

Fiche TD avec le logiciel  : CroissCirc

---

## Visualisation de la croissance horaire de chênes

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

---

L'objectif est de faire une animation graphique pour suivre la croissance horaire de la surface basale du tronc d'une population de 10 chênes. On veut représenter simultanément des variables météorologiques, à savoir le déficit en pression de vapeur et les précipitations.

### Contents

<b>1</b>	<b>Origine et nature des données</b>	<b>2</b>
<b>2</b>	<b>Traitement des données</b>	<b>8</b>
<b>3</b>	<b>Annexes</b>	<b>10</b>
	<b>Références</b>	<b>12</b>

# 1 Origine et nature des données

## 1.1 Les données de croissance

### 1.1.1 Dendromètres électroniques

LES données ont déjà été décrites<sup>1</sup>. Elles sont issues du suivi avec des microdendromètres de 10 chênes sessiles, *Quercus petraea*, de l'unité territoriale du Saulnois (57590) dans le département de la Moselle en France (site CHS57A) en 2021. Les variations relatives des périmètres sont exprimés en mm avec une précision de la mesure qui est de l'ordre du  $\mu\text{m}$ . Comme ce sont des données relatives on les forcera à 0 en début de suivi. Les instants du suivi horaire sont représentés par des objets de la classe `POSIXct` dans le fuseau horaire UTC+2 correspondant à l'heure d'été légale en France métropolitaine. On ne considérera que les données de la saison de végétation<sup>2</sup>, soit 186 jours du 9 avril au 12 octobre 2021.

```
chmin <- "https://pbil.univ-lyon1.fr/R/donnees/"
load(url(paste0(chmin, "microdendroCHS57/microd.Rda")))
# date de début du débourrement
ddd <- ISOdatetime(year = 2021, month = 4, day = 9,
                  hour = 0, min = 0, sec = 0,
                  tz = "Etc/GMT-2")
# fin saison végétative
fsv <- ddd
library(lubridate)
yday(fsv) <- yday(fsv) + 186
microdV <- subset(microd, temps >= ddd & temps < fsv)
# Pour partir de 0
for(j in seq_len(10)) microdV[, j] <- microdV[, j] - microdV[1, j]
# Les noms des arbres dans l'ordre de microdV
nomsa <- substr(colnames(microdV)[1:10], 8, 10)
```

QUAND on s'intéresse à la croissance des arbres on préfère raisonner en terme d'accroissement de la surface basale (BAI en anglais pour *Basal Surface Increment*). En effet, si on assimile le tronc des arbres à un cylindre de révolution ayant pour base un cercle de rayon  $r$ , sa surface,  $S$ , est donnée par  $S(r) = \pi r^2$  et sa circonférence,  $c$ , par  $c(r) = 2\pi r$ . On peut donc exprimer la surface en fonction de la circonférence comme  $S(c) = \frac{c^2}{4\pi}$ . En dérivant on voit qu'un (petit) incrément de la surface,  $dS$ , et un (petit) incrément du périmètre,  $dc$ , sont liés par la relation  $2\pi dS = cdc$ . Dès lors, l'impact d'une variation de circonférence sur la variation de la surface basale croît linéairement avec la circonférence (et donc le diamètre) des arbres. Par exemple, pour un arbre de 20 cm de diamètre une augmentation de 0.1 cm de son périmètre correspond à une augmentation de 1 cm<sup>2</sup> de sa surface basale tandis que cette même augmentation de 0.1 cm du périmètre se traduira par une augmentation de 5 cm<sup>2</sup> de la surface basale d'un arbre de 100 cm de diamètre. On voit tout l'intérêt de travailler avec les surfaces basales, mais le problème c'est que les microdendromètres ne donnent que les variations relatives des périmètres, on ne connaît pas leur valeur dans l'absolu, on ne peut donc pas calculer les surfaces basales.

<sup>1</sup>Voir la fiche « [m]anipulation de données temporelles appliquée au suivi horaire de la croissance de dix chênes pendant un an » à <https://pbil.univ-lyon1.fr/R/pdf/microdendroCHS57.pdf>

<sup>2</sup>En moyenne pour ce site, voir les figures 3 et 4 de [1], mais il devrait être possible d'être plus précis avec les données phénologiques de l'année 2021 pour ce site.

