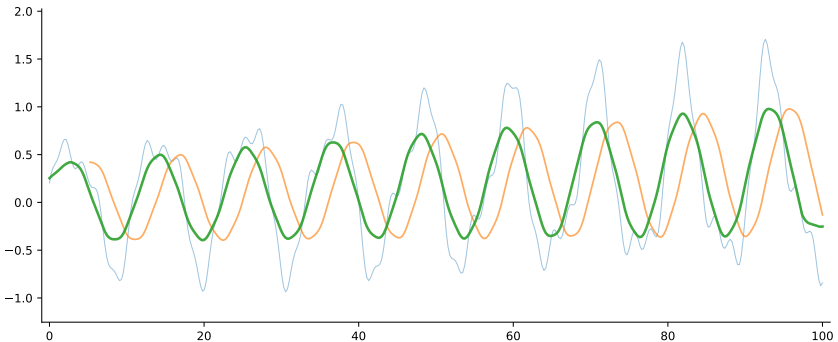
Séries temporelles

Cours 2 – Objets `R` et premiers filtres



M1 Actuariat – Université Gustave Eiffel

Solesne Bourguin – 2026–2027

Fil conducteur de la séance

La séance précédente a installé les questions statistiques. Aujourd'hui, on apprend à manipuler proprement les séries dans R, puis on donne un langage commun aux transformations simples.

Fil conducteur de la séance

La séance précédente a installé les questions statistiques. Aujourd'hui, on apprend à manipuler proprement les séries dans `R`, puis on donne un langage commun aux transformations simples.

1. Comprendre ce que contient un objet `ts` : valeurs et horloge.
2. Contrôler le calendrier : début, fin, fréquence, `time()` et `cycle()`.
3. Extraire, transformer, différencier, et repérer les valeurs manquantes.
4. Comprendre pourquoi `lag()` demande de la prudence au début.
5. Introduire une nouvelle partie : les filtres finis, qui généralisent moyennes mobiles et différences.

Ce qu'on garde du cours 1

On a appris à ne pas commencer par un modèle. On commence par lire la série.

Lecture statistique

- chronogramme ;
- tendance ;
- saisonnalité ;
- résidu ;
- dépendance aux retards.

Transformations

- logarithme ;
- retard B ;
- différence $1 - B$;
- différence saisonnière $1 - B^s$;
- stationnarité visée pour le reste.

Ce qu'on garde du cours 1

On a appris à ne pas commencer par un modèle. On commence par lire la série.

Lecture statistique

- chronogramme ;
- tendance ;
- saisonnalité ;
- résidu ;
- dépendance aux retards.

Transformations

- logarithme ;
- retard B ;
- différence $1 - B$;
- différence saisonnière $1 - B^s$;
- stationnarité visée pour le reste.

Aujourd'hui, la question devient : comment coder ces objets sans perdre l'information temporelle ?

Une série temporelle dans R : pas seulement un vecteur

Un vecteur de nombres donne les valeurs. Une série temporelle régulière doit aussi dire où ces valeurs se placent dans le temps.

Une série temporelle dans R : pas seulement un vecteur

Un vecteur de nombres donne les valeurs. Une série temporelle régulière doit aussi dire où ces valeurs se placent dans le temps.

Un objet `ts` contient :

1. des valeurs observées ;
2. une date de début ;
3. une date de fin ;
4. une fréquence d'observation.

Une série temporelle dans R : pas seulement un vecteur

Un vecteur de nombres donne les valeurs. Une série temporelle régulière doit aussi dire où ces valeurs se placent dans le temps.

Un objet `ts` contient :

1. des valeurs observées ;
2. une date de début ;
3. une date de fin ;
4. une fréquence d'observation.

Une série temporelle dans R, c'est donc un vecteur plus une horloge. Si l'horloge est fausse, une partie de l'analyse devient fausse.

La fréquence dans `ts()` : une convention de calendrier

Dans `ts()`, la fréquence signifie : nombre d'observations par unité de temps.

Type de données	**Fréquence R**	**Exemple de date**
annuelles	`frequency = 1`	`1980`
trimestrielles	`frequency = 4`	`c(1980, 3)` : 3e trimestre
mensuelles	`frequency = 12`	`c(1980, 7)` : juillet 1980

La fréquence dans `ts()` : une convention de calendrier

Dans `ts()`, la fréquence signifie : nombre d'observations par unité de temps.

Type de données	**Fréquence R**	**Exemple de date**
annuelles	`frequency = 1`	`1980`
trimestrielles	`frequency = 4`	`c(1980, 3)` : 3e trimestre
mensuelles	`frequency = 12`	`c(1980, 7)` : juillet 1980

Ici, « fréquence » ne signifie pas fréquence spectrale. Cela signifie seulement combien d'observations `R` doit placer dans une année.

Exemple : ce que R sait déjà de co2

La série co2 est fournie avec R. Elle est déjà stockée comme objet ts.

```
data(co2)
start(co2)
end(co2)
frequency(co2)
tsp(co2)
```

Exemple : ce que R sait déjà de co2

La série co2 est fournie avec R. Elle est déjà stockée comme objet ts.

```
data(co2)
start(co2)
end(co2)
frequency(co2)
tsp(co2)
```

`start()`, `end()` et `frequency()` contrôlent séparément le début, la fin et le nombre d'observations par année.

`tsp()` donne ces trois informations en une seule sortie : début, fin, fréquence.

Exemple : ce que R sait déjà de co2

La série co2 est fournie avec R. Elle est déjà stockée comme objet ts.

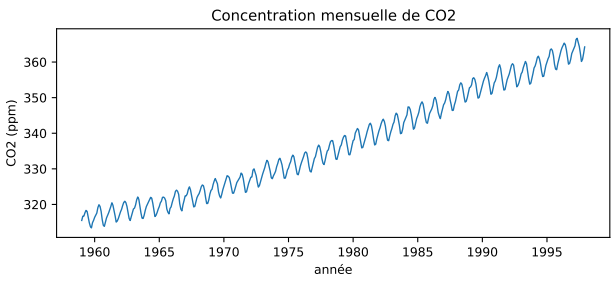
```
data(co2)
start(co2)
end(co2)
frequency(co2)
tsp(co2)
```

start(), end() et frequency() contrôlent séparément le début, la fin et le nombre d'observations par année.

tsp() donne ces trois informations en une seule sortie : début, fin, fréquence.

