Concentration de CO2 dans l'atmosphère depuis 1958

Thomas Pérot

12 juin 2023

Contents

Introduction

Dans ce document nous allons présenter et analyser des données d'évolution de la concentration en CO2 atmosphérique provenant du site de l'observatoire de Mauna Loa situé sur l'île d'Hawaï. Ces données sont décrites dans le document suivant :

C. D. Keeling, S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, and H. A. Meijer, Exchanges of atmospheric CO2 and 13CO2 with the terrestrial biosphere and oceans from 1978 to 2000. I. Global aspects, SIO Reference Series, No. 01-06, Scripps Institution of Oceanography, San Diego, 88 pages, 2001.

Chargement des données

Les données sont disponibles sur le site suivant : Scripps CO2 Program.

Nous avons téléchargé les données le 12 juin 2023.

Les premières 57 lignes du fichier donnent des informations sur les données. Les lignes 2 à 19 indiquent l'endroit d'où proviennent les mesures, la source des données et d'autres informations sur les données. Les lignes 23 à 37 indiquent comment citer les données. Les lignes 41 à 56 décrivent le tableau de données.

Le tableau de données commence à la ligne 58 avec l'entête des colonnes. Ensuite il y a deux lignes qui sont des informations sur les colonnes, soit par rapport au format, soit par rapport à l'unité, soit sur comment ont été obtenues les valeurs. Les données commencent réellement à partir de la ligne 61 avec janvier 1958

```
#data_url = "https://scrippsco2.ucsd.edu/assets/data/
#atmospheric/stations/in_situ_co2/monthly/monthly_in_situ_co2_mlo.csv"
data_csv = "monthly_in_situ_co2_mlo.csv"
data = read.csv("monthly_in_situ_co2_mlo.csv",skip=57)
head(data)
```

```
##      Yr Mn     Date   Date.1      CO2 seasonally         fit seasonally.1
## 1    NA NA                NA            adjusted             adjusted fit
## 2    NA NA    Excel       NA    [ppm]     [ppm]        [ppm]        [ppm]
## 3 1958  1    21200 1958.041   -99.99     -99.99      -99.99       -99.99
```

```
## 4 1958  2    21231 1958.126   -99.99     -99.99      -99.99       -99.99
## 5 1958  3    21259 1958.203   315.71     314.44      316.19       314.90
## 6 1958  4    21290 1958.288   317.45     315.16      317.30       314.98
##          CO2.1    seasonally.2  Sta
## 1       filled adjusted filled
## 2        [ppm]           [ppm]
## 3       -99.99          -99.99  MLO
## 4       -99.99          -99.99  MLO
## 5       315.71          314.44  MLO
## 6       317.45          315.16  MLO
```

Dans un premier temps nous allons donc récuperer l'entête des colonnes, puis charger les données sans l'entête et renommer les colonnes conformément à l'entête. Enfin, la description du fichier nous indique que les valeurs -99.99 correspondent à des données manquantes. Nous allons les remplacer par des NA et nous afficherons l'ensemble des valeurs manquantes.

```
header = read.csv("monthly_in_situ_co2_mlo.csv",skip=57,nrows=1)
data = read.csv("monthly_in_situ_co2_mlo.csv",skip=60,header=FALSE)
names(data)<-names(header)
data[data==-99.99]<-NA

na_records = apply(data, 1, function (x) {any(is.na(x))})
data[na_records,]
```

```
##       Yr Mn  Date   Date.1    CO2 seasonally    fit seasonally.1  CO2.1
## 1   1958  1 21200 1958.041     NA         NA     NA           NA     NA
## 2   1958  2 21231 1958.126     NA         NA     NA           NA     NA
## 6   1958  6 21351 1958.455     NA         NA 317.26       315.14 317.26
## 10  1958 10 21473 1958.789     NA         NA 312.42       315.40 312.42
## 74  1964  2 23422 1964.126     NA         NA 320.03       319.37 320.03
## 75  1964  3 23451 1964.205     NA         NA 320.74       319.41 320.74
## 76  1964  4 23482 1964.290     NA         NA 321.83       319.45 321.83
## 785 2023  5 45061 2023.370 423.78     420.37     NA           NA 423.78
## 786 2023  6 45092 2023.455     NA         NA     NA           NA     NA
## 787 2023  7 45122 2023.537     NA         NA     NA           NA     NA
## 788 2023  8 45153 2023.622     NA         NA     NA           NA     NA
## 789 2023  9 45184 2023.707     NA         NA     NA           NA     NA
## 790 2023 10 45214 2023.789     NA         NA     NA           NA     NA
## 791 2023 11 45245 2023.874     NA         NA     NA           NA     NA
## 792 2023 12 45275 2023.956     NA         NA     NA           NA     NA
##     seasonally.2  Sta
## 1             NA  MLO
## 2             NA  MLO
## 6         315.14  MLO
## 10        315.40  MLO
## 74        319.37  MLO
## 75        319.41  MLO
## 76        319.45  MLO
## 785       420.37  MLO
## 786           NA  MLO
## 787           NA  MLO
## 788           NA  MLO
## 789           NA  MLO
## 790           NA  MLO
## 791           NA  MLO
```

```
## 792           NA  MLO
```

Conversion des colonnes en date

Problème rencontré : les colonnes 3 et 4 sont des dates mais le format de la date n'est pas indiqué. Après avoir recherché les différentes possibilités voici la description des quatre premières colonnes :

- colonne 1 : année ;
- colonne 2 : mois ;
- colonne 3 : date en nombre de jours depuis une origine, la date d'origine étant 1899-12-30 (comme dans Excel ce qui était indiqué par le mot Excel ligne 60 du fichier) ;
- colonne 4 : date en année décimal.

Nous allons convertir la colonne 3 en date en indiquant la date 1899-12-30 comme origine.

```
class(data$Date)
```

```
## [1] "integer"
```

```
data$Date<-as.Date(data$Date, origin = "1899-12-30")
class(data$Date)
```

```
## [1] "Date"
```

```
head(data[,c("Yr","Mn","Date","CO2")])
```

```
##     Yr Mn       Date    CO2
## 1 1958  1 1958-01-15     NA
## 2 1958  2 1958-02-15     NA
## 3 1958  3 1958-03-15 315.71
## 4 1958  4 1958-04-15 317.45
## 5 1958  5 1958-05-15 317.51
## 6 1958  6 1958-06-15     NA
```

```
tail(data[,c("Yr","Mn","Date","CO2")])
```

```
##       Yr Mn       Date CO2
## 787 2023  7 2023-07-15  NA
## 788 2023  8 2023-08-15  NA
## 789 2023  9 2023-09-15  NA
## 790 2023 10 2023-10-15  NA
## 791 2023 11 2023-11-15  NA
## 792 2023 12 2023-12-15  NA
```

Cette conversion renvoie le 15 du mois ce qui est conforme à la description dans le fichier "The monthly values have been adjusted to 24:00 hours on the 15th of each month". Nous n'avons pas réussi à faire de même avec la colonne 4.

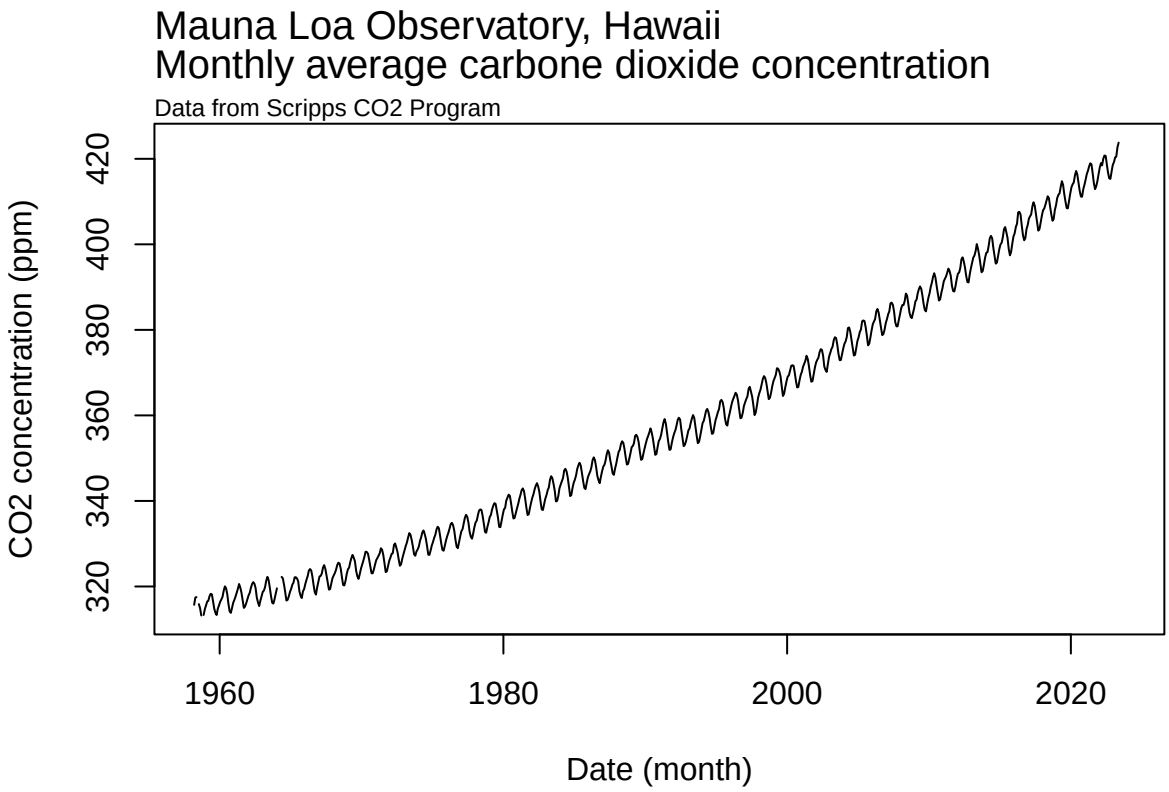
Représentations graphiques des données

Réprésentation de l'évolution de la concentration de CO2 atmosphérique en fonction du temps