### 1.1.2 Dendromètres mécaniques

HEUREUSEMENT, dans le cas de ce site, on dispose des données de dendromètres classiques relevées par Gilles SINICCO *toutes les semaines* entre 2015 et 2021<sup>3</sup>. On a donc une très bonne idée du diamètre des troncs des arbres au début de la période qui nous intéresse. On renseigne le vecteur `dia0` avec les diamètres initiaux des arbres, exprimés ici en cm, en respectant l'ordre des colonnes du tableau `microdV`.

```
load(url(paste0(chmin, "dendroCHS57/diacm.Rda")))
# on ne garde que l'année 2021
diacm <- subset(diacm, year(Date) == 2021)
diacm$week <- week(diacm$Date)
microdV$week <- week(microdV$temps) # Début semaine 15
dia0 <- numeric(10) ; names(dia0) <- nomsa
diacm0 <- subset(diacm, week == microdV[1, "week"])
for(i in 1:10){
  ii <- which(substr(names(diacm0), 7, 9) == nomsa[i])
  dia0[i] <- diacm0[1, ii]
}
```

ON peut maintenant calculer pour chaque arbre et chaque heure la surface basale exprimée en  $\text{cm}^2$  :

```
baiV <- microdV
for(j in 1:10) baiV[, j] <- pi*(microdV[, j]/(10*pi) + dia0[j])^2/4
```

MAIS comme finalement ce qui nous intéresse c'est l'accroissement de la surface basale (BAI), on fait tout partir de zéro :

```
for(j in 1:10) baiV[, j] <- baiV[, j] - baiV[1, j]
```

POUR lisser le signal, je vais considérer la somme des variations relatives des surfaces basales des 10 chênes. C'est une sorte de moyenne, sauf que je donne plus de poids aux plus grands chênes. Ce qui justifie de faire cette opération c'est l'incroyable synchronisme de la réponse des 10 chênes<sup>4</sup>.

```
baiV$popcm2 <- apply(baiV[, 1:10], 1, sum)
```

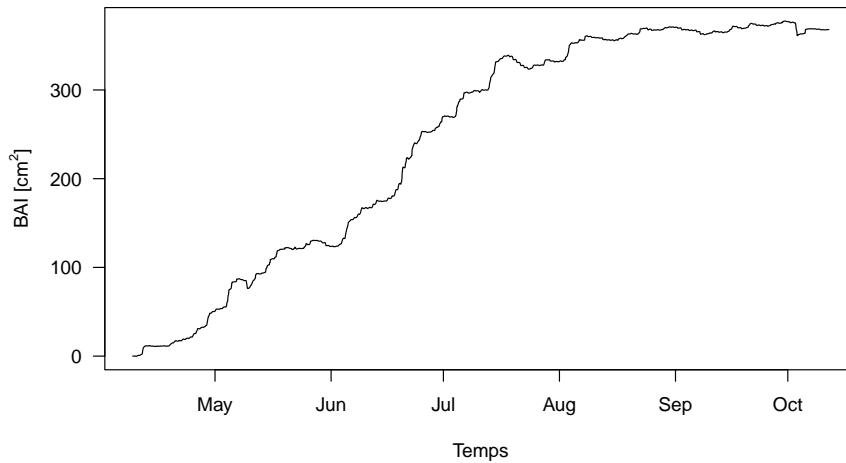
LES données de croissance que l'on veut illustrer sont données dans la figure ci-après. On note que la croissance est régulière avec parfois quelques creux. À cette échelle on ne voit pas bien les fluctuations journalières.

```
par(mar = c(5, 5, 4, 2) + 0.1)
plot(baiV$temps, baiV$popcm2, type = "l", las = 1,
      xlab = "Temps", ylab = expression(paste("BAI [", cm^2, "]")),
      main = "Croissance de 10 chênes CHS57 en 2021")
```

<sup>3</sup>Voir la fiche « [m]anipulation de données calendaires appliquée au suivi hebdomadaire de la croissance de dix chênes pendant sept ans » à <https://pbil.univ-lyon1.fr/R/pdf/dendroCHS57.pdf>

<sup>4</sup>Voir par exemple la figure page 14 de <https://pbil.univ-lyon1.fr/R/pdf/microdendroCHS57.pdf>

### Croissance de 10 chênes CHS57 en 2021



LES 10 chênes ont gagnés (au total) de l'ordre de 400 cm<sup>2</sup> de surface basale au cours de la saison, soit un carré de 20 cm de coté, ou encore à peu près la surface d'une feuille au format A4. Si on compte des troncs de 20 m et une densité du bois de 0.8, cela fait 640 kg, soit de l'ordre d'une tonne. Ce qui serait intéressant c'est d'exprimer les choses en termes de tonnes de gaz carbonique fixés par hectare de forêt.