Avant de tracer ou de transformer une série, on vérifie toujours son horloge.

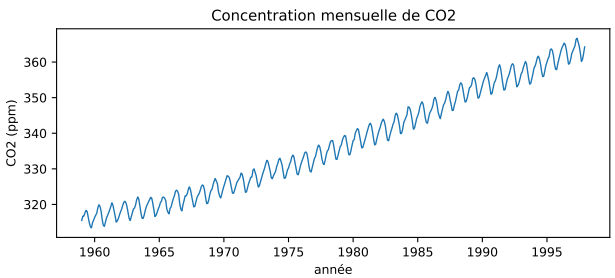
Chronogramme de co2



Deux phénomènes sautent aux yeux :

- une tendance croissante de long terme ;
- une oscillation annuelle régulière.

Chronogramme de co2



Deux phénomènes sautent aux yeux :

- une tendance croissante de long terme ;
- une oscillation annuelle régulière.

La série est mensuelle : si la fréquence n'était pas codée correctement, les graphiques et les transformations saisonnières seraient mal interprétés.

Construire soi-même un objet `ts`

Quand on part d'un simple vecteur `x`, il faut lui ajouter une horloge.

```
y <- ts(x, start = c(2020, 1), frequency = 12)
```

Construire soi-même un objet `ts`

Quand on part d'un simple vecteur `x`, il faut lui ajouter une horloge.

```
y <- ts(x, start = c(2020, 1), frequency = 12)
```

Cette commande dit à `R` :

- les valeurs sont dans `x` ;
- la première observation est janvier 2020 ;
- il y a 12 observations par année.

Construire soi-même un objet `ts`

Quand on part d'un simple vecteur `x`, il faut lui ajouter une horloge.

```
y <- ts(x, start = c(2020, 1), frequency = 12)
```

Cette commande dit à `R` :

- les valeurs sont dans `x` ;
- la première observation est janvier 2020 ;
- il y a 12 observations par année.

La difficulté n'est pas technique : la difficulté est de ne pas mentir à `R`. Une série mensuelle codée comme annuelle perd son cycle mensuel.

Lire une commande R sans paniquer

Une commande R a souvent beaucoup d'arguments possibles, mais on n'en utilise pas toujours beaucoup.

Dans

```
ts(x, start = c(2020, 1), frequency = 12),
```

`x` est le premier argument, puis `start` et `frequency` sont des arguments nommés.

Lire une commande R sans paniquer

Une commande R a souvent beaucoup d'arguments possibles, mais on n'en utilise pas toujours beaucoup.

Dans

```
ts(x, start = c(2020, 1), frequency = 12),
```

`x` est le premier argument, puis `start` et `frequency` sont des arguments nommés.

Un argument non écrit prend souvent une valeur par défaut.

La page d'aide liste toutes les possibilités ; le cours utilise seulement celles dont on a besoin.

Lire une commande R sans paniquer

Une commande R a souvent beaucoup d'arguments possibles, mais on n'en utilise pas toujours beaucoup.

Dans

```
ts(x, start = c(2020, 1), frequency = 12),
```

`x` est le premier argument, puis `start` et `frequency` sont des arguments nommés.

Un argument non écrit prend souvent une valeur par défaut.

La page d'aide liste toutes les possibilités ; le cours utilise seulement celles dont on a besoin.

Quand une commande paraît plus courte que sa page d'aide, ce n'est pas une contradiction : c'est justement parce que beaucoup d'arguments sont optionnels.

Deux fonctions pour lire l'horloge : `time()` et `cycle()`

Une fois la série codée, on peut regarder deux informations différentes.

```
head(time(co2), 12)
head(cycle(co2), 12)
```

Deux fonctions pour lire l'horloge : `time()` et `cycle()`

Une fois la série codée, on peut regarder deux informations différentes.

```
head(time(co2), 12)
head(cycle(co2), 12)
```

`time(x)` donne les dates numériques associées aux observations.

`cycle(x)` donne la position dans le cycle saisonnier.

Deux fonctions pour lire l'horloge : `time()` et `cycle()`

Une fois la série codée, on peut regarder deux informations différentes.

```
head(time(co2), 12)
head(cycle(co2), 12)
```

`time(x)` donne les dates numériques associées aux observations.

`cycle(x)` donne la position dans le cycle saisonnier.

Pour une série mensuelle, `cycle(x)` vaut 1 pour janvier, 2 pour février, etc. Ce n'est pas la même information que `time(x)`.

Même série, deux questions différentes

Pour co2, les premières dates numériques sont

$$1959, \quad 1959 + \frac{1}{12}, \quad 1959 + \frac{2}{12}, \quad \ldots$$

Les premiers cycles sont

$$1, \quad 2, \quad 3, \quad \ldots, \quad 12.$$

Même série, deux questions différentes

Pour `co2`, les premières dates numériques sont

$$1959, \quad 1959 + \frac{1}{12}, \quad 1959 + \frac{2}{12}, \quad \ldots$$

Les premiers cycles sont

$$1, \quad 2, \quad 3, \quad \ldots, \quad 12.$$

`time()` répond : à quelle date absolue se trouve l'observation ?

`cycle()` répond : où se trouve l'observation dans l'année ?

Même série, deux questions différentes

Pour `co2`, les premières dates numériques sont

$$1959, \quad 1959 + \frac{1}{12}, \quad 1959 + \frac{2}{12}, \quad \ldots$$

Les premiers cycles sont

$$1, \quad 2, \quad 3, \quad \ldots, \quad 12.$$

`time()` répond : à quelle date absolue se trouve l'observation ?

`cycle()` répond : où se trouve l'observation dans l'année ?

Cette distinction devient importante dès qu'on veut construire des effets mensuels ou trimestriels.