D'après la description des données contenue dans le fichier csv la colonne 5 "CO2" correspond à la concentration en CO2 : "Column 5 below gives monthly Mauna Loa CO2 concentrations in micro-mol CO2 per mole (ppm)". D'après le document cité en introduction, ce ne sont pas des données brutes mais des données recalculées à une échelle mensuelle. Nous pouvons représenter ces données sur un graphe semblable à celui du site Scripps CO2 Program.

```
plot(data$Date,data$CO2,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)")
```

```
mtext("Mauna Loa Observatory, Hawaii", side=3,line=2,cex=1.25,adj=0)
mtext("Monthly average carbone dioxide concentration", side=3,line=1,cex=1.25,adj=0)
mtext("Data from Scripps CO2 Program", side=3,lin=0,cex=.75,adj=0)
```



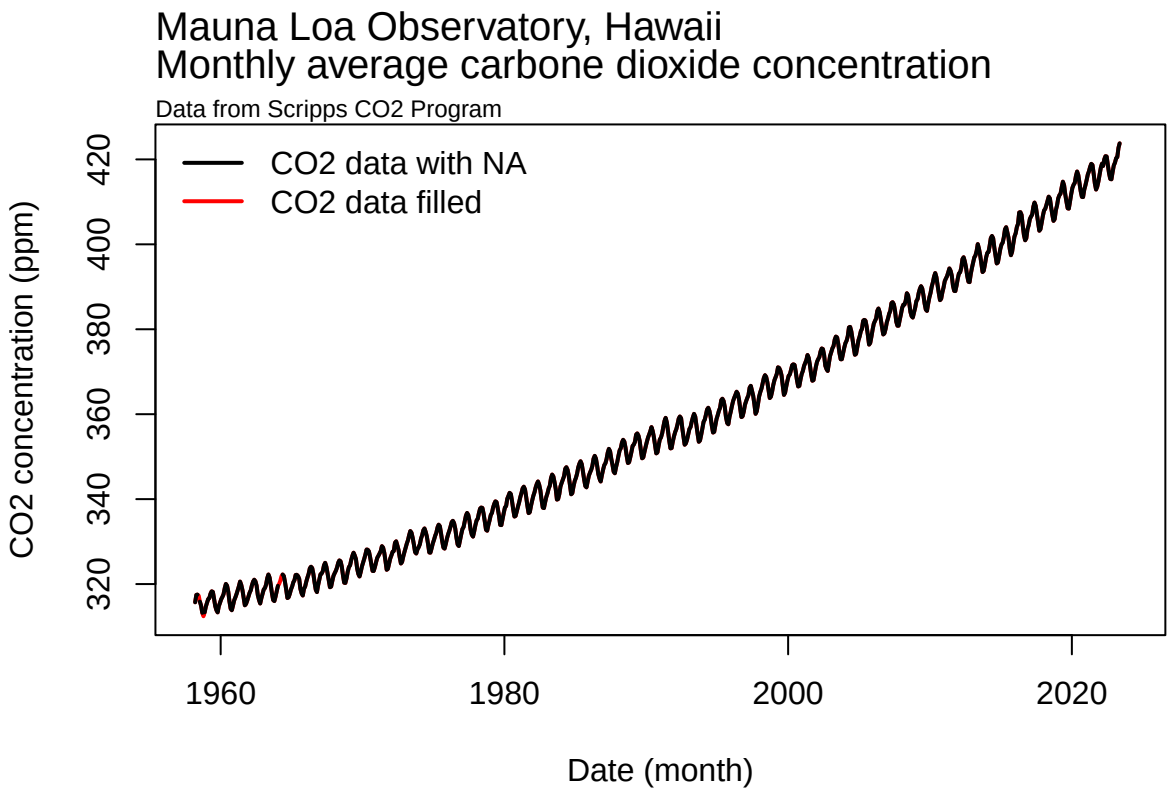
Nous voyons clairement deux structures dans les données, une forte tendance à l'augmentation au cours du temps et une oscillation des valeurs autour de cette tendance à une échelle intra-annuelle.

Caractérisation de l'oscillation périodique et de la tendance

Transformation des données en série temporelle

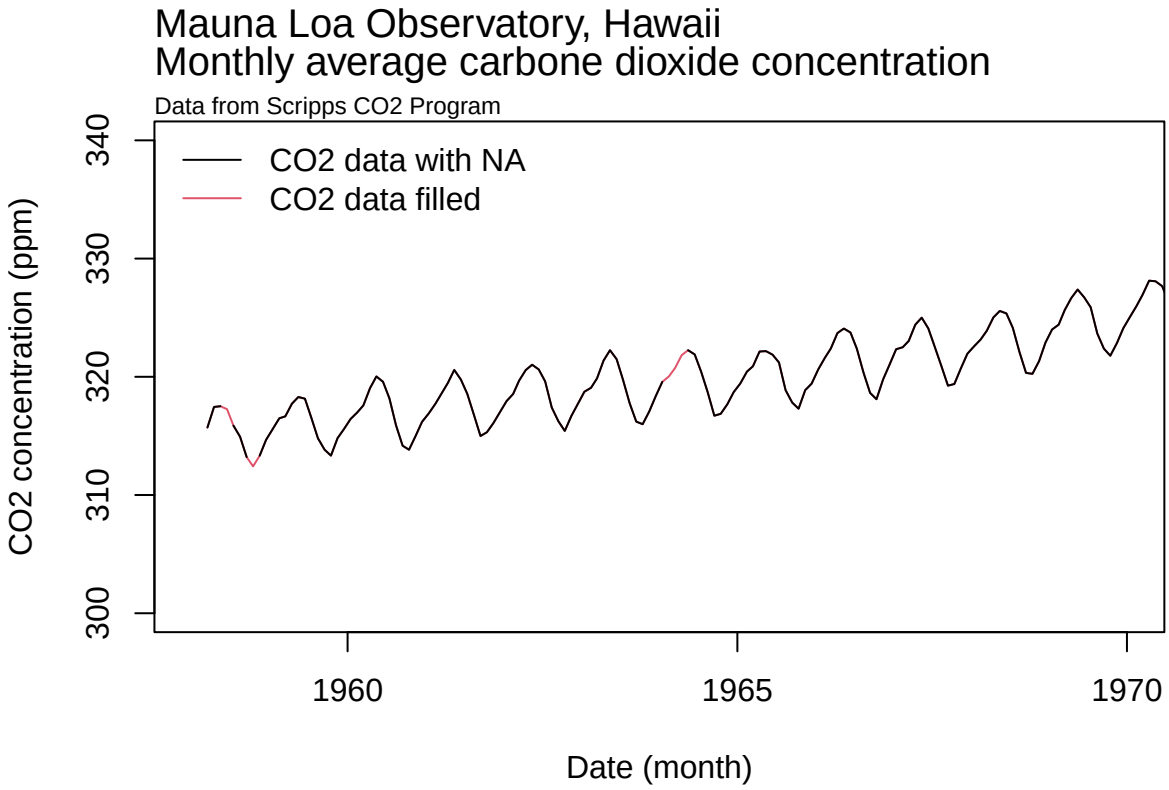
Pour caractériser ces deux composantes, nous allons considérer les données comme une série temporelle. Nous allons transformer les données en un objet ts (time series). Pour cela nous devons utiliser une série de données sans valeurs manquantes. Plutôt que la colonne 5 nous allons donc utiliser la colonne 9 "CO2.1" décrite dans le fichier csv et qui correspond aux mêmes données que la colonne 5 mais où les données manquantes ont été remplacées en utilisant une "smoothed version of the data generated from a stiff cubic spline function plus 4-harmonic functions with linear gain". Cette procédure est décrite en détail dans le rapport cité en introduction de ce document. Nous pouvons le vérifier en superposant les données remplies et les données avec données manquantes :

```
plot(data$Date,data$CO2.1,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)",col="red",lwd=2)
mtext("Mauna Loa Observatory, Hawaii", side=3,line=2,cex=1.25,adj=0)
mtext("Monthly average carbone dioxide concentration", side=3,line=1,cex=1.25,adj=0)
mtext("Data from Scripps CO2 Program", side=3,lin=0,cex=.75,adj=0)
points(data$Date,data$CO2,type="l",col=1,lwd=2)
legend("topleft",c("CO2 data with NA","CO2 data filled"),lty=1,col=c(1,"red"),lwd=2,bty="n")
```



Nous constatons qu'il y a très peu de données manquantes et qu'elles se situent toutes entre 1958 et 1964 (nous les avions déjà identifiées plus haut).

```
plot(data$Date,data$CO2.1,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)",col=2,
     xlim=as.Date(c("1958-01-01","1970-01-01")),ylim=c(300,340))
mtext("Mauna Loa Observatory, Hawaii", side=3,line=2,cex=1.25,adj=0)
mtext("Monthly average carbone dioxide concentration", side=3,line=1,cex=1.25,adj=0)
mtext("Data from Scripps CO2 Program", side=3,lin=0,cex=.75,adj=0)
points(data$Date,data$CO2,type="l",col=1)
legend("topleft",c("CO2 data with NA","CO2 data filled"),lty=1,col=c(1,2),bty="n")
```