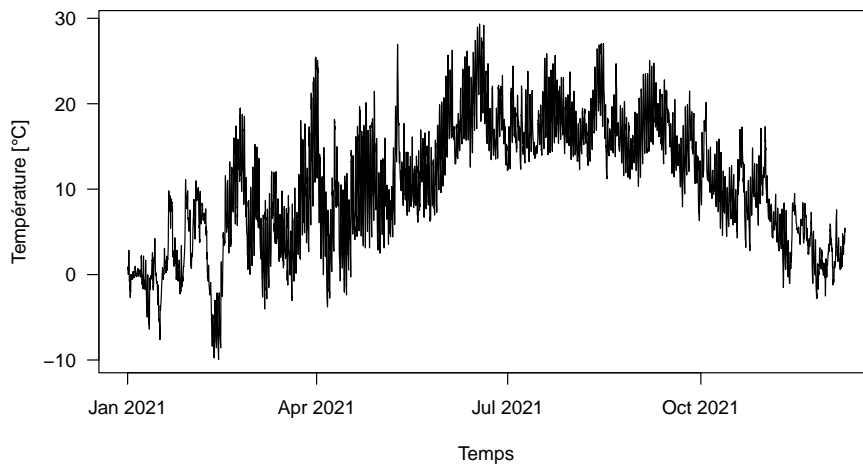
## 1.2 Les données de température

POUR les données de températures je vais utiliser celles mesurées *in situ* par les micro-dendromètres. Comme il y a 10 arbres équipés dans ce site, je vais prendre la médiane des 10 températures relevées. Ce sont des données horaires dans le fuseau UTC+2 correspondant à l'heure d'été légale en France métropolitaine.

```
load(url(paste0(chmin, "microdendroCHS57/meteo.Rda")))
# Calcul température médiane
meteo$T <- apply(meteo[, 1:10], 1, median)
save(meteo, file = "CroissCirc/meteo.Rda")

load(url(paste0(chmin, "CroissCirc/meteo.Rda")))
plot(meteo$temps, meteo$T, type = "l", las = 1,
      xlab = "Temps", ylab = "Température [°C]",
      main = "Les données de température\ndes micro-dendromètres")
```

### Les données de température des micro-dendromètres



### 1.3 Les données HOBO ?

LE site CHS57 est équipé d'un capteur HOBO qui permet d'enregistrer la température et l'humidité relative toutes les heures, mais il n'a pas encore été relevé. Les données seront en principe disponibles en janvier 2024.

### 1.4 Les données d'humidité relative

POUR les données sur l'humidité relative j'ai utilisé le paquet `riem` [4] qui permet de récupérer les données météorologiques de l'archive de l'IEM (*Iowa Environment Mesonet*<sup>5</sup>). Ce sont les archives des bulletins METAR des aéroports radio-diffusés à l'attention des pilotes des avions<sup>6</sup>. J'ai récupéré les données de l'aéroport de NANCY-ESSEY pour l'année 2021 ainsi :

```
library(riem)
stations <- riem_stations(network = "FR_ASOS")
# LFSN -> Nancy-Éssey
Nancy2021 <- riem_measures(
  station = "LFSN",
  date_start = "2021-01-01",
  date_end = "2021-12-31")
Nancy2021 <- as.data.frame(Nancy2021)
comment(Nancy2021) <- paste("Téléchargé le :", Sys.time())
save(Nancy2021, file = "CroissCirc/Nancy2021.Rda")
```

ON sélectionne les données qui nous intéressent, à savoir l'instant de l'observation valid en UTC que l'on convertit en UTC+2 et l'humidité relative `relh` en pourcentage. On note au passage que les données sur les précipitations ne sont pas renseignées.

```
load(url(paste0(chmin, "CroissCirc/Nancy2021.Rda")))
comment(Nancy2021)
```

