



## 2 - Analyse graphique

ALAIN QUARTIER-LA-TENTE

# Objectifs de cette séquence

---

Présenter différents outils permettant d'analyser les séries temporelles

# Questions de positionnement

---

Quelles sont les outils de visualisation possibles d'une série temporelle ?

Comment analyser la dépendance temporelle ?

Comment analyser la saisonnalité ?

Pourquoi il est parfois nécessaire de transformer une série ?

# Sommaire

---

## 1. Analyse graphique

- 1.1 Chronogramme
- 1.2 Les composantes d'une série temporelle
- 1.3 Monthplot
- 1.4 Seasonal plot
- 1.5 Lag plot
- 1.6 ACF
- 1.7 Densité spectrale

## 2. Transformation de la série

## 3. Conclusion

# Chronogramme

---

La première analyse graphique possible est le chronogramme, permet :

- avoir une vue d'ensemble
- identifier des ruptures, points atypiques, valeurs manquantes

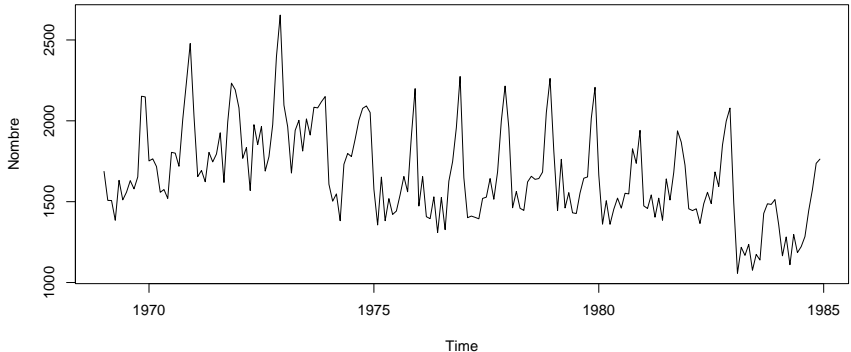
Sous **R** : `plot()` (`=plot.ts()`) ou `forecast::autoplot()` pour une version `ggplot2`.

Exemple avec le nombre mensuel de conducteurs de voitures tués en Grande-Bretagne entre 1969 et 1984.

# Exemple (1)

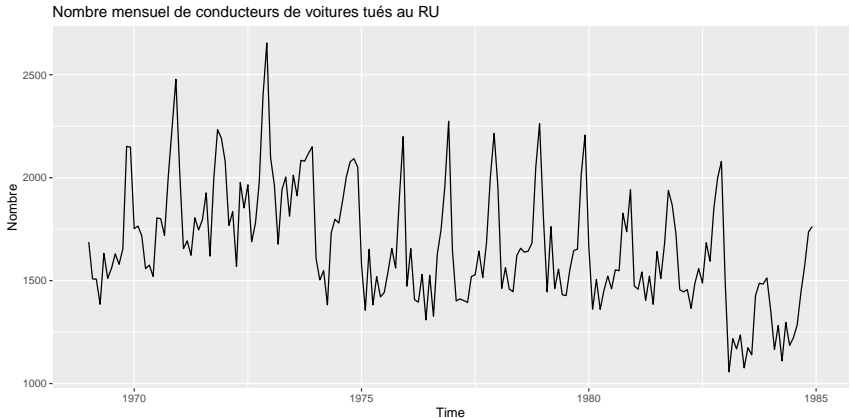
```
plot(UKDriverDeaths,  
     main = "Nombre mensuel de conducteurs de voitures tués au RU",  
     ylab = "Nombre")
```