Extraire une sous-série avec `window()`

On ne regarde pas toujours toute une série. On peut vouloir zoomer sur une période, ou séparer apprentissage et test.

```
co2.9092 <- window(co2, start = c(1990, 1),
                        end   = c(1992, 12))
plot.ts(co2.9092)
```

Extraire une sous-série avec `window()`

On ne regarde pas toujours toute une série. On peut vouloir zoomer sur une période, ou séparer apprentissage et test.

```
co2.9092 <- window(co2, start = c(1990, 1),
                        end   = c(1992, 12))
plot.ts(co2.9092)
```

`window()` extrait une période en conservant la structure temporelle de l'objet `ts`.

Extraire une sous-série avec `window()`

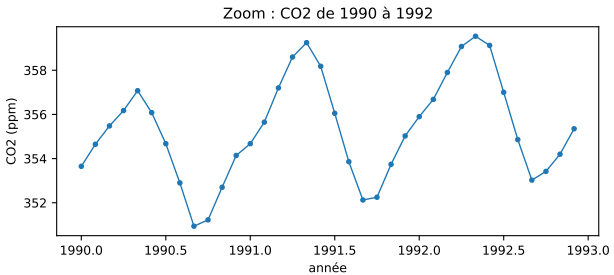
On ne regarde pas toujours toute une série. On peut vouloir zoomer sur une période, ou séparer apprentissage et test.

```
co2.9092 <- window(co2, start = c(1990, 1),
                        end   = c(1992, 12))
plot.ts(co2.9092)
```

`window()` extrait une période en conservant la structure temporelle de l'objet `ts`.

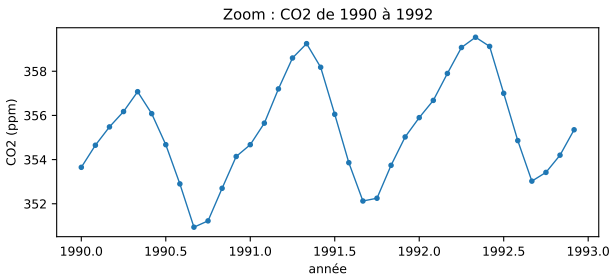
C'est préférable à une extraction brute de vecteur quand on veut garder les dates et la fréquence.

Zoom sur co2 : ce que l'échelle change



Sur trois années seulement, le motif annuel devient beaucoup plus lisible.

Zoom sur co2 : ce que l'échelle change

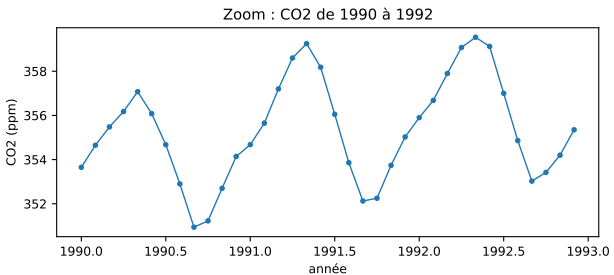


Sur trois années seulement, le motif annuel devient beaucoup plus lisible.

Le graphique global met surtout en avant la tendance croissante.

Le zoom met surtout en avant la saisonnalité annuelle.

Zoom sur co2 : ce que l'échelle change



Sur trois années seulement, le motif annuel devient beaucoup plus lisible.

Le graphique global met surtout en avant la tendance croissante.

Le zoom met surtout en avant la saisonnalité annuelle.

Le choix de la fenêtre temporelle fait déjà partie de l'analyse.

Retard mathématique et `lag()` dans `R`

Mathématiquement, l'opérateur retard B signifie une chose très précise :

$$(BX)_t = X_{t-1}.$$

Autrement dit, à la date t, on remplace la valeur courante par la valeur observée une date avant.

Retard mathématique et `lag()` dans `R`

Mathématiquement, l'opérateur retard B signifie une chose très précise :

$$(BX)_t = X_{t-1}.$$

Autrement dit, à la date t, on remplace la valeur courante par la valeur observée une date avant.

La fonction `lag()` de `R` sert aussi à décaler une série, mais elle décale une série `ts` avec son index temporel. Elle ne produit pas simplement un vecteur "précédent" à coller à côté du vecteur initial.

Retard mathématique et `lag()` dans `R`

Mathématiquement, l'opérateur retard B signifie une chose très précise :

$$(BX)_t = X_{t-1}.$$

Autrement dit, à la date t, on remplace la valeur courante par la valeur observée une date avant.

La fonction `lag()` de `R` sert aussi à décaler une série, mais elle décale une série `ts` avec son index temporel. Elle ne produit pas simplement un vecteur "précédent" à coller à côté du vecteur initial.

C'est pour cela qu'au début on évite d'identifier mécaniquement `lag(x)` avec BX. Pour les graphiques de retard, on construit explicitement les couples à représenter.

Lag plot : quelle question pose-t-on ?

Un lag plot de retard 1 sert à regarder si une grande valeur est souvent suivie d'une grande valeur, et si une petite valeur est souvent suivie d'une petite valeur.

Lag plot : quelle question pose-t-on ?

Un lag plot de retard 1 sert à regarder si une grande valeur est souvent suivie d'une grande valeur, et si une petite valeur est souvent suivie d'une petite valeur.

On représente les points

$$(x_{t-1}, x_t).$$

Chaque point compare donc une observation avec l'observation précédente.

Lag plot : quelle question pose-t-on ?

Un lag plot de retard 1 sert à regarder si une grande valeur est souvent suivie d'une grande valeur, et si une petite valeur est souvent suivie d'une petite valeur.

On représente les points

$$(x_{t-1}, x_t).$$

Chaque point compare donc une observation avec l'observation précédente.

Date	valeur précédente	valeur courante	point du graphique
$t = 2$	x_1	x_2	(x_1, x_2)
$t = 3$	x_2	x_3	(x_2, x_3)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t = n$	x_{n-1}	x_n	(x_{n-1}, x_n)

Lag plot : quelle question pose-t-on ?

Un lag plot de retard 1 sert à regarder si une grande valeur est souvent suivie d'une grande valeur, et si une petite valeur est souvent suivie d'une petite valeur.

On représente les points

$$(x_{t-1}, x_t).$$

Chaque point compare donc une observation avec l'observation précédente.