<sup>5</sup><https://mesonet.agron.iastate.edu/>

<sup>6</sup>Voir la fiche « [l]es données météorologiques des aéroports » à <https://pbil.univ-lyon1.fr/R/pdf/METAR.pdf>

```
[1] "Téléchargé le : 2023-12-22 09:50:20.37088"
# Pas de données sur les précipitations
range(Nancy2021$p01i)
[1] 0 0
Nancy2021 <- Nancy2021[, c("valid", "relh")]
Nancy2021$valid <- with_tz(Nancy2021$valid, tzzone = "Etc/GMT-2")
head(Nancy2021$valid)
[1] "2021-01-01 02:00:00 +02" "2021-01-01 02:30:00 +02" "2021-01-01 03:00:00 +02"
[4] "2021-01-01 03:30:00 +02" "2021-01-01 04:00:00 +02" "2021-01-01 04:30:00 +02"
any(is.na(Nancy2021$relh)) # Pas de NA mais attention :
[1] FALSE
table(diff(Nancy2021$valid))
  -30   30   60   90  120  660
    1 17432    2    2    2    1
```

LES données sont en général renseignées toutes les 30 minutes, mais on note quelques trous. Il va falloir les boucher. On commence par définir une variable qualitative indicatrice des heures possibles :

```
Nancy2021$fachour <-
  with(Nancy2021, paste(format(valid, "%Y"),
                        format(valid, "%j"),
                        format(valid, "%H"),
                        sep = "-"))
Nancy2021$fachour <- as.factor(Nancy2021$fachour)
head(Nancy2021$fachour)
[1] 2021-001-02 2021-001-02 2021-001-03 2021-001-03 2021-001-04 2021-001-04
8722 Levels: 2021-001-02 2021-001-03 2021-001-04 2021-001-05 ... 2021-365-01
```

ON peut calculer maintenant l'humidité relative moyenne pour chaque heure. On constate qu'il y a 4 trous de 2 heures et 1 trou de 11 heures.

```
moy <- function(x){
  x <- x[!is.na(x)]
  if(length(x) == 0L) return(NA) # au lieu de NaN avec mean()
  return(mean(x))
}
relh.h <- tapply(Nancy2021$relh, Nancy2021$fachour, moy)
Nancy2021.h <- as.data.frame(list(
  valid = as.POSIXct(names(relh.h), tz = "Etc/GMT-2",
                     format = "%Y-%j-%H"),
  relh = relh.h))
table(diff(Nancy2021.h$valid))
  1   2  11
8716 4   1
```

POUR boucher les trous on va commencer par les faire apparaître de façon explicite. Le principe est de générer toutes les heures possibles, il nous suffit alors de faire une jointure à gauche pour les insérer (avec des NA pour les valeurs des variables) dans la table.

```
allh <- seq.POSIXt(min(Nancy2021.h$valid), max(Nancy2021.h$valid), by = "hour")
Nancy2021.h <- merge(data.frame(list(valid = allh)), Nancy2021.h,
                    by.x = "valid", by.y = "valid", all.x = TRUE)
table(diff(Nancy2021.h$valid))
  1
8735
```

ON impute les valeurs manquantes par simple interpolation linéaire grâce au paquet `imputeTS` [2].

```
library(imputeTS)
Nancy2021.h$relh <- na_interpolation(Nancy2021.h$relh)
any(is.na(Nancy2021.h$relh))
[1] FALSE
```

## 1.5 Les données de précipitation

COMME on a pu le constater, les données sur les précipitations ne sont pas disponibles dans les archives de l'IEM. Je me suis basé sur les données synoptiques de Météo-France<sup>7</sup>. J'ai récupéré les données de la base aérienne 133 de NANCY-OCHEY pour l'année 2021.

```
load("../synop/data/synop2.Rda")
pluie <- subset(synop, year == 2021 & Nom == "NANCY-OCHEY")
rm(synop)
# Les variables qui m'intéressent
target <- c("Date", "Température", "Humidité",
            "Précipitations.dans.les.3.dernières.heures",
            "valid", "year", "doy", "doyf")
pluie <- pluie[, target]
# Dans l'ordre chronologique
pluie <- pluie[order(pluie$doyf), ]
# Passage en heure d'été
pluie$valid <- with_tz(pluie$valid, tzzone = "Etc/GMT-2")
# Conversion Kelvin -> degrés Celcius
pluie$T <- as.numeric(pluie$Température) - 273.15
# Vide -> NA
pluie$Précipitations.dans.les.3.dernières.heures <-
  as.numeric(pluie$Précipitations.dans.les.3.dernières.heures)
pluie[which(pluie[, 1:4] == "", arr.ind = TRUE)] <- NA
# Time stamp
comment(pluie) <- paste("Généré le", Sys.time())
save(pluie, file = "CroissCirc/pluie.Rda")
```