Nombre mensuel de conducteurs de voitures tués au RU



## Exemple (2)

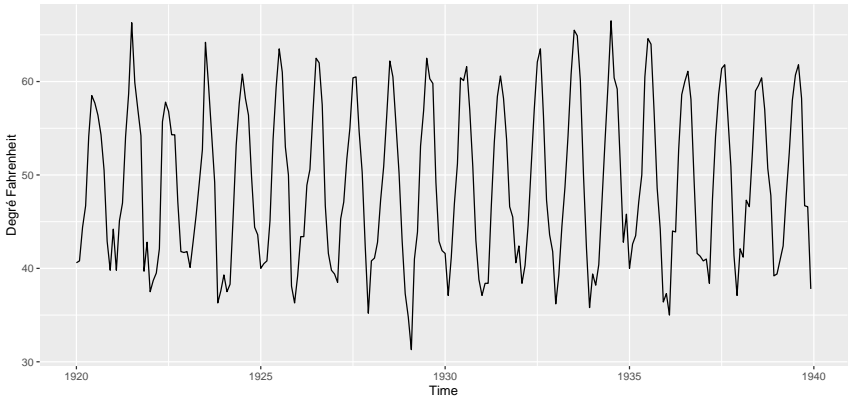
```
library(forecast);library(ggplot2)
autoplot(UKDriverDeaths) +
  labs(title = "Nombre mensuel de conducteurs de voitures tués au RU",
       y = "Nombre")
```



## Exemple (3)

```
autoplot(nottem) +  
  ggtitle("Température mensuelle moyenne au Château de Nottingham") +  
  ylab("Degré Fahrenheit")
```

Température mensuelle moyenne au Château de Nottingham





# Les différentes composantes

---

On considère généralement qu'une série temporelle peut s'écrire en fonction de différentes composantes :

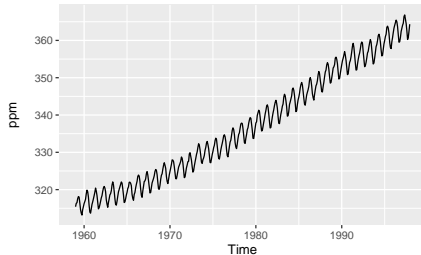
- tendance : évolution de long terme de la série
- cycle : mouvement cyclique (d'au moins 2 ans) autour de la tendance à une à une fréquence non fixée
- saisonnalité : fréquence fixe est connue (infra-annuelle)
- les autres effets de calendrier (jours ouvrables, etc.)
- irrégulier : composante d'erreur (idéalement faible)

# Exemple (1)

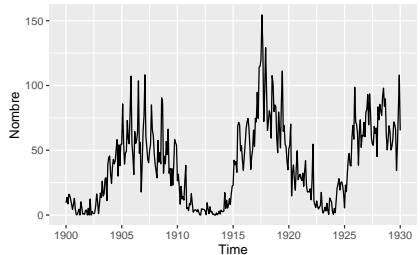
```
library(patchwork)
(
  autoplot(co2) +
    labs(title = latex2exp::TeX(
      "Concentration mensuelle en CO$_2$ à Mauna Loa"
    ),
    y = "ppm") +
  autoplot(window(sunspots, start = 1900, end = 1930)) +
    labs(title = "Nombre mensuel de taches solaires", y = "Nombre")
) / (
  autoplot(austres) +
    labs(title = "Nombre trimestriel de résidents australiens",
    y = "milliers") +
  autoplot(diff(EuStockMarkets[, "CAC"])) +
    labs(
      title = "Variations quotidiennes du cours\nde clôture du CAC 40",
      y = "Variation dans les prix"
    )
)
```

## Exemple (2)

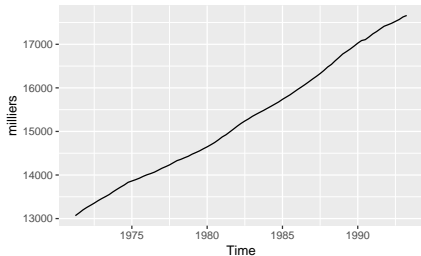
Concentration mensuelle en  $\text{CO}_2$  à Mauna Loa



