Rapport computationnel sur le sujet 1

1. Nom de la fiche

Sujet 1 : Concentration de CO_2 dans l'atmosphère depuis 1958

2. Contexte

Plusieurs autres fiches étaient intéressantes et m'auraient permis de découvrir de nouveaux packages, fonctions ou méthodologies mais ne pouvant consacrer autant de temps que je l'aurais voulu à cet exercice, j'ai choisi le sujet 1 qui semblait se rapprocher le plus de ce que je connaissais.

3. Objectifs

Les trois objectifs de cette fiche sont les suivants :

1. Réalisez un graphique qui vous montrera une oscillation périodique superposée à une évolution systématique plus lente.
2. Séparez ces deux phénomènes. Caractérisez l'oscillation périodique. Proposez un modèle simple de la contribution lente, estimez ses paramètres et tentez une extrapolation jusqu'à 2025 (dans le but de pouvoir valider le modèle par des observations futures).
3. Déposer dans FUN votre résultat

4. Source du fichier

- Les données utilisées pour cette fiche sont issues du site

https://scrippsco2.ucsd.edu/data/atmospheric_co2/mlo.html

- Le nom du fichier, téléchargé le 27/11/2022, est :

weekly_in_situ_co2_mlo.csv

5. Analyse

Dans un premier temps, on charge les librairies dont nous aurons besoin :

- lubridate pour la gestion des dates
- MASS pour la fonction de régression rlm
- stplus pour la décomposition des effets saisonniers

On ouvre ensuite le fichier avec readLines pour comprendre sa structure. Pour supprimer les premières lignes, on cherche la position de la date de départ avec grep. La valeur de la ligne est stockée dans la variable start_row.

On peut ensuite ouvrir à nouveau le fichier en se limitant aux données qui nous intéressent, renommer les colonnes et convertir la colonne Date en classe Date.

```
library(lubridate)
library(MASS)
library(stlplus)
data_raw <- readLines(
       "C:/Disque D/Org/weekly_in_situ_co2_mlo_20221127.csv")
start_row <- grep("1958-03-29", data_raw)
data_co2 <- read.table(
      "C:/Disque D/Org/weekly_in_situ_co2_mlo_20221127.csv",
  sep = ",", dec = ".", as.is = T, header = F, skip = (start_row - 1))
names(data_co2) <- c("Date", "MicromolCO2")
data_co2$Date <- as.Date(data_co2$Date)
```

Les données se présentent sous deux colonnes : la date et la mesure de CO_2.

```
head(data_co2)
tail(data_co2)
```

```
        Date MicromolCO2
1 1958-03-29      316.19
2 1958-04-05      317.31
```

```
3 1958-04-12      317.69
4 1958-04-19      317.58
5 1958-04-26      316.48
6 1958-05-03      316.95
           Date MicromolCO2
3294 2022-09-24      414.82
3295 2022-10-01      415.12
3296 2022-10-08      414.85
3297 2022-10-15      415.31
3298 2022-10-22      415.60
3299 2022-10-29      416.08
```

Comme vous téléchargerez votre fichier plus tard que moi, pour avoir les mêmes résultats, pensez à filtrer les dates pour avoir un jeu de données qui se termine le 29/10/2022.

Les mesures sont normalement hebdomadaires. Pour vérifier que c'est bien le cas, on peut calculer l'écart entre chaque date

```
table(difftime(tail(as.Date(data_co2$Date), -1), head(as.Date(data_co2$Date),
```

```
   7   14   21   28   35   42   63  133
3269   17    5    2    2    1    1    1
```

On note que plusieurs mesures sont manquantes. Pour conserver un écart de 7 jours, nous allons ajouter des dates avec une mesure NA.

```
range_date <-range(data_co2$Date)
seq_date <- seq(range_date[1], range_date[2], by = "week")
df_na <- data.frame(Date = seq_date,
                    MicromocCO2 = rep(NA, length(seq_date)))
data_co2 <- merge(df_na, data_co2, all.x = TRUE, by = 'Date')[, c(1, 3)]
```

