

Manipulation de données calendaires appliquée
au suivi hebdomadaire de la croissance de dix
chênes pendant sept ans

P^r Jean R. LOBRY

Table des matières

1	Des données utilisant le format Date	2
1.1	Le format Date	2
1.2	Les données	2
1.3	Calcul des surfaces basales	5
2	Tendance générale	7
2.1	Résumé linéaire	7
2.2	Écarts au modèle linéaire	8
2.3	Pentes des droites	9
3	Composante saisonnière	10
3.1	Représentation au cours de l'année	10
3.2	Résumé logistique	12
4	Utilisation des vitesses instantannées	23
4.1	Calcul des vitesses instantannées	23
4.2	Statistiques circulaires	27
4.3	Contraction des troncs par grand froid	28
5	Animation graphique	30
6	Annexes	32
6.1	Remerciements	32
6.2	Importation au format Date	33
6.3	Code 	34
	Références	36

1 Des données utilisant le format Date

1.1 Le format Date

DANS cette fiche on cherche à illustrer toute la puissance qui est à notre disposition dans `R` quand les données temporelles sont au format `Date`. Les données calendaires manipulées ici sont *déjà* dans ce format, mais on montrera en annexe page 32 comment les obtenir au moment de l'importation. Voyons maintenant un exemple concret :

```
exempleDate <- as.Date("2022-07-01")
class(exempleDate)
[1] "Date"
```

NOUS avons utilisé la fonction `as.Date()` pour convertir une chaîne de caractères au format ISO 8601, très utilisé parce que l'ordre lexicographique y est le même que l'ordre chronologique, en un objet de la classe `Date`. Si on demande à afficher la valeur de notre objet on retrouve le format ISO 8601 :

```
exempleDate
[1] "2022-07-01"
```

POUR voir comment les dates sont codées en interne on utilise la fonction `dput()` qui permet d'avoir une représentation en texte ASCII d'un objet :

```
dput(exempleDate)
structure(19174, class = "Date")
```

LES dates au format `Date` sont codées en interne par le nombre de jours¹ écoulés depuis le 1^{er} janvier 1970, le temps zéro d'UNIX. Au 1^{er} juillet 2022 il y avait donc 19174 jours écoulés depuis cette date.

1.2 Les données

LES données ont été collectées avec des dendromètres (cf. figure 1 page 3) par Gilles SINCCO, technicien forestier territorial de l'ONF, entre 2015 et 2021 sur le site CHS57A (donc majoritairement des chênes sessiles *Quercus petraea*) de l'unité territoriale du Saulnois (57590) dans le département de la Moselle en France. Les mesures ont été faites, à dix exceptions près, tous les samedis :

```
load(url("https://pbil.univ-lyon1.fr/R/donnees/dendroCHS57/diacm.Rda"))
class(diacm$Date)
[1] "Date"
sort(table(format(diacm$Date, "%a")))
Thu Fri Sun Sat
 1   4   5 355
```

À partir d'un objet de la classe `Date`, la fonction `format()` avec l'argument `%a` permet donc de récupérer très facilement le nom abrégé du jour de la semaine. La liste des spécificateurs de format disponibles est détaillée dans la table 1 page 6. Si on veut le nom du jour en entier c'est l'argument `%A` qu'il faut utiliser :

¹Si pour une série temporelle on a besoin d'une meilleure résolution que le jour, il faudra utiliser des objets de la classe abstraite `POSIXt` dont dérivent les classes `POSIXct` (codage du temps en secondes depuis le 1^{er} janvier 1970) et `POSIXlt` (codage du temps sous la forme d'une liste d'éléments censés être plus lisibles par un être humain).



```
sort(table(format(diacm$Date, "%A")))
Thursday Friday Sunday Saturday
1         4         5        355
```

UNE autre solution possible, sans doute plus facile à mémoriser et plus lisible, est d'utiliser la fonction `weekdays()`, mais elle ne fait rien d'autre que d'appeler la fonction `format()` :

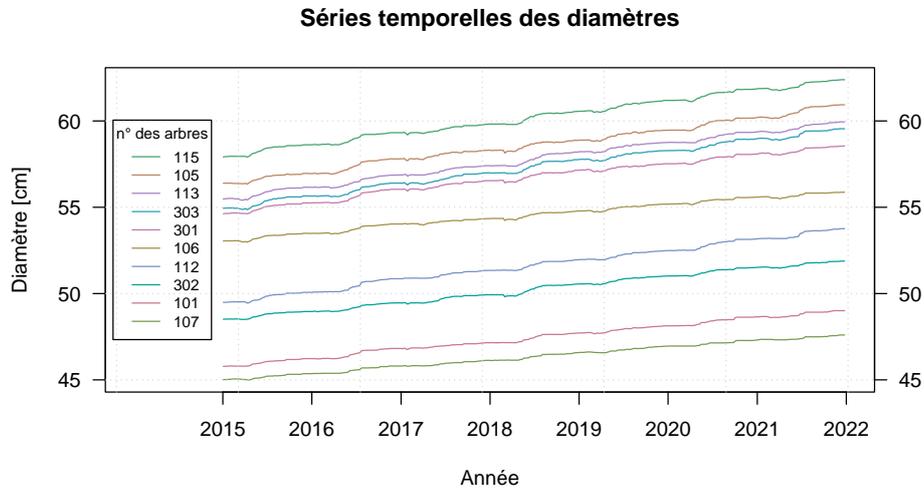
```
sort(table(weekdays(diacm$Date)))
Thursday Friday Sunday Saturday
1         4         5        355
```

LES diamètres des 10 chênes suivis, exprimés en centimètres avec une précision d'un dixième de millimètre, sont portés en colonne ici, par exemple pour les cinq premiers arbres :

```
head(diacm[, 1:5])
  CHS57.101 CHS57.105 CHS57.106 CHS57.107 CHS57.115
1    45.78    56.39    53.04    45.01    57.92
2    45.79    56.40    53.05    45.02    57.93
3    45.79    56.39    53.06    45.02    57.94
4    45.80    56.40    53.05    45.03    57.94
5    45.80    56.40    53.06    45.03    57.95
6    45.81    56.39    53.05    45.05    57.96
```

LE code ci-dessous est un exemple de graphique produit avec en abscisse une variable de la classe `Date`, donc typiquement une série temporelle. Nous utilisons la fonction `as.Date()` pour modifier les limites temporelles, `xlim`, pour commencer en 2014 de façon à gagner de la place pour afficher la légende. Nous l'utilisons également pour générer la série temporelle `xseq` de 2015 à 2022 avec un pas d'un an pour afficher les années où il y a des données. Nous avons besoin également de la fonction `format()` avec l'option `%Y` pour récupérer l'année à partir d'un objet de la classe `Date`.

```
xlim <- as.Date(c("2014-01-01", "2021-12-31"))
plot.new() ; plot.window(xlim = xlim, ylim = range(diacm[, 1:10]))
title(xlab = "Année", main = "Séries temporelles des diamètres",
      ylab = "Diamètre [cm]")
mycol <- hcl.colors(10, "Dark 2")
for(j in seq_len(10)) lines(diacm$Date, diacm[, j], col = mycol[j])
xseq <- seq(as.Date("2015-01-01"), as.Date("2022-01-01"), by = "year")
axis(2, las = 1) ; axis(4, las = 1) ; box()
axis(1, at = xseq, labels = format(xseq, "%Y"))
noms <- sapply(colnames(diacm)[1:10], \(x) unlist(strsplit(x, split = "\\."))[2])
myo <- rev(order(colMeans(diacm[, 1:10])))
grid()
legend("left", inset = 0.01, legend = noms[myo], lty = 1, col = mycol[myo],
      cex = 0.75, title = "n° des arbres")
```



ON voit ici la lente et régulière croissance du diamètre des troncs des dix chênes : au cours des sept années de suivi un gain de 2-3 cm est observé. Les courbes ne sont pas parfaitement linéaires à cause des fluctuations saisonnières, on y reviendra dans la section 3 page 10.

1.3 Calcul des surfaces basales

SI on suppose, est c'est assez raisonnable, que le tronc des arbres est un cylindre de révolution alors la surface de sa base, S , en fonction du diamètre, d , est donnée par :

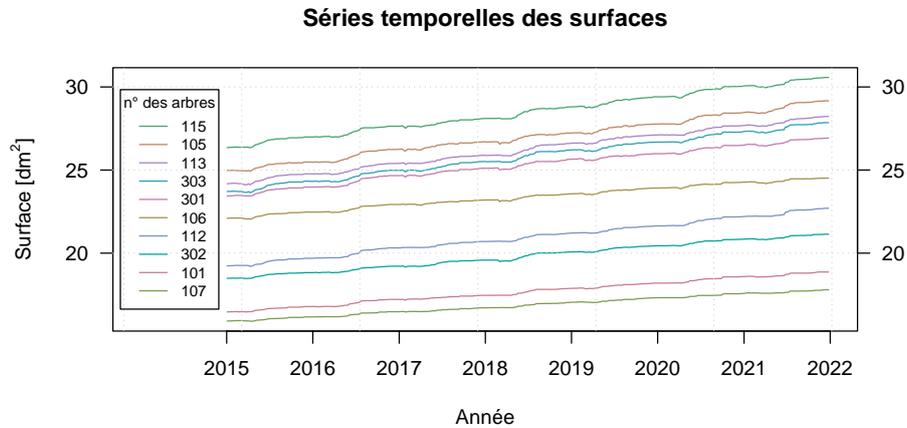
$$S(d) = \pi \frac{d^2}{4}$$

```

surf <- diacm
surf[ , 1:10] <- apply(surf[ , 1:10], 2, \ (x) pi*x^2/400) # en dm^2
par(mar = c(5, 5, 4, 2) + 0.1)
plot(surf$Date, surf[,1], type = "n", ylim = range(surf[ , 1:10]),
     las = 1, xlab = "Année", main = "Séries temporelles des surfaces",
     ylab = expression(paste("Surface [", dm^2, "]")), las = 1, xaxt = "n", xlim = xlim)
mycol <- hcl.colors(10, "Dark 2")
for(j in seq_len(10)) lines(surf$Date, surf[ , j], col = mycol[j])
axis(1, at = xseq, labels = format(xseq, "%Y")); axis(4, las = 1)
grid()
legend("left", inset = 0.01, legend = noms[myo], lty = 1, col = mycol[myo],
      cex = 0.75, title = "n° des arbres")
    
```

Format	Signification	Exemple 2022-07-01
%a	Nom abrégé du jour de la semaine dans le système d'exploitation local	Fri
%A	Nom complet du jour de la semaine dans le système d'exploitation local	Friday
%b	Nom abrégé du mois dans le système d'exploitation local	Jul
%B	Nom complet du mois dans le système d'exploitation local	July
%c	Date et heure dans le format standard du système d'exploitation local	Fri Jul 1 00:00:00 2022
%C	« Siècle » : la partie entière de l'année divisée par 100	20
%d	Le jour du mois compris entre 01 et 31	01
%D	La date dans un format du type %m%d%y : pour le standard C99 c'est le format exact, mais il n'est pas honoré par tous les systèmes d'exploitation	07/01/22
%e	Le jour du mois compris entre 1 et 31, avec une espace en amont quand il n'y a qu'un seul chiffre	→ 1←
%F	Équivalent à %Y-%m-%d, c'est à dire le format ISO 8601 pour les dates	2022-07-01
%g	Les deux derniers chiffres de l'année basée sur les semaines (voir %V)	22
%h	Équivalent à %b	Jul
%j	Rang du jour dans l'année compris entre 001 et 366	182
%m	Rang du mois dans l'année compris entre 01 et 12	07
%u	Rang du jour dans la semaine compris entre 1 et 7, lundi vaut 1	5
%U	Rang de la semaine dans l'année compris entre 00 et 53 en utilisant dimanche comme le jour 1 de la semaine (et typiquement comme le premier dimanche de l'année comme le jour 1 de la semaine 1). Convention locale aux USA.	26
%V	Rang de la semaine dans l'année compris entre 01 et 53 comme défini dans la norme internationale ISO 8601. Si la semaine (commençant le lundi) qui contient le 1 ^{er} janvier comporte au moins 4 autres jours dans la nouvelle année alors c'est la semaine 1. Sinon, c'est la dernière semaine de l'année précédente et c'est la semaine suivante qui est la semaine 1	26
%w	Rang du jour dans la semaine compris entre 0 et 6, dimanche vaut 0	5
%x	Date dans le format du système d'exploitation local	07/01/2022
%y	L'année sans le siècle compris entre 00 et 99	22
%Y	L'année avec le siècle. Notez que s'il n'y avait pas d'année 0 dans le calendrier grégorien originel, la norme ISO 8601 le définit comme valide (interprété comme 1 av. J.-C.). La norme précise également que les années antérieures à 1582 (début du calendrier grégorien) ne doivent être utilisées qu'avec l'accord de toutes les parties impliquées.	2022

TABLE 1 : Liste des formats de conversions utilisables avec la fonction `format()` pour les objets de la classe `Date`. Ils sont documentés dans l'aide de la fonction `strptime()`. La fonction `format()` renvoie une chaîne de caractères, pour exploiter des valeurs numériques il faudra les transtyper avec `as.numeric()` ou `as.integer()`.



POUR visualiser ce que cela représente en pratique, notons qu'un carré de 50 cm de côté a une surface de 25 dm². Au cours des 7 années de suivi, les dix chênes ont augmenté de 3-4 dm² la surface basale de leur tronc.