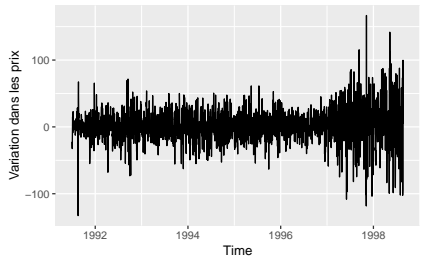
Nombre mensuel de taches solaires



Nombre trimestriel de résidents australiens




Variations quotidiennes du cours de clôture du CAC 40



# Monthplot

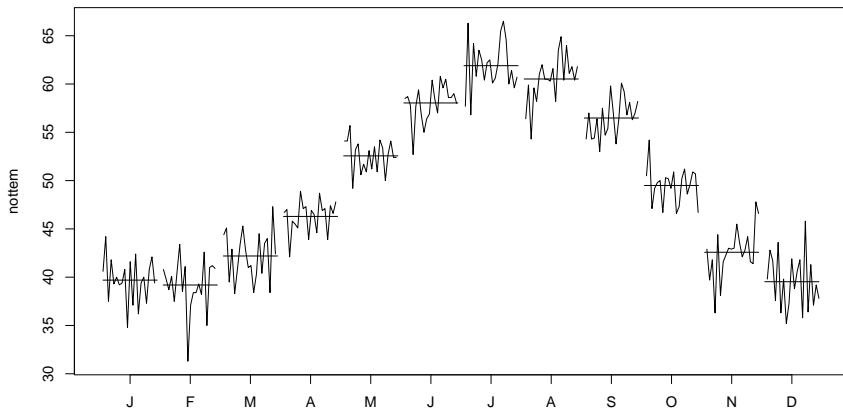
---

Permet d'étudier le schéma saisonnier et l'évolution de la saisonnalité au cours du temps

 : `monthplot()`, `forecast::ggmonthplot()` ou `feasts::gg_subseries()` pour la version tsibble.

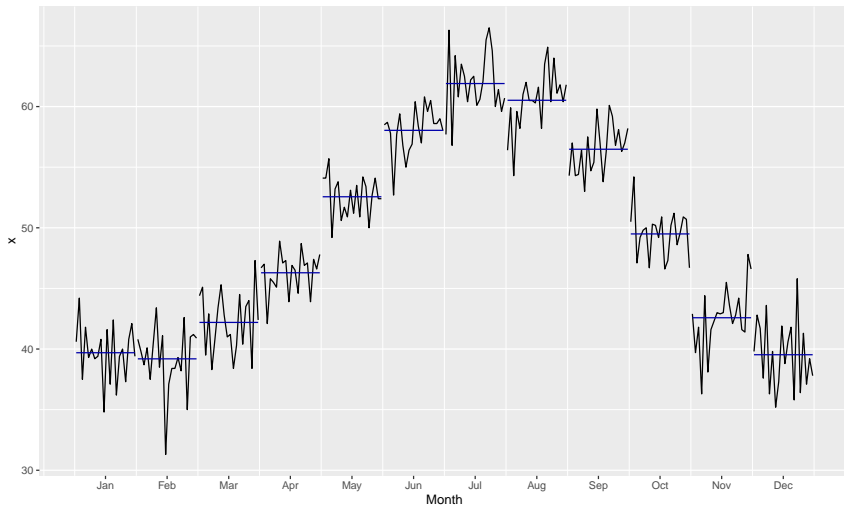
# Exemple (1)

```
monthplot(nottem)
```



# Exemple (2)

```
ggmonthplot(nottem)
```



# Seasonal plot

---

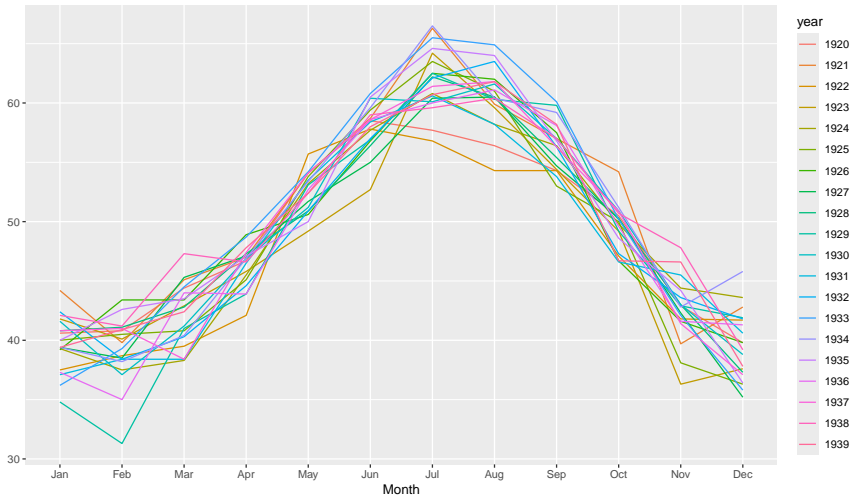
Permet d'étudier le schéma saisonnier et les écarts importants par rapport à ce schéma

