

TP_5

2024-04-19

1) Création d'une série temporelle

a) Chargement des modules

```
library("weathercan")
```

```
## Warning: le package 'weathercan' a été compilé avec la version R 4.3.3
```

```
library("tidyverse")
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.3.3
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.4      v readr      2.1.5
```

```
## v forcats    1.0.0      v stringr    1.5.1
```

```
## v ggplot2    3.5.0      v tibble     3.2.1
```

```
## v lubridate  1.9.3      v tidyr      1.3.1
```

```
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("lubridate")
```

```
library("tsibble")
```

```
## Warning: le package 'tsibble' a été compilé avec la version R 4.3.3
```

```
##
```

```
## Attachement du package : 'tsibble'
```

```
##
```

```
## L'objet suivant est masqué depuis 'package:lubridate':
```

```
##
```

```
##     interval
```

```
##
```

```
## Les objets suivants sont masqués depuis 'package:base':
```

```
##
```

```
##     intersect, setdiff, union
```

```
library("fable")
```

```
## Warning: le package 'fable' a été compilé avec la version R 4.3.3
```

```
## Le chargement a nécessité le package : fabletools
```

```
## Warning: le package 'fabletools' a été compilé avec la version R 4.3.3
```

```
library("fpp3")
```

```
## Warning: le package 'fpp3' a été compilé avec la version R 4.3.3

## -- Attaching packages ----- fpp3 0.5 --
## v tsibbledata 0.4.1      v feasts      0.3.2

## Warning: le package 'tsibbledata' a été compilé avec la version R 4.3.3

## Warning: le package 'feasts' a été compilé avec la version R 4.3.3

## -- Conflicts ----- fpp3_conflicts --
## x lubridate::date()      masks base::date()
## x dplyr::filter()        masks stats::filter()
## x tsibble::intersect()   masks base::intersect()
## x tsibble::interval()   masks lubridate::interval()
## x dplyr::lag()           masks stats::lag()
## x tsibble::setdiff()     masks base::setdiff()
## x tsibble::union()       masks base::union()

library("ggplot2")
library("dplyr")
library("forecast")

## Warning: le package 'forecast' a été compilé avec la version R 4.3.3

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

library("readxl")
library("cowplot")

## Warning: le package 'cowplot' a été compilé avec la version R 4.3.3

##
## Attachement du package : 'cowplot'
##
## L'objet suivant est masqué depuis 'package:lubridate':
##
##      stamp
```

b) Transformation des données

D'abord, on doit transformer la colonne « time » qui est de type numérique, et dans laquelle les dates sont des nombres réels au format : date. On extrait l'année entière de la colonne « time » dans la variable « year » et la partie décimale de la colonne « time » représentant les mois dans « month_decimal ». Ensuite, on convertit cette partie décimale en mois en multipliant par 12 et en ajoutant 1, puis en arrondissant. Enfin, on transforme le tout en un objet date, puis on reformate cette date en une chaîne de caractères au format « année-mois ».

```
hawai <- read_excel("C:/Users/DELL/Desktop/Cours/Modélisation des agrosystemes/TP5_Aissi/hawai.xlsx")
hawai <- hawai %>%
  mutate(year = floor(time),
         month_decimal = time - year,
         month = round((month_decimal * 12) + 1),
         time = format(as.Date(paste(year, month, "01", sep = "-")), "%Y-%m"))
hawai <- hawai %>%
  select(time, CO2)
head(hawai)
```

```
## # A tibble: 6 x 2
##   time      CO2
##   <chr>    <dbl>
## 1 1958-03  316.
## 2 1958-04  317.
## 3 1958-05  317.
## 4 1958-06  317.
## 5 1958-07  316.
## 6 1958-08  315.
```

