
```

Les abréviations utilisées sont assez transparentes, par exemple `BOURGENB` pour `BOURG-EN-BRESSE`, sauf dans les cas suivants :

Nom du dossier		Nom dans le fichier `*.csv`
BASSENS	→	BORDBASS
CRAPONNE	→	LYONWEST
PESSAC	→	BORDPESS
SAINT-QUENTIN	→	QUENTIN
SAINT-QUENTIN-EN-YVELINES	→	SQY
THIZY	→	RHONENOR

Quand il y a plusieurs sites pour une même ville ils sont numérotés en chiffres arabes ou romains pour les distinguer, mais pas de façon cohérente entre les deux nomenclatures.

Nom du dossier		Nom dans le fichier `*.csv`
LYON	→	LYON
LYON2	→	LYONII
LYON3	→	LYONIII
NICE	→	NICE
NICE2	→	NICE2
NIMES	→	NIMES
NIMES2	→	NIMES2
PARIS	→	PARIS
PARIS2	→	PARISII

1.3 Nomenclature dans le nom des fichiers journaliers

Dans le dossier `BDD_daily` le nom des fichiers `*.xls[x]` portent enfin la trace d'une troisième nomenclature[1] qui ressemble à la nomenclature plus compacte mais il n'y a plus que 130 sites documentés.

```
length(polj)
```

```
[1] 130
```

[1] *e.g.* `Particle_Extract_BOURGENB_2006-01-01_to_2006-12-31.xls`

```
head(names(polj))
```
```
[1] "AGEN"     "AIXENPRO" "AJACCIO"  "ALES"     "AMBERIEU" "AMIENS"
```

Les sites suivants sont présents dans les données journalières mais pas dans les les fichier `*.csv`.

```
names(polj)[!names(polj) %in% allVillesCSV]
```
```
[1] "CORTE"    "DIEULEFI" "LAFLECHE" "SACLAYSP" "SEDAN"
```

Pour `CORTE`, `DIEULEFI` et `SEDAN` cela correspond aux fichiers `*.csv` vides, et on peut vérifier que c'est également bien le cas pour les données journalières : toutes les colonnes sont en valeurs manquantes sauf la première (la date) et la dernière pour le total général qui est artificiellement portée à zéro :

```
sapply(polj[c("CORTE", "DIEULEFI", "SEDAN")], \(x) all(is.na(x[,-c(1, 145)])))
```
```
  CORTE DIEULEFI    SEDAN
   TRUE     TRUE     TRUE
```

Reste les cas de `LAFLECHE` et de `SACLAYSP`. Pour `LAFLECHE` c'est un fichier vide de données et cela ne porte pas trop à conséquence. Le cas de `SACLAYSP` est plus mystérieux puisqu'il comporte des données mais n'a pas son pendant dans les données horaires.

```
sapply(polj[c("LAFLECHE", "SACLAYSP")], \(x) all(is.na(x[,-c(1, 145)])))
```
```
LAFLECHE SACLAYSP
    TRUE    FALSE
```

Les sites suivants sont présents dans les fichier `*.csv` mais pas dans les données journalières.

```
allVillesCSV[!allVillesCSV %in% names(polj)]
```
```
CAMBO-LES-BAINS           CORTE     DIEU-LE-FIT   SAINT-QUENTIN           SEDAN
     "CAMBO-LE"              NA              NA       "QUENTIN"              NA
         TAVAUX           THIZY           TORCY
       "TAVAUX"      "RHONENOR"         "TORCY"
```

On a déjà vu le cas de `CORTE`, `DIEU-LE-FIT` et `SEDAN` pour lesquels il n'y a pas de données horaires. Pour les 5 autres ils ne sont pas vides de données, on ne comprends pas bien pourquoi ils ne figurent pas dans les données journalières.