Il nous faut également supprimer les premières et dernières données (manquantes) allant de janvier 1958 à février 1958 puis de juin 2023 à décembre 2023.

```
data.trunc<-data[!is.na(data$CO2.1),]
head(data.trunc[,c("Date","CO2","CO2.1")],10)
```

```
##          Date    CO2  CO2.1
## 3  1958-03-15 315.71 315.71
## 4  1958-04-15 317.45 317.45
## 5  1958-05-15 317.51 317.51
## 6  1958-06-15     NA 317.26
## 7  1958-07-15 315.87 315.87
## 8  1958-08-15 314.93 314.93
## 9  1958-09-15 313.21 313.21
## 10 1958-10-15     NA 312.42
## 11 1958-11-15 313.33 313.33
## 12 1958-12-15 314.67 314.67
```

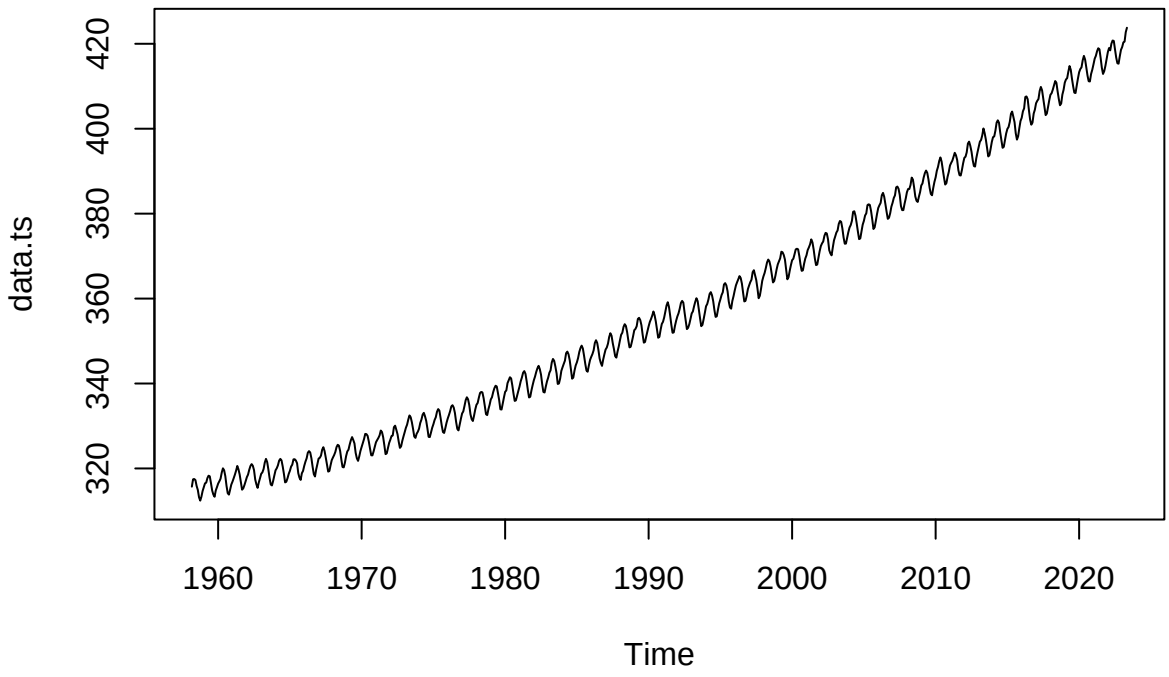
```
tail(data.trunc[,c("Date","CO2","CO2.1")],10)
```

```
##           Date    CO2  CO2.1
## 776 2022-08-15 416.76 416.76
## 777 2022-09-15 415.41 415.41
## 778 2022-10-15 415.31 415.31
## 779 2022-11-15 417.04 417.04
## 780 2022-12-15 418.57 418.57
## 781 2023-01-15 419.24 419.24
## 782 2023-02-15 420.33 420.33
```

```
## 783 2023-03-15 420.51 420.51
## 784 2023-04-15 422.73 422.73
## 785 2023-05-15 423.78 423.78
```

Nous pouvons maintenant transformer la colonne 9 en un objet ts. Nous commençons par trier les données pour être sûr que les lignes sont bien ordonnées selon la date. La série temporelle correspond à des données mensuelles sur plusieurs années. La première donnée est le mois de mars 1958. La fin est le mois de mai 2023. Nous créons notre objet ts en indiquant le début et la fréquence de la série.

```
data.trunc<-data.trunc[order(data.trunc$Date),]
data.ts<-ts(data=data.trunc$CO2.1,start=c(1958,3),frequency=12)
plot(data.ts)
```



Décomposition de la série temporelle

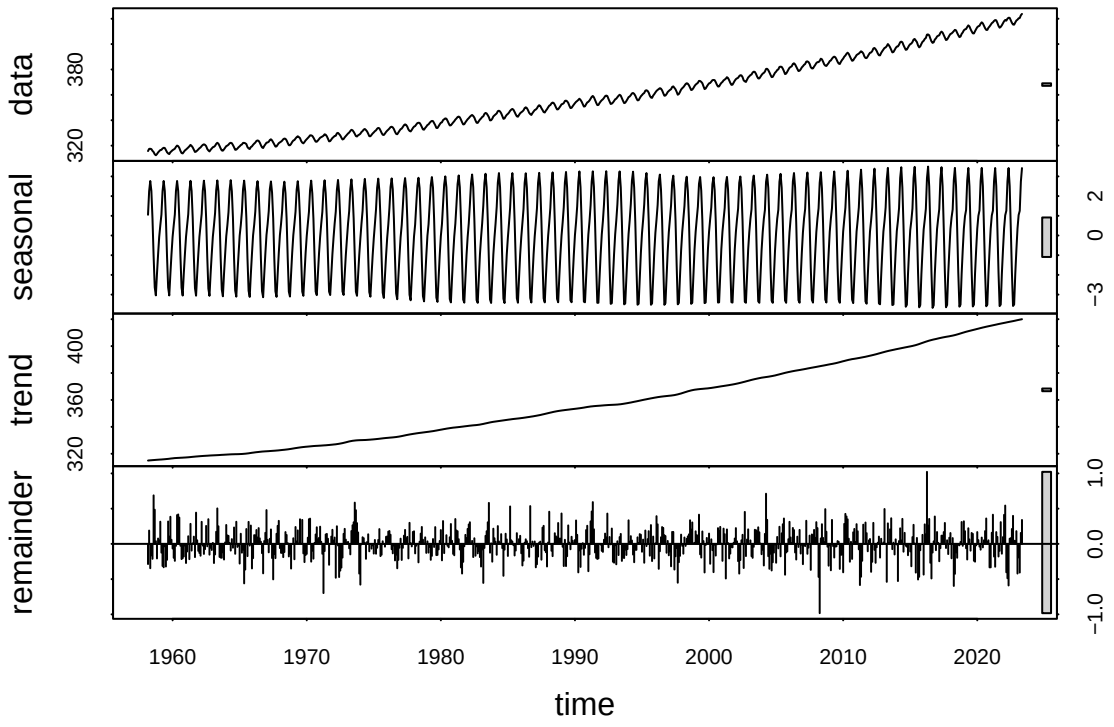
La fonction stl() du package stats permet de décomposer la série temporelle en utilisant un modèle loess pour ajuster la tendance. Nous avons fixé le paramètre s.window à 13 ce qui signifie que la fonction utilise une fenêtre de 13 annés consécutives pour estimer chaque valeur de la composante saisonnière. Nous avons préféré cela plutôt que la valeur par défaut ("periodic") qui considère que la composante saisonnière ne varie pas au cours du temps. La fonction renvoie la composante saisonnière, la tendance et les résidus.

```
require(stats)
#decom.stl<-stl(data.ts,s.window="periodic")
decom.stl<-stl(data.ts,s.window=13)
summary(decom.stl)
```

```
##  Call:
##  stl(x = data.ts, s.window = 13)
```