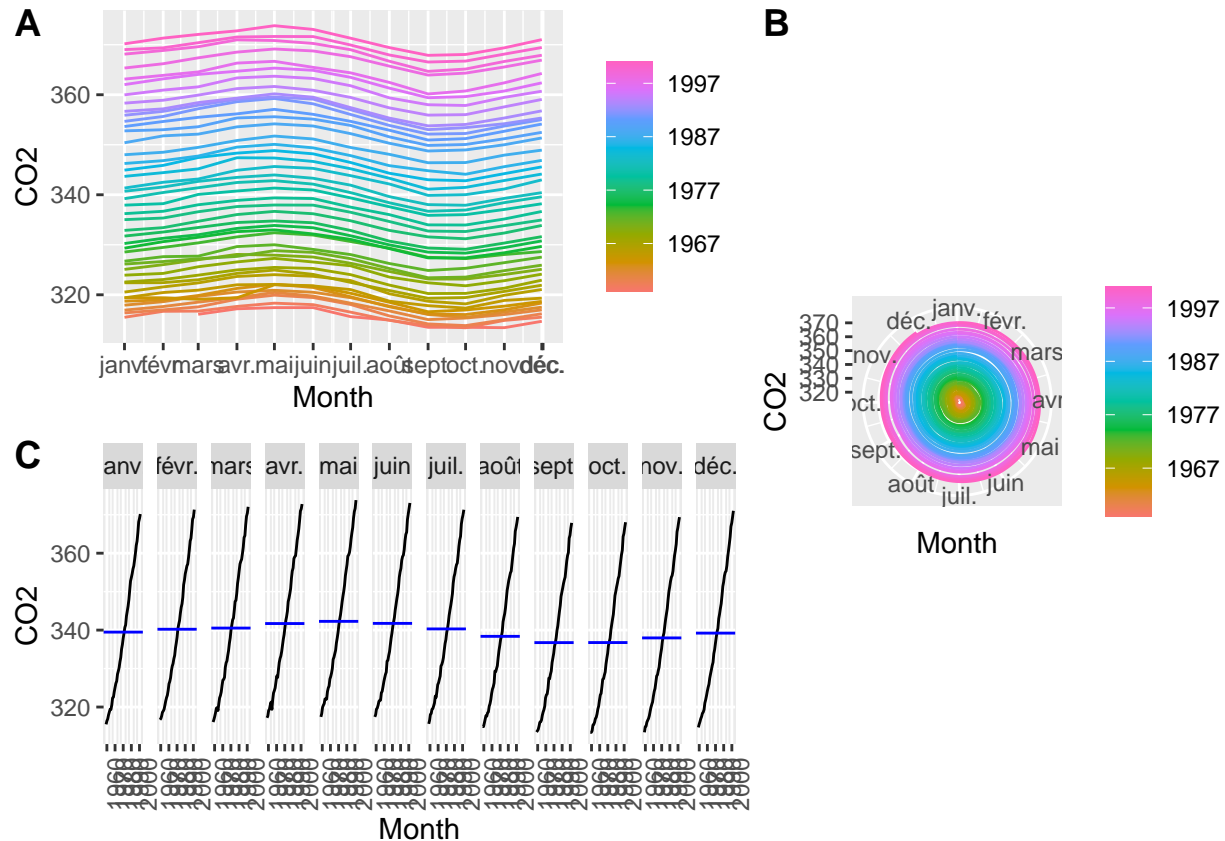
```
hawai_ts <- hawai |>
  mutate(Month = yearmonth(time)) |>
  as_tsibble(index = Month) |>
  select(Month, CO2)
head(hawai_ts)
```

```
## # A tsibble: 6 x 2 [1M]
##       Month      CO2
##       <mth>    <dbl>
## 1 1958 mars    316.
## 2 1958 avr.    317.
## 3 1958 mai     317.
## 4 1958 juin    317.
## 5 1958 juil.   316.
## 6 1958 août    315.
```

c) Visualisation

On visualise les tendances cycliques avec `feasts::gg_season()` et `feasts::gg_subseries()`.

```
ggA <- hawai_ts |>
  gg_season(`CO2`) + xlab("Month")
ggB <- hawai_ts |>
  gg_season(`CO2`, polar = TRUE) + xlab("Month")
ggC <- hawai_ts |>
  gg_subseries(`CO2`) + xlab("Month")
left_col <- plot_grid(ggA, ggC, labels = c('A', 'C'), ncol = 1)
plot_grid(left_col, ggB, labels = c('', 'B'),
  ncol = 2, rel_widths = c(1.5, 1))
```