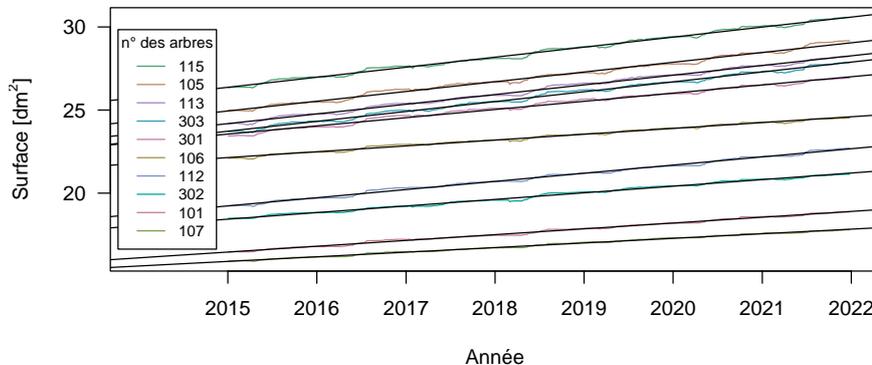
2 Tendances générales

2.1 Résumé linéaire

AU vu des séries temporelles des surfaces on imagine que la tendance générale pourra bien être résumée avec un simple modèle linéaire :

```
plot(surf$Date, surf[,1], type = "n", ylim = range(surf[, 1:10]),
     las = 1, xlab = "Année", main = "Séries temporelles des surfaces et résumé linéaire",
     ylab = expression(paste("Surface [", dm^2, "]")), las = 1, xaxt = "n", xlim = xlim)
mycol <- hcl.colors(10, "Dark 2")
for(j in seq_len(10)){
  x <- surf$Date ; y <- surf[, j]
  lines(x, y, col = mycol[j])
  abline(lm(y~as.numeric(x)))
}
axis(1, at = xseq, labels = format(xseq, "%Y"))
legend("left", inset = 0.01, legend = noms[myo], lty = 1, col = mycol[myo],
      cex = 0.75, bg = "white", title = "n° des arbres")
```

Séries temporelles des surfaces et résumé linéaire



CE qui est frappant ici c'est à quel point le résumé linéaire est bon : il y a bien des écarts mais on est quasiment dans l'épaisseur du trait. Ceci signifie que pour la période étudiée il n'y a pas eu d'années de forte ou de faible croissance², mais au contraire une croissance très régulière, et ce pour tous les arbres.

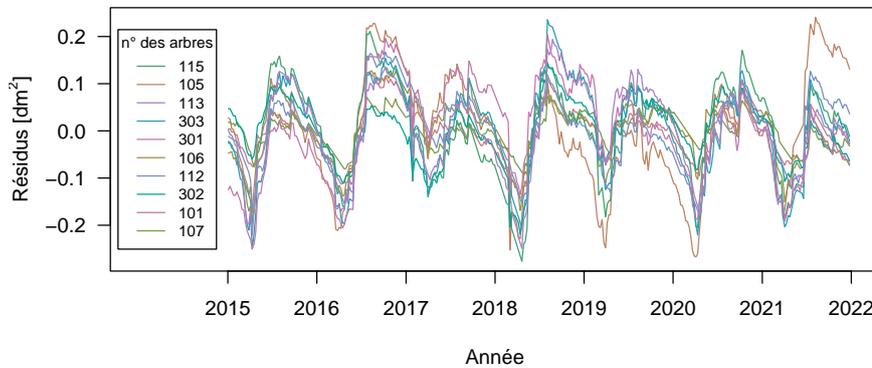
2.2 Écarts au modèle linéaire

BIEN entendu, si on regarde de plus près les résidus du modèle linéaire, on constate qu'il y a une très forte auto-corrélation temporelle d'iceux. Ceci est la conséquence de la composante saisonnière qui, dans un premier temps, ne nous intéresse pas dans cette section sur la tendance générale.

```
residuals <- surf
residuals[, 1:10] <- apply(residuals[, 1:10], 2,
                          \x) lm(x-as.numeric(residuals$Date))$residuals)
plot(residuals$Date, residuals[,1], type = "n", ylim = range(residuals[, 1:10]),
     las = 1, xlab = "Année", main = "Résidus du résumé linéaire",
     ylab = expression(paste("Résidus [", dm^2, "]")), las = 1, xaxt = "n", xlim = xlim)
for(j in seq_len(10)) lines(residuals$Date, residuals[, j], col = mycol[j])
axis(1, at = xseq, labels = format(xseq, "%Y"))
legend("left", inset = 0.01, legend = noms[myo], lty = 1, col = mycol[myo],
      cex = 0.75, title = "n° des arbres")
```

²Cette assertion n'est vraie qu'en première approximation, on verra avec une analyse plus fine dans la section 3.2.1 page 14 qu'elle doit être nuancée.

Résidus du résumé linéaire



2.3 Pentés des droites

SI on calcule les pentes des droites de régression on trouve des valeurs très petites. C'est ici qu'il faut se souvenir que les dates sont exprimées en jours et qu'il faut donc multiplier par le nombre de jours dans une année pour avoir les accroissements annuels moyens.

```
(pentes <- apply(surf[, 1:10], 2, \ (x) lm(x-as.numeric(surf$Date))$coef[2]))
CHS57.101 CHS57.105 CHS57.106 CHS57.107 CHS57.115 CHS57.302
0.0009563015 0.0016050578 0.0009603561 0.0007588403 0.0016525548 0.0010861328
CHS57.303 CHS57.112 CHS57.113 CHS57.301
0.0016269633 0.0013453974 0.0015959968 0.0013496115
(pentes <- pentes*365.25)
CHS57.101 CHS57.105 CHS57.106 CHS57.107 CHS57.115 CHS57.302 CHS57.303 CHS57.112
0.3492891 0.5862473 0.3507700 0.2771664 0.6035956 0.3967100 0.5942483 0.4914064
CHS57.113 CHS57.301
0.5829378 0.4929456
```

POUR vérifier qu'il n'y a pas d'erreur on fait un rapide calcul pour l'arbre n° 301. Son accroissement annuel moyen est de l'ordre de $0.5 \text{ dm}^2 \text{ y}^{-1}$, donc un gain de l'ordre de $7 \times 0.5 = 3.5 \text{ dm}^2$ durant les sept années de suivi. C'est bien ce qui est observé pour l'écart entre la dernière et la première valeur pour cet arbre :

```
surf[nrow(surf), 10] - surf[1, 10]
[1] 3.493122
```

SI on regarde la distribution des accroissements annuels moyens on constate qu'ils sont assez variables d'un arbre à l'autre puisque l'on a des valeurs qui peuvent varier du simple au double, avec une médiane à $0.5 \text{ dm}^2 \text{ y}^{-1}$:

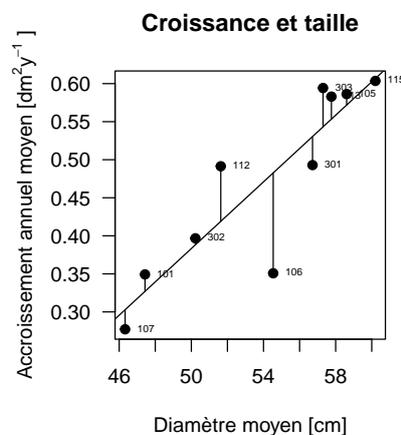
```
summary(pentes)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.2772 0.3623 0.4922 0.4725 0.5854 0.6036
```

SI on choisit comme valeur représentative un accroissement annuel moyen $0.5 \text{ dm}^2 \text{ y}^{-1}$, qu'est ce que cela représente concrètement ? Supposons que notre arbre fasse 25 m de haut, l'accroissement volumétrique annuel moyen est alors

de $0.5 \times 250 = 125 \text{ dm}^3$. En prenant une densité volumétrique de 0.8, cela correspond à un gain moyen de 100 kg par an.

La relation entre l'accroissement annuel moyen et le diamètre moyen des arbres montre que, pour des chênes de cette classe de diamètre, le gain annuel de biomasse est d'autant plus important que le diamètre est élevé :

```
d moy <- colMeans(diacm[ , 1:10]) ; par(pty = "s", mar = c(5, 5, 4, 2))
plot(d moy, pentes, pch = 19, las = 1, xlab = "Diamètre moyen [cm]",
      ylab = expression(paste("Accroissement annuel moyen [", dm^2*y^-1, " ]")),
      main = "Croissance et taille")
text(d moy, pentes, noms, pos = 4, cex = 0.5, xpd = NA)
reslm <- lm(pentes~d moy) ; abline(reslm)
segments(d moy, pentes, d moy, predict(reslm))
```



LES résidus de la droite de régression montrent que l'arbre n° 106 est moins productif que ce qui serait attendu pour son diamètre moyen alors que l'arbre n° 112 l'est plus.

3 Composante saisonnière

3.1 Représentation au cours de l'année

AVEC l'option %j de la fonction `format()` il est très facile de récupérer le rang du jour dans l'année d'une variable au format `Date`. Il est donc facile de représenter nos séries temporelles en fonction du numéro du jour de l'année pour mettre en évidence les composantes saisonnières. Le problème est que le rang du jour dans l'année n'est pas très parlant pour un être humain habitué au calendrier grégorien. Pour faciliter la lecture, nous allons enrichir nos graphiques en faisant figurer les limites des mois. Pour ce faire nous allons générer avec la fonction `sprintf()` tous les 1^{ers} du mois :

```
(pdm <- as.Date(sprintf("2022-%0.2i-01", 1:12)))
[1] "2022-01-01" "2022-02-01" "2022-03-01" "2022-04-01" "2022-05-01" "2022-06-01"
[7] "2022-07-01" "2022-08-01" "2022-09-01" "2022-10-01" "2022-11-01" "2022-12-01"
```

La fonction `format()` nous permet de récupérer en clair le nom des mois correspondants, en entier ou en abrégé :

```
format(pdm, "%B")
[1] "January" "February" "March" "April" "May" "June"
[7] "July" "August" "September" "October" "November" "December"
(mois.abb <- format(pdm, "%b"))
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

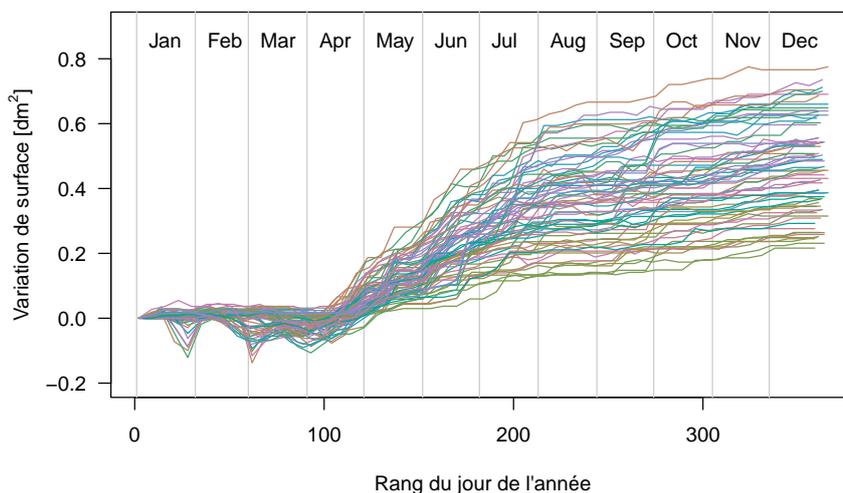
IL nous est également facile avec la fonction `format()` de calculer le rang du jour des 1^{ers} de chaque mois :

```
(imonth <- as.integer(format(pdm, "%j")))
[1] 1 32 60 91 121 152 182 213 244 274 305 335
```

NOUS avons fait ces calculs pour l'année 2022 qui n'est pas bissextile, il y aura donc un petit décalage de un jour pour les années bissextile à partir du mois de mars. On négligera cette petite erreur ici. Dans la représentation graphique suivante on translate verticalement toutes les courbes de façon à partir de zéro pour la première mesure de l'année :