LES données sont renseignées toutes les 3 heures avec quelques trous. Il devrait être possible en principe de reconstituer les données horaires puisque les précipitations des 3, 2 et 1 dernières heures sont renseignées.

```
load(url(paste0(chmin, "CroissCirc/pluie.Rda")))
table(diff(pluie$valid))
  3   6   9  12  15  27
2853 19   2   2   1   1
```

JE me suis contenté d'utiliser les précipitations des trois dernières heures et j'ai fait une interpolation linéaire pour boucher les trous. J'ai aussi forcé les valeurs négatives (-0.1 mm) à 0.

```
# Pas de valeurs négatives
pluie$Précipitations.dans.les.3.dernières.heures[
  pluie$Précipitations.dans.les.3.dernières.heures < 0] <- 0
# Jointure à gauche
allh <- seq.POSIXt(min(pluie$valid), max(pluie$valid), by = "hour")
pluie <- merge(data.frame(list(valid = allh)), pluie,
               by.x = "valid", by.y = "valid", all.x = TRUE)
# Interpolation
pluie$Précipitations.dans.les.3.dernières.heures <-
  na_interpolation(pluie$Précipitations.dans.les.3.dernières.heures)
```

## 1.6 Les données pollen ?

JE n'ai pas les données sur la concentration aérienne en pollen du réseau RNSA. Si on pouvait les avoir avec une résolution temporelle infra-journalière ça pourrait être intéressant de les faire figurer. On a aussi les données sur l'orientation du vent avec les données METAR.

<sup>7</sup>Le fichier `donnees-synop-essentielles-omm.csv` disponible à [https://donneespubliques.meteofrance.fr/?fond=produit&id\\_produit=90&id\\_rubrique=32](https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=90&id_rubrique=32)

## 2 Traitement des données

### 2.1 Calcul du VPD


LE point de départ est celui des équations d'un article [5] qui cherche à modéliser la dispersion des grains de pollen de chêne. Le VPD pour *Vapor Pressure Deficit*, soit le déficit de pression de vapeur, est donné par l'équation A1 page 192 de [5]. C'est simplement l'écart entre la pression à saturation et la pression courante. Plus il est grand, plus il est facile de se refroidir en transpirant.

```
# Ta is the temperature in Celcius
esat <- function(Ta){
  ifelse(Ta < 0,
    6.1078*exp(22.44294*Ta/(272.44 + Ta)),
    6.1078*exp(17.08085*Ta/(234.175 + Ta)))
}
# current water vapor pressure (ecurr, hPa):
ecurr <- function(Ta, RH) esat(Ta)*RH/100
# Vapor Pressure Deficit (hPa)
VPD <- function(Ta, RH) esat(Ta) - ecurr(Ta, RH)
```

POUR le calculer j'ai besoin de connaître la température et l'humidité relative. Je fais donc une jointure pour compléter ma table `meteo`. J'en profite pour rapatrier aussi les données sur les précipitations.