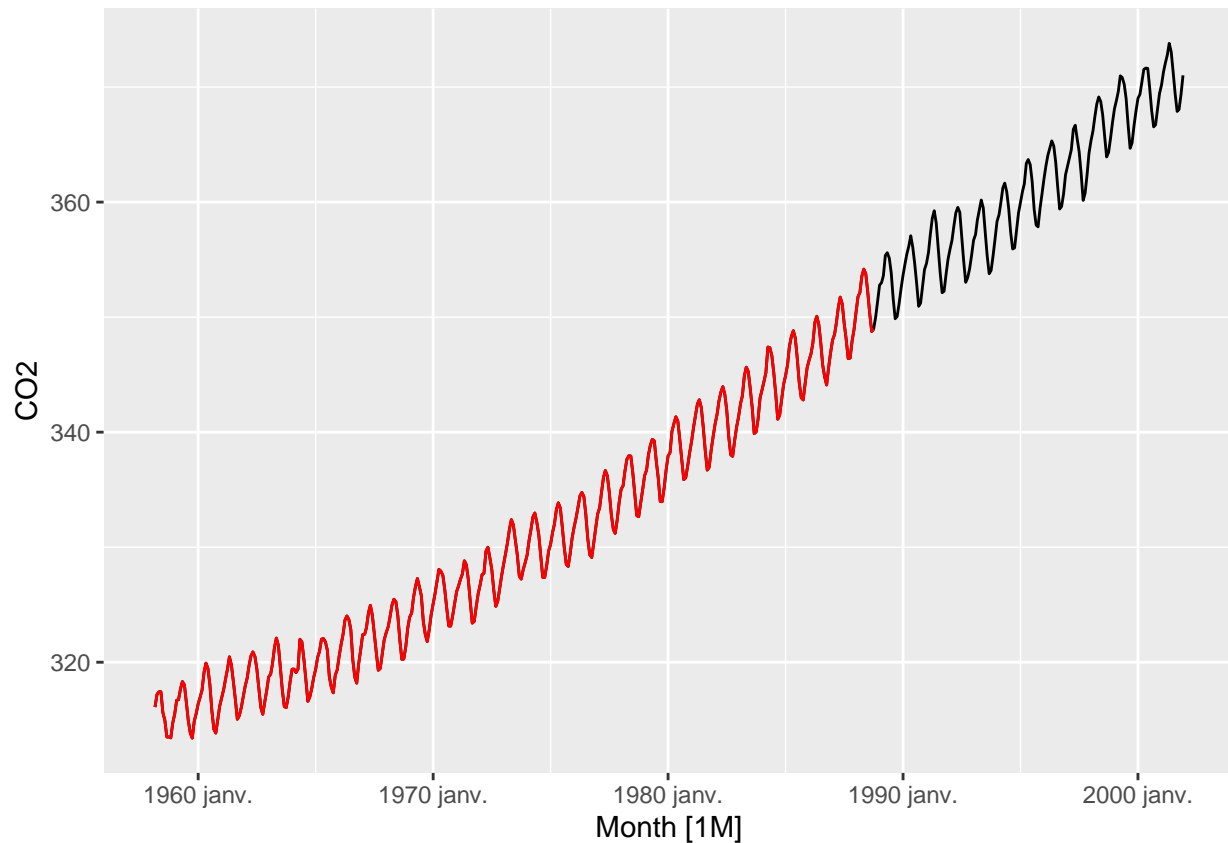
On remarque une légère oscillation des valeurs de CO2 entre les mois, et une tendance croissante au fil des années.

2) Séparation de la série en parties d'entraînement et en partie test

On calcule l'index de séparation entre les données d'entraînement et de test, puis on extrait les premières lignes de l'ensemble de données jusqu'à l'index de séparation calculé, créant ainsi l'ensemble de données d'entraînement, le reste étant les données de test.

```
prop_train <- 0.7
split_index <- floor(nrow(hawai_ts) * prop_train)
train_data <- hawai_ts[1:split_index, ]
test_data <- hawai_ts[(split_index + 1):nrow(hawai_ts), ]
autoplot(hawai_ts) +
  autolayer(train_data, colour = "red")
```

```
## Plot variable not specified, automatically selected `vars = C02`
## Plot variable not specified, automatically selected `vars = C02`
```