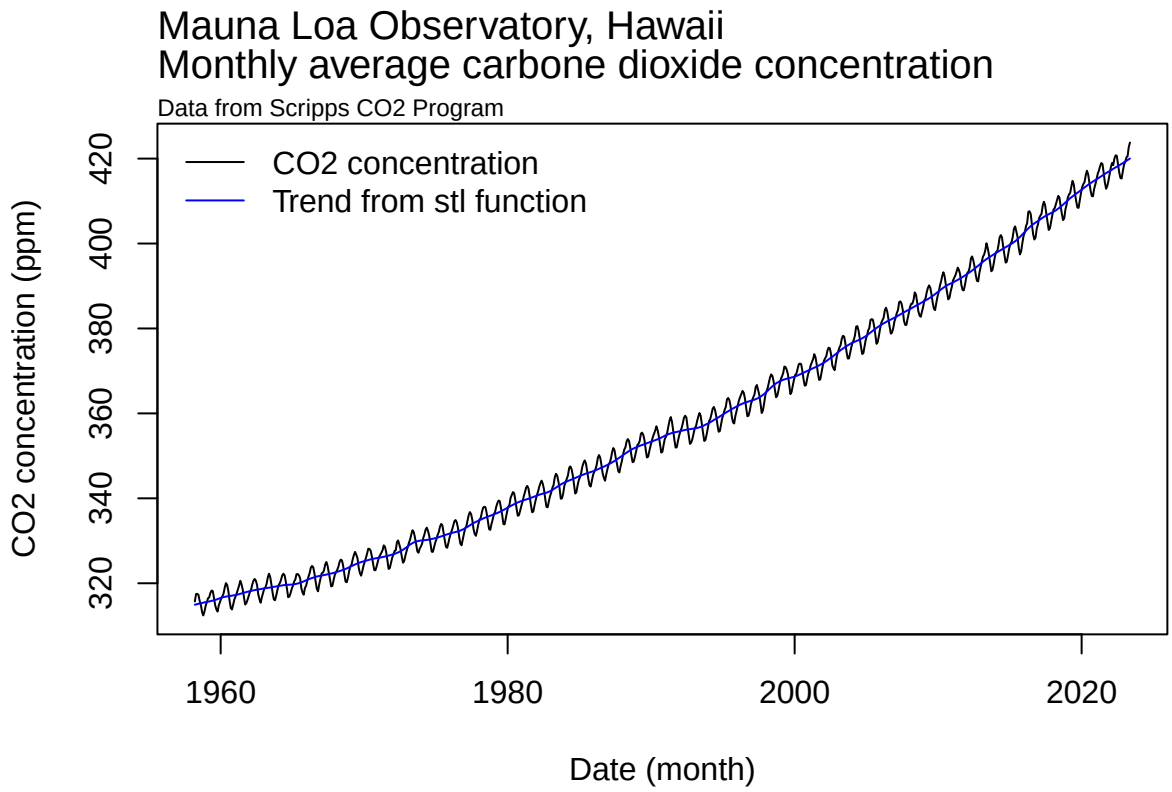
```
##
##  Time.series components:
##     seasonal             trend            remainder
##  Min.   :-3.674953   Min.   :314.9440   Min.   :-0.9832754
##  1st Qu.:-1.880815   1st Qu.:330.2705   1st Qu.:-0.1494056
##  Median : 0.350430   Median :354.4387   Median : 0.0077596
##  Mean   : 0.009404   Mean   :357.9442   Mean   :-0.0010652
##  3rd Qu.: 2.142851   3rd Qu.:382.7630   3rd Qu.: 0.1417021
##  Max.   : 3.500580   Max.   :420.0169   Max.   : 1.0224083
##  IQR:
##      STL.seasonal STL.trend STL.remainder data
##       4.0237      52.4925    0.2911       52.4750
##    %   7.7         100.0       0.6         100.0
##
##  Weights: all == 1
##
##  Other components: List of 5
##  $ win  : Named num [1:3] 13 21 13
##  $ deg  : Named int [1:3] 0 1 1
##  $ jump : Named num [1:3] 2 3 2
##  $ inner: int 2
##  $ outer: int 0
```

```
plot(decom.stl)
```



Nous pouvons récupérer les différentes composantes de la série temporelle et représenter les données avec la tendance.

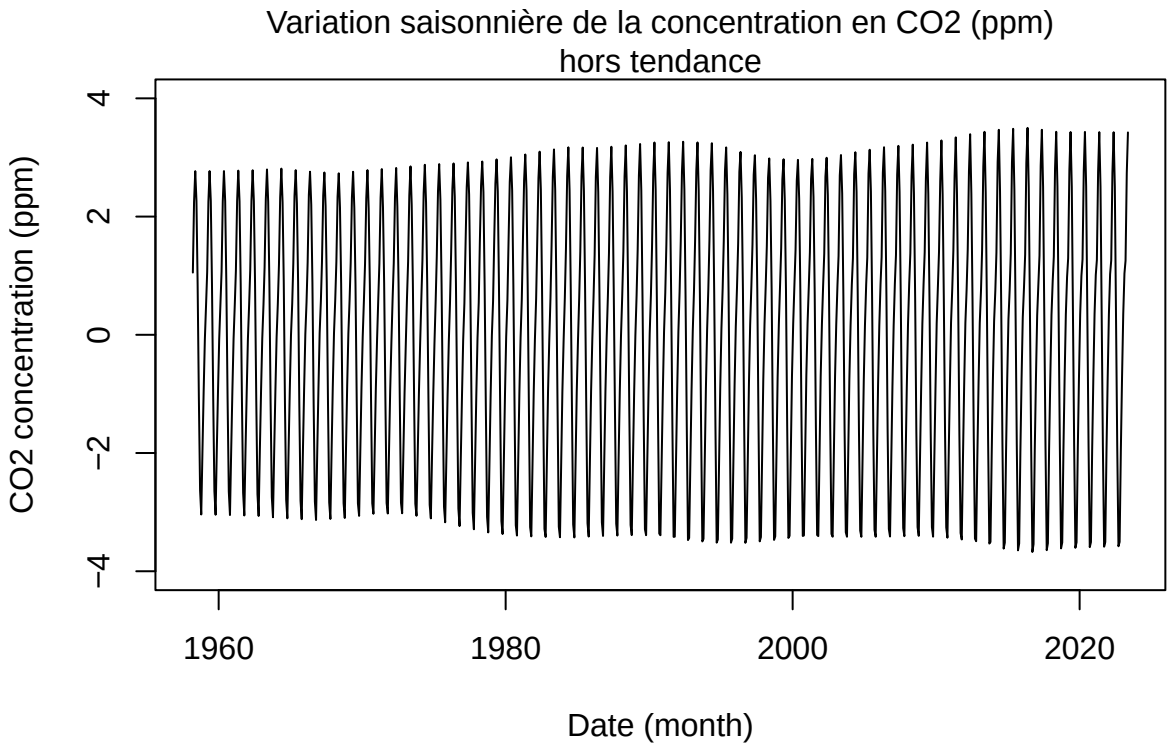
```
data.trunc$seasonSTL<-as.numeric(decom.stl[1]$time.series[,1])
data.trunc$trendSTL<-as.numeric(decom.stl[1]$time.series[,2])
data.trunc$residSTL<-as.numeric(decom.stl[1]$time.series[,3])

plot(data.trunc$Date,data.trunc$CO2.1,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)")
mtext("Mauna Loa Observatory, Hawaii", side=3,line=2,cex=1.25,adj=0)
mtext("Monthly average carbone dioxide concentration", side=3,line=1,cex=1.25,adj=0)
mtext("Data from Scripps CO2 Program", side=3,lin=0,cex=.75,adj=0)
points(data.trunc$Date,data.trunc$trendSTL,col="blue",type="l")
legend("topleft",c("CO2 concentration",
                   "Trend from stl function"),lty=1,col=c(1,"blue"),bty="n")
```



Caractérisation de la saisonalité

Nous pouvons également représenter la saisonnalité seule.

```
plot(data.trunc$Date,data.trunc$seasonSTL,type="l",xlab="Date (month)",
     ylab="CO2 concentration (ppm)",ylim=c(-4,4))
mtext("Variation saisonnière de la concentration en CO2 (ppm)",line=1)
mtext("hors tendance",line=0)
```



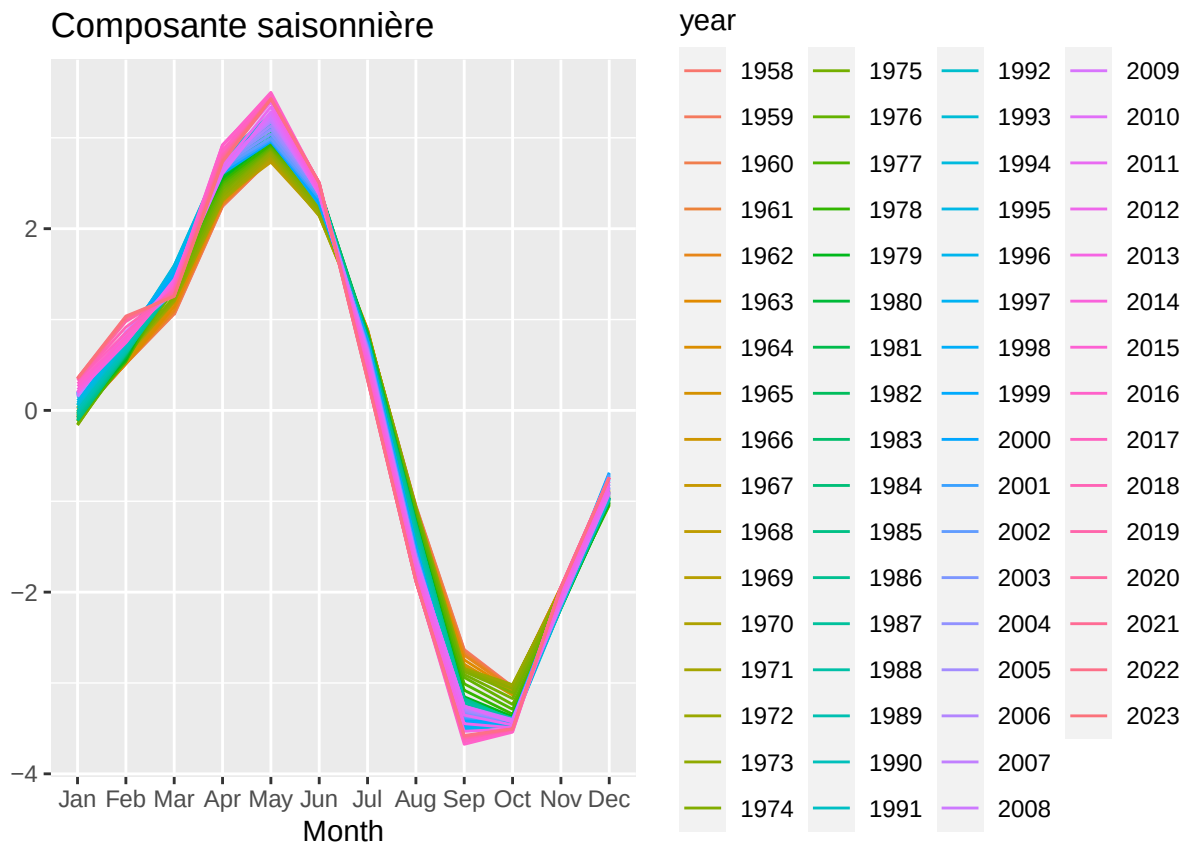
Nous pouvons donner quelques statistiques de cette composante saisonnières. Dans une année le CO2 atmosphérique peut varier de -3.67 ppm à 3.5 ppm par rapport à la moyenne annuelle. L'écart-type de la composante saisonnière est de 2.12 ppm. La variance de la composante saisonnière n'est pas constante au cours du temps. Nous voyons sur le graphe précédent que cette variation intra-annuelle augmente légèrement au cours du temps.