```
todo <- c("CAMBO-LES-BAINS", "SAINT-QUENTIN", "TAVAUX", "THIZY", "TORCY")
sapply(pol2h[todo], \(x) sum(!is.na(x[ , 4:15])))
```
```
CAMBO-LES-BAINS   SAINT-QUENTIN          TAVAUX           THIZY           TORCY
         171002          153000           65340           18384           48300
```

2 Importation des données

2.1 Source

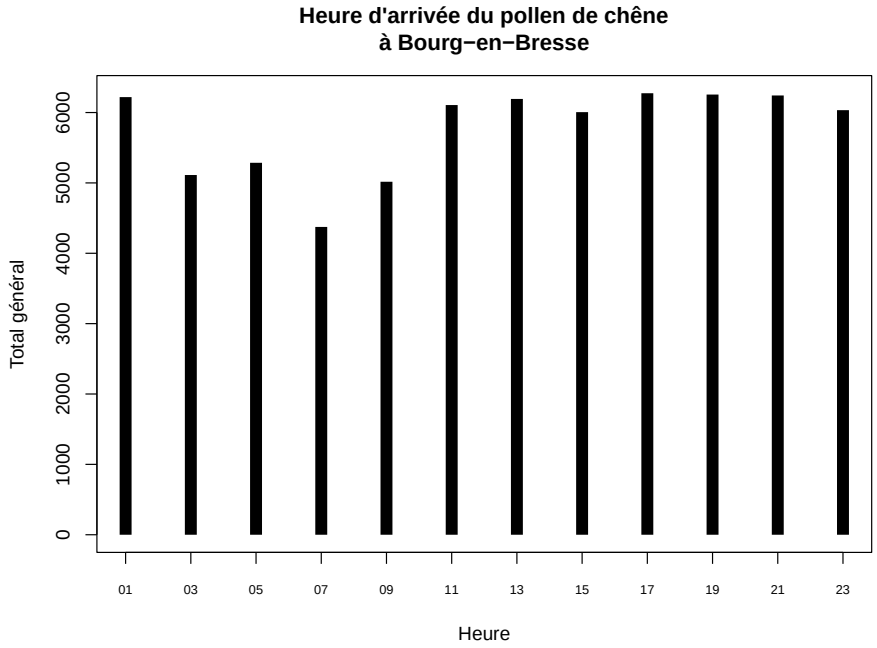
Les données sont disponibles sous licence libre[2]. Le dossier `BDD` a un pas de temps de 2 heures, le dossier `BDD_daily` a un pas de temps de 24 heures, et le fichier `documentation.xlsx` donne quelques informations. La version récupérée ici est celle du 17 juillet 2025.

2.2 Les données avec un pas de 2 heures

Les données sont regroupées par ville, il y a 133 dossiers en tout. Dans chaque dossier, les données sont regroupées par année, ce sont des fichiers de type `*.csv` avec des points-virgules comme séparateur de colonne. On définit la fonction `readVille2h()` pour importer les données d'une ville. On teste avec Bourg-en-Bresse pour représenter l'heure d'arrivée moyenne du pollen de chêne.

```
readVille2h <- function(the_ville, verbose = FALSE){
  cnames <- c("Date", "Site", "Species", "X0100", "X0300", "X0500", "X0700",
"X0900", "X1100", "X1300", "X1500", "X1700", "X1900", "X2100",
"X2300", "Total") ; nc <- length(cnames)
  dfstart <- as.data.frame(matrix(NA, nrow = 0, ncol = nc))
  colnames(dfstart) <- cnames
  fnames <- dir(paste0("data/BDD/", the_ville), full.names = TRUE)
  for(fic in fnames){
    if(verbose) print(fic)
    if(file.size(fic) == 0L) next # fichier vide
    tmp <- read.table(fic, header = TRUE, sep = ";", dec = ".")
    tmp$Analyses <- as.Date(tmp$Analyses, format = "%d/%m/%Y")
    names(tmp)[1:3] <- c("Date", "Site", "Species")
    dfstart <- rbind(dfstart, tmp)
  }
  dfstart$Species <- as.factor(dfstart$Species)
  return(dfstart)
}
testBB <- readVille2h("BOURG-EN-BRESSE")
x <- seq(1, 23, by = 2)
y <- colSums(subset(testBB, Species == "QUERCUS")[ ,4:15], na.rm = TRUE)
par(lend = "butt")
plot(x, y, type = "h", lwd = 10, axes = F, xlab = "Heure",
     ylab = "Total général", ylim = c(0, max(y)),
     main = "Heure d'arrivée du pollen de chêne\nà Bourg-en-Bresse")
axis(1, at = x, labels = sprintf("%02d", x), cex.axis = 0.7) ; axis(2) ; box()
```

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



La boucle suivante a été utilisée pour lire toutes les données de toutes les villes et les ranger dans la liste de tableaux `pol2h`.

```
allVilles <- dir("data/BDD/") ; nv <- length(allVilles)
pol2h <- vector(mode = "list", length = nv)
names(pol2h) <- allVilles
for(i in seq_len(nv)){
  print(allVilles[i])
  pol2h[[i]] <- readVille2h(allVilles[i])
}
comment(pol2h) <- paste("Généré le :", Sys.time())
save(pol2h, file = "pol2h.Rda")
```

2.3 Les données avec un pas de 24 heures

Les données sont dans `data/BDD_daily` mais dans un format différent : du `*.xsl` jusqu'en 2023 et du `*.xslx` en 2024. Les fichiers sont tous à plat dans le dossier sans structuration par ville. On définit une petit fonction utilitaire utilitaire `f2v()` pour extraire le nom des villes des noms des fichiers `*.xls[x]`.