Date	valeur précédente	valeur courante	point du graphique
$t = 2$	x_1	x_2	(x_1, x_2)
$t = 3$	x_2	x_3	(x_2, x_3)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t = n$	x_{n-1}	x_n	(x_{n-1}, x_n)

Si le nuage est très allongé autour d'une diagonale croissante, cela suggère une forte dépendance entre deux dates consécutives.

Lag plot : construction explicite et sûre

Pour représenter les points (x_{t-1}, x_t), on peut convertir temporairement la série en vecteur numérique.

```
x <- as.numeric(co2)
n <- length(x)
plot(x[-n], x[-1],
     xlab = "valeur précédente",
     ylab = "valeur courante")
```

Lag plot : construction explicite et sûre

Pour représenter les points (x_{t-1}, x_t), on peut convertir temporairement la série en vecteur numérique.

```
x <- as.numeric(co2)
n <- length(x)
plot(x[-n], x[-1],
     xlab = "valeur précédente",
     ylab = "valeur courante")
```

`x[-n]` enlève la dernière valeur : on garde $x_1, \ldots, x_{n-1}$.

`x[-1]` enlève la première valeur : on garde $x_2, \ldots, x_n$.

Lag plot : construction explicite et sûre

Pour représenter les points (x_{t-1}, x_t), on peut convertir temporairement la série en vecteur numérique.

```
x <- as.numeric(co2)
n <- length(x)
plot(x[-n], x[-1],
     xlab = "valeur précédente",
     ylab = "valeur courante")
```

`x[-n]` enlève la dernière valeur : on garde $x_1, \ldots, x_{n-1}$.

`x[-1]` enlève la première valeur : on garde $x_2, \ldots, x_n$.

Les deux vecteurs ont la même longueur et sont alignés comme voulu : le point numéro i est (x_i, x_{i+1}).

Différences ordinaires et saisonnières avec `diff()`

La fonction `diff()` calcule des différences entre une valeur et une valeur passée.

```
d1   <- diff(co2)
d12  <- diff(co2, lag = 12)
```

Différences ordinaires et saisonnières avec `diff()`

La fonction `diff()` calcule des différences entre une valeur et une valeur passée.

```
d1   <- diff(co2)
d12  <- diff(co2, lag = 12)
```

`diff(co2)` calcule les différences successives :

$$x_t - x_{t-1}.$$

`diff(co2, lag = 12)` calcule les différences saisonnières :

$$x_t - x_{t-12}.$$

Différences ordinaires et saisonnières avec `diff()`

La fonction `diff()` calcule des différences entre une valeur et une valeur passée.

```
d1   <- diff(co2)
d12  <- diff(co2, lag = 12)
```

`diff(co2)` calcule les différences successives :

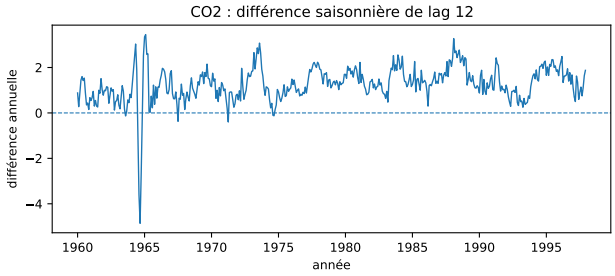
$$x_t - x_{t-1}.$$

`diff(co2, lag = 12)` calcule les différences saisonnières :

$$x_t - x_{t-12}.$$

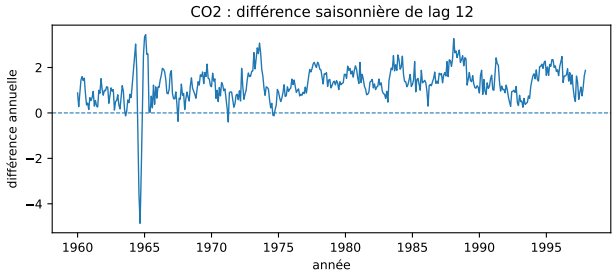
Une différence change la grandeur étudiée : on ne regarde plus un niveau, mais une variation.

Différence saisonnière de co2



Ici, chaque mois est comparé au même mois de l'année précédente.

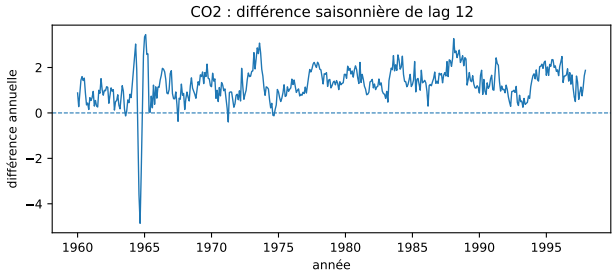
Différence saisonnière de co2



Ici, chaque mois est comparé au même mois de l'année précédente.

Le cycle annuel est fortement réduit, car on soustrait des observations placées au même endroit du cycle.

Différence saisonnière de co2



Ici, chaque mois est comparé au même mois de l'année précédente.

Le cycle annuel est fortement réduit, car on soustrait des observations placées au même endroit du cycle.

Il peut rester une structure de long terme : différencier saisonnièrement n'est pas une garantie de stationnarité.

Valeurs manquantes : repérer avant de modéliser

Dans R, une valeur manquante est notée NA. Ce n'est pas zéro, et ce n'est pas une petite valeur.

```
any(is.na(co2))
sum(is.na(co2))
```

Valeurs manquantes : repérer avant de modéliser

Dans R, une valeur manquante est notée NA. Ce n'est pas zéro, et ce n'est pas une petite valeur.

```
any(is.na(co2))
sum(is.na(co2))
```

`any(is.na(x))` dit s'il existe au moins une valeur manquante.

`sum(is.na(x))` compte le nombre de valeurs manquantes.

Valeurs manquantes : repérer avant de modéliser

Dans R, une valeur manquante est notée NA. Ce n'est pas zéro, et ce n'est pas une petite valeur.

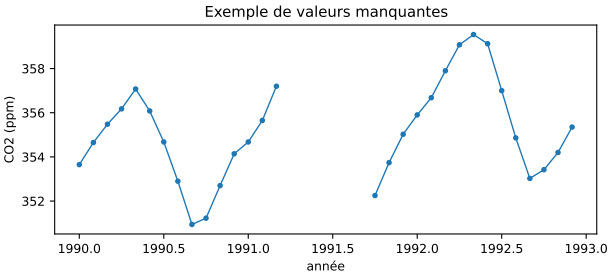
```
any(is.na(co2))
sum(is.na(co2))
```

`any(is.na(x))` dit s'il existe au moins une valeur manquante.