**R** : `forecast::seasonplot()`, `forecast::ggseasonplot()` ou `feasts::gg_season()` pour la version tsibble.

# Exemple

```
ggseasonplot(nottem)
```

Seasonal plot: nottem

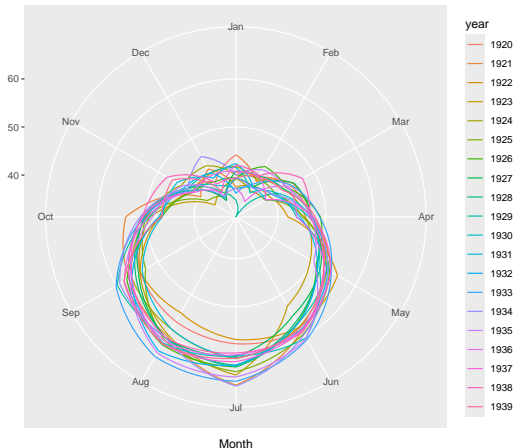




# Exemple : avec option polar = TRUE

```
ggseasonplot(nottem, polar = TRUE)
```

Seasonal plot: nottem



# Lag plot

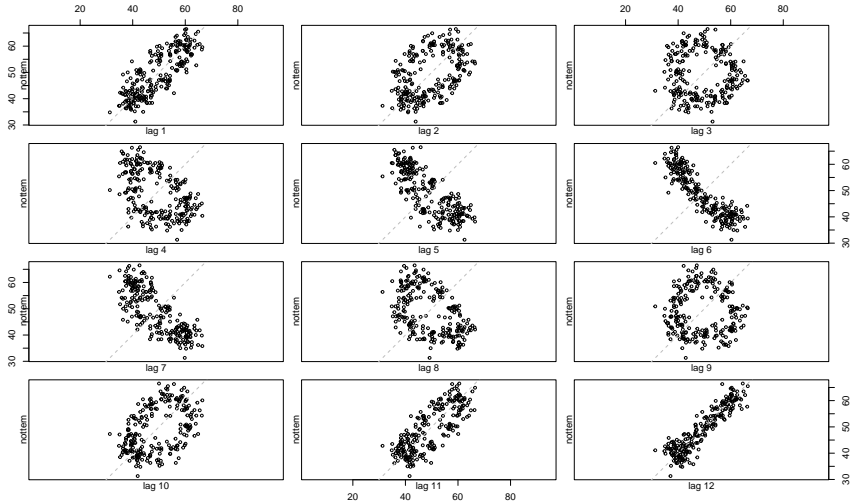
---

Permet d'étudier la dépendance d'une série avec son passé

**R** : `lag.plot()`, `forecast::gglagplot()` ou `feasts::gg_lag()` pour la version tsibble.

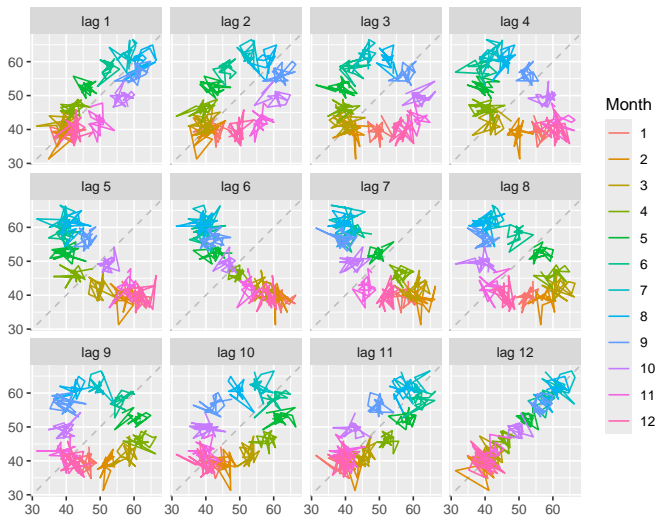
# Exemple (1)

```
lag.plot(nottem, lags = 12)
```