Pour mieux voir cet effet saisonnier nous pouvons le représenter en superposant toutes les années dans un graphe annuel avec la fonction ggseasonplot() du packages forecast. Pour mieux identifier où se situe les pics et les creux nous allons faire cette représentation à partir de la composante saisonnière obtenue avec la fonction stl().

```
#install.packages("forecast")
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
ggseasonplot(decom.stl[1]$time.series[,1],main="Composante saisonnière")
```



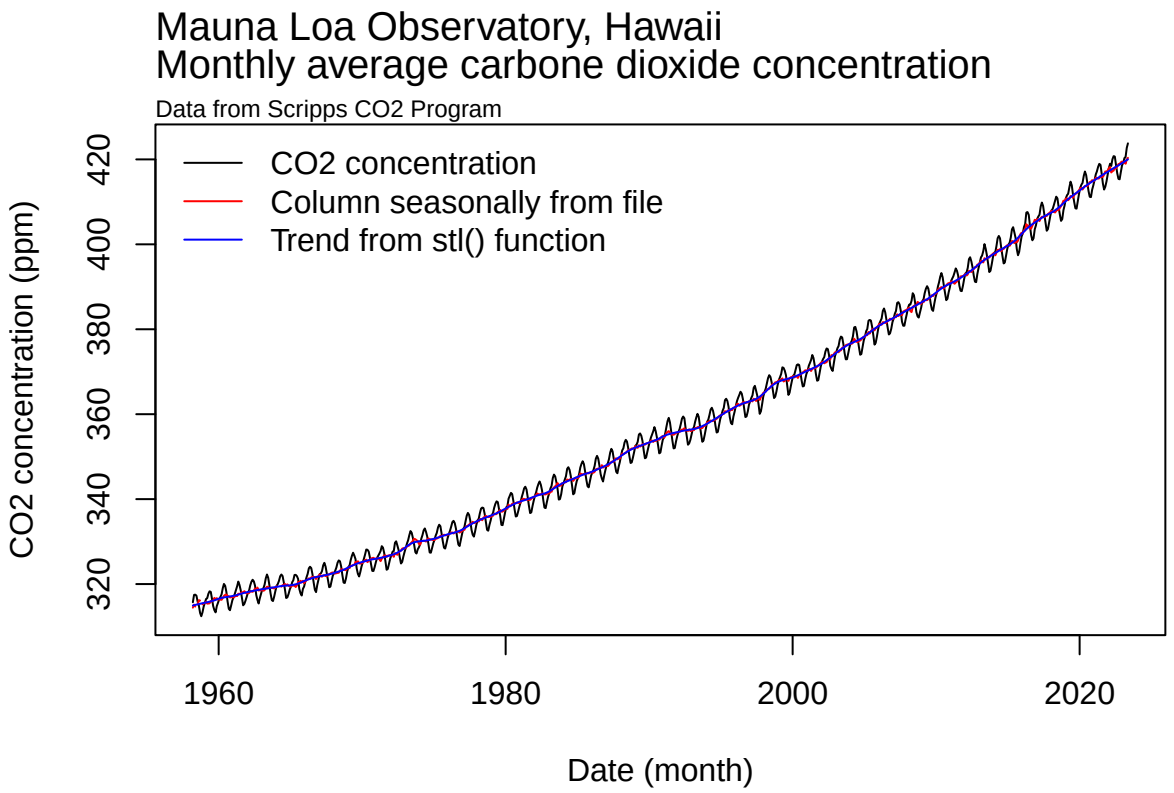
Nous voyons qu'au sein d'une année le pic de concentration de CO2 a lieu vers le mois de mai et les valeurs les plus basses sont vers le mois de septembre octobre. Les données reflètent les variations de la concentration de CO2 atmosphérique dans l'hemisphère nord. Dans l'hemisphère nord, à partir du printemps, les plantes captent le CO2 atmosphérique expliquant une partie de ce cycle saisonnier (voir le document cité en introduction pour plus de détails).

Caractérisation de la tendance

La tendance obtenue avec la fonction stl() nous permet de donner quelques chiffres sur l'évolution de la concentration en CO2 atmosphérique. Entre 1958 et 2023 la concentration de CO2 atmosphérique est passé de 314.94 ppm à 420.02 ppm, soit une augmentation de 33.4% en 65 ans. Cette augmentation est principalement lié à l'utilisation de carburant fossile par l'homme (voir le document cité en introduction pour plus de détails).

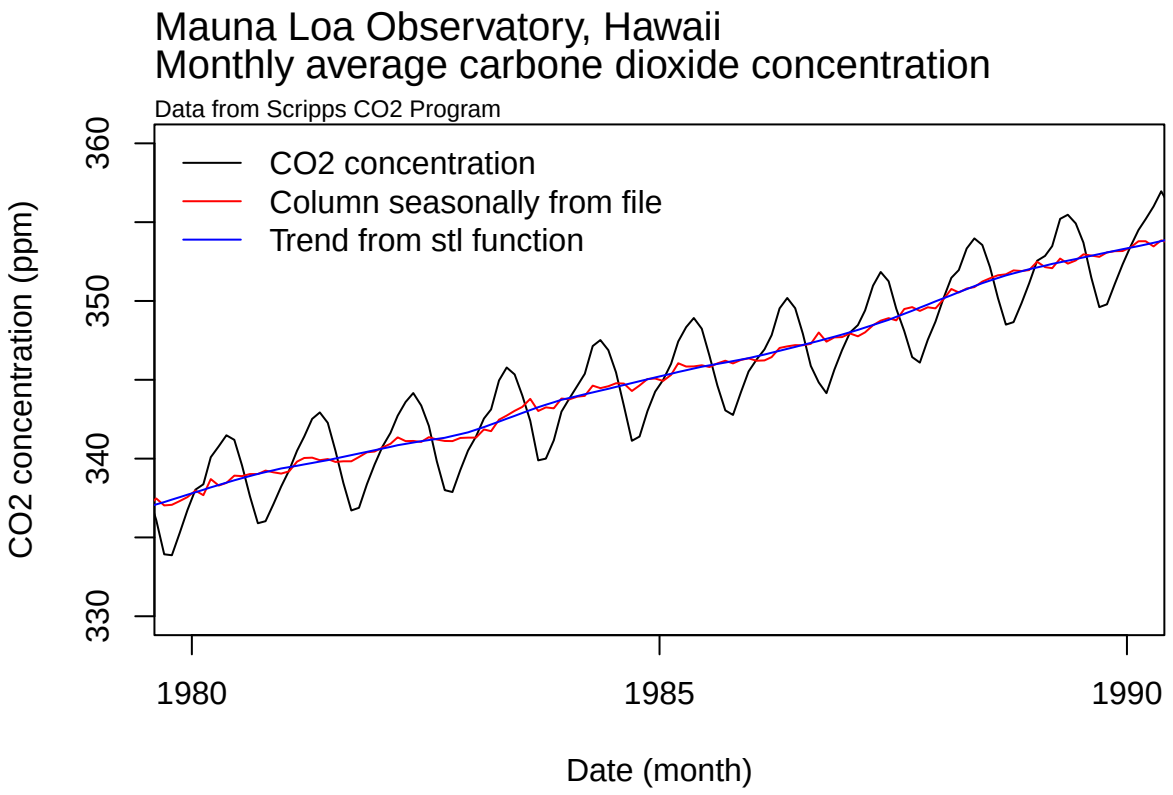
La tendance obtenue avec la fonction stl() correspond au même type de données que la colonne 6 "seasonally" : "Column 6 gives the same data after a seasonal adjustment to remove the quasi-regular seasonal cycle". Nous pouvons le vérifier en superposant les deux variables sur un même graphe.

```
plot(data.trunc$Date,data.trunc$CO2.1,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)")
mtext("Mauna Loa Observatory, Hawaii", side=3,line=2,cex=1.25,adj=0)
mtext("Monthly average carbone dioxide concentration", side=3,line=1,cex=1.25,adj=0)
mtext("Data from Scripps CO2 Program", side=3,lin=0,cex=.75,adj=0)
points(data.trunc$Date,data.trunc$seasonally,col="red",type="l")
points(data.trunc$Date,data.trunc$trendSTL,col="blue",type="l")
legend("topleft",c("CO2 concentration","Column seasonally from file",
                   "Trend from stl() function"),lty=1,col=c(1,"red","blue"),bty="n")
```



Cependant, en zoomant nous voyons que la tendance que nous obtenons avec la fonction stl() correspond à un lissage plus fort que la colonne 6 “seasonally”. Les méthodes utilisées ne sont pas les mêmes.

```
plot(data.trunc$Date,data.trunc$CO2.1,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)",xlim=as.Date(c("1980-01-01","1990-01-01")),
     ylim=c(330,360))
mtext("Mauna Loa Observatory, Hawaii", side=3,line=2,cex=1.25,adj=0)
mtext("Monthly average carbone dioxide concentration", side=3,line=1,cex=1.25,adj=0)
mtext("Data from Scripps CO2 Program", side=3,lin=0,cex=.75,adj=0)
points(data.trunc$Date,data.trunc$seasonally,col="red",type="l")
points(data.trunc$Date,data.trunc$trendSTL,col="blue",type="l")
legend("topleft",c("CO2 concentration","Column seasonally from file",
                   "Trend from stl function"),lty=1,col=c(1,"red","blue"),bty="n")
```