`sum(is.na(x))` compte le nombre de valeurs manquantes.

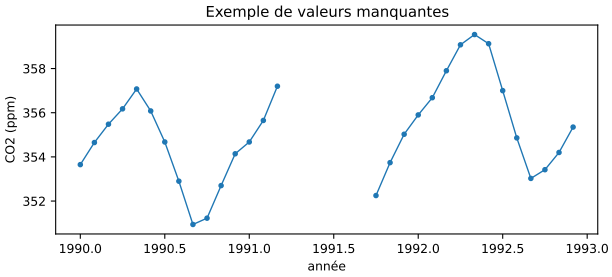
Avant un ajustement, on veut savoir si le modèle travaille sur des observations réelles ou sur une série trouée.

Une lacune n'est pas une valeur nulle



On a introduit artificiellement quelques `NA` dans la série.

Une lacune n'est pas une valeur nulle

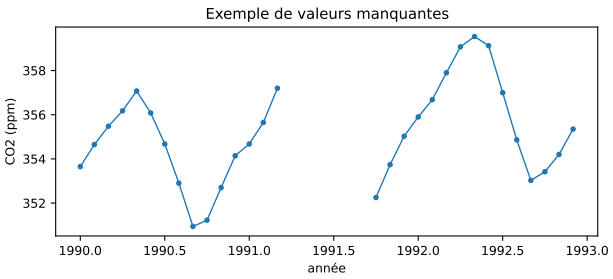


On a introduit artificiellement quelques `NA` dans la série.

Le trou dans la courbe représente une absence d'information.

Il ne faut pas confondre absence d'observation et observation égale à zéro.

Une lacune n'est pas une valeur nulle



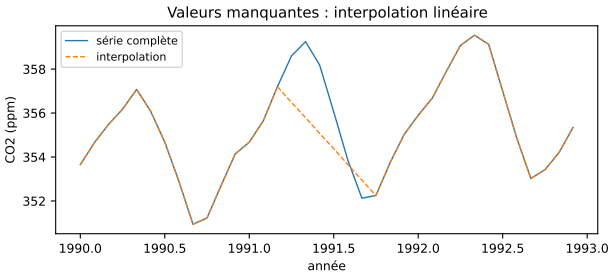
On a introduit artificiellement quelques `NA` dans la série.

Le trou dans la courbe représente une absence d'information.

Il ne faut pas confondre absence d'observation et observation égale à zéro.

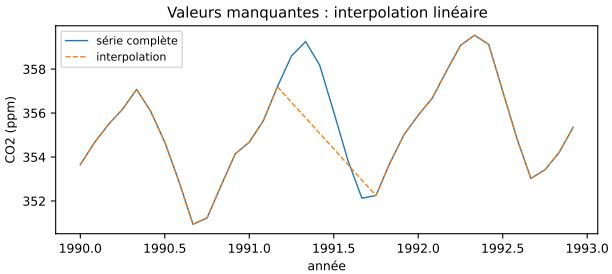
Un graphique peut être interrompu non parce que le phénomène s'arrête, mais parce que la mesure manque.

Interpoler : utile, mais pas neutre



Une interpolation linéaire remplace les valeurs manquantes par des valeurs reconstruites.

Interpoler : utile, mais pas neutre

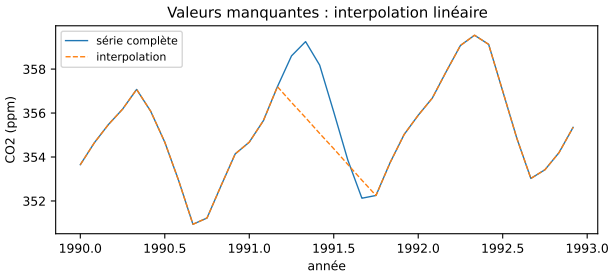


Une interpolation linéaire remplace les valeurs manquantes par des valeurs reconstruites.

Cela peut être raisonnable pour un graphique ou une courte lacune isolée.

Cela devient risqué pour une longue période manquante ou une série instable.

Interpoler : utile, mais pas neutre



Une interpolation linéaire remplace les valeurs manquantes par des valeurs reconstruites.

Cela peut être raisonnable pour un graphique ou une courte lacune isolée.

Cela devient risqué pour une longue période manquante ou une série instable.

Une valeur interpolée est plausible, mais elle n'est pas observée. On doit garder cette distinction en tête.

Filtres linéaires finis

On passe maintenant des commandes `R` à une question mathématique : que font réellement les transformations locales d'une série ?

Pourquoi introduire les filtres ?

Au cours 1, on a surtout appris à **lire** une série : chronogramme, tendance, saisonnalité, résidus, transformations simples comme le logarithme.

Pourquoi introduire les filtres ?

Au cours 1, on a surtout appris à **lire** une série : chronogramme, tendance, saisonnalité, résidus, transformations simples comme le logarithme.

Les outils du cours 1

- décrivent la série ;
- changent parfois son échelle ;
- aident à voir tendance et saisonnalité.

Les filtres

- fabriquent une nouvelle série ;
- combinent plusieurs dates voisines ;
- permettent de contrôler ce qu'on enlève ou conserve.

Pourquoi introduire les filtres ?

Au cours 1, on a surtout appris à **lire** une série : chronogramme, tendance, saisonnalité, résidus, transformations simples comme le logarithme.

Les outils du cours 1

- décrivent la série ;
- changent parfois son échelle ;
- aident à voir tendance et saisonnalité.

Les filtres

- fabriquent une nouvelle série ;
- combinent plusieurs dates voisines ;
- permettent de contrôler ce qu'on enlève ou conserve.

L'idée n'est pas d'ajouter une technique de plus, mais de donner un langage commun aux opérations comme lisser, différencier, enlever une saisonnalité ou comparer au passé.

De l'intuition à la formule

Beaucoup d'opérations utiles en séries temporelles ont la même forme : à la date t, on remplace X_t par une combinaison de valeurs prises autour de t.