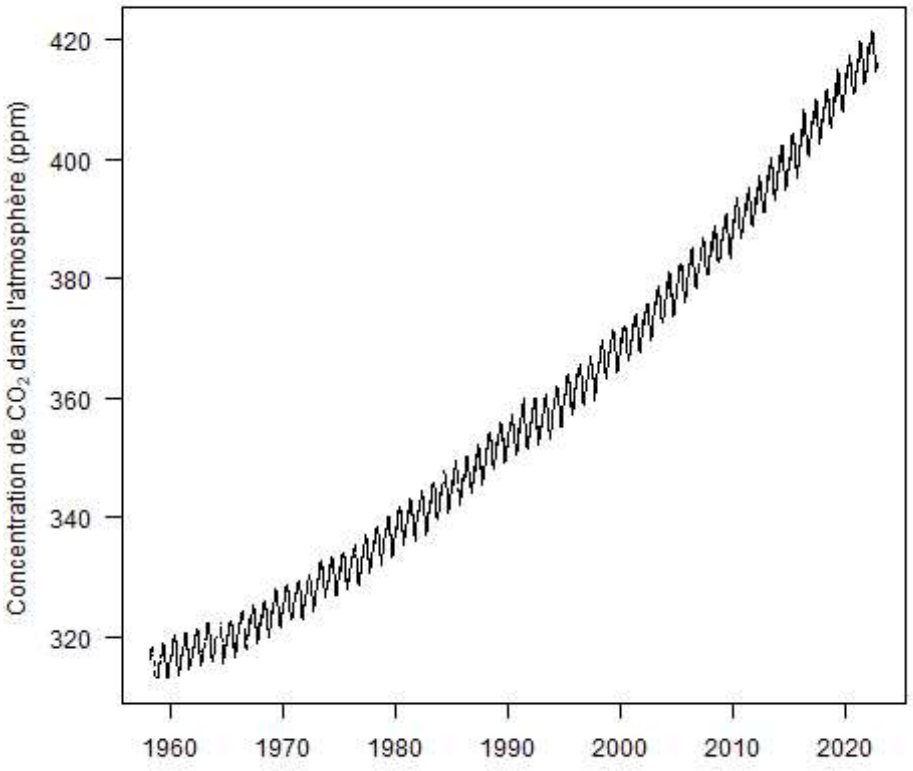
On vérifie que les dates manquantes sont bien remplacées par une date avec une mesure NA.

```
head(data_co2, 15)
```

```
         Date MicromolCO2
1  1958-03-29      316.19
2  1958-04-05      317.31
3  1958-04-12      317.69
4  1958-04-19      317.58
5  1958-04-26      316.48
6  1958-05-03      316.95
7  1958-05-10          NA
8  1958-05-17      317.56
9  1958-05-24      317.99
10 1958-05-31          NA
11 1958-06-07          NA
12 1958-06-14          NA
13 1958-06-21          NA
14 1958-06-28          NA
15 1958-07-05      315.85
```

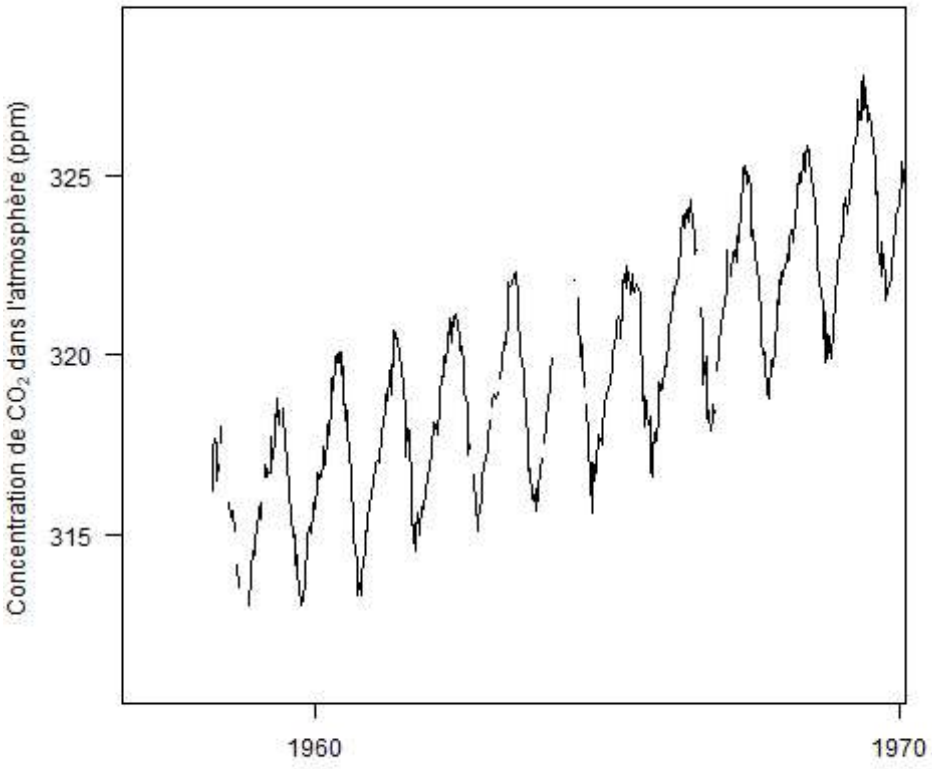
Le graphe résultant, qui répond à l'objectif 1, montre bien la superposition d'une oscillation périodique à une évolution systématique plus lente :

```
plot(data_co2$Date, data_co2$MicromolCO2,
     type = "l", las = 1,
     xlab = "", ylab = expression(paste("Concentration de ",CO[2]," dans l'atr
```