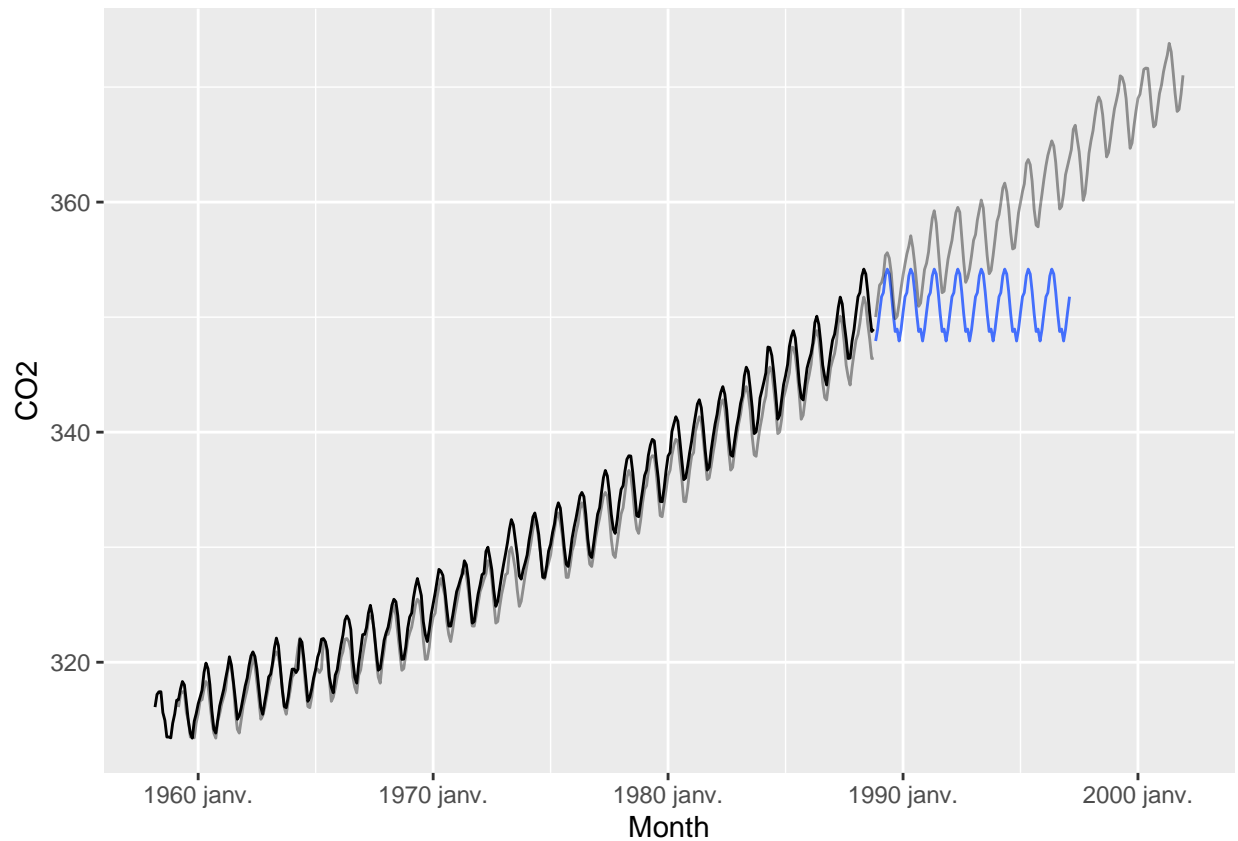
3) Création des modèles prévisionnels et analyse des résidus

a) Méthode SNAIVE

```
CO2_fit <- train_data |>
  model(SNaive = SNAIVE(`CO2`))
CO2_fc <- CO2_fit |> forecast(h = 100)
CO2_fc |>
  autoplot(train_data, level = NULL) +
  autolayer(fitted(CO2_fit), alpha = 0.4) +
  autolayer(test_data, alpha = 0.4)

## Plot variable not specified, automatically selected `.vars = .fitted`
## Plot variable not specified, automatically selected `.vars = CO2`

## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



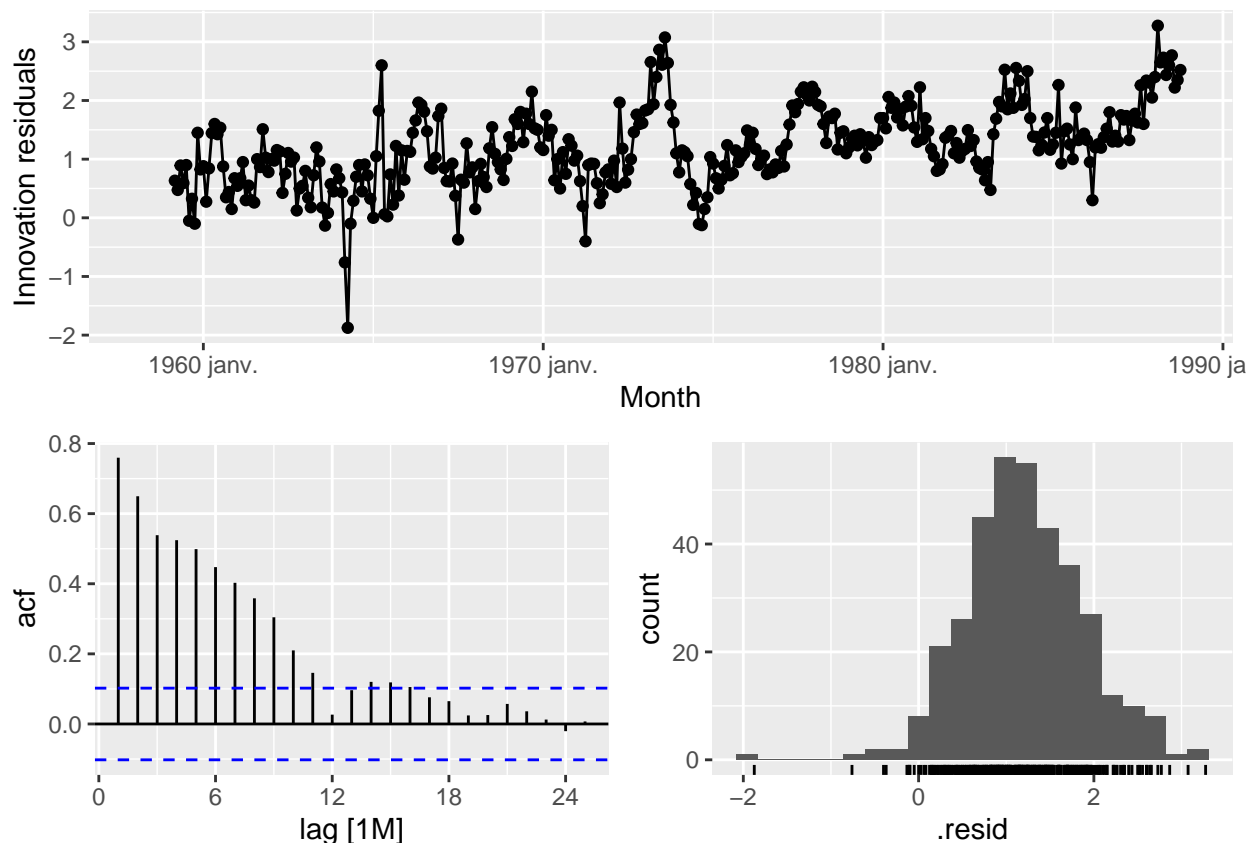
Puisque dans la méthode SNAIVE, la valeur prédite est celle provenant du cycle précédent, les prédictions n'ont pas pris en compte la tendance croissante des autres cycles à travers les années. Le même cycle est alors en train de se répéter, et la prévision aura peu d'intérêt.