Prédiction de la concentration de CO2 jusqu'en 2025

Modélisatin de la tendance avec un modèle simple

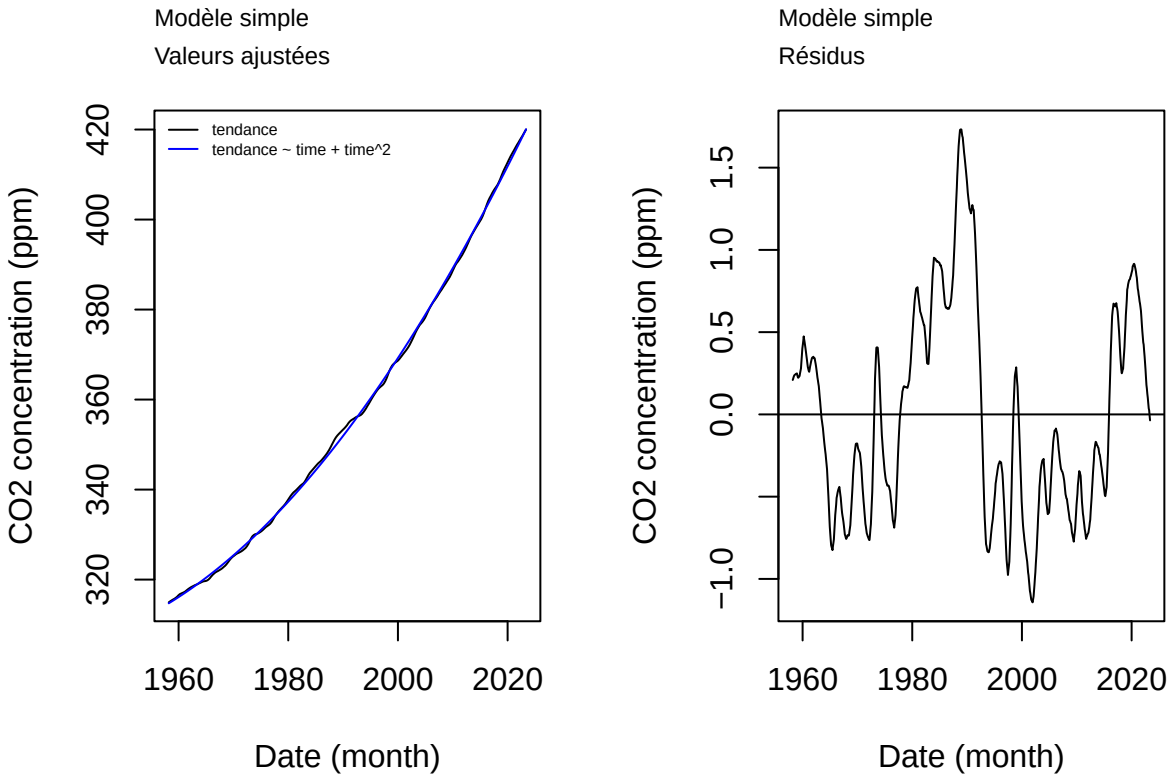
Lorsque l'on zoome sur les données comme sur le graphe précédent, un modèle linéaire semble convenir pour représenter la tendance d'augmentation du CO2 dans le temps. Cependant en observant l'ensemble des données il apparaît clairement que la concentration de CO2 augmente de plus en plus vite. Un premier modèle simple pourrait être d'ajuster un modèle linéaire ayant une composante linéaire et une composante quadratique (un polynôme de degré 2). Pour simplifier nous allons ajuster ce modèle sur la tendance extraite précédemment avec la fonction stl(), variable nommée trendSTL dans le data frame. Pour ajuster le modèle nous allons convertir le champ date en numérique et utiliser cette variable comme variable explicative du modèle. Nous nommons cette variable "time".

```
data.trunc$time<-as.numeric(data.trunc$Date)
```

Nous ajustons le modèle et récupérons les valeurs ajustées et les résidus que nous représentons.

```
modeleSimple0<-lm(data=data.trunc, trendSTL ~ time + I((time)^2))

data.trunc$modeleSimple0Fit<-fitted(modeleSimple0)
data.trunc$modeleSimple0Res<-resid(modeleSimple0)

par(mfrow=c(1,2))
plot(data.trunc$Date,data.trunc$trendSTL,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)")
mtext("Modèle simple", side=3,line=2,cex=0.75,adj=0)
mtext("Valeurs ajustées", side=3,line=1,cex=0.75,adj=0)
points(data.trunc$Date,data.trunc$modeleSimple0Fit,col="blue",type="l")
```

```
legend("topleft",c("tendance","tendance ~ time + time^2"),lty=1,
       col=c("black","blue"),bty="n",cex=0.5)
plot(data.trunc$Date,data.trunc$modeleSimple0Res,type="l",main="",
     xlab="Date (month)",ylab="CO2 concentration (ppm)")
mtext("Modèle simple", side=3,line=2,cex=0.75,adj=0)
mtext("Résidus", side=3,line=1,cex=0.75,adj=0)
abline(h=0)
```



Ce modèle est simpliste car, par exemple, nous n'avons pas tenu compte des observations répétées par année, c'est à dire de la présence de pseudoréplications, ni de l'autocorrélation temporelle. Néanmoins, même s'il n'est pas parfait, ce modèle très simple semble bien convenir aux données.

Prédiction jusqu'en 2025 à l'aide du modèle simple

Pour prédire la concentration de CO2 en 2025, nous allons utiliser le modèle précédent et l'appliquer jusqu'en 2025. Au delà de mai 2023, nous n'avons plus de données. Il s'agira d'extrapolations à l'aide du modèle simple. Nous utilisons la fonction predict.lm() qui permet d'obtenir l'intervalle de confiance des prédictions.

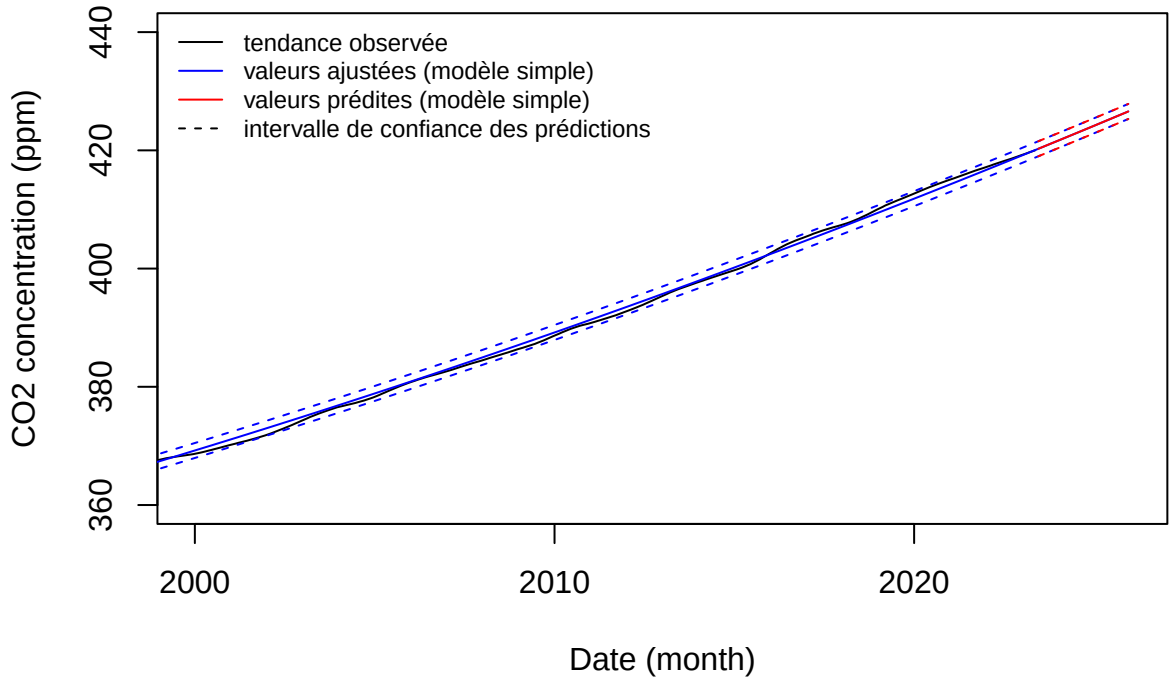
Nous commençons par construire un data.frame avec une colonne "time" sur l'ensemble de la période de mars 1958 à décembre 2025. Puis nous utilisons ce data.frame pour prédire les valeurs de CO2 à l'aide du modèle simple et de la fonction predict.lm(). Nous créons un data.frame spécifiquement pour les données au-delà de mai 2023 qui correspond aux valeurs prédites extrapolées.

```
data.trunc$time<-as.numeric(data.trunc$Date)
myNewData<-data.frame(time=as.numeric(seq.Date(from=as.Date("1958-03-15"),
                                    to=as.Date("2025-12-15"),by="month")))
predictions<-data.frame(predict.lm(modeleSimple0,myNewData,
```

```
                                    interval="prediction",type="response"))
myNewData<-cbind(myNewData,predictions)
myNewDataExtrapolation<-myNewData[myNewData$time > as.numeric(as.Date("2023-05-15")),]
```