```
surf$doy <- as.integer(format(surf$Date, "%j"))
surf$Year <- as.integer(format(surf$Date, "%Y"))
par(mar = c(5, 5, 4, 2) + 0.1) ; plot.new()
plot.window(xlim = range(surf$doy), ylim = c(-0.2, 0.9))
title(main = "Évolution de la surface au cours de l'année",
      ylab = expression(paste("Variation de surface [", dm^2, "]")),
      xlab = "Rang du jour de l'année")
axis(1) ; axis(2, las = 1) ; box()
abline(v = imonth, col = grey(0.8))
text(imonth, 0.9*par("usr")[4], mois.abb, pos = 4)
for(the_year in unique(surf$Year)){
  tmp <- surf[surf$Year == the_year, ]
  for(j in seq_len(10)) lines(tmp$doy, tmp[,j] - tmp[1, j], col = mycol[j])
}
```

Évolution de la surface au cours de l'année

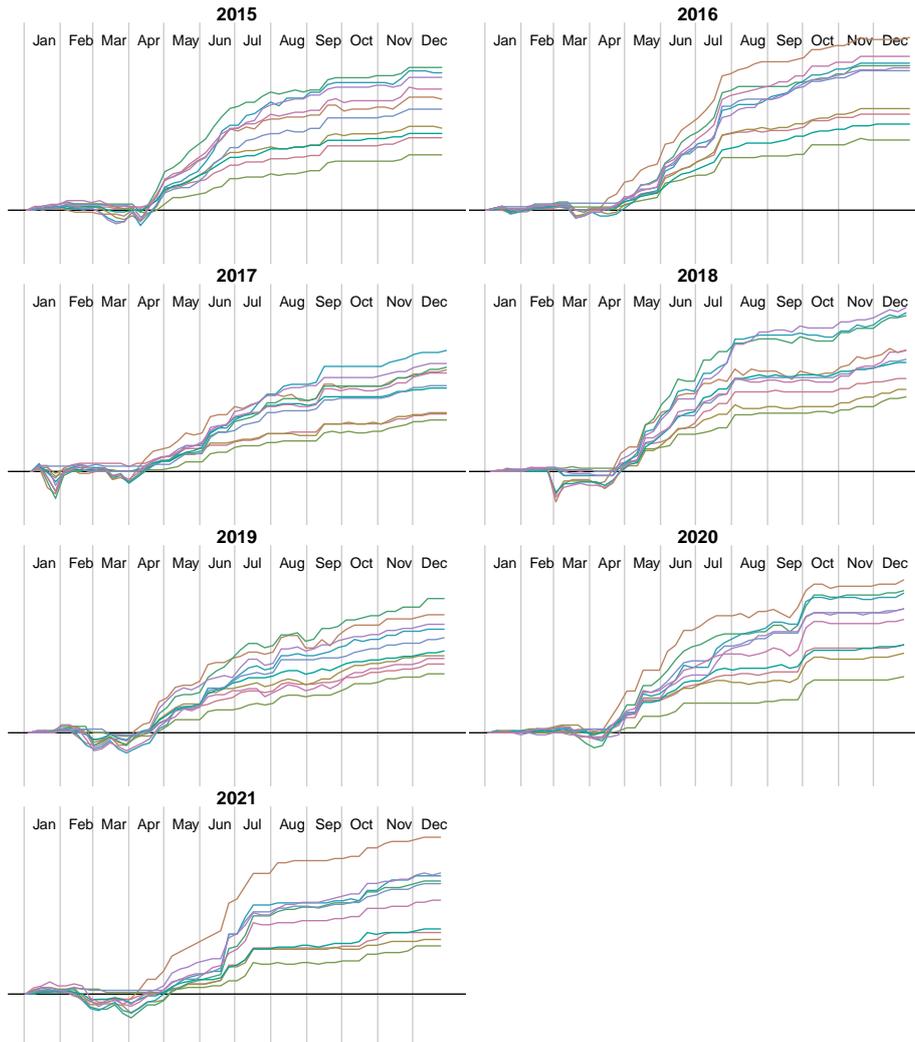


ON visualise très bien la composante saisonnière ici. La croissance ne commence pas avant le mois d'avril chaque année et n'est plus très importante au dernier trimestre. Pour y voir plus clair on peut représenter les courbes de croissance année par année :

```

par(mar = c(0, 0, 1, 0) + 0.1, mfrow = c(4, 2))
xlim <- range(surf$doy) ; ylim <- c(-0.2, 0.8)
for(the_year in unique(surf$Year)){
  tmp <- surf[surf$Year == the_year, ]
  plot.new() ; plot.window(xlim, ylim) ; title(main = the_year)
  abline(v = imonth, col = grey(0.8))
  abline(h = 0)
  text(imonth, 0.9*par("usr")[4], mois.abb, pos = 4)
  for(j in seq_len(10)) lines(tmp$doy, tmp[,j] - tmp[1, j], col = mycol[j])
}

```



ON voit qu'il peut y avoir des phases de contraction en début d'année, par exemple fin janvier en 2017. La croissance n'est pas toujours régulière, il peut y avoir des phases d'accélération, par exemple fin septembre-début octobre en 2020. On voit également que la croissance des arbres est assez synchrone.

3.2 Résumé logistique

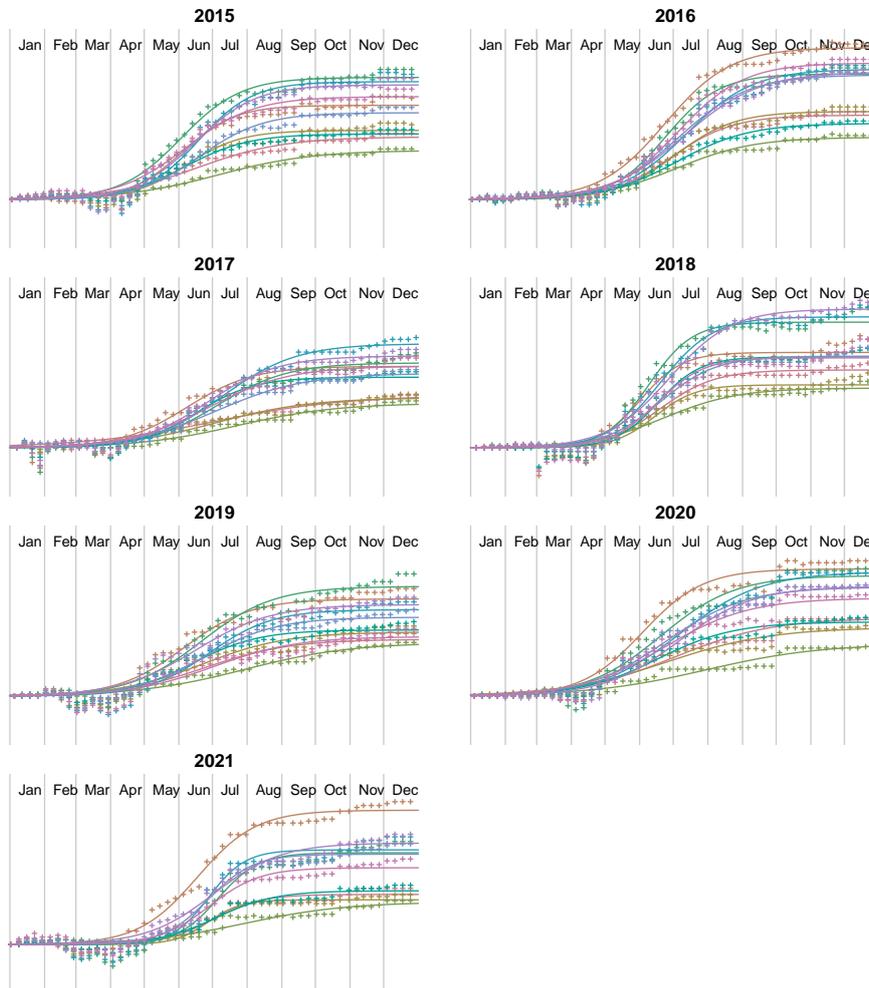
ON utilise maintenant le modèle de croissance logistique de VERHULST [3, 4, 5, 6] pour lisser les séries temporelles d'accroissement de la surface basale.

$$y = f(x) = \frac{y_{\infty}}{1 + e^{\frac{x_{0.5} - x}{\sigma}}} \quad (1)$$

La signification des paramètres est la suivante. y_{∞} (**Asym**) représente la valeur de l'asymptote horizontale à droite, obtenue pour les très grandes valeurs de la variable de la fonction. $x_{0.5}$ (**xmid**) est un paramètre de position qui donne la valeur du point d'inflexion. La valeur de la courbe au point d'inflexion vaut la moitié de la valeur de l'asymptote horizontale à droite. s (**scale**) contrôle la rapidité de la transition, plus il est petit, plus elle sera brutale. La figure suivante représente l'ajustement du modèle logistique aux données :

```
par(mar = c(0, 1, 1.5, 0) + 0.1, mfrow = c(4, 2), oma = c(0, 0, 3, 0))
diacm$Year <- format(diacm$Date, "%Y")
tablog <- as.data.frame(matrix(NA, ncol = 6, nrow = 70))
colnames(tablog) <- c("Tree_name", "Year", "Asym", "xmid", "scal", "diacm")
surf$doy <- as.integer(format(surf$Date, "%j"))
surf$Year <- as.integer(format(surf$Date, "%Y"))
ii <- 1
for(the_year in unique(surf$Year)){
  tmp <- surf[surf$Year == the_year, ]
  plot.new() ; plot.window(xlim, ylim) ; title(main = the_year)
  abline(v = imonth, col = grey(0.8))
  text(imonth, 0.9*par("usr")[4], mois.abb, pos = 4)
  for(j in seq_len(10)){
    tablog[ii, "Tree_name"] <- colnames(surf)[j]
    tablog[ii, "Year"] <- the_year
    tmp$y <- tmp[,j] - tmp[1, j]
    points(tmp$doy, tmp$y, col = mycol[j], pch = 3, cex = 0.5)
    resnls <- nls(y ~ SSlogis(doy, Asym, xmid, scal), data = tmp)
    temps <- 1:366
    lines(temps, predict(resnls, list(doy = temps)), col = mycol[j])
    tablog[ii, 3:5] <- coef(resnls)
    tablog[ii, "diacm"] <- mean(diacm[diacm$Year == the_year, j])
    ii <- ii + 1
  }
}
title(main = "Ajustement du modèle logistique", outer = TRUE,
      cex.main = 2)
```

Ajustement du modèle logistique



NOTEZ que comme les valeurs prédites par le modèle logistique sont strictement positives, nous gommons automatiquement ici les événements de décroissance parfois observés en début d'année.

3.2.1 Croissance annuelle

L'ASYMPTOTE du modèle logistique représente la croissance annuelle de la surface basale. On retrouve pour les arbres ce que l'on avait déjà vu section 2 page 7 avec par exemple l'arbre n° 107 qui est le moins productif, mais cette fois on visualise aussi la variabilité inter-annuelle de la croissance des arbres. Pour les années il semble il y avoir une alternance d'années de faible et de forte croissance.

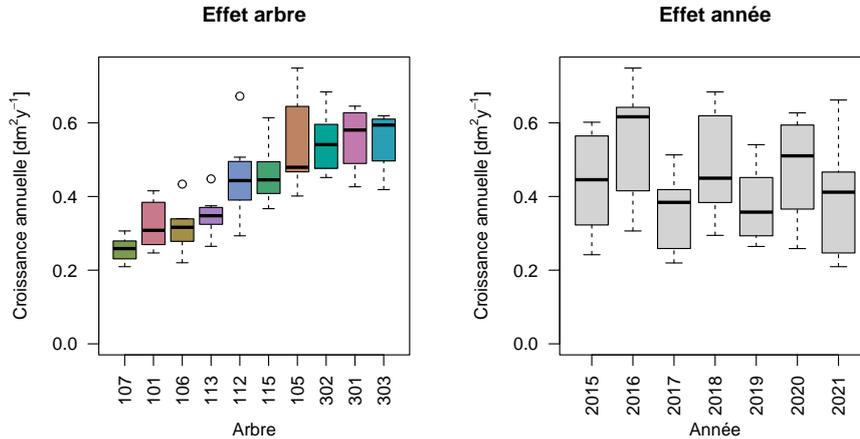
```
years <- unique(tablog$Year)
par(mfrow = c(1, 2), mar = c(5, 5, 4, 2), oma = c(0, 0, 1, 0))
ylim <- c(0, max(tablog$Asym))
ylab <- expression(paste("Croissance annuelle [", dm^2*y^-1, "]"))
myo <- rank(tapply(tablog$Asym, tablog$Tree_name, median))
boxplot(Asym~Tree_name, tablog, names = noms, las = 2, xlab = "Arbre",
        ylab = ylab, main = "Effet arbre", col = mycol, ylim = ylim,
```

```

at = myo)
boxplot(Asym~Year, tablog, names = years, las = 2, xlab = "Année",
        ylab = ylab, main = "Effet année", ylim = ylim)
title(main = "Asymptote du modèle logistique", outer = TRUE)

```

Asymptote du modèle logistique



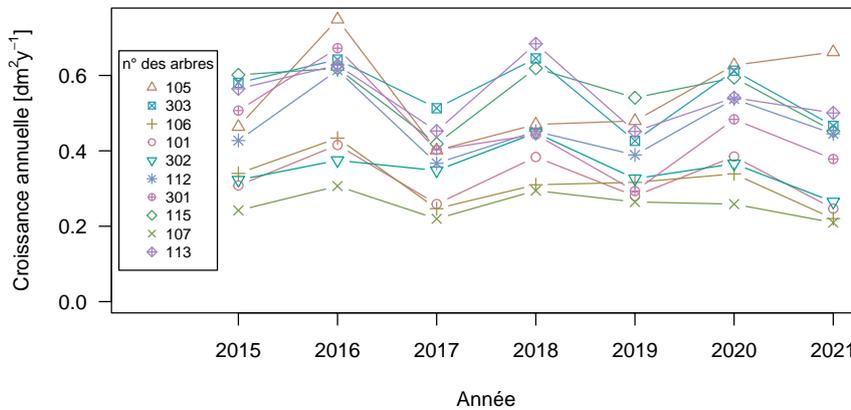
POUR confirmer l'alternance d'années plus ou moins fastes pour la croissance de la surface basale on décide de faire une représentation graphique qui mette en évidence le comportement de chaque arbre :

```

xseq <- 2014:2021
par(mar = c(5, 5, 4, 2), oma = c(0, 0, 1, 0))
plot.new() ; plot.window(xlim = range(xseq), ylim = c(0, max(tablog$Asym)))
title(xlab = "Année", main = "Croissance annuelle de la surface basale", ylab = ylab)
trees <- unique(tablog$Tree_name) ; nt <- length(trees)
for(j in seq_len(nt)) {
  tmp <- tablog[tablog$Tree_name == trees[j], ]
  lines(tmp$Year, tmp$Asym, col = mycol[j], pch = j, type = "b")
}
axis(2, las = 1) ; box()
axis(1, at = xseq[-1], labels = xseq[-1])
legend("left", inset = 0.01, legend = noms[myo], col = mycol[myo],
      cex = 0.75, title = "n° des arbres", pch = (1:10)[myo])

```

Croissance annuelle de la surface basale



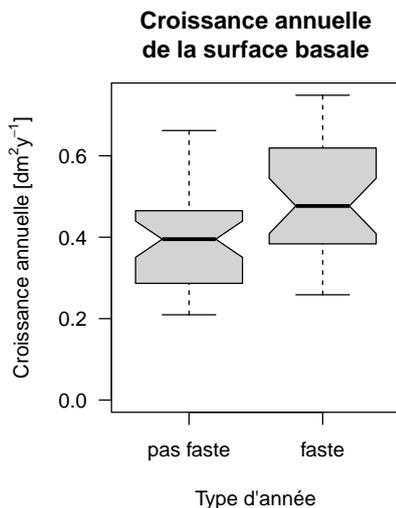
Il a donc bien globalement une alternance d'années plus ou moins fastes pour la croissance de la surface basale des arbres mais on note des exceptions individuelles comme par exemple l'arbre n° 105 qui ne fait qu'augmenter de 2017 à 2021.

La figure ci-après permet de relativiser l'importance de l'effet de l'alternance entre les années plus ou moins fastes. En terme de taille d'effet on observe une différence de l'ordre de $0.1 \text{ dm}^2\text{y}^{-1}$, ce qui pour une croissance moyenne de $0.5 \text{ dm}^2\text{y}^{-1}$ correspond à des variations de l'ordre de 20 %, ce qui est faible par rapport à la variabilité inter-annuelle de l'effort de reproduction. On note également que les intervalles de confiance pour la médiane sont chevauchants, nous sommes dans l'incapacité de rejeter l'hypothèse nulle de l'absence d'effet.