```
gg_tsresiduals(CO2_fit)
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



On remarque une très forte autocorrélation dans le graphique « acf ». Voyons maintenant les tests statistiques de Shapiro et de Ljung-box :

```
C02_res <- C02_fit |>
  augment()
C02_res |>
  features(.innov, ljung_box, lag = 19)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>   <dbl>   <dbl>
## 1 SNaive   914.       0
```

```
shapiro.test(C02_res$.innov)
```

```
##
## Shapiro-Wilk normality test
##
## data: C02_res$.innov
## W = 0.99012, p-value = 0.01695
```

Les résultats des tests de Ljung-Box, et de Shapiro donnent une p-value de 0, qui est très petite, et il est peu probable que les résidus forment un bruit blanc. -> Rejet de l'hypothèse nulle selon laquelle les données ne sont pas corrélées.

```
accuracy(C02_fc, hawai_ts)
```

```
## # A tibble: 1 x 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
```

```
##    <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 SNaive Test    6.25  7.07  6.25  1.74  1.74  5.11  5.14  0.948
```

L'évaluation du modèle effectuée avec la fonction `fabletools::accuracy()` donne aussi des erreurs très élevées. Le modèle prévisionnel par SNAIVE n'est alors pas approprié.

b) Méthode SES avec tendance

On peut laisser R optimiser notre choix avec le modèle ETS par défaut (error, tend and seasonal). L'optimisation est lancée avec la fonction `fable::ETS()`, sans utiliser les paramètres.

```
C02_model <- train_data |>
  model(ETS(`C02`))
report(C02_model)
```