## Exemple (2)

```
gglagplot(nottem, lags = 12)
```



# Analyse

---

- Corrélation positive entre  $X_t$  et  $X_{t-1}$  : température du mois  $t$  est proche de celle du mois précédent
- Corrélation positive entre  $X_t$  et  $X_{t-12}$  : température du mois  $t$  est proche de celle du mois  $t$  de l'année précédente
- Corrélation négative entre  $X_t$  et  $X_{t-6}$  : il y a des saisons
- Anneaux pour les retards 2, 3, 4, 8, 9 et 10. Ex pour les retards 3 : en avril/mai/juin la température est plus élevée que trois mois avant (au-dessus de la bissextre)

# ACF

---

Autocorrélogrammes : autocorrélations associées aux graphiques précédents

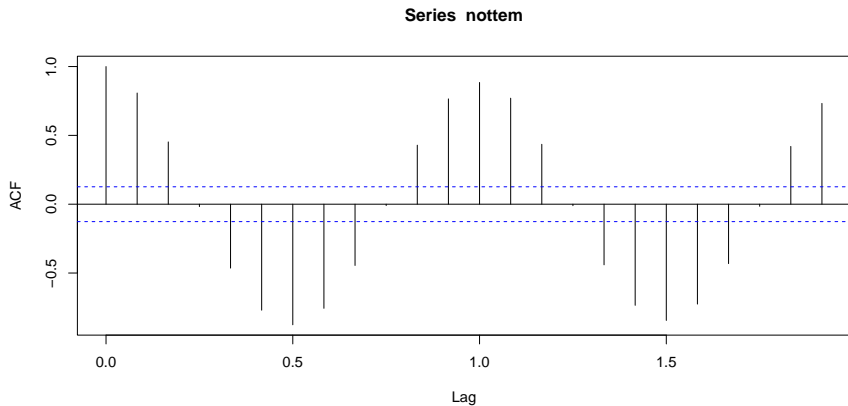
$$\hat{\gamma}(k) = \frac{1}{n} \sum_{t=k+1}^n (y_t - \bar{y})(y_{t-k} - \bar{y}) \text{ et } \hat{\rho}(k) = \frac{\hat{\gamma}(k)}{\hat{\gamma}(0)}$$

Permet d'identifier les tendances et la saisonnalité

**R** : `acf()`, `forecast::ggAcf()` ou `autoplot(feast::ACF())` pour la version `tsibble`.

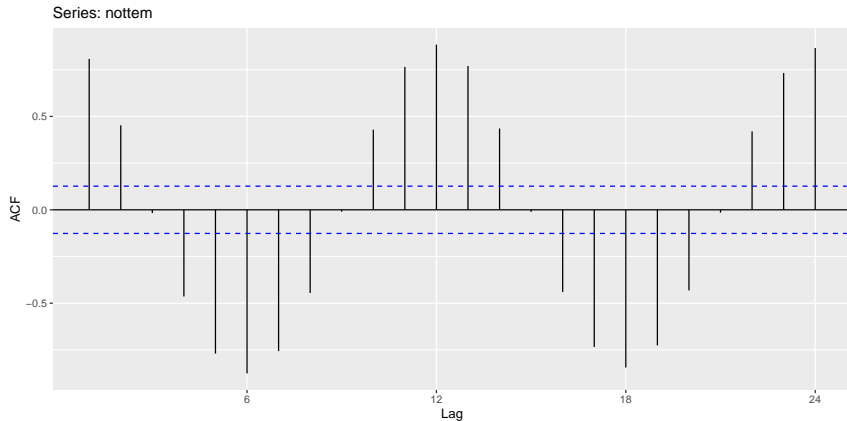
# Exemple (1)

```
acf(nottem)
```



## Exemple (2)

```
ggAcf(nottem)
```



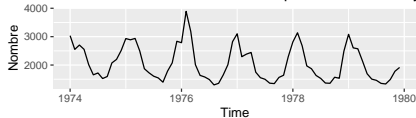


# Exercice

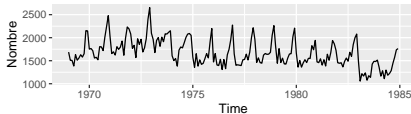
1. Nombre de passagers aériens mensuel



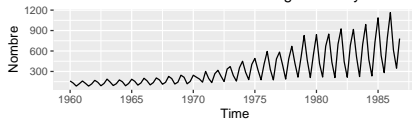
2. Nombre de décès de maladies pulmonaires au Royaume-Uni