```

tablog$facfaste <- as.factor(tablog$Year)
levels(tablog$facfaste) <- c(rep(c("pas faste", "faste"), 3), "pas faste")
par(mar = c(5, 5, 4, 2), mfrow = c(1, 2))
boxplot(Asym~facfaste, tablog, notch = TRUE, ylim = c(0, max(tablog$Asym)),
        las = 1, xlab = "Type d'année", ylab = ylab,
        main = "Croissance annuelle\nde la surface basale")

```



3.2.2 Date du pic de croissance

Le paramètre de position du modèle logistique est l'abscisse du point d'inflexion, c'est-à-dire la date du pic de croissance de la surface basale. Pour tous les arbres la date du pic de croissance est en juin ou juillet. Mais, si pour la majorité des arbres le pic est fin juin, on note deux exceptions avec l'arbre n° 105 qui a un pic précoce début juin et l'arbre n° 107 qui a un pic tardif fin juillet. Pour ce qui est de l'effet année on ne note pas de structure temporelle particulière, simplement des années où, dans l'ensemble, les pics de croissance sont précoces comme en 2015 et 2018 et des années où ils sont tardifs comme en 2016 et 2017. On pourra ici faire le lien avec la représentation circulaire de la section 4.2 page 27.

```

years <- unique(tablog$Year)
par(mfrow = c(1, 2), mar = c(5, 5, 4, 2), oma = c(0, 0, 1, 0))
ylab <- "Date du pic de croissance\n[rang du jour dans l'année]"

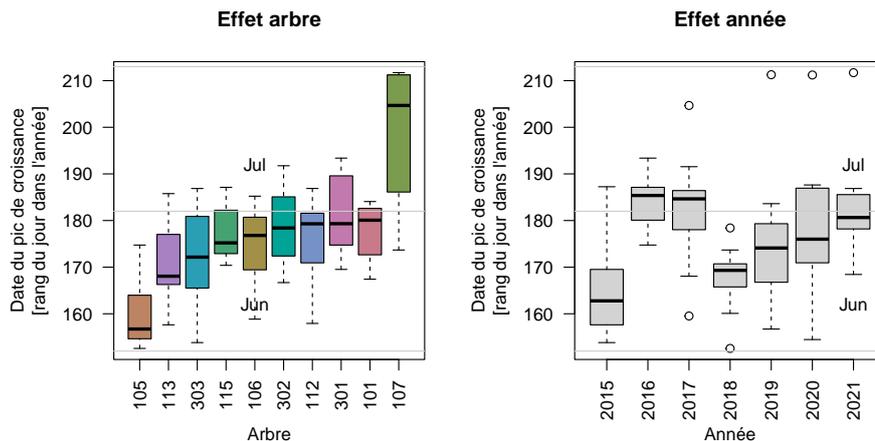
```

```

myo <- rank(tapply(tablog$xmids, tablog$Tree_name, median))
boxplot(xmid~Tree_name, tablog, names = noms, las = 2, xlab = "Arbre",
        ylab = ylab, main = "Effet arbre", col = mycol,
        at = myo)
abline(h = imonth, col = grey(0.8)) ; text(5, imonth + 10, mois.abb)
boxplot(xmid~Year, tablog, names = years, las = 2, xlab = "Année",
        ylab = ylab, main = "Effet année")
abline(h = imonth, col = grey(0.8)) ; text(7, imonth + 10, mois.abb)
title(main = "Paramètre de position du modèle logistique", outer = TRUE)

```

Paramètre de position du modèle logistique

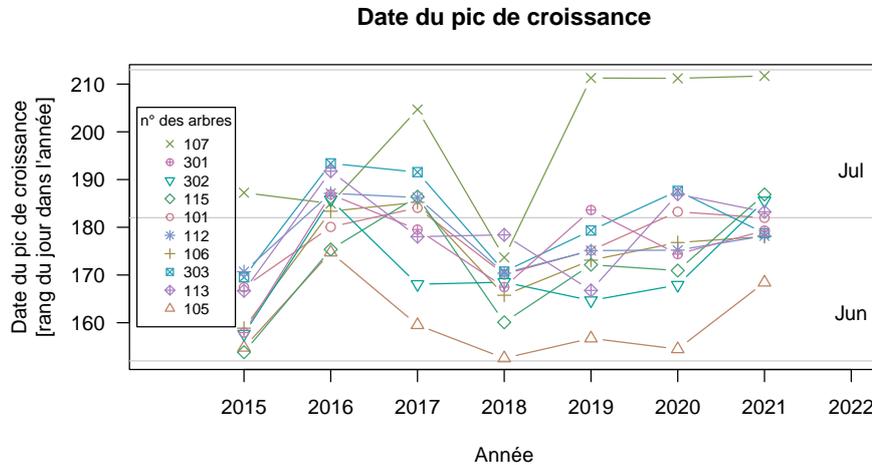


La représentation des données individuelles permet de confirmer le caractère précoce de l'arbre n° 105 et tardif de l'arbre n° 107 et de voir qu'en 2016 et 2018 ils n'étaient pas si atypiques que les autres années :

```

xseq <- 2014:2022
myopic <- rev(order(tapply(tablog$xmids, tablog$Tree_name, mean)))
par(mar = c(5, 5, 4, 2), oma = c(0, 0, 1, 0))
plot.new() ; plot.window(xlim = range(xseq), ylim = range(tablog$xmids))
title(xlab = "Année", main = "Date du pic de croissance", ylab = ylab)
axis(2, las = 1) ; box() ; axis(1, at = xseq[-1], labels = xseq[-1])
abline(h = imonth, col = grey(0.8)) ; text(2022, imonth + 10, mois.abb)
for(j in seq_len(nt)) {
  tmp <- tablog[tablog$Tree_name == trees[j], ]
  lines(tmp$Year, tmp$xmids, col = mycol[j], pch = j, type = "b")
}
legend("left", inset = 0.01, legend = noms[myopic], col = mycol[myopic],
      cex = 0.75, title = "n° des arbres", pch = (1:10)[myopic])

```

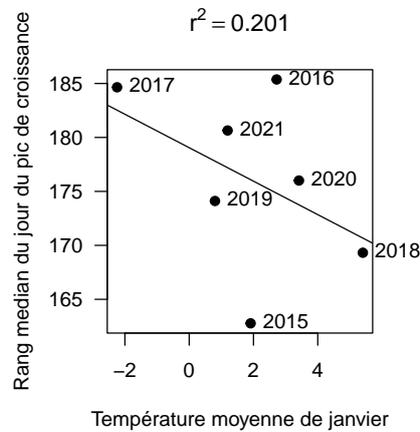


PEUT-ON faire le lien avec les conditions météo ? On commence par définir une petite fonction utilitaire qui calcule la température sur une période donnée pour une année donnée. On exprime le temps par le rang du jour de l'année, notre période est comprise entre les jours `deb` et `fin`, inclus. On l'illustre dans le cas du mois de janvier 2012 :

```
load(url("https://pbil.univ-lyon1.fr/R/donnees/dendroCHS57/met.Rda"))
futil <- fonction(the_year, deb, fin){
  qui <- with(met, which(Year == the_year & doy >= deb & doy <= fin))
  return(with(met[qui, ], mean(T)))
}
futil(2015, 1, 31)
[1] 1.911935
```

ON calcule pour chaque année dans `xmids` la médiane de la date du pic de croissance et on la confronte à la température moyenne du mois de janvier de l'année correspondante :

```
(xmids <- with(tablog, tapply(xmid, Year, median)))
      2015      2016      2017      2018      2019      2020      2021
162.7748 185.3732 184.6481 169.3159 174.1101 176.0063 180.6425
n <- length(xmids) ; tmeans <- numeric(n)
for(i in seq_len(n)) tmeans[i] <- futil(names(xmids)[i], 1, 31)
r2 <- signif(cor(tmeans, xmids)^2, 3)
main <- bquote(r^2 == .(r2))
par(pty = "s")
plot(tmeans, xmids, pch = 19, main = main, las = 1,
      xlab = "Température moyenne de janvier",
      ylab = "Rang median du jour du pic de croissance")
abline(lm(xmids~tmeans))
text(tmeans, xmids, names(xmids), pos = 4, xpd = NA)
```



IL Y A une très légère corrélation négative : les mois de janvier les plus chauds ont tendance à correspondre à des dates du pic de croissance plus précoces. Mais ce n'est pas bien convainquant. Peut-on faire mieux ? On va généraliser le procédé en calculant notre r^2 non pas sur le seul mois de janvier, mais sur une fenêtre glissante plus ou moins large.

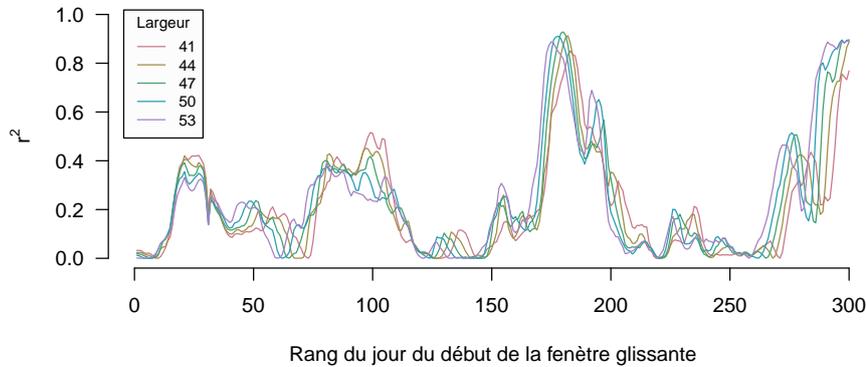
```

doymax <- 300
large <- c(41, 44, 47, 50, 53) ; ncol = length(large)
myres <- matrix(0, nrow = doymax, ncol = ncol)
for(i in seq_len(doymax)){
  for(j in seq_len(ncol)){
    for(k in seq_len(n)) tmeans[k] <- futil(names(xmids)[k], i, i + large[j] - 1)
    myres[i, j] <- cor(tmeans, xmids, use = "pairwise.complete.obs")^2
  }
}
save(myres, doymax, large, ncol, file = "myres.Rda")

load("myres.Rda")
plot.new() ; plot.window(xlim = c(1, doymax), ylim = c(0, 1))
mycol <- hcl.colors(ncol, "Dark 2")
for(j in seq_len(ncol)) points(1:doymax, myres[ , j], type = "l", col = mycol[j])
axis(1) ; axis(2, las = 1)
legend("topleft", inset = 0.02, bg = grey(0.99), legend = large,
      lty = 1, col = mycol, title = "Largeur", cex = 0.75)
title(xlab = "Rang du jour du début de la fenêtre glissante")
title(ylab = expression(r^2))
title(main = "Température moyenne et date du pic de croissance")

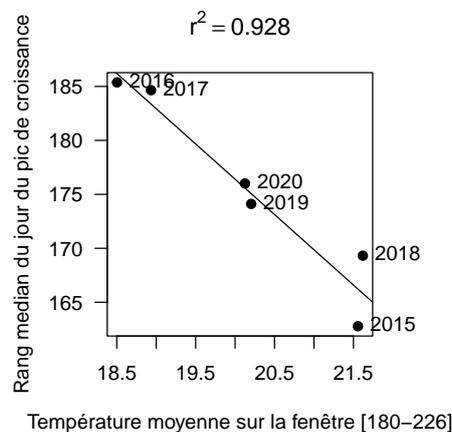
```

Température moyenne et date du pic de croissance



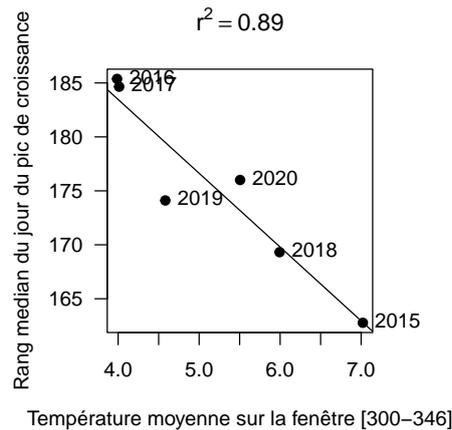
N'EST-CE PAS merveilleux ? Nous observons un pic du r^2 qui flirte avec les 0.9 pour une date qui correspond approximativement avec le pic de croissance. Mais encore faut-il vérifier que la corrélation aille dans le bon sens.