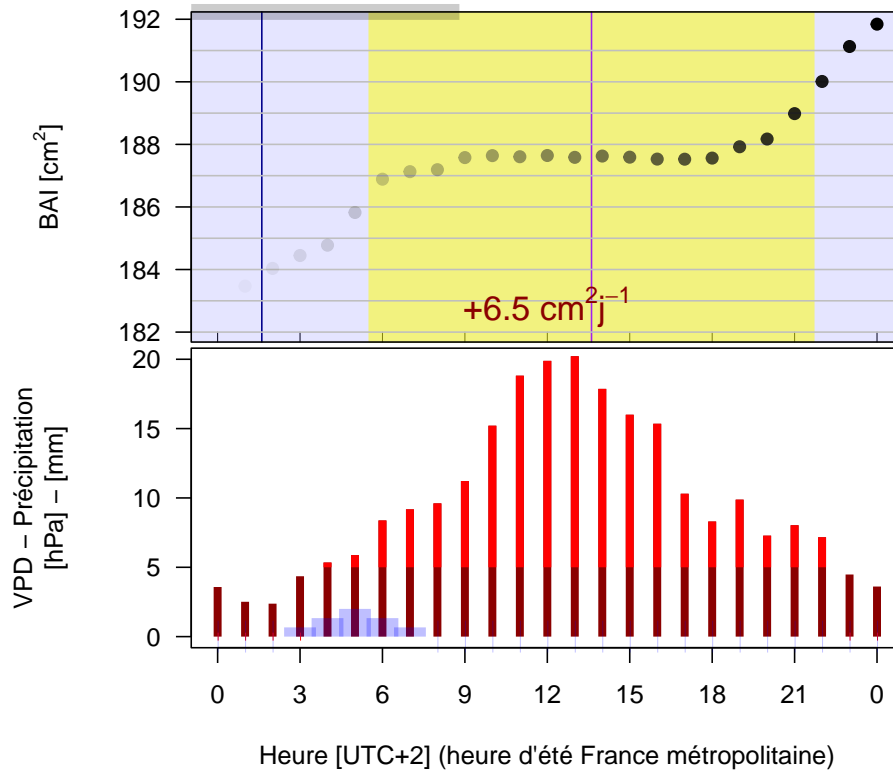
```
meteo <- merge(meteo, Nancy2021.h,
              by.x = "temps", by.y = "valid")
meteo$VPD <- VPD(meteo$T, meteo$relh)
Pstr <- "Précipitations.dans.les.3.dernières.heures"
meteo <- merge(meteo, pluie[, c("valid", Pstr)],
              by.x = "temps", by.y = "valid")
```

### 2.2 Représentation graphique

LE code  produisant le graphique étant assez long, on le trouvera en annexe section 3.2 page 11. L'objectif de la fonction `plot24h()` est de faire une représentation graphique couvrant une période de 24 heures à partir d'une heure donnée :



### Croissance de 10 chênes CHS57A le 2021-06-18 Vendredi 18 juin , rang du jour : 169



Le graphique du haut représente l'évolution de la surface basale au cours du temps. La principale difficulté est que l'échelle en ordonnée ne va pas arrêter de bouger au cours du temps, ce qui peut être trompeur si on n'y prend garde. Pour pallier cet inconvénient j'ai utilisé plusieurs artifices. Il y a tout d'abord les barres de progression en gris (en haut à gauche et en bas à droite) qui permettent de savoir où l'on se situe par rapport à l'ensemble de la série. Il y a ensuite les graduations horizontales tous les  $\text{cm}^2$  qui, quand elles se densifient, indiquent que l'on est dans une période de forte variation. Enfin, la vitesse moyenne journalière de variation de la surface basale est indiquée dans une police d'autant plus grande qu'elle est élevée (en valeur absolue). Les couleurs en arrière plan indiquent s'il fait jour, en jaune, ou s'il fait nuit, en bleu. Les heures du lever et du coucher du soleil ont été calculées avec le paquet `suncalc` [6]. La barre verticale pourpre indique le midi solaire (au voisinage de 14h00 puisque nous sommes en UTC+2) et la barre verticale bleu le nadir (minuit solaire, au voisinage de 2 heures du matin.)

Le graphique du bas représente l'évolution du VPD (voir section 2.1 page 8) et des précipitations au cours du temps. Le déficit en pression de vapeur, VPD, est en rouge. C'est un indicateur de la sécheresse de l'air ambiant, plus il est élevé plus l'air est sec. J'ai utilisé un rouge plus vif au delà de 5 hPa, à titre de comparaison dans [5] on estime qu'il n'y a pas d'émission pollinique en dessous de 1.273 hPa. Les précipitations sont en bleu, exprimées en mm de

pluie au cours des trois dernières heures. Contrairement au graphique du haut, l'échelle en ordonnée est imposée et commune à tous les graphiques.