De l'intuition à la formule

Beaucoup d'opérations utiles en séries temporelles ont la même forme : à la date t, on remplace X_t par une combinaison de valeurs prises autour de t.

Intention	**Valeurs utilisées**	**Exemple**
comparer au passé	présent et passé	$X_t - X_{t-1}$
lisser	plusieurs valeurs voisines	$(X_t + X_{t-1} + X_{t-2})/3$
comparer à la même saison	présent et même saison passée	$X_t - X_{t-12}$

De l'intuition à la formule

Beaucoup d'opérations utiles en séries temporelles ont la même forme : à la date t, on remplace X_t par une combinaison de valeurs prises autour de t.

Intention	**Valeurs utilisées**	**Exemple**
comparer au passé	présent et passé	$X_t - X_{t-1}$
lisser	plusieurs valeurs voisines	$(X_t + X_{t-1} + X_{t-2})/3$
comparer à la même saison	présent et même saison passée	$X_t - X_{t-12}$

Un filtre sert à écrire ces opérations dans un même cadre, puis à prévoir leur effet avant même de les appliquer aux données.

Définition : filtre linéaire fini

On formalise les opérations qui combinent un nombre fini de valeurs.

Soit $a = (a_j)_{j\in\mathbb{Z}}$ une suite réelle nulle sauf pour un nombre fini d'indices. Le filtre linéaire fini de coefficients a transforme $X = (X_t)$ en $Y = a(B)X$ défini par

$$Y_t = (a(B)X)_t = \sum_{j\in\mathbb{Z}} a_j X_{t-j}.$$

Définition : filtre linéaire fini

On formalise les opérations qui combinent un nombre fini de valeurs.

Soit $a = (a_j)_{j\in\mathbb{Z}}$ une suite réelle nulle sauf pour un nombre fini d'indices. Le filtre linéaire fini de coefficients a transforme $X = (X_t)$ en $Y = a(B)X$ défini par

$$Y_t = (a(B)X)_t = \sum_{j\in\mathbb{Z}} a_j X_{t-j}.$$

Le coefficient a_j est le poids donné à la valeur X_{t-j}.

Définition : filtre linéaire fini

On formalise les opérations qui combinent un nombre fini de valeurs.

Soit $a = (a_j)_{j\in\mathbb{Z}}$ une suite réelle nulle sauf pour un nombre fini d'indices. Le filtre linéaire fini de coefficients a transforme $X = (X_t)$ en $Y = a(B)X$ défini par

$$Y_t = (a(B)X)_t = \sum_{j\in\mathbb{Z}} a_j X_{t-j}.$$

Le coefficient a_j est le poids donné à la valeur X_{t-j}.

La somme est finie : à ce stade, il n'y a pas de question de convergence.

Premier exemple : une moyenne mobile

Une moyenne mobile est un filtre qui remplace une valeur par une moyenne de valeurs voisines.

Par exemple, la moyenne mobile passée sur 3 valeurs est

$$Y_t = \frac{X_t + X_{t-1} + X_{t-2}}{3}.$$

C'est le filtre de coefficients

$$a_0 = a_1 = a_2 = \frac{1}{3}.$$

Premier exemple : une moyenne mobile

Une moyenne mobile est un filtre qui remplace une valeur par une moyenne de valeurs voisines.

Par exemple, la moyenne mobile passée sur 3 valeurs est

$$Y_t = \frac{X_t + X_{t-1} + X_{t-2}}{3}.$$

C'est le filtre de coefficients

$$a_0 = a_1 = a_2 = \frac{1}{3}.$$

Pourquoi cela lisse ? Une valeur très haute ou très basse est mélangée avec ses voisines : les variations rapides sont atténuées.

Exemples de filtres finis

Filtre	**Coefficients non nuls**	**Série transformée**
identité	$a_0 = 1$	$Y_t = X_t$
retard	$a_1 = 1$	$Y_t = X_{t-1}$
différence ordinaire	$a_0 = 1,\ a_1 = -1$	$Y_t = X_t - X_{t-1}$
différence saisonnière	$a_0 = 1,\ a_s = -1$	$Y_t = X_t - X_{t-s}$
moyenne mobile passée	$a_0 = a_1 = a_2 = 1/3$	$Y_t = (X_t + X_{t-1} + X_{t-2})/3$

Exemples de filtres finis

Filtre	**Coefficients non nuls**	**Série transformée**
identité	$a_0 = 1$	$Y_t = X_t$
retard	$a_1 = 1$	$Y_t = X_{t-1}$
différence ordinaire	$a_0 = 1,\ a_1 = -1$	$Y_t = X_t - X_{t-1}$
différence saisonnière	$a_0 = 1,\ a_s = -1$	$Y_t = X_t - X_{t-s}$
moyenne mobile passée	$a_0 = a_1 = a_2 = 1/3$	$Y_t = (X_t + X_{t-1} + X_{t-2})/3$

Les différences vues avec `diff()` sont donc aussi des filtres. Elles apparaissent ici parce qu'on veut comprendre mathématiquement ce qu'elles enlèvent ou conservent.

Convention d'indice : passé ou futur ?

Dans la formule

$$Y_t = \sum_j a_j X_{t-j},$$

le signe de j indique où se trouve la valeur utilisée.

Convention d'indice : passé ou futur ?

Dans la formule

$$Y_t = \sum_j a_j X_{t-j},$$

le signe de j indique où se trouve la valeur utilisée.

Si $j > 0$, alors X_{t-j} est une valeur passée.

$$a_2 X_{t-2}$$

utilise la valeur deux dates avant t.

Si $j < 0$, alors $X_{t-j} = X_{t+|j|}$ est une valeur future.

$$a_{-1} X_{t+1}$$

utilise la valeur une date après t.

Convention d'indice : passé ou futur ?

Dans la formule

$$Y_t = \sum_j a_j X_{t-j},$$

le signe de j indique où se trouve la valeur utilisée.

Si $j > 0$, alors X_{t-j} est une valeur passée.

$$a_2 X_{t-2}$$

utilise la valeur deux dates avant t.