Le graphe suivant représente la tendance “observée” sur la période connue, les valeurs de la tendance obtenues par ajustement du modèle simple et enfin les valeurs de la tendance prédite au-delà de mai 2023.

```
plot(data.trunc$Date,data.trunc$trendSTL,type="l",main="",xlab="Date (month)",
     ylab="CO2 concentration (ppm)",col="black",xlim=as.Date(c("2000-01-01","2026-01-01")),
     ylim=c(360,440))
points(myNewData$time,myNewData$fit,col="blue",type="l")
points(myNewData$time,myNewData$lwr,col="blue",type="l",lty=2)
points(myNewData$time,myNewData$upr,col="blue",type="l",lty=2)
points(myNewDataExtrapolation$time,myNewDataExtrapolation$fit,col="red",type="l")
points(myNewDataExtrapolation$time,myNewDataExtrapolation$lwr,col="red",type="l",lty=2)
points(myNewDataExtrapolation$time,myNewDataExtrapolation$upr,col="red",type="l",lty=2)
legend("topleft",c("tendance observée","valeurs ajustées (modèle simple)",
                   "valeurs prédites (modèle simple)",
                   "intervalle de confiance des prédictions"),
       col=c("black","blue","red","black"),lty=c(1,1,1,2),bty="n",cex=0.75)
```



Nous pouvons estimer à l’aide du modèle simple la concentration moyenne attendue en CO2 pour l’année 2025. Pour cela nous récupérons dans le data frame des extrapolations les données de l’année 2025 et nous calculons quelques statistiques pour l’année 2025.

```
myNewData2025<-myNewDataExtrapolation[myNewDataExtrapolation$time >=
               as.numeric(as.Date("2025-01-01")) &
```

```
myNewDataExtrapolation$time <= as.numeric(as.Date("2025-12-31")),]
dim(myNewData2025)
```

```
## [1] 12  4
```

```
round(mean(myNewData2025$fit),2)
```

```
## [1] 425.41
```

```
round(mean(myNewData2025$fit)-mean(myNewData2025$lwr),2);
```

```
## [1] 1.27
```

```
round(mean(myNewData2025$fit)-mean(myNewData2025$upr),2)
```

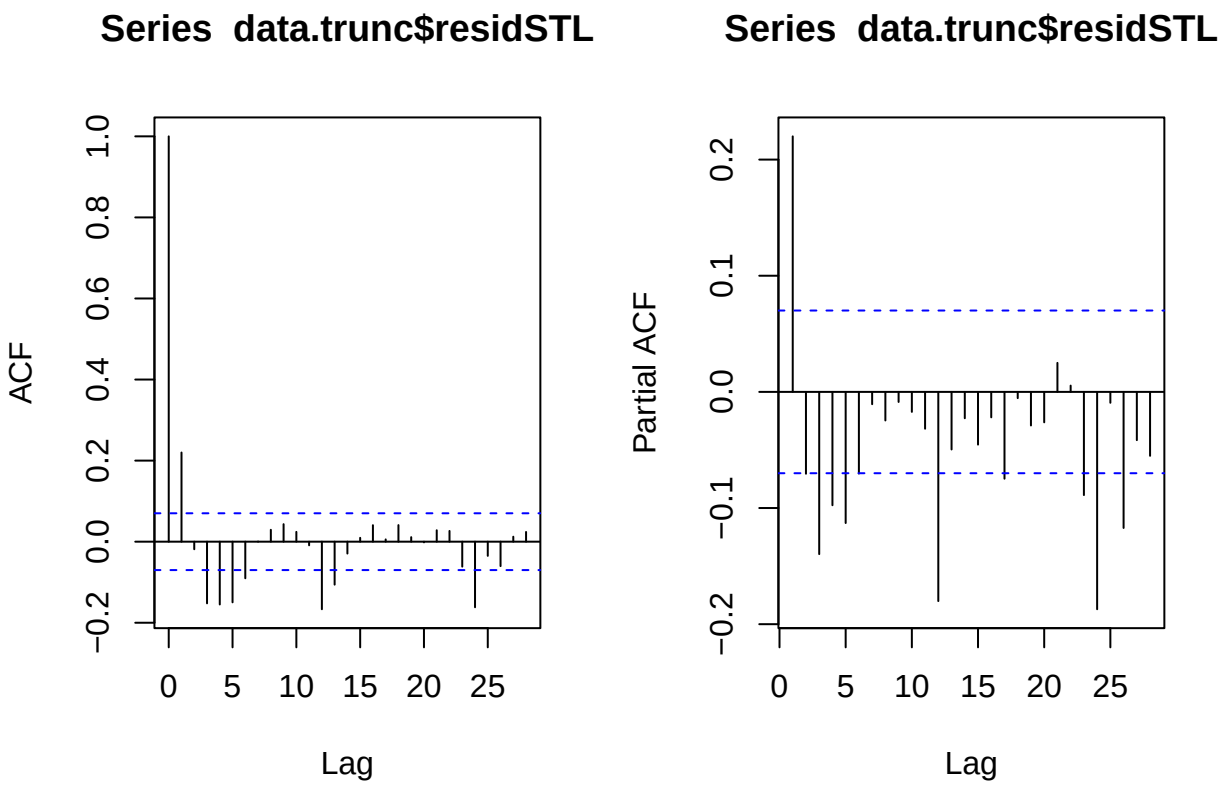
```
## [1] -1.27
```

D'après le modèle simple, en première approximation la valeur de concentration moyenne en CO2 en 2025 serait de 425.41 ppm plus ou moins 1.27 ppm.

Modélisation à l'aide d'un modèle SARIMA

Pour aller un peu plus loin nous allons tenter d'utiliser un modèle stochastique de type SARIMA ("Seasonal Autoregressive Integrated Moving Average") pour faire les prédictions. En effet, nous avons vu précédemment que la série temporelle présentait à la fois une tendance et une saisonnalité. Nous pouvons également vérifier que les résidus obtenus avec la fonction stl() présentent une autocorrélation et une autocorrélation partielle à l'aide des fonctions acf et pcaf.

```
par(mfrow=c(1,2))
acf(data.trunc$residSTL)
pacf(data.trunc$residSTL)
```



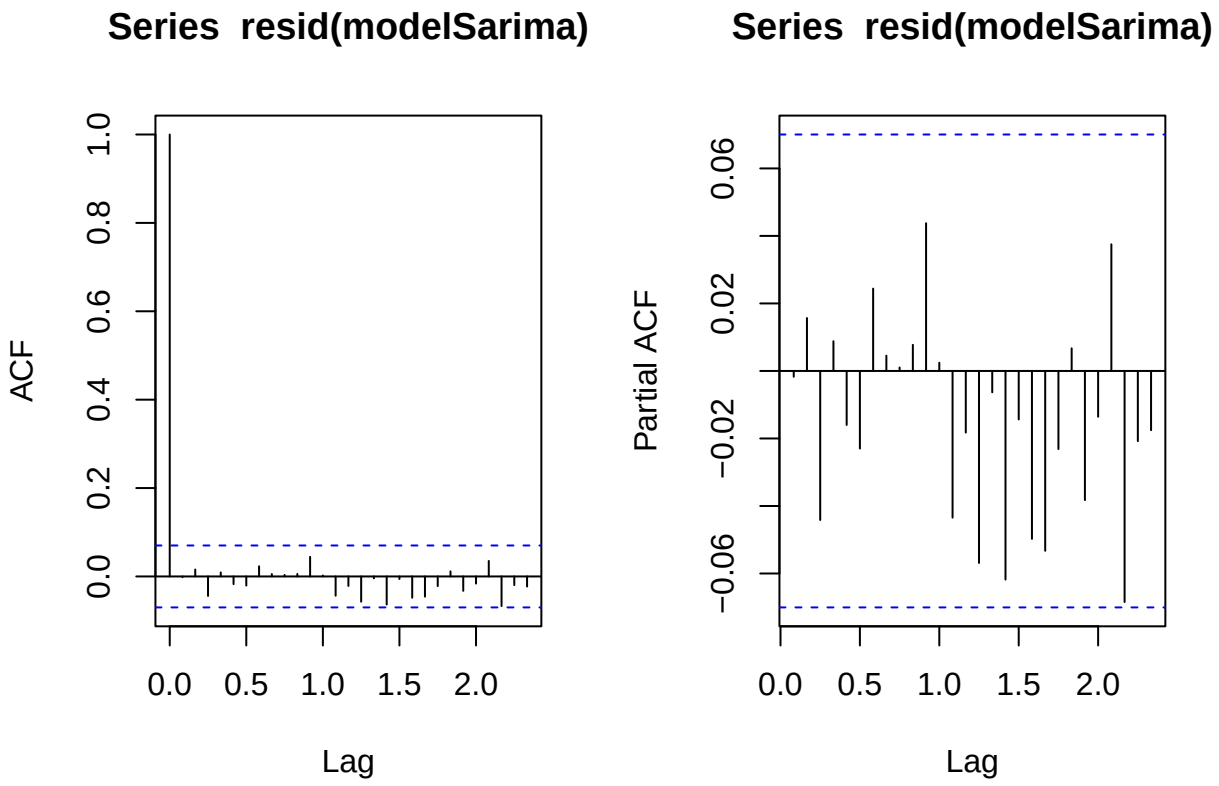
Pour simplifier le processus de construction du modèle nous allons utiliser la fonction auto.arima() du package forecast qui permet de choisir le type de modèle approprié et d'estimer les différents paramètres du modèle stochastique. Même si cela rallonge le temps de calcul, nous fixons les paramètres stepwise et approximation à FALSE pour obtenir le meilleur modèle possible.

```
#install.packages("forecast")
library(forecast)

modelSarima<-auto.arima(data.ts,approximation=FALSE,stepwise=FALSE)
summary(modelSarima)
```

```
## Series: data.ts
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.2609  -0.6228  -0.8561
## s.e.  0.0794   0.0639   0.0186
##
## sigma^2 = 0.09726:  log likelihood = -195.23
## AIC=398.46   AICc=398.51   BIC=417.04
##
## Training set error measures:
##                       ME      RMSE       MAE         MPE       MAPE     MASE
## Training set 0.02416939 0.3086648 0.2419771 0.006575062 0.06769306 0.149703
##                       ACF1
## Training set -0.001729324
```