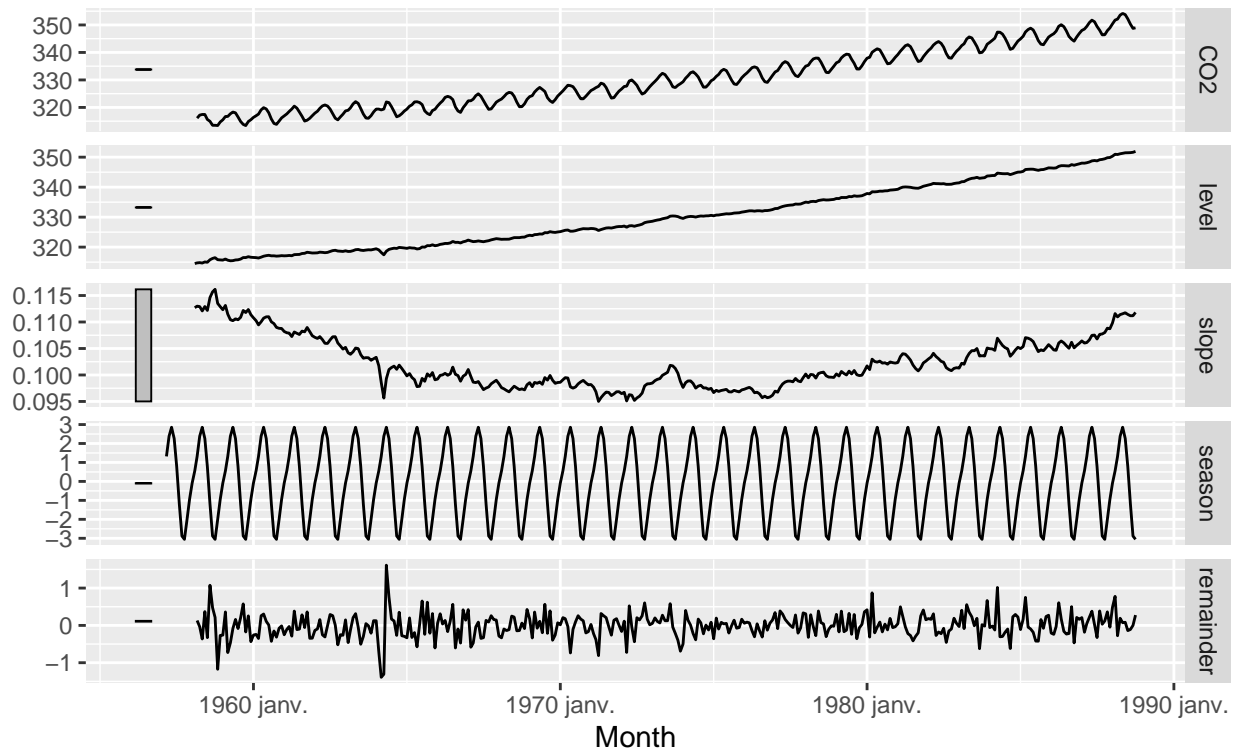
```
## Series: C02
## Model: ETS(A,A,A)
## Smoothing parameters:
##   alpha = 0.6802687
##   beta  = 0.00225957
##   gamma = 0.0001015036
##
## Initial states:
##   l[0]    b[0]    s[0]    s[-1]    s[-2]    s[-3]    s[-4]
## 314.5366 0.112694 0.5563618 -0.08727554 -0.9497972 -2.004136 -3.056502
##   s[-5]    s[-6]    s[-7]    s[-8]    s[-9]    s[-10]    s[-11]
## -2.878968 -1.183077 0.78395 2.259468 2.856629 2.384125 1.319221
##
## sigma^2: 0.114
##
##      AIC      AICc      BIC
## 1392.687 1394.436 1459.125
```

```
components(C02_model) |>
  autoplot()
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```


ETS(A,A,A) decomposition

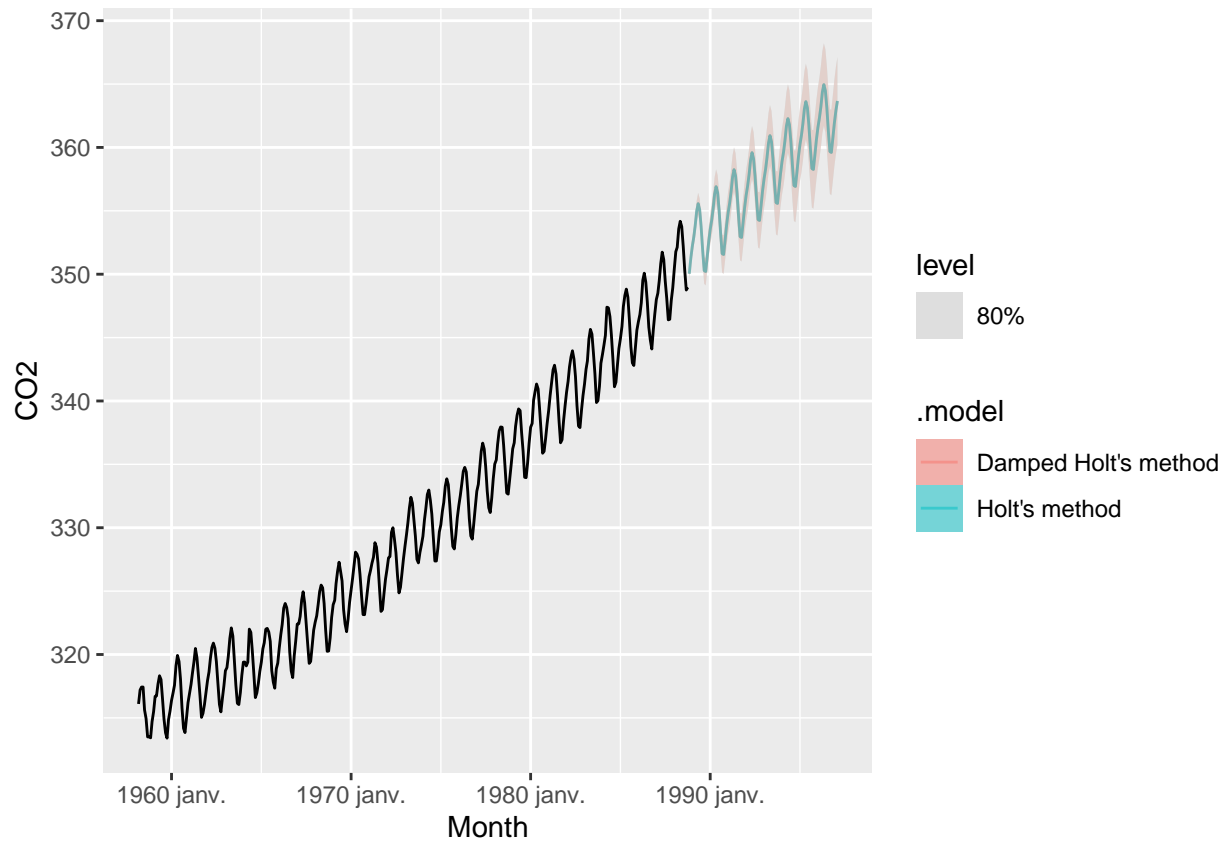
$CO_2 = \text{lag}(\text{level}, 1) + \text{lag}(\text{slope}, 1) + \text{lag}(\text{season}, 12) + \text{remainder}$



Le modèle ajusté est de type ETS (A, A, A), où : “A” représente une composante d’erreur additive pour chacune des tendances, saison, et erreur. Dans un modèle avec tendance et saisonnalité additive, les données « levels » sont additionnées aux les données « season » pour obtenir la prévision.

La fonction de `fable::ETS()` permet d’estimer automatiquement les paramètres de lissage ainsi que les paramètres d’état en sélectionnant le paramètre de tendance “A” pour la méthode Holt et Holt adoucité :