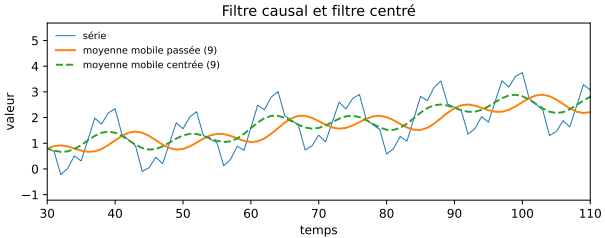
Si $j < 0$, alors $X_{t-j} = X_{t+|j|}$ est une valeur future.

$$a_{-1} X_{t+1}$$

utilise la valeur une date après t.

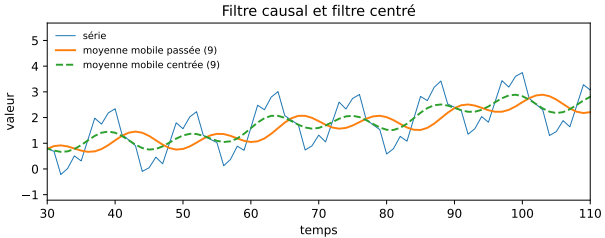
Cette convention est celle qui rend la notation compatible avec $B^j X_t = X_{t-j}$.

Filtre causal ou filtre centré ?



Un filtre est **causal** s'il n'utilise que le présent et le passé : $a_j = 0$ pour $j < 0$.

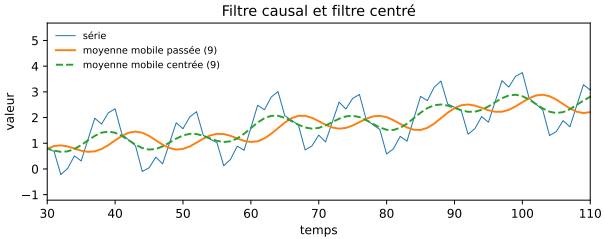
Filtre causal ou filtre centré ?



Un filtre est **causal** s'il n'utilise que le présent et le passé : $a_j = 0$ pour $j < 0$.

Un filtre est **centré** si ses poids sont répartis autour de la date t.

Filtre causal ou filtre centré ?



Un filtre est **causal** s'il n'utilise que le présent et le passé : $a_j = 0$ pour $j < 0$.

Un filtre est **centré** si ses poids sont répartis autour de la date t.

Un filtre centré est naturel pour décrire une série historique. Pour prévoir en temps réel, il est impossible d'utiliser X_{t+1}.

Linéarité : pourquoi les filtres sont faciles à analyser

Pour toutes séries X, W et tous réels α, β,

$$a(B)(\alpha X + \beta W) = \alpha a(B)X + \beta a(B)W.$$

Linéarité : pourquoi les filtres sont faciles à analyser

Pour toutes séries X, W et tous réels α, β,

$$a(B)(\alpha X + \beta W) = \alpha a(B)X + \beta a(B)W.$$

Si une série se décompose en tendance + saisonnalité + bruit, on peut étudier séparément l'effet du filtre sur chaque composante.

Linéarité : pourquoi les filtres sont faciles à analyser

Pour toutes séries X, W et tous réels α, β,

$$a(B)(\alpha X + \beta W) = \alpha a(B)X + \beta a(B)W.$$

Si une série se décompose en tendance + saisonnalité + bruit, on peut étudier séparément l'effet du filtre sur chaque composante.

C'est la raison pour laquelle on regarde maintenant l'effet des filtres sur des composantes simples : constantes, tendances et motifs périodiques.

Somme des coefficients : moyenne ou différence ?

Supposons d'abord que la série soit constante : $X_t = c$ pour tout t.

$$Y_t = \sum_j a_j X_{t-j} = \sum_j a_j c = c \sum_j a_j.$$

Somme des coefficients : moyenne ou différence ?

Supposons d'abord que la série soit constante : $X_t = c$ pour tout t.

$$Y_t = \sum_j a_j X_{t-j} = \sum_j a_j c = c \sum_j a_j.$$

Moyenne mobile : $a_0 = a_1 = a_2 = 1/3$.

$$\sum_j a_j = 1, \qquad Y_t = \frac{c + c + c}{3} = c.$$

Elle conserve un niveau constant.

Différence : $a_0 = 1$, $a_1 = -1$.

$$\sum_j a_j = 0, \qquad Y_t = c - c = 0.$$

Elle annule un niveau constant.

Deux filtres importants : les différences

Les différences apparaissent ici parce qu'elles sont à la fois des commandes `R` très utilisées et des filtres linéaires très simples.

Deux filtres importants : les différences

Les différences apparaissent ici parce qu'elles sont à la fois des commandes `R` très utilisées et des filtres linéaires très simples.

Différence ordinaire :

$$(1 - B)X_t = X_t - X_{t-1}.$$

Elle compare une valeur à la valeur précédente.

Différence saisonnière :

$$(1 - B^s)X_t = X_t - X_{t-s}.$$

Elle compare une valeur à la même position dans le cycle précédent.

Deux filtres importants : les différences

Les différences apparaissent ici parce qu'elles sont à la fois des commandes R très utilisées et des filtres linéaires très simples.

Différence ordinaire :

$$(1 - B)X_t = X_t - X_{t-1}.$$

Elle compare une valeur à la valeur précédente.

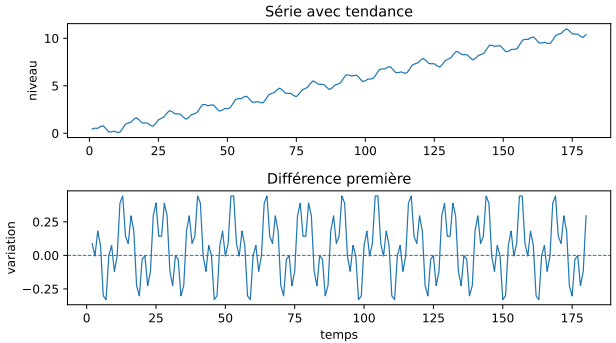
Différence saisonnière :

$$(1 - B^s)X_t = X_t - X_{t-s}.$$

Elle compare une valeur à la même position dans le cycle précédent.