```
for(i in seq_len(n)) tmeans[i] <- futil(names(xmids)[i], 180, 180+47-1)
r2 <- signif(cor(tmeans, xmids, use = "pairwise.complete")^2, 3)
main <- bquote(r^2 == .(r2))
par(pty = "s")
plot(tmeans, xmids, pch = 19, main = main, las = 1,
      xlab = "Température moyenne sur la fenêtre [180-226]",
      ylab = "Rang median du jour du pic de croissance")
abline(lm(xmids~tmeans))
text(tmeans, xmids, names(xmids), pos = 4, xpd = NA)
```



TOUT ceci semble bel et bon : plus la température est importante sur notre fenêtre, plus la date du pic de croissance est précoce. Mais il y a un « petit » problème ici : à part pour les années 2016 et 2017, la date du pic de croissance est *antérieure* au début de la fenêtre. Nous sommes en train de vouloir expliquer le présent avec des données du futur ! Une forte corrélation n'est pas forcément synonyme d'un lien de causalité. Pour donner un exemple encore plus caricatural :

```
for(i in seq_len(n)) tmeans[i] <- futil(names(xmids)[i], 300, 300+47-1)
r2 <- signif(cor(tmeans, xmids, use = "pairwise.complete")^2, 3)
main <- bquote(r2 == .(r2))
par(pty = "s")
plot(tmeans, xmids, pch = 19, main = main, las = 1,
      xlab = "Température moyenne sur la fenêtre [300-346]",
      ylab = "Rang median du jour du pic de croissance")
abline(lm(xmids~tmeans))
text(tmeans, xmids, names(xmids), pos = 4, xpd = NA)
```



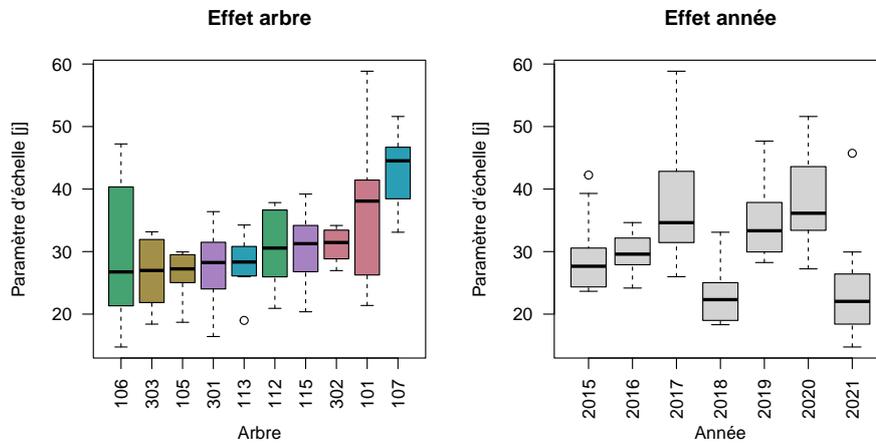
NOUS avons donc une excellente corrélation entre les dates du pic de croissance en été et les températures moyennes en novembre-décembre, alors que la croissance est très largement terminée. Il faut donc être *extrêmement* prudent dans l'interprétation de ce type d'approche : en multipliant les fenêtres on multiplie les occasions d'avoir une forte corrélation par pur hasard.

3.2.3 Durée de la période de croissance

LE paramètre d'échelle du modèle logistique contrôle la vitesse de transition entre les deux asymptote, plus il est petit plus la période de croissance est courte. C'est surtout l'arbre n° 107 qui ressort avec une période longue et les années 2018 et 2021 avec des périodes courtes.

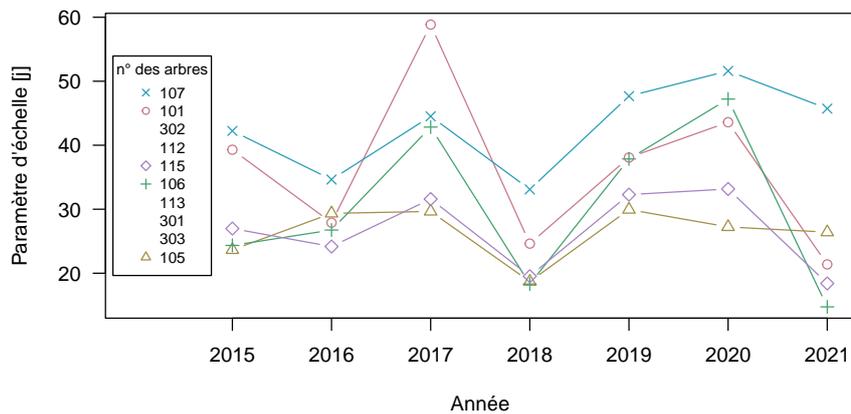
```
par(mfrow = c(1, 2), mar = c(5, 4, 4, 2), oma = c(0, 0, 1, 0))
ylab <- "Paramètre d'échelle [j]"
myo <- rank(tapply(tablog$scal, tablog$Tree_name, median))
boxplot(scal~Tree_name, tablog, names = noms, las = 2, xlab = "Arbre",
        ylab = ylab, main = "Effet arbre", col = mycol, at = myo)
boxplot(scal~Year, tablog, names = years, las = 2, xlab = "Année",
        ylab = ylab, main = "Effet année")
title(main = "Paramètre d'échelle du modèle logistique", outer = TRUE)
```

Paramètre d'échelle du modèle logistique



```
xseq <- 2014:2021
myscal <- rev(order(tapply(tablog$scal, tablog$Tree_name, mean)))
par(mar = c(5, 5, 4, 2), oma = c(0, 0, 1, 0))
plot.new() ; plot.window(xlim = range(xseq), ylim = range(tablog$scal))
title(xlab = "Année", main = "Durée de la phase de croissance", ylab = ylab)
axis(2, las = 1) ; box() ; axis(1, at = xseq[-1], labels = xseq[-1])
for(j in seq_len(nt)) {
  tmp <- tablog[tablog$Tree_name == trees[j], ]
  lines(tmp$Year, tmp$scal, col = mycol[j], pch = j, type = "b")
}
legend("left", inset = 0.01, legend = noms[myscal], col = mycol[myscal],
      cex = 0.75, title = "n° des arbres", pch = (1:10)[myscal])
```

Durée de la phase de croissance



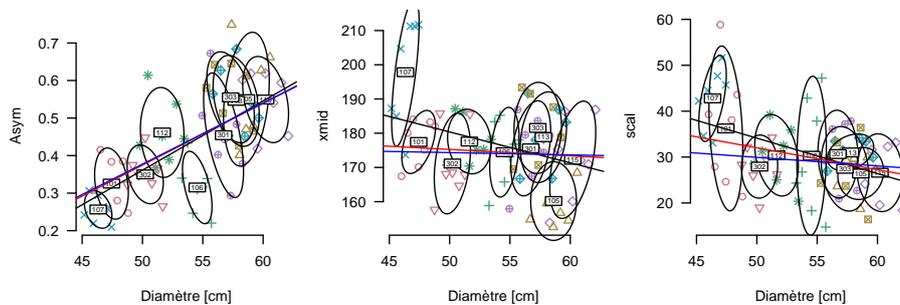
3.2.4 Paramètres de croissance et diamètre du tronc

POUR synthétiser l'information on va représenter les paramètres de croissance de chaque arbre chaque année en fonction du diamètre annuel moyen :

```

fact <- as.factor(sapply(tablog$Tree_name, \(x) unlist(strsplit(x, split = "\\.")))[2]))
library(ade4)
plotp <- function(param){
  par(pty = "s", bty = "n")
  plot(tablog$diacm, tablog[, param], col = mycol, las = 1,
       xlab = "Diamètre [cm]", pch = 1:10, ylab = param)
  s.class(tablog[, c("diacm", param)], fact, add.plot = TRUE, axesell = FALSE,
         cstar = 0, cpoint = 0, clab = 0.75)
  abline(lm(tablog[, param]~tablog[, "diacm"]))
  cent7 <- which(tablog$Tree_name == "CHS57.107")
  cent1 <- which(tablog$Tree_name == "CHS57.101")
  abline(lm(tablog[-cent7, param]~tablog[-cent7, "diacm"]), col = "red")
  abline(lm(tablog[-c(cent1, cent7), param]~tablog[-c(cent1, cent7), "diacm"]), col = "blue")
}
par(mfrow = c(1, 3), mar = c(4, 4, 0, 1), oma = c(0, 0, 3, 0))
plotp("Asym") ; plotp("xmld") ; plotp("scal")
title(main = "Synthèse des paramètres de croissance par arbre
      en fonction du diamètre", outer = TRUE,
      cex.main = 1.5)
    
```

Synthèse des paramètres de croissance par arbre en fonction du diamètre



LES paramètres de croissance sont relativement stables pour chaque arbre d'une année à l'autre, malgré l'alternance d'années plus ou moins fastes. On retrouve la corrélation positive entre le diamètre et la croissance annuelle (déjà vue dans la section 2 page 7) avec l'arbre n° 105 qui est en deçà de la tendance générale. La corrélation négative entre le diamètre et la date du pic de croissance (ligne noire) est surtout le fait de l'arbre n° 107 et disparaît quand on ne le prend plus en compte (ligne rouge). La date du pic de croissance est plutôt homogène d'un arbre à l'autre (autour du 175^e jour de l'année soit au voisinage du solstice d'été) avec les n^{os} 105 et 107 qui se singularisent par des dates plus précoces ou plus tardives, respectivement. Il pourrait y avoir une légère corrélation négative entre le diamètre et la durée de la phase de croissance, en tout cas les deux plus petits arbres, n^{os} 107 et 101, semblent avoir une phase de croissance plus longue. La corrélation devient négligeable si on retire ces deux arbres de l'analyse (ligne bleu).

4 Utilisation des vitesses instantannées

4.1 Calcul des vitesses instantannées

SI ON NOTE, pour un arbre donné, t_i la date de la i^{e} observation ($i \in \{1, 2, \dots, n\}$) de la surface basale, S_i , alors la vitesse instantannée, V_i , est définie par :

$$\forall i \in \{2, 3, \dots, n\} : V_i = \frac{S_i - S_{i-1}}{t_i - t_{i-1}} = \frac{\Delta S_i}{\Delta t_i}$$

La fonction `diff()` appliquée au vecteur des dates va nous calculer les écarts de temps en jours. La majorité des écarts vaut 7 jours puisqu'il s'agit d'un suivi hebdomadaire, avec quelques exceptions quand la mesure n'a pas été faite un samedi.