```
CO2_holt <- train_data |>
  model(
    `Holt's method` = ETS(CO2 ~ error("A") +
                          trend("A") + season("A")),
    `Damped Holt's method` = ETS(CO2 ~ error("A") +
                                  trend("A") + season("A"))
  )
CO2_holt |>
  forecast(h = 100) |>
  autoplot(train_data, level = 80, alpha = 0.5)
```

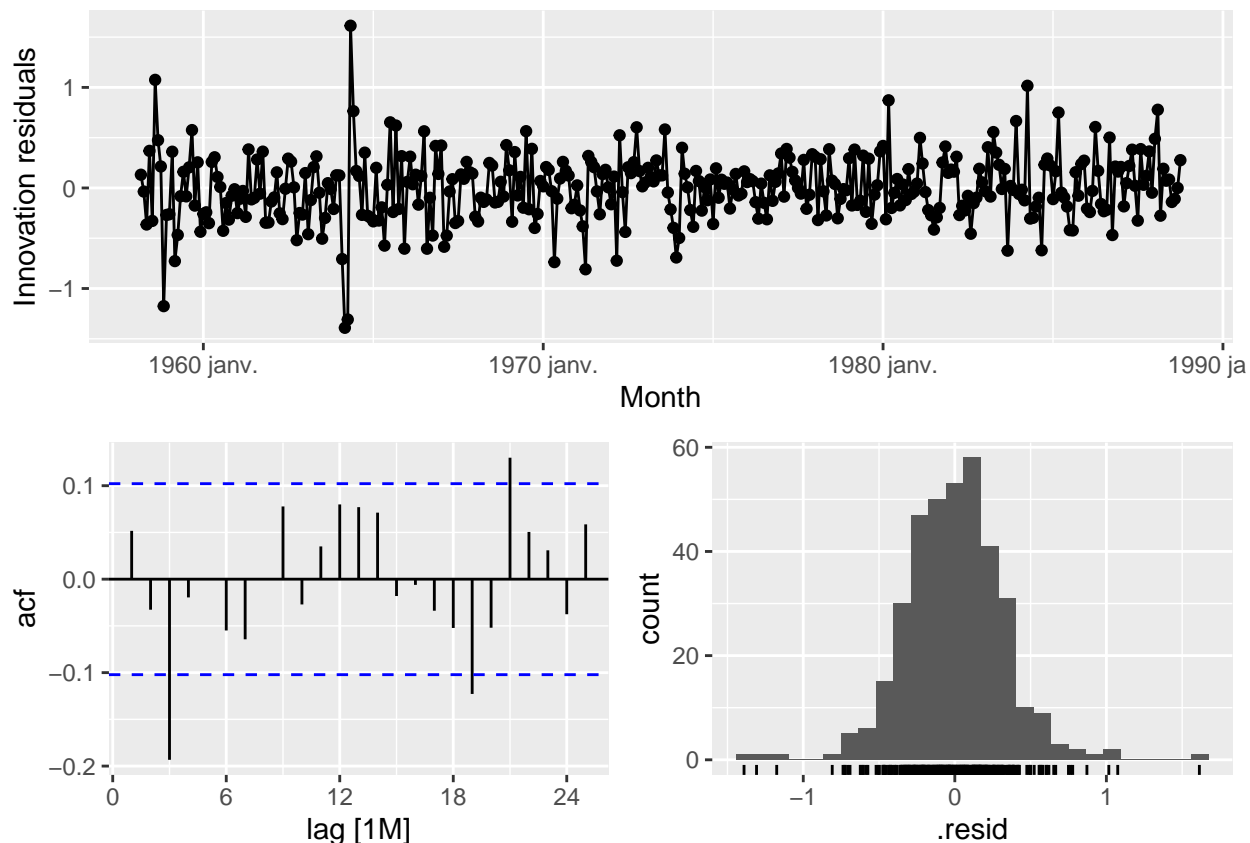