Nous pouvons vérifier qu'il n'y a plus d'autocorrélation dans les résidus du modèle SARIMA.

```
par(mfrow=c(1,2))
acf(resid(modelSarima))
pacf(resid(modelSarima))
```



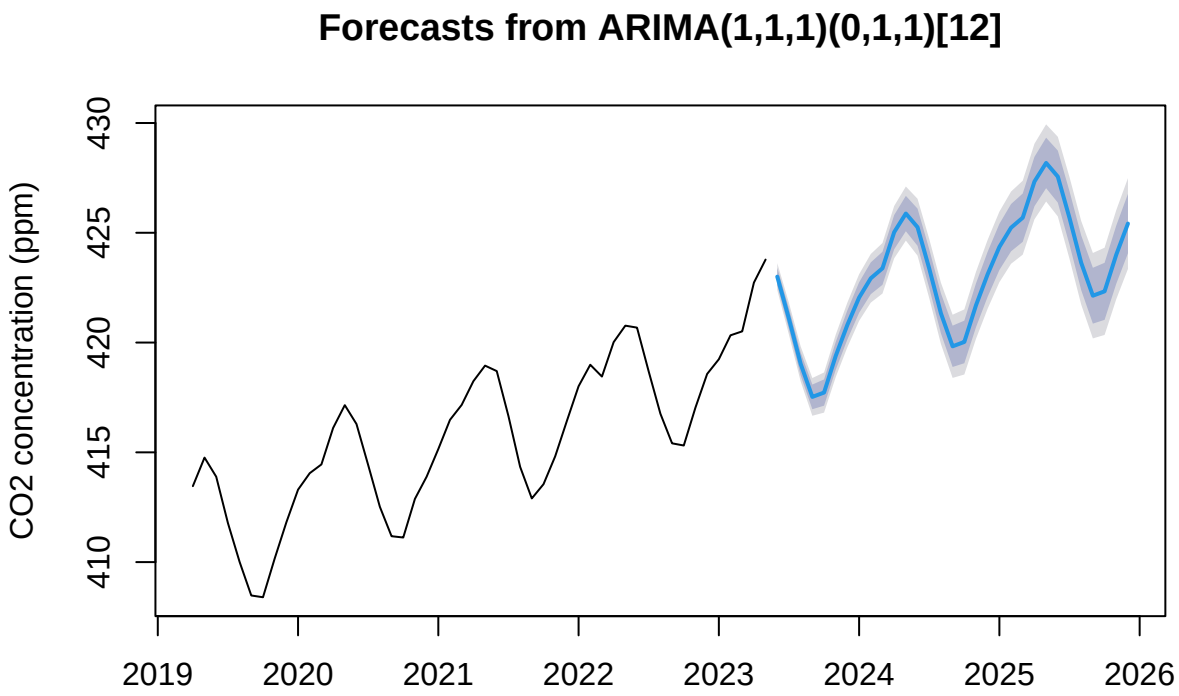
Pour prédire la concentration de CO2 jusqu'en 2025 nous allons utiliser le modèle SARIMA et la fonction forecast() du package forecast. Pour aller jusqu'à décembre 2025 nous allons faire des prédictions sur 31 valeurs supplémentaires (de juin 2023 à décembre 2025).

```
predictionsSARIMA<-data.frame(forecast(modelSarima,h=31))
tail(predictionsSARIMA)
```

```
##           Point.Forecast    Lo.80    Hi.80    Lo.95    Hi.95
## Jul 2025       425.6725 424.4567 426.8882 423.8131 427.5318
## Aug 2025       423.6345 422.3903 424.8787 421.7316 425.5374
## Sep 2025       422.1342 420.8625 423.4060 420.1892 424.0792
## Oct 2025       422.3330 421.0343 423.6316 420.3469 424.3191
## Nov 2025       424.0034 422.6785 425.3284 421.9771 426.0298
## Dec 2025       425.4190 424.0683 426.7698 423.3532 427.4848
```

Nous pouvons représenter les prédictions à l'aide de la fonction plot.forecast() du package forecast qui représente les prédictions avec les intervalles de confiance à 80 et 95%. La fonction représente également les données qui ont servi à construire le modèle. Pour plus de lisibilité nous choisissons de ne représenter que les 50 dernières valeurs passées avec le paramètre include=50.

```
plot(forecast(modelSarima,h=31),include=50,ylab="CO2 concentration (ppm)")
```



De la même façon que pour le modèle simple nous pouvons estimer la concentration de CO2 en 2025. Nous récupérons les dates dans le data frame des prédictions du modèle SARIMA et nous créons un data.frame uniquement pour 2025. Les dates sont contenues dans le nom des lignes avec un format utilisant une abréviation du mois en anglais suivi de l'année. Nous devons donc fixer la langue des dates en anglais pour récupérer la date dans le bon format puis la remettre en français.

```
Sys.setlocale("LC_TIME", "English")
```

```
## [1] "English_United States.1252"
```

```
predictionsSARIMA$Date<-as.Date(paste("15 ",row.names(predictionsSARIMA),sep=""),format="%d %b %Y")
Sys.setlocale("LC_TIME", "French")
```

```
## [1] "French_France.1252"
```

```
predictionsSARIMA2025<-predictionsSARIMA[predictionsSARIMA$Date >= as.Date("2025-01-01") &
predictionsSARIMA$Date <= as.Date("2025-12-31"),]
```

D'après le modèle SARIMA, la valeur de concentration moyenne en CO2 en 2025 serait de 425.13 ppm plus ou moins 1.83 ppm.

Le modèle SARIMA prédit donc une concentration de CO2 très proche, très légèrement inférieure, par rapport au modèle simple polynomial. L'intervalle de confiance des prédictions est légèrement plus large avec le modèle SARIMA.

Conclusion

Dans cette analyse nous avons chargé un jeu de données correspondant à l'évolution de la concentration en CO2 atmosphérique provenant du site de Mauna Loa sur l'île d'Hawaii. Pour construire le jeu de données à

partir du fichier csv nous avons exclu des lignes correspondant à des informations autres que les données. Nous avons également géré des données manquantes codées par une valeur numérique. Enfin nous avons convertit une colonne en un format date reconnu par R. Cette dernière étape a demandé un peu de temps car les informations sur le format des colonnes représentant la date n'étaient pas précises. Une fois la préparation du jeu de données réalisée nous avons représenté les données et constaté la présence d'une tendance et d'une saisonnalité fortes. Nous avons caractérisé les oscillations périodiques et la tendance en considérant les données comme une serie temporelle. Pour cela nous avons utilisé les fonction ts() et stl() du package stats. Ensuite, pour prédire la concentration en CO2 jsuqu'en 2025, nous avons construit un premier modèle simple de la tendance à l'aide d'un modèle linéaire correspondant à un polynôme de degré 2 du temps. Nous avons ensuite proposé un modèle stochastique SARIMA plus complexe mais plus adapté aux données qui permet de prédire la tendance et la saisonnalité. Pour cela nous avons utilisé les fonctions auto.arima() et forecast() du package forecast. Les deux approches donnent des résultats très similaires probablement parceque nous avons fait des prédictions à très court terme. L'avantage du modèle SARIMA est qu'il peut prédire également les variations saisonnières de la concentration de CO2 atmosphérique.

Informations sur l'environnement de travail

```
sessionInfo()
```

```
## R version 4.3.0 (2023-04-21 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=French_France.utf8  LC_CTYPE=French_France.utf8
## [3] LC_MONETARY=French_France.utf8 LC_NUMERIC=C
## [5] LC_TIME=French_France.1252
##
## time zone: Europe/Paris
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] forecast_8.21
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.3      compiler_4.3.0    highr_0.10        Rcpp_1.0.10
##  [5] parallel_4.3.0    scales_1.2.1      yaml_2.3.7        fastmap_1.1.1
##  [9] lattice_0.21-8    ggplot2_3.4.2     R6_2.5.1          labeling_0.4.2
## [13] generics_0.1.3    curl_5.0.0        lmtest_0.9-40     knitr_1.42
## [17] tibble_3.2.1      munsell_0.5.0     nnet_7.3-18       timeDate_4022.108
## [21] pillar_1.9.0      rlang_1.1.1       quantmod_0.4.22   utf8_1.2.3
## [25] urca_1.3-3        xfun_0.39         quadprog_1.5-8    cli_3.6.1
## [29] withr_2.5.0       magrittr_2.0.3    xts_0.13.1        digest_0.6.31
## [33] grid_4.3.0        rstudioapi_0.14   nlme_3.1-162      lifecycle_1.0.3
## [37] fracdiff_1.5-2    vctrs_0.6.2       evaluate_0.21     glue_1.6.2
## [41] farver_2.1.1      tseries_0.10-54   zoo_1.8-12        fansi_1.0.4
## [45] colorspace_2.1-0  TTR_0.24.3        rmarkdown_2.21    tools_4.3.0
## [49] pkgconfig_2.0.3   htmltools_0.5.5
```