```
deltat <- diff(surf$Date)
head(deltat)
Time differences in days
[1] 7 7 7 7 7 7
table(deltat)
deltat
 5  6  7  8  9
 1  8 346 8  1
```

COMME nous avons noté dans la section 1.2 page 2 qu'il y avait 10 mesures qui n'avaient pas été faites un samedi, on s'attendrait plutôt à avoir 20 écarts différents de 7 jours (alors que l'on en compte 18 ici), sauf s'il y a deux jours exceptionnels consécutifs. Il est facile de vérifier cela avec la fonction `rle()` (*run length encoding*) :

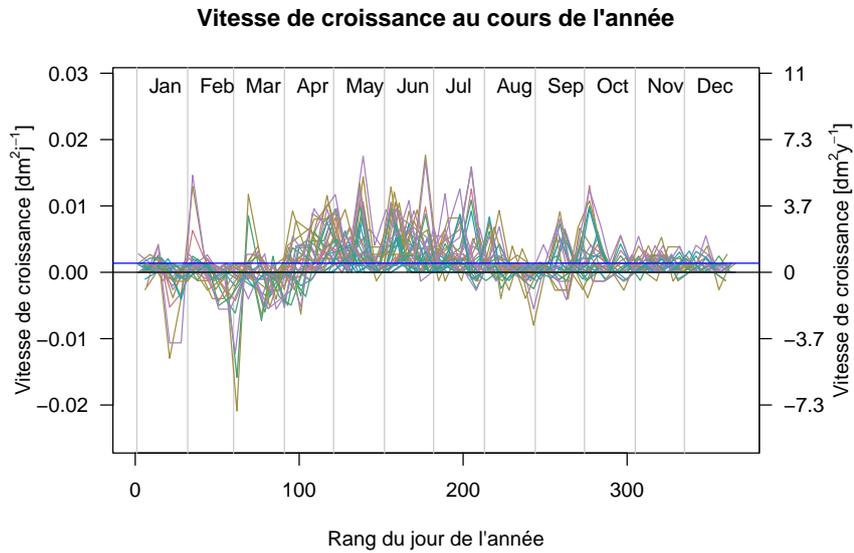
```
resrle <- rle(format(surf$Date, "%a"))
jours <- resrle$lengths
names(jours) <- resrle$values
jours
Sat Fri Sat Fri Sat Sun Sat Sun Sat Fri Sat Fri Sat Sun Sat Thu Sat Sun Sat
35  1 50  1 96  1 14  1 25  1 16  1 52  1 53  1  7  2  7
```

CECI se lit de la façon suivante : il y a eu 35 samedis consécutifs puis 1 vendredi puis 50 samedis consécutifs puis 1 vendredi etc. On constate qu'il y a eu 2 dimanches consécutifs en fin de série, c'est donc normal de n'avoir que 18 écarts différents de 7 jours.

LES vitesses de croissance sont élevées majoritairement au printemps et pendant l'été, parfois à l'automne. On observe parfois une contraction du tronc pendant les mois de janvier et février³. La ligne horizontale bleu représente l'accroissement annuel typique de $0.5 \text{ dm}^2 \text{ y}^{-1}$, on voit que les vitesses de pointe observées l'été dépassent largement 10 fois cette moyenne annuelle : ce graphique illustre le caractère saisonnier de la croissance des arbres dans la zone tempérée.

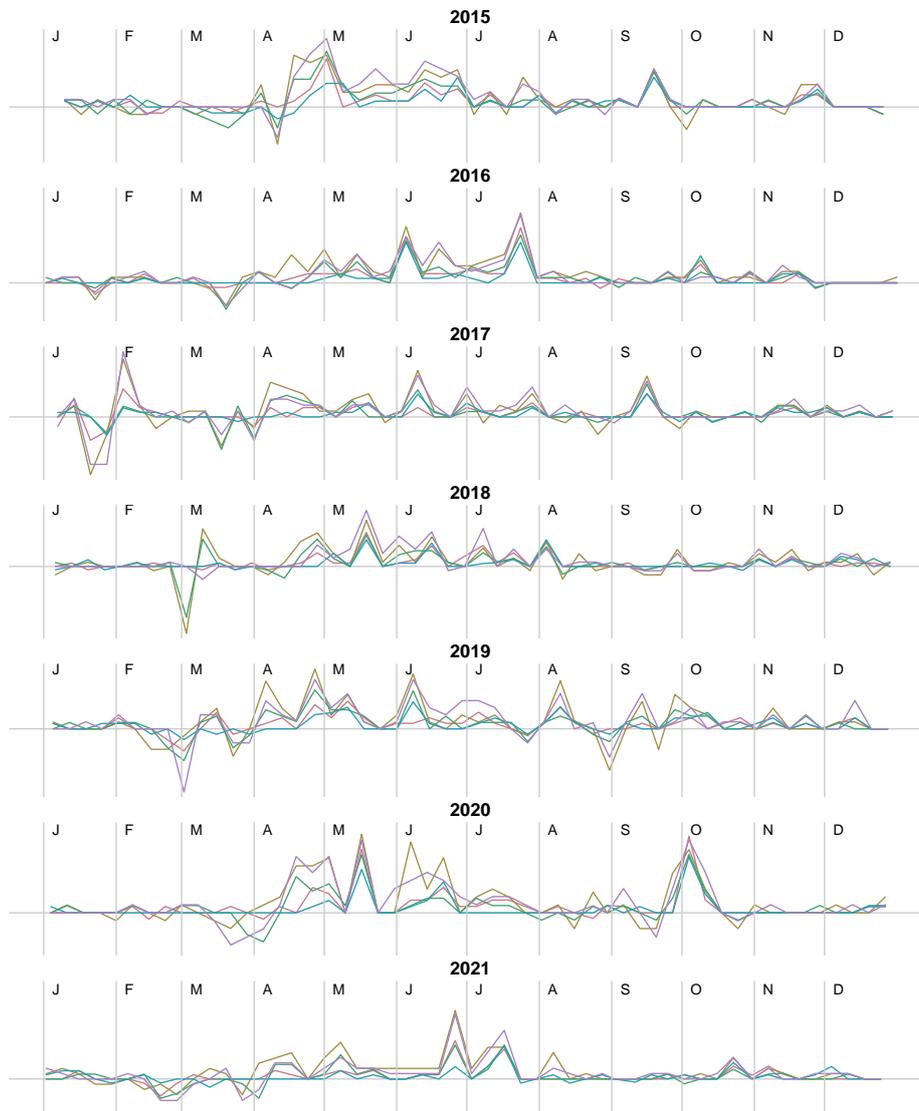
```
V <- as.data.frame(apply(surf[, 1:10], 2, \(\x) diff(x)/as.numeric(deltat)))
V$Date <- surf$Date[-1]
V$doy <- as.integer(format(V$Date, "%j"))
V$Year <- as.integer(format(V$Date, "%Y"))
par(mar = c(5, 5, 4, 6) + 0.1)
ylim <- 1.2*range(V[,1:10])
plot(1:366, 1:366, type = "n", ylim = ylim,
     las = 1, xlab = "Rang du jour de l'année", main = "Vitesse de croissance au cours de l'année",
     ylab = expression(paste("Vitesse de croissance [", dm^2*j^-1, "]"))
abline(v = imonth, col = grey(0.8))
text(imonth, 0.9*par("usr")[4], mois.abb, pos = 4)
axis(4, las = 1, at = pretty(ylim), labels = round(pretty(ylim)*365.25, 1))
for(the_year in unique(V$Year)){
  tmp <- V[V$Year == the_year, ]
  for(j in seq_len(10)) lines(tmp$doy, tmp[,j], col = mycol[j])
}
abline(h = 0.5/365.25, col = "blue") ; abline(h = 0)
mtext(expression(paste("Vitesse de croissance [", dm^2*y^-1, "]")), side = 4, line = 3)
```

³On y reviendra dans la section 4.3 page 28



La figure est trop chargée en courbes pour pouvoir en faire une lecture fine. On décide de représenter les années une par une et une :

```
par(mfrow = c(7, 1), mar = c(0, 0, 1.5, 0))
for(the_year in unique(V$Year)){
  tmp <- V[V$Year == the_year, ]
  plot.new() ; plot.window(xlim = c(1, 366), ylim = range(tmp[, 1:10]))
  abline(h = 0, col = grey(0.8))
  title(main = the_year)
  for(j in seq_len(10)) lines(tmp$doy, tmp[,j], col = mycol[j])
  abline(v = imonth, col = grey(0.8))
  text(imonth, 0.9*par("usr")[4], substr(mois.abb, 1, 1), pos = 4)
}
```



QUAND on regarde les courbes des vitesses on a l'impression qu'elles ont un aspect en dents de scie, comme si une semaine de forte croissance était suivie par une semaine de plus faible croissance. C'est ce que nous étudierons plus en détail dans la section ?? page ??. On note la très grande synchronicité des arbres entre-eux. Il y a donc un sens à calculer la vitesse instantanée moyenne des arbres :

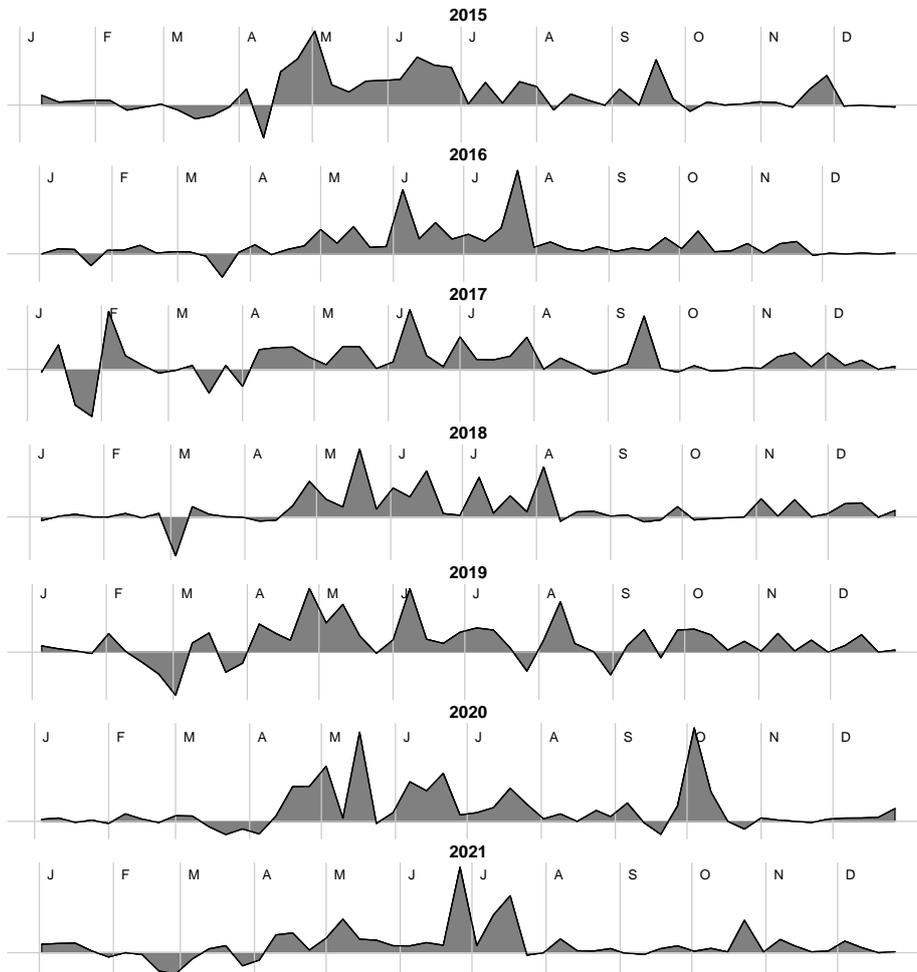
```

polycurve <- function(x, y, base.y = min(y), ...) {
  polygon(x = c(min(x), x, max(x)), y = c(base.y, y, base.y), ...)
}
par(mfrow = c(7, 1), mar = c(0, 0, 1.5, 0), bty = "n")
years <- unique(V$Year) ; ny <- length(years)
V$moy <- apply(V[, 1:10], 1, mean)
for(i in seq_len(ny)){
  tmp <- V[V$Year == years[i], ]
  plot(tmp$doy, tmp$moy, ylim = range(tmp$moy), type = "l", main = years[i],
       xaxt = "n", yaxt = "n")
  # polycurve(tmp$doy, tmp$moy, col = "pink")
  polycurve(tmp$doy, tmp$moy, base.y = 0, col = grey(0.5))
}
    
```

```

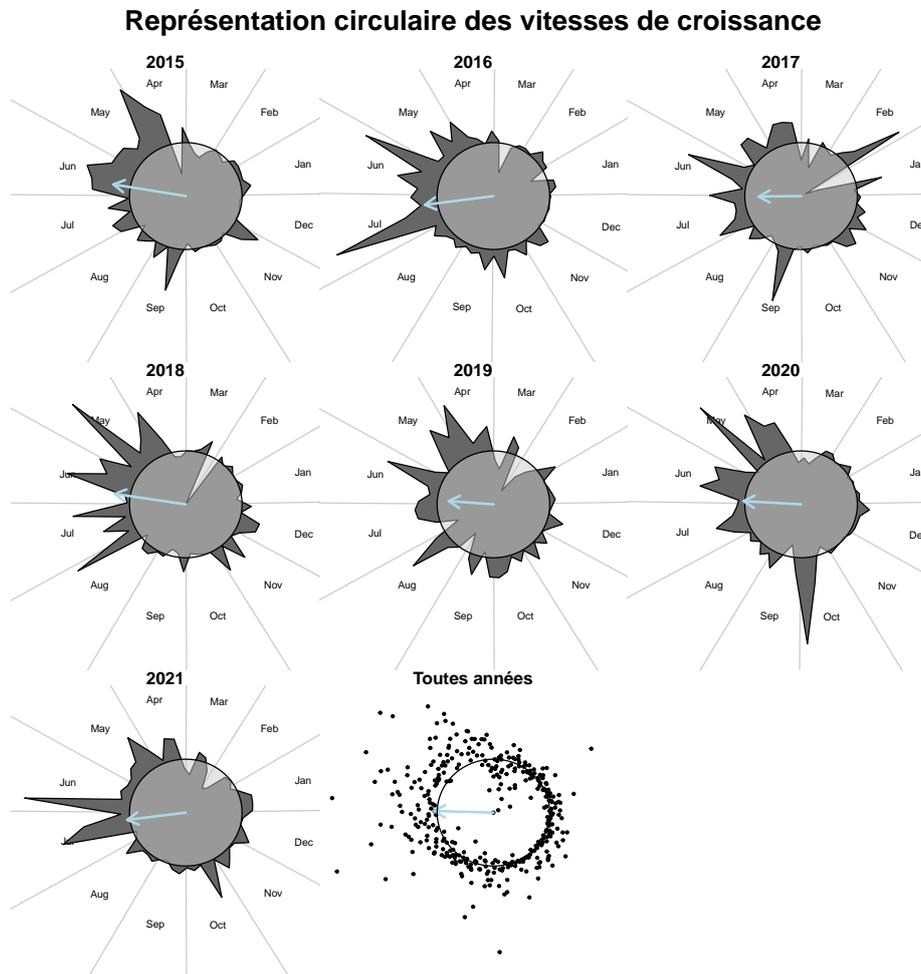
abline(h = 0, col = grey(0.8))
abline(v = imonth, col = grey(0.8))
text(imonth, 0.85*par("usr")[4], substr(mois.abb, 1, 1), pos = 4)
lines(tmp$doy, tmp$moy, lwd = 1)
}

```



4.2 Statistiques circulaires

QUAND on s'intéresse comme ici à un phénomène périodique, il peut être intéressant d'utiliser des représentations et des statistiques circulaires. Le code `R` permettant de produire la figure suivante est donné en annexe page 34. L'idée est de représenter les données V_i par un vecteur du plan partant de l'origine dont l'angle indique l'avancée dans le cycle et la longueur représente la valeur observée. Pour que la représentation ait un sens il faut translater toutes les valeurs V_i de façon à ce que la valeur minimale soit à zéro. Dans le graphique ci-dessous le cercle représente les croissances nulles, tout ce qui est en gris foncé représente les périodes de croissance. On voit qu'il y a un biais vers la gauche, correspondant au mois juin et juillet. On peut quantifier ce biais en calculant le vecteur moyen, représenté en bleu ici (mais pas à la même échelle).



SELON les années, le pic de croissance a lieu fin juin ou début juillet. Toutes années confondues, le maximum de croissance est en moyenne le 181^e jour de l'année, soit environ une semaine après le solstice d'été :