```
tidy(CO2_holt)
```

```
## # A tibble: 34 x 3
##   .model      term      estimate
##   <chr>      <chr>      <dbl>
## 1 Holt's method alpha    0.680
## 2 Holt's method beta     0.00226
## 3 Holt's method gamma    0.000102
## 4 Holt's method l[0]    315.
## 5 Holt's method b[0]     0.113
## 6 Holt's method s[0]     0.556
## 7 Holt's method s[-1]   -0.0873
## 8 Holt's method s[-2]   -0.950
## 9 Holt's method s[-3]   -2.00
## 10 Holt's method s[-4]  -3.06
## # i 24 more rows
```

-Le paramètre de lissage “beta” est de 0.99, ce qui signifie que la pente change rapidement, -Le paramètre de lissage “alpha” est aussi de 0.99, ce qui signifie qu’il donne beaucoup de poids aux observations récentes, -L’optimisation de “Phi” donne une valeur de 0.8, une valeur suffisamment faible pour que l’adoucissement soit fort.

```
gg_tsresiduals(CO2_model)
```



D'après le graphique, l'autocorrélation semble être assez significative et cyclique. Il est donc peu probable qu'il s'agit d'un bruit blanc.

```
C02_res <- C02_holt |>
  augment()
C02_res |>
  features(.innov, ljung_box, lag = 19)
```

```
## # A tibble: 2 x 3
##   .model          lb_stat lb_pvalue
##   <chr>          <dbl>   <dbl>
## 1 Damped Holt's method    35.4    0.0125
## 2 Holt's method          35.4    0.0125
```

```
shapiro.test(C02_res$.innov)
```

```
##
## Shapiro-Wilk normality test
##
## data:  C02_res$.innov
## W = 0.96899, p-value = 2.231e-11
```

La p-value étant de 2.231e-11, on peut conclure qu'il est peu probable que les résidus forment un bruit blanc, et que les données ne suivent pas une distribution normale. Les résidus contiennent de l'autocorrélation, ce qui devrait être évité. Ceci est toutefois dû aux points allant au-delà du seuil de 0.1 et de -0.1, que l'on peut observer sur le graphique d'autocorrélation. De plus, le graphique de la distribution des résidus montre des valeurs aberrantes, ainsi qu'une distribution plutôt pointue, qui donnerait un test de Kurtosis probablement élevé.

```
library(moments)
kurtosis_value <- kurtosis(CO2_res$.innov, na.rm = TRUE)
print(kurtosis_value)
```

```
## [1] 5.835065
```

En effet, la valeur kurtosis est de 5.83 (>3), ce qui indique que les données sont plus pointues que la distribution normale.

```
accuracy(CO2_model)
```

```
## # A tibble: 1 x 10
##   .model  .type      ME  RMSE  MAE      MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>   <chr>    <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ETS(CO2) Training -0.00106 0.330 0.248 -0.000786 0.0755 0.203 0.240 0.0517
```

Les erreurs sont plutôt faible (Surtout RMSE qui est proche de 0). Le modèle ETS(CO2) semble alors avoir des performances acceptables en termes de précision.