## 2.3 Animations

### 2.3.1 Faible résolution

GRÂCE au paquet `gifski` [3] il est facile de produire une image animée. Je commence par une petite représentation à faible résolution sur 10 jours pour avoir une image pas trop lourde sur le site<sup>8</sup>.

```
starts <- microdV$temps[2000:2240] ; ns <- length(starts)
for(i in seq_len(ns)){
  ichar <- sprintf("%06d", i)
  png(paste0("low/image", ichar, ".png"), width = 7, height = 4, units = "in", res = 36)
  par(cex = 0.5)
  plot24h(start = starts[i])
  dev.off()
}
library(gifski)
png_files <- list.files(path = "low/", pattern = ".png", full.names = TRUE)
gifski::gifski(png_files, gif_file = "low.gif",
              delay = 0.2)
```

### 2.3.2 Haute résolution

JE fais ensuite une représentation à plus haute résolution pour l'ensemble des données. Pour visualiser le résultat il faut ouvrir avec votre navigateur le lien : <https://pbil.univ-lyon1.fr/R/donnees/CroissCirc/high.gif>.

```
starts <- microdV$temps
starts <- starts[1:(length(starts) - 24)] # pas le dernier jour
ns <- length(starts)
for(i in seq_len(ns)){
  ichar <- sprintf("%06d", i)
  png(paste0("high/image", ichar, ".png"), width = 7, height = 6, units = "in", res = 72)
  plot24h(start = starts[i])
  dev.off()
}
library(gifski)
png_files <- list.files(path = "high/", pattern = ".png", full.names = TRUE)
gifski::gifski(png_files, gif_file = "high.gif",
              delay = 0.2)
```

## 3 Annexes

### 3.1 Remerciements

UN GRAND MERCI à Nicolas DELPIERRE, maître de conférences universitaire au laboratoire « Écologie Systématique Évolution » de l'université Paris-Saclay, et à Gilles SINICCO, technicien forestier territorial de l'ONF de l'unité territoriale du Saulnois (57590) dans le département de la Moselle, d'avoir accepté de partager leur données.

<sup>8</sup>[https://pbil.univ-lyon1.fr/R/enseignement\\_div.php?contents=html/quercus](https://pbil.univ-lyon1.fr/R/enseignement_div.php?contents=html/quercus)