```
365.25*(pi + atan(mean(V$y)/mean(V$x)))/(2*pi)
[1] 180.9154
format(as.Date("2022-06-21"), "%j")
[1] "172"
```

4.3 Contraction des troncs par grand froid

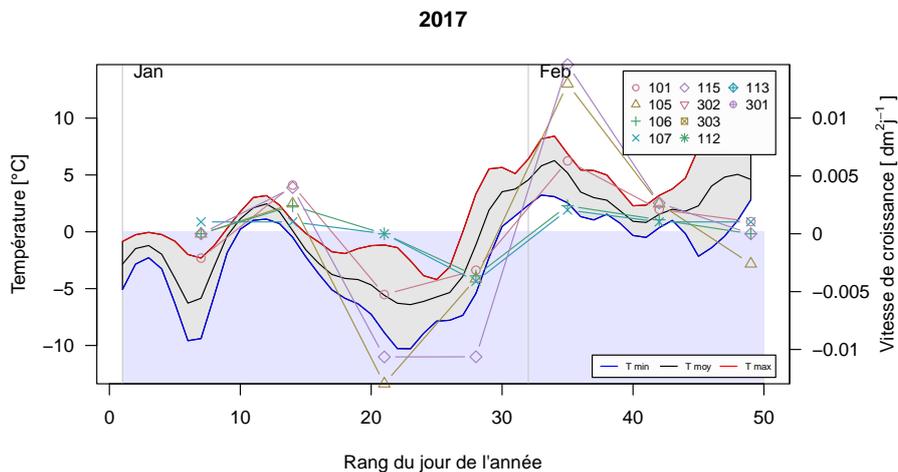
Ce n'est pas une observation récente puisque Joseph FRIERICH écrivait⁴ [1, p. 155] dès 1897 : « comme les dendromètres ont été laissés sur la plupart sur

⁴Nachdem die Zuwachsmesser meistens über Winter an den Bäumen belassen wurden, konnte auch der Einfluss grösserer Kälte auf den Baumumfang beobachtet werden. Die Aufschreibungen während dieser Zeit gelangten wegen Raumersparniss nicht zum Abdrucke. Im Allgemeinen bewirken Temperaturen unter Nullgrad selbstverständlich eine Verringerung des Baumumfanges, welche bei länger andauerndem starken Froste die Grösse einer Jahrringbreite erreichen kann.

des arbres pendant l'hiver, l'influence d'un froid conséquent sur la circonférence des troncs a également pu être étudiée. Les observations hivernales n'ont pas été imprimées ici pour gagner de la place. En général, les températures inférieures à zéro provoquent naturellement une réduction du diamètre, qui peut atteindre la taille d'une largeur de cerne en cas de gel sévère prolongé. »

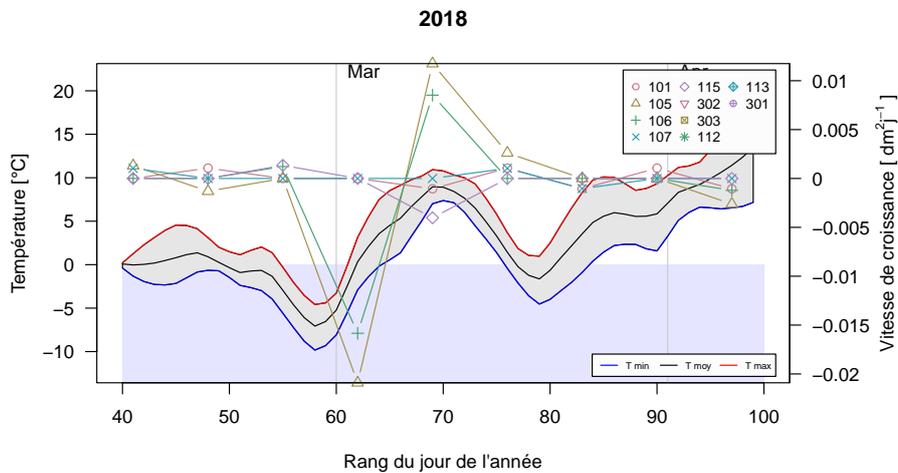
ON définit une fonction utilitaire pour représenter simultanément les températures et les vitesses de croissances de la surface basale (le code `R` de la fonction `plotmetV()` est donné en annexe page 35). C'est en 2017 fin janvier début février que l'on observe le plus spectaculaire épisode de contraction-expansion-retour à la normale. Tous les arbres sont concernés sauf le n° 112. Cela correspond à une période très froide avec des températures maximales durablement négatives.

```
plotmetV("2017", mindoy = 1, maxday = 50, span = 0.2, pos = "bottomright",
pos2 = "topright")
```



ON observe début mars 2018 un épisode de contraction-expansion-retour à la normale mais qui ne concerne que quatre arbres, les n°s 105, 106, 301 et 302, les autres étant peu affectés. Il y a une période froide qui précède cet évènement.

```
plotmetV("2018", mindoy = 40, maxday = 100, span = 0.2, pos = "bottomright",
pos2 = "topright")
```



5 Animation graphique

L'OBJECTIF est de faire une petite image GIF animée pour représenter l'évolution de la surface terrière des 10 chênes au cours du temps, avec quelques indications météorologiques en sus (température et précipitations journalières, durée de la phase diurne). Pour être visualisée, il suffit d'ouvrir le fichier⁵ avec son navigateur favori. Le code suivant n'a pas besoin d'être exécuté, je l'ai laissé simplement pour avoir une trace de comment j'ai généré le fichier des données météo utilisées.

```
load("../SIM2/data/meteoLBBE.Rda")
meteo <- meteoLBBE[["CHS57"]]
rm(meteoLBBE)
meteo <- subset(meteo, DATE >= min(surf$Date) & DATE <= max(surf$Date))
selected <- c("DATE", "T_Q", "PRELIQ_Q")
meteo <- meteo[, selected]
save(meteo, file = "dendroCHS57/meteo.Rda")
```

ON récupère les données météo qui nous intéressent directement à partir de la sauvegarde précédente. Ce sont des données quotidiennes, on fait la jointure à gauche avec les données de croissance en utilisant la date comme clef commune. Les données de croissance étant hebdomadaires on se retrouve avec des NA que l'on impute par interpolation linéaire.

```
load(url("https://pbil.univ-lyon1.fr/R/donnees/dendroCHS57/meteo.Rda"))
surf$surf_m2 <- rowSums(surf[, 1:10])/100
dta <- merge(meteo, surf[, c("Date", "surf_m2")], by.x = "DATE", by.y = "Date", all.x = TRUE)
library(imputeTS)
dta$surf_m2 <- na_interpolation(dta$surf_m2)
```

La longueur du jour est calculé pour chaque date (aux coordonnées du site CHS57A) pour pouvoir la représenter de façon symbolique avec un soleil plus ou moins intense en arrière-plan du graphique.

⁵<http://pbil.univ-lyon1.fr/R/donnees/dendroCHS57/high.gif>

```
library(suncalc)
soleil <- getSunlightTimes(dta$DATE,
                          lat = 48 + 52/60 + 18/3600,
                          lon = 6 + 29/60 + 02/3600,
                          tz = "Etc/GMT-2")[c("sunrise", "sunset")]
dta$sunl <- as.numeric(soleil$sunset - soleil$sunrise)
```

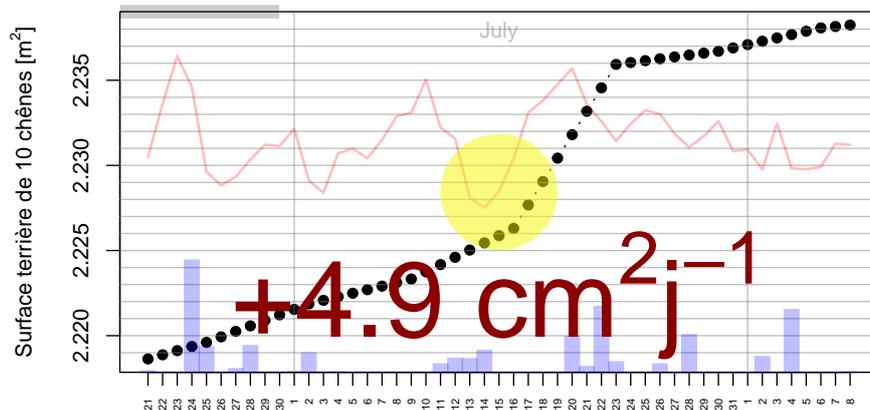
O DÉFINIT la fonction graphique `plotday()` pour représenter les données à partir d'un jour donné, par exemple ici à partir du solstice d'été en 2016.

```
library(lubridate)
library(plotrix) # pour cercle
plotday <- function(the_day, long = 49, scaleP = max(dta$PRELIQ_Q),
                   colP = rgb(0, 0, 1, 0.25),
                   scaleT = max(dta$T_Q),
                   colT = rgb(1, 0, 0, 0.25),
                   colmonth = rgb(0.5, 0.5, 0.5, 0.5),
                   colbar = rgb(0, 0, 0, 0.2),
                   faccex = 1,
                   facsol = 1){
  ii <- which(dta$DATE == the_day)
  with(dta[ii:(ii + long - 1), ],{
    par(mar = c(2, 5, 4, 1) + 0.1)
    main <- paste(sprintf("%02d", mday(the_day)),
                  month(the_day, label = TRUE), year(the_day), "-",
                  "CHS57A : unité territoriale du Saulnois")
    ylab <- bquote(paste("Surface terrière de 10 chênes [", m^2, "]"))
    plot(DATE, surf_m2, pch = 19, type = "b", main = main, xaxt = "n",
         ylab = ylab, xlab = "")
    pu <- par("usr")

    axis(1, at = DATE, label = mday(DATE), cex.axis = 0.5, las = 2)
    # Grille horizontale
    mins <- signif(min(surf_m2), 4)
    hgrid <- seq(mins, max(c(surf_m2, mins + 0.001)), by = 0.001)
    abline(h = hgrid, col = colmonth)
    # Visualisation des précipitations
    xleft <- DATE - 0.5
    ybottom <- rep(pu[3], long)
    xright <- DATE + 0.5
    ytop <- pu[3] + (pu[4] - pu[3])*PRELIQ_Q/scaleP
    rect(xleft, ybottom, xright, ytop, col = colP, border = colP, xpd = NA)
    # Visualisation des températures
    ytop <- pu[3] + (pu[4] - pu[3])*T_Q/scaleT
    points(DATE, ytop, type = "l", col = colT, lwd = 2)
    # Visualisation des mois
    idbm <- which(mday(DATE) == 1)
    abline(v = DATE[idbm], col = colmonth)
    imdm <- which(mday(DATE) == 15)
    text(DATE[imdm], pu[4], month(DATE[imdm], label = TRUE, abbr = FALSE), pos = 1, col = colmonth)
    # Barres de progression
    pcy <- (max(surf_m2) - min(dta$surf_m2))/diff(range(dta$surf_m2))
    par(lend = "butt")
    segments(pu[2], pu[3], pu[2], pu[3] + pcy*(pu[4] - pu[3]),
             lwd = 10, xpd = NA, col = colbar)
    pcx <- ii/nrow(dta)
    segments(pu[1], pu[4], pu[1] + pcx*(pu[2] - pu[1]), pu[4],
             lwd = 10, xpd = NA, col = colbar)
    # Calcul de la vitesse moyenne en cm2/j
    jseq <- 1:long
    slope <- lm(100^2*surf_m2~jseq)$coef[2]
    # je veux le signe et un chiffre après la virgule
    slopech <- sprintf("%+.1f ", slope)
    # 'gade comme c'est beau comme ça
    txt <- bquote(paste(.(slopech), cm^2*j^-1))
    text((pu[1] + pu[2])/2, pu[3], txt, pos = 3,
         col = ifelse(slope > 0, "darkred", "darkblue"),
         cex = abs(slope)/faccex)
    # Longueur du jour
    ldj <- mean(sunl)
    pcm <- ldj/max(dta$sunl)
    colsol <- rgb(pcm, pcm, 0, 0.5)
    draw.circle(mean(pu[1:2]), mean(pu[3:4]), facsol*sqrt(ldj), col = colsol, border = colsol)
  })
}
```

```
}
plotday(as.Date("2016-06-21"))
```

21 Jun 2016 – CHS57A : unité territoriale du Saulnois



ON fait mouliner cette fonction graphique à basse résolution et sur un sous-ensemble des données pour illustrer le site web sans trop l'alourdir.

```
days <- dta$DATE[1:1000] ; ns <- length(days)
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)
  plotday(days[i])
  dev.off()
}
library(gifski)
png_files <- list.files(path = "low/", pattern = ".png", full.names = TRUE)
gifski::gifski(png_files, gif_file = "dendroCHS57.gif",
               delay = 0.1)
```

ON fait ensuite mouliner la fonction à haute résolution sur l'ensemble des données pour avoir le rendu final (adresse dans la note de bas de page précédente).