En zoomant, on peut observer les données manquantes :

```
plot(data_co2$Date, data_co2$MicromolCO2,
     type = "l", las = 1,
     xlab = "",
     ylab = expression(paste("Concentration de ",CO[2]," dans l'atmosphère (pp
     ylim = c(311, 329), xlim = c(-4683.2972, -146.7261))
```



Afin de décomposer la partie oscillatoire de la tendance globale, on utilise la fonction stlplus du package éponyme. La fonction a besoin d'un objet de classe ts ou un vecteur de données numériques (concentration en CO_2) et un vecteur avec les dates à considérer (dates hebdomadaires).

J'ai aussi essayé d'utiliser la fonction ts mais les arrondis faisaient que les dates générées ne correspondaient pas exactement aux dates de la colonne data_co2$Date.

Pour générer le vecteur temps, j'ai donc remplacé le code ci-dessous :

```
c_date <- ts(data_co2$MicromolCO2,
 freq=365.25/7,
 start=decimal_date(ymd("1958-03-29"))).
```

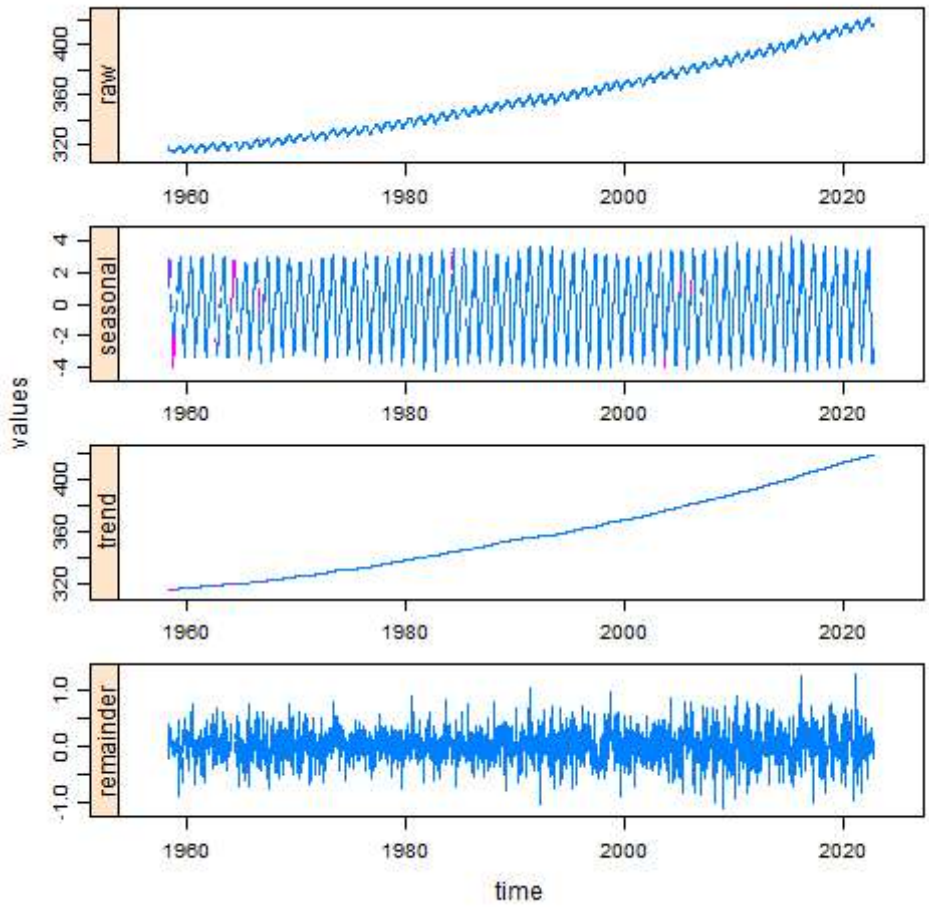
par celui-ci qui utilise des fonctions du package lubridate :

```
c_date <- decimal_date(ymd(data_co2$Date))
```

On utilise ensuite ce vecteur avec la fonction stlplus. J'ai dû jouer avec les paramètres, en particulier s.window, afin de réduire les résidus sans faire apparaître de schémas répétitifs et obtenir la décomposition ci-dessous. Le paramètre n.p correspond à la périodicité des données (52 valeurs par an).