```
allfnames <- dir("data/BDD_daily/")
head(allfnames)
```

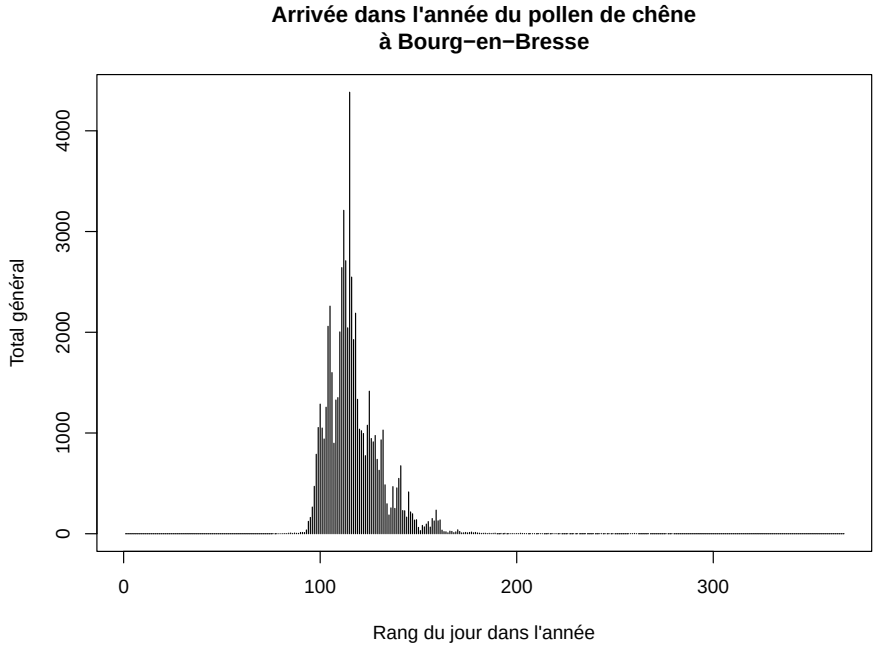
```
[1] "Particle_Extract_AGEN_2003-01-01_to_2003-12-31.xls"
[2] "Particle_Extract_AGEN_2004-01-01_to_2004-12-31.xls"
[3] "Particle_Extract_AGEN_2005-01-01_to_2005-12-31.xls"
[4] "Particle_Extract_AGEN_2006-01-01_to_2006-12-31.xls"
[5] "Particle_Extract_AGEN_2007-01-01_to_2007-12-31.xls"
[6] "Particle_Extract_AGEN_2008-01-01_to_2008-12-31.xls"
```

```
f2v <- function(fname){
  require(seqinr)
  res <- substr(fname, 18, 255)
  istop <- which(s2c(res) == "_")[1]
  res <- substr(res, 1, istop - 1)
  return(res)
}
allVilles24h <- unique(sapply(allfnames, f2v))
head(allVilles24h)
```

```
[1] "AGEN"     "AIXENPRO" "AJACCIO"  "ALES"     "AMBERIEU" "AMIENS"
```

On définit la fonction `readVille24h()` pour lire les données quotidiennes d'une ville. On la teste pour représenter l'arrivée annuelle du pollen de chêne à Bourg-en-Bresse.

```
library(readxl)
cnames <- colnames(read_excel("data/BDD_daily/Particle_Extract_AGEN_2003-01-01_to_2003-12-31.xls"))
nc <- length(cnames) ; cnames[1] <- "Date"
dfstart <- as.data.frame(matrix(NA, nrow = 0, ncol = nc))
colnames(dfstart) <- cnames
readVille24h <- function(the_ville, verbose = FALSE){
  allfnames <- dir("data/BDD_daily/", full.names = TRUE)
  ific <- grep(paste0(the_ville, "_"), allfnames)
  for(fic in allfnames[ific]){
    if(verbose) print(fic)
    if(file.size(fic) == 0L) next
    tmpj <- read_excel(fic)
    tmpj <- as.data.frame(tmpj)
    tmpj[ , 1] <- as.Date(tmpj[ , 1])
    names(tmpj)[1] <- "Date"
    dfstart <- rbind(dfstart, tmpj)
  }
  return(dfstart)
}
testBB24h <- readVille24h("BOURGENB")
x <- 1:366 ; library(lubridate)
y <- with(testBB24h, tapply(QUERCUS, yday(Date), sum, na.rm = TRUE))
plot(x, y, type = "h", xlab = "Rang du jour dans l'année",
  ylab = "Total général",
  main = "Arrivée dans l'année du pollen de chêne\nà Bourg-en-Bresse")
```



La boucle suivante a été utilisée pour lire toutes les données de toutes les villes et les ranger dans la liste de tableaux `pol2j`.

```
nv24h <- length(allVilles24h)
polj <- vector(mode = "list", length = nv24h)
names(polj) <- allVilles24h
for(i in seq_len(nv24h)){
```

```
  print(allVilles24h[i])
  polj[[i]] <- readVille24h(allVilles24h[i])
}
comment(polj) <- paste("Généré le : ", Sys.time())
save(polj, file = "polj.Rda")
```