```
days <- dta$DATE[1:2500] ; ns <- length(days)
for(i in seq_len(ns)){
  ichar <- sprintf("%06d", i)
  png(paste0("high/image", ichar, ".png"), width = 6, height = 3, units = "in", res = 144)
  par(cex = 0.5)
  plotday(days[i])
  dev.off()
}
png_files <- list.files(path = "high/", pattern = ".png", full.names = TRUE)
gifski::gifski(png_files, gif_file = "high.gif",
               delay = 0.1)
```

6 Annexes

6.1 Remerciements

MERCI à Gilles SINICCO d'avoir accepté de partager ses données de suivi dendrométrique. Les données météorologiques ont été extraites de la base

SAFRAN (Système d'Analyse Fournissant des Renseignements Adaptés à la Nivologie) du CNRM (Centre National de la Recherche Météorologique).

6.2 Importation au format Date

ON a illustré ici toute la puissance du format `Date`, tout ceci est bel et bon, mais que fait-on si on ne dispose pas de ce format ? J'explique ici comment les données ont été importées dans `R`. Le point de départ est un fichier de type tableur⁶ ressemblant à ceci :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CHS57A													
2	N° Arbre	dendro simple/double	Date Dernière Mesure	Dernière Mesure (cm)	06/12/2014	13/12/2014	20/12/2014	27/12/2014	03/01/2015	10/01/2015	17/01/2015	24/01/2015	31/01/2015	07/02/2015
3	CH557 101		19/12/2013	45,32	45,76	45,77	45,77	45,78	45,78	45,79	45,79	45,80	45,80	45,81
4	CH557 105		19/12/2013	55,77	56,37	56,38	56,38	56,39	56,39	56,40	56,39	56,40	56,40	56,39
5	CH557 106		19/12/2013	52,55	53,03	53,03	53,04	53,04	53,04	53,05	53,06	53,05	53,06	53,05
6	CH557 107		19/12/2013	44,64	44,98	44,99	45,00	45,00	45,01	45,02	45,02	45,03	45,03	45,05
7	CH557 115	d	19/12/2013	57,30	57,91	57,91	57,91	57,92	57,92	57,93	57,94	57,94	57,95	57,96
8	CH557 302		19/12/2013	47,99	48,49	48,50	48,50	48,51	48,51	48,52	48,52	48,52	48,53	48,52
9	CH557 303	double	19/12/2013	54,19	54,91	54,92	54,93	54,93	54,94	54,95	54,95	54,96	54,96	54,96
10	CH557 112		19/12/2013	48,97	49,46	49,47	49,48	49,48	49,49	49,49	49,50	49,50	49,51	49,52
11	CH557 113		19/12/2013	54,74	55,46	55,46	55,46	55,47	55,48	55,49	55,50	55,51	55,51	55,52
12	CH557 301		19/12/2013	54,01	54,61	54,61	54,61	54,61	54,62	54,64	54,64	54,64	54,65	54,67
13					52,098	52,104	52,108	52,113	52,118	52,128	52,131	52,135	52,14	52,145
14	0,115					0,006	0,004	0,005	0,005	0,010	0,003	0,004	0,005	0,005

J'AI commencé par l'enregistrer sous la forme d'un fichier texte, `dendroCHS57.csv`, en utilisant des tabulations comme séparateur de colonnes et en ne conservant que les informations qui m'intéressaient :

	A	B	C	D	E	F	G	H	I
1	Tree_name	03/01/2015	10/01/2015	17/01/2015	24/01/2015	31/01/2015	07/02/2015	14/02/2015	21/02/2015
2	CHS57 101	45,78	45,79	45,79	45,80	45,80	45,81	45,80	45,79
3	CHS57 105	56,39	56,40	56,39	56,40	56,40	56,39	56,38	56,38
4	CHS57 106	53,04	53,05	53,06	53,05	53,06	53,05	53,06	53,06
5	CHS57 107	45,01	45,02	45,02	45,03	45,03	45,03	45,05	45,05
6	CHS57 115	57,92	57,93	57,94	57,94	57,95	57,96	57,95	57,95
7	CHS57 302	48,51	48,52	48,52	48,52	48,53	48,52	48,52	48,53
8	CHS57 303	54,94	54,95	54,95	54,96	54,96	54,96	54,95	54,94
9	CHS57 112	49,49	49,49	49,50	49,50	49,51	49,52	49,52	49,52
10	CHS57 113	55,48	55,49	55,50	55,51	55,51	55,52	55,50	55,49
11	CHS57 301	54,62	54,64	54,64	54,64	54,65	54,67	54,67	54,67

J'AI ensuite importé les données dans `R` en précisant que la première ligne portait le nom des colonnes, que ces dernières étaient séparées par des tabulations et que le séparateur décimal est une virgule :

⁶DENDRO_CHS_57A_SUIVI.xlsx

```
fname <- "https://pbil.univ-lyon1.fr/R/donnees/dendroCHS57/dendroCHS57.csv"
pre <- read.table(fname, header = TRUE, sep = "\\t", dec = ",")
head(pre[, 1:5])
```

Tree_name	X03.01.2015	X10.01.2015	X17.01.2015	X24.01.2015
1 CHS57 101	45.78	45.79	45.79	45.80
2 CHS57 105	56.39	56.40	56.39	56.40
3 CHS57 106	53.04	53.05	53.06	53.05
4 CHS57 107	45.01	45.02	45.02	45.03
5 CHS57 115	57.92	57.93	57.94	57.94
6 CHS57 302	48.51	48.52	48.52	48.52

J'ai transposé la table de façon à avoir les séries temporelles en colonne :

```
diacm <- as.data.frame(t(pre[, -1]))
colnames(diacm) <- make.names(pre$Tree_name)
head(diacm[, 1:5])
```

	CHS57.101	CHS57.105	CHS57.106	CHS57.107	CHS57.115
X03.01.2015	45.78	56.39	53.04	45.01	57.92
X10.01.2015	45.79	56.40	53.05	45.02	57.93
X17.01.2015	45.79	56.39	53.06	45.02	57.94
X24.01.2015	45.80	56.40	53.05	45.03	57.94
X31.01.2015	45.80	56.40	53.06	45.03	57.95
X07.02.2015	45.81	56.39	53.05	45.05	57.96

RESTE la partie la plus intéressante : comment transformer les dates au format `Date`? Dans ma table `diacm` les dates sont données par le nom des lignes, par exemple `X03.01.2015` pour la première. On utilise la fonction `as.Date()` pour convertir ces chaînes de caractères en objets de la classe `Date` en précisant que le format est la forme `X%d.%m.%Y`⁷ :

```
diacm$Date <- as.Date(rownames(diacm), format = "X%d.%m.%Y")
head(diacm[, c(1:5, 11)])
```

	CHS57.101	CHS57.105	CHS57.106	CHS57.107	CHS57.115	Date
X03.01.2015	45.78	56.39	53.04	45.01	57.92	2015-01-03
X10.01.2015	45.79	56.40	53.05	45.02	57.93	2015-01-10
X17.01.2015	45.79	56.39	53.06	45.02	57.94	2015-01-17
X24.01.2015	45.80	56.40	53.05	45.03	57.94	2015-01-24
X31.01.2015	45.80	56.40	53.06	45.03	57.95	2015-01-31
X07.02.2015	45.81	56.39	53.05	45.05	57.96	2015-02-07

IL ne ne reste plus qu'à sauvegarder notre tableau de données `diacm` au format `XDR` [2], qui est un format binaire compatible multi-plateformes. On enlève au préalable le nom des lignes qui ne nous est plus utile puisque la date est maintenant dans la colonne `Date`.

```
rownames(diacm) <- NULL # enlever le nom des colonnes
save(diacm, file = "dendroCHS57/diacm.Rda")
```

6.3 Code

6.3.1 Code de la section 4.2

```
circle <- function (x = 0, y = 0, r = 1, theta = c(0, 360), n = 100, ...)
{
  a <- seq(theta[1], theta[2], length = n)
  xx <- x + r * cos(a * 2 * pi/360)
  yy <- y + r * sin(a * 2 * pi/360)
  if (!identical(theta, c(0, 360))) {
    xx <- c(xx, x)
    yy <- c(yy, y)
  }
  polygon(xx, yy, ...)
}
seqyear <- 2015:2021
```

⁷Se référer à la table page 6 pour une description détaillée des spécificateurs de format.

```
ndj <- as.numeric(format(as.Date(paste0(seqyear, "-12-31")), "%j"))
names(ndj) <- seqyear
V$ndj <- ndj[sapply(V$Year, \(x) which(names(ndj) == x))]
V$angle <- 2*pi*V$doy/V$ndj
V$x <- cos(V$angle)*(V$moy - min(V$moy))
V$y <- sin(V$angle)*(V$moy - min(V$moy))
par(mfrow = c(3, 3), mar = c(0, 0, 1, 0), oma = c(0, 0, 4, 0))
lims <- c(-0.02, 0.015) ; coly <- hcl.colors(7, "Dark 2")
for(i in seq_len(ny)){
  tmp <- V[V$Year == years[i], ]
  plot.new() ; plot.window(xlim = lims, ylim = lims)
  for(j in 1:12){
    alpha <- 2*pi*imonth[j]/ndj[i]
    segments(0, 0, 10*cos(alpha), 10*sin(alpha), col = grey(0.8))
    text(0.015*cos(alpha + pi/12), 0.015*sin(alpha + pi/12), mois.abb[j], cex = 0.75)
  }
  polygon(tmp$x, tmp$y, col = grey(0.4))
  title(main = years[i])
  circle(r = 0 - min(V$moy), col = rgb(0.8, 0.8, 0.8, 0.5))
  arrows(0, 0, 10*mean(tmp$x), 10*mean(tmp$y), length = 0.1, lwd = 2, col = "lightblue")
}
plot(V$x, V$y, type = "p", pch = 19, cex = 0.5, xlim = lims, ylim = lims,
      main = "Toutes années", xaxt = "n", yaxt = "n", bty = "n")
arrows(0, 0, 10*mean(V$x), 10*mean(V$y), length = 0.1, lwd = 2, col = "lightblue")
circle(r = 0 - min(V$moy))
title(main = "Représentation circulaire des vitesses de croissance",
      outer = TRUE, cex.main = 2)
```

6.3.2 Code de la section 4.3

```
load(url("https://pbil.univ-lyon1.fr/R/donnees/dendroCHS57/met.Rda")) # données météo
plotmetV <- function(the_year, mindoy = 1, maxdoy = 120, pos = "bottom",
                    span = 0.1, pos2 = "topright"){
  m <- met[met$Year == the_year, ]
  m <- m[m$doy < maxdoy & m$doy >= mindoy, ]
  xlim <- c(mindoy, maxdoy)
  ylim <- c(min(m$Tmin), max(m$Tmax))
  par(mar = c(5, 4, 4, 6) + 0.1)
  plot.new() ; plot.window(xlim = xlim, ylim = ylim)
  rect(mindoy, par("usr")[3], maxdoy, 0, border = NA, col = rgb(0,0,1,0.1))
  title(main = the_year, ylab = "Température [°C]",
        xlab = "Rang du jour de l'année") ; axis(1)
  axis(2, las = 1)
  abline(v = imonth, col = grey(0.8))
  text(imonth, 0.95*par("usr")[4], mois.abb, pos = 4)
  T.lo <- loess(T ~ doym, span = span)
  dseq <- mindoy:(maxdoy - 1)
  Tmin.lo <- loess(Tmin ~ doym, span = span)
  Tmin.lo.p <- predict(Tmin.lo, data.frame(doy = dseq))
  Tmax.lo <- loess(Tmax ~ doym, span = span)
  Tmax.lo.p <- predict(Tmax.lo, data.frame(doy = dseq))
  polygon(c(dseq, rev(dseq)), c(Tmin.lo.p, rev(Tmax.lo.p)), col = grey(0.9))
  lines(dseq, Tmin.lo.p, col = "blue")
  lines(dseq, Tmax.lo.p, col = "red")
  lines(dseq, predict(T.lo, data.frame(doy = dseq)))
  legend(pos, inset = 0.02, lwd = 1, col = c("blue", "black", "red"),
        legend = c("T min", "T moy", "T max"), cex = 0.5, ncol = 3, bg = grey(0.99))
  pu3 <- par("usr")[3] ; pu4 <- par("usr")[4] ; dpu <- pu4 - pu3
  v <- V[V$Year == the_year & V$doy < maxdoy & V$doy >= mindoy, ]
  minv <- min(v[,1:10]) ; maxv <- max(v[,1:10])
  dpy <- maxv - minv
  for(j in seq_len(10)){
    points(v$doy, pu3 + dpu*(v[,j] - minv)/dpy, col = mycol[j],
          pch = j, type = "b", xpd = NA)
  }
  ax4 <- pretty(unlist(v[,1:10]))
  axis(4, at = pu3 + dpu*(ax4 - minv)/dpy, labels = ax4, las = 1)
  ylab <- expression(paste("Vitesse de croissance [ ", dm^2*j^-1, " ]"))
  mtext(ylab, side = 4, line = 4)
  legend(pos2, inset = 0.02, pch = 1:10, col = mycol,
```

```
    legend = noms, cex = 0.75, ncol = 3, bg = grey(0.99))  
  }  
}
```

Références

- [1] J. Friedrich. Über den Einfluss der Witterung auf den Baumzuwachs. *Mitteilungen Forstlichen Bundes-Versuchsanstalt Wien*, 22 :1–160, 1897.
- [2] Sun Microsystems. XDR : external data representation standard. RFC 1014. Technical report, Network Working Group, 1987.
- [3] P.-F. Verhulst. Notice sur la loi que la population suit dans son accroissement. *Correspondance mathématique et physique*, 10 :113–121, 1838.
- [4] P.-F. Verhulst. Recherches mathématiques sur la loi d'accroissement de la population. *Nouveaux mémoires de l'académie royale des sciences, des lettres et des beaux-arts de Belgique*, 18 :1–38, 1845. Lu à la séance du 30 novembre 1844.
- [5] P.-F. Verhulst. Note sur la loi d'accroissement de la population. *Bulletins de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles*, 13, 1846.
- [6] P.-F. Verhulst. Deuxième mémoire sur la loi d'accroissement de la population. *Mémoires de l'académie royale des sciences, des lettres et des beaux-arts de Belgique*, 20 :1–32, 1847. Lu à la séance de l'Académie royale du 15 mai 1846.