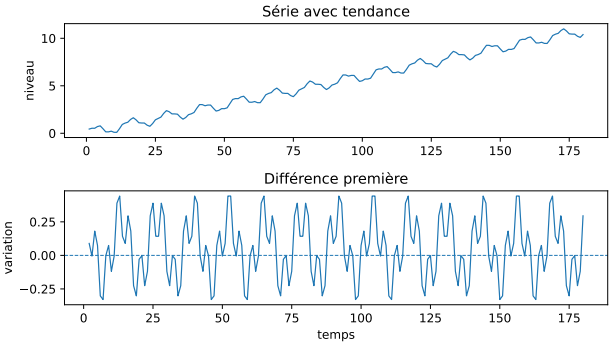
On les étudie maintenant comme des filtres pour comprendre pourquoi elles peuvent réduire une tendance ou une saisonnalité.

Différence ordinaire et tendance



La différence ordinaire ne sert pas seulement à calculer une variation : elle modifie la tendance.

Différence ordinaire et tendance



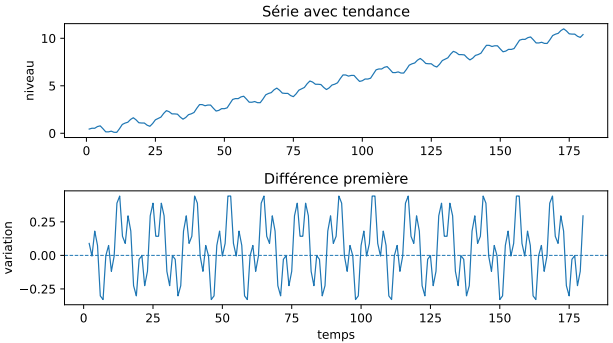
La différence ordinaire ne sert pas seulement à calculer une variation : elle modifie la tendance.

Si $p(t)$ est un polynôme de degré $d \geq 1$, alors

$$\Delta p(t) = p(t) - p(t-1)$$

est un polynôme de degré $d - 1$.

Différence ordinaire et tendance



La différence ordinaire ne sert pas seulement à calculer une variation : elle modifie la tendance.

Si $p(t)$ est un polynôme de degré $d \geq 1$, alors

$$\Delta p(t) = p(t) - p(t-1)$$

est un polynôme de degré $d - 1$.

Une tendance linéaire devient constante après une différence ; elle disparaît après deux différences.

Différence saisonnière et saisonnalité

Supposons que s_t soit une composante exactement périodique de période s :

$$s_t = s_{t-s}.$$

Différence saisonnière et saisonnalité

Supposons que s_t soit une composante exactement périodique de période s :

$$s_t = s_{t-s}.$$

La différence saisonnière est le filtre $1 - B^s$. Appliqué à cette composante, il donne

$$(1 - B^s)s_t = s_t - s_{t-s} = 0.$$

Différence saisonnière et saisonnalité

Supposons que s_t soit une composante exactement périodique de période s :

$$s_t = s_{t-s}.$$

La différence saisonnière est le filtre $1 - B^s$. Appliqué à cette composante, il donne

$$(1 - B^s)s_t = s_t - s_{t-s} = 0.$$

La raison est simple : on soustrait deux valeurs situées au même endroit du cycle saisonnier. Pour un motif parfaitement répété, elles sont égales.

Différence saisonnière et saisonnalité

Supposons que s_t soit une composante exactement périodique de période s :

$$s_t = s_{t-s}.$$

La différence saisonnière est le filtre $1 - B^s$. Appliqué à cette composante, il donne

$$(1 - B^s)s_t = s_t - s_{t-s} = 0.$$

La raison est simple : on soustrait deux valeurs situées au même endroit du cycle saisonnier. Pour un motif parfaitement répété, elles sont égales.

En pratique, une saisonnalité réelle n'est pas toujours parfaitement répétée. La différence saisonnière la réduit souvent, mais ne garantit pas à elle seule que la série obtenue soit stationnaire.

Synthèse : ce qu'il faut savoir faire après cette séance

1. Contrôler l'horloge d'un objet `ts` avec `start()`, `end()`, `frequency()` et `tsp()`.
2. Distinguer date numérique et position saisonnière avec `time()` et `cycle()`.
3. Extraire une fenêtre temporelle avec `window()` sans perdre le calendrier.
4. Calculer une différence ordinaire ou saisonnière avec `diff()`.
5. Repérer les `NA` et comprendre qu'une interpolation n'est pas une observation.
6. Comprendre une moyenne mobile et une différence comme deux exemples de filtres linéaires finis.
7. Distinguer un filtre causal d'un filtre centré.

Synthèse : ce qu'il faut savoir faire après cette séance

1. Contrôler l'horloge d'un objet `ts` avec `start()`, `end()`, `frequency()` et `tsp()`.
2. Distinguer date numérique et position saisonnière avec `time()` et `cycle()`.
3. Extraire une fenêtre temporelle avec `window()` sans perdre le calendrier.
4. Calculer une différence ordinaire ou saisonnière avec `diff()`.
5. Repérer les `NA` et comprendre qu'une interpolation n'est pas une observation.
6. Comprendre une moyenne mobile et une différence comme deux exemples de filtres linéaires finis.
7. Distinguer un filtre causal d'un filtre centré.

La prochaine étape sera de passer des filtres finis aux filtres infinis, puis de voir comment un filtre transforme un bruit blanc en série dépendante.

Avant le TD/TP

Le TD/TP 2 reprend les manipulations de cette séance dans `R`.

- vérifier l'horloge de `co2` et `UKgas` ;
- tracer un chronogramme et zoomer avec `window()` ;
- comparer `time()` et `cycle()` ;
- calculer une différence saisonnière ;
- créer et inspecter des valeurs manquantes.

Avant le TD/TP

Le TD/TP 2 reprend les manipulations de cette séance dans `R`.

- vérifier l'horloge de `co2` et `UKgas` ;
- tracer un chronogramme et zoomer avec `window()` ;
- comparer `time()` et `cycle()` ;
- calculer une différence saisonnière ;
- créer et inspecter des valeurs manquantes.

Avant de modéliser, on commence par contrôler le calendrier et comprendre les transformations appliquées.