Le premier graphe correspond aux données brutes. Le deuxième et le troisième graphes sont la réponse à la première partie de l'objectif 2, à savoir la décomposition de la partie oscillatoire périodique et de la tendance à long terme. Le dernier graphe correspond aux résidus entre les données brutes et l'ajustement de l'effet saisonnier et la tendance.

```
data_decompose <- stlplus(data_co2$MicromolCO2, c_date,
n.p = 52, s.window = 6, s.degree = 2)
plot(data_decompose, scales = list(relation = "free"))
```



De manière à trouver la période du signal oscillant, on cherche les changements de signe du signal. Chaque changement de signe indique un passage à 0 du signal. La différence de temps entre deux passages à 0 donne la demi-période du signal. Il suffit de doubler la médiane des demi-périodes pour avoir une valeur médiane de la période du signal oscillant.

```
df_season <- data.frame(Date = c_date,
                        Signal = data_decompose$data$seasonal)
c_sign <- sign(head(df_season$Signal, -1) * tail(df_season$Signal, -1))
c_sign <- c(c_sign[1], c_sign)
row_minus <- which(c_sign < 0)
print(paste(
  "La période de l'oscillation est de ",
  round(median(diff(df_season$Date[row_minus]))*2, 2),
  " année.", sep = ""))
```

```
[1] "La période de l'oscillation est de 0.96 année."
```

Pour déterminer l'amplitude du signal, on découpe le signal par période et, pour chaque période, on recherche les valeurs min et max. On donne ensuite les valeurs médianes min et max de l'amplitude.

```
cut_date <- cut(df_season$Date,
                      breaks = df_season$Date[row_minus[seq(1, length(row_minu
interval_date <- split(df_season, cut_date)
df_min_max <- do.call("rbind",
                      lapply(interval_date,
                             function(X){ data.frame(Date = X$Date[1],
                                                     val_min = min(X$Signal),
                                                     val_max = max(X$Signal))}
med_min <- median(df_min_max$val_min)
med_max <- median(df_min_max$val_max)
print(paste("La valeur minimale médiane du signal oscillant est de ",
            round(med_min, 2), " ppm.", sep = ""))
print(paste("La valeur maximale médiane du signal oscillant est de ",
            round(med_max, 2), " ppm.", sep = ""))
```

```
[1] "La valeur minimale médiane du signal oscillant est de -3.68 ppm."
[1] "La valeur maximale médiane du signal oscillant est de 3.14 ppm."
```

Le signal oscillant a donc une période proche d'un an et est relativement symétrique.

Le dernier objectif consiste à modéliser la contribution lente et à proposer une extrapolation jusqu'en 2025. Après avoir crée un data.frame avec la contribution lente, on ajuste les données avec un polynôme d'ordre 2 (pour être tout à fait honnête, j'ai trouvé ce modéle sur internet à cette adresse : https://mgimond.github.io/ES218/Week10a.html).

On affiche ensuite les paramètres du modèle et la prévision pour 2025.

```
df_trend <- data.frame(Date = c_date,
                       Trend = data_decompose$data$trend)
fit_trend <- rlm(Trend ~ Date + I(Date^2), df_trend)
print(fit_trend)
```

```
Call:
rlm(formula = Trend ~ Date + I(Date^2), data = df_trend)
Converged in 6 iterations

Coefficients:
  (Intercept)          Date     I(Date^2)
 5.050680e+04 -5.200213e+01  1.346666e-02

Degrees of freedom: 3371 total; 3368 residual
Scale estimate: 0.724
```

D'après le modèle, la prévision de concentration de CO_2 en ppm pour 2025 sera de :

```
trend_2025 <- round(predict(fit_trend, data.frame(Date = 2025)) ,1)
print(trend_2025[[1]])
```

```
[1] 424.2
```

Ci-dessous, la représentation graphique de la courbe de tendance, du modèle et la prévision en 2025.

```
plot(df_trend, type = "l", las = 1, cex.axis = 1.2, cex.lab = 1.2,
     xlab = "",
     ylab = expression(paste(CO[2], " concentration trend  (ppm)")),
     xlim = c(1956, 2026),
     ylim = c(300, 450)
     )
points(df_trend$Date, predict(fit_trend), type = "l", col ="red")
points (2025, trend_2025, pch = 20, cex = 1.6, col = "red")
legend(x = "topleft", c("Trend", "Model", "Forecast in 2025"),
       col = c("black", "red", "red"),
       pch = c(45, 45, 20), pt.cex = 2, cex = 1.2, inset = 0.02, bg = "white",
```