## 3.2 Code source du graphique

```
library(suncalc) # pour calculer le lever et coucher du soleil
Sys.setlocale("LC_ALL", "fr_FR.UTF-8")
plot24h <- function(start = "2021-04-09 00:00:00"){
  start <- as.POSIXct(start)
  ii <- which(microdV$temps == start)
  iseq <- ii:(ii + 24) # un jour
  x <- baiV[iseq, "temps"]
  y <- baiV[iseq, "popcm2"]
  par(mfrow = c(2, 1))
  # Graphique de croissance
  main <- paste("Croissance de 10 chênes CHS57A le",
               format(microdV[ii[1], "temps"], "%F"), "\n",
               format(microdV[ii[1], "temps"], "%A %d %B"),
               ", rang du jour :", yday(microdV[ii[1], "temps"]))
  par(mar = c(0, 6, 4, 0.5) + 0.1)
  plot.new() ; plot.window(xlim = range(x), ylim = range(y))
  pu <- par("usr")
  title(main = main)
  axis(2, las = 1)
  ylab <- expression(paste("BAI [", cm^2, "]"))
  title(ylab = ylab, line = 4, xpd = NA)
  synop <- seq(0, 24, by = 3)
  ix <- which(hour(x) %in% synop)
  rug(x[ix], xpd = NA)

  # Fond d'écran avec nuit et jours
  solcol <- rgb(1, 1, 0, 0.5)
  nuitcol <- rgb(0, 0, 1, 0.1)
  # Nuit par défaut
  rect(pu[1], pu[3], pu[2], pu[4], col = nuitcol, border = nuitcol)
  # Ajour du soleil
  addsoleil <- function(jour){
    soleil <- getSunlightTimes(as.Date(jour),
                               lat = 48 + 52/60 + 18/3600,
                               lon = 6 + 29/60 + 02/3600,
                               tz = "Etc/GMT-2")[c("sunrise", "sunset", "solarNoon", "nadir")]
    t1 <- soleil[,1] # sunrise
    t2 <- soleil[,2] # sunset
    t3 <- soleil[,3] # midi solaire
    t4 <- soleil[,4] # minuit solaire
    rect(t1, pu[3], t2, pu[4], col = solcol, border = solcol)
    abline(v = t3, col = "purple")
    abline(v = t4, col = "darkblue")
  }
  addsoleil(x[1]) ; addsoleil(x[length(x)])
  # Grille tous les cm2 pour faciliter la lecture
  grille <- seq(floor(min(y)), max(y), by = 1)
  abline(h = grille, col = grey(0.75))
  box()
  # Données de croissance
  gseq <- seq(0.25, 0, le = 25)
  colgrey <- rgb(gseq, gseq, gseq, seq(0, 1, le = 25))
  points(x, y, pch = 19, type = "b", col = colgrey)
  # Barres de progression
  pcy <- max(y)/max(baiV[, "popcm2"])
  par(lend = "butt")
  colbar <- rgb(0, 0, 0, 0.2)
  segments(pu[2], pu[3], pu[3] + pcy*(pu[4] - pu[3]),
           lwd = 10, xpd = NA, col = colbar)
  pcx <- ii/nrow(baiV)
  segments(pu[1], pu[4], pu[1] + pcx*(pu[2] - pu[1]), pu[4],
           lwd = 10, xpd = NA, col = colbar)
  # Calcul de la vitesse en cm2/j
  jseq <- (0:24)/24
  slope <- lm(y-jseq)$coef[2]
  # je veux le signe et un chiffre après la virgule
  slopech <- sprintf("%.1f ", slope)
  # 'gade comme c'est beau comme ça
  txt <- bquote(paste(.(slopech), cm^2*j^-1))
  text((pu[1] + pu[2])/2, pu[3], txt, pos = 3,
       col = ifelse(slope > 0, "darkred", "darkblue"),
       cex = abs(slope)/5)
  # abline(lm(y-x), lty = 3) bof bof
```

```
# Graphique météo
ixm <- which(meteo$temps %in% x)
ym <- meteo[ixm, "VPD"]
xm <- meteo[ixm, "temps"]
par(mar = c(5, 6, 0, 0.5) + 0.1)
plot.new() ; plot.window(xlim = range(x), ylim = c(0, 20))
axis(2, las = 1)
title(ylab = "VPD - Précipitation\n[hPa] - [mm]", line = 4)
axis(1, at = x[ix], label = hour(x[ix]))
title(xlab = "Heure [UTC+2] (heure d'été France métropolitaine)")

points(xm, ym, type = "h", lwd = 5, col = "darkred",
       xpd = NA)
segments(xm, ifelse(ym < 5, 0, 5), xm,
         ifelse(ym < 5, 0, ym), col = "red", lwd = 5, xpd = NA)
# Précipitations mm
yp <- meteo[ixm, Pstr]
col <- rgb(0, 0, 1, 0.25)
points(xm, yp, type = "h", lwd = 20, col = col,
       xpd = NA)
box()

# Lien vers PDF explicatif
title(sub = "https://pbil.univ-lyon1.fr/R/pdf/CroissCirc.pdf", cex.sub = 0.5)
}
plot24h(start = "2021-06-18 00:00:00")
```

## References

- [1] F. Lebourgeois, J. Differt, A. Granier, N. Bréda, and E. Ulrich. Premières observations phénologiques des peuplements du réseau national de suivi à long terme des écosystèmes forestiers (RENECOFOR). *Revue forestière française*, 54(5):407–418, 2002.
- [2] S. Moritz and T. Bartz-Beielstein. imputeTS: Time Series Missing Value Imputation in R. *The R Journal*, 9(1):207–218, 2017.
- [3] Jeroen Ooms. *gifski: Highest Quality GIF Encoder*, 2023. R package version 1.12.0-1.
- [4] M. Salmon. *riem: Accesses Weather Data from the Iowa Environment Mesonet*, 2022. R package version 0.3.0.
- [5] S. Schueler and K.H. Schlünzen. Modeling of oak pollen dispersal on the landscape level with a mesoscale atmospheric model. *Environmental Modelling & Assessment*, 11(3):179–194, 2006.
- [6] B. Thieurmél and A. Elmarhraoui. *suncalc: Compute Sun Position, Sunlight Phases, Moon Position and Lunar Phase*, 2022. R package version 0.5.1.