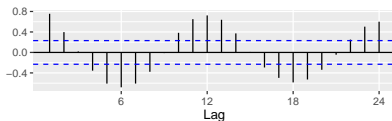
3. Nombre de conducteurs de voitures tués au RU



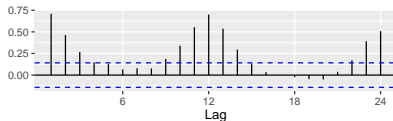
4. Consommation trimestrielle de gaz au Royaume-Uni



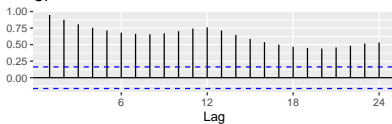
A.



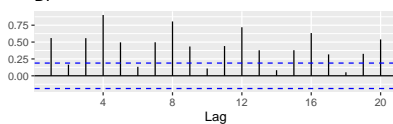
B.



C.



D.



# Solution

```
p1 = autoplot(AirPassengers) + labs(  
  y = "Nombre",  
  title = "1. Nombre de passagers aériens mensuel")  
acf_c = ggAcf(AirPassengers) + labs(title = "C.", y = NULL)  
p2 = autoplot(ldeaths) + labs(  
  y = "Nombre",  
  title = "2. Nombre de décès de maladies pulmonaires au Royaume-Uni")  
acf_a = ggAcf(ldeaths) + labs(title = "A.", y = NULL)  
p3 = autoplot(UKDriverDeaths) + labs(  
  y = "Nombre",  
  title = "3. Nombre de conducteurs de voitures tués au RU")  
acf_b = ggAcf(UKDriverDeaths) + labs(title = "B.", y = NULL)  
p4 = autoplot(UKgas) + labs(  
  y = "Nombre",  
  title = "4. Consommation trimestrielle de gaz au Royaume-Uni")  
acf_d = ggAcf(UKgas) + labs(title = "D.", y = NULL)  
patchwork::wrap_plots(p1, p2, p3, p4, acf_a, acf_b, acf_c, acf_d,  
  ncol = 2)
```

# Analyse spectrale

---

**Représentation classique d'une série** : fluctuations de la série en fonction du temps. . .

**Représentation spectrale** : une série peut être représentée dans le domaine des fréquences. Un spectre montre les fluctuations de la série par fréquence.

$$\text{fréquence} = \frac{2\pi}{\text{période}}$$

Interprétation : **décomposition de la variance de la série** selon les différentes fréquences

Deux méthodes d'estimation : périodogramme et spectre autorégressif

NB : fonctions définies à une constante multiplicative près

# Périodogramme (1)

Sous **R** : `spec.pgram()` ou `spectrum(method = "pgram")`

Soit  $x_1, \dots, x_n$  des observations d'une série temporelle. On suppose  $x_1, \dots, x_n$  les valeurs d'une fonction en  $1, \dots, n$  :

$$x_t = \frac{1}{\sqrt{n}} \sum_{\pi < \omega_j \leq \pi} a_j e^{it\omega_j}, \quad \omega_j = \frac{2\pi j}{n}$$

Rmq : c'est bien possible d'écrire sous cette forme car  $e_j = n^{-1/2}(e^{i\omega_j}, \dots, e^{in\omega_j})$  base orthonormée de  $\mathbb{C}^n$

Il vient :

$$x = \sum_{j|\pi < \omega_j \leq \pi} a_j e_j \quad \text{avec} \quad a_j = \langle x, e_j \rangle = \frac{1}{\sqrt{n}} \sum_{t=1}^n x_t e^{-it\omega_j}$$

## Périodogramme (2)

La valeur du périodogramme  $I(\omega_j)$  est :

$$I(\omega_j) = |a_j|^2 = \frac{1}{n} \left| \sum_{t=1}^n x_t e^{-it\omega_j} \right|^2$$

On a donc une décomposition de la variance :

$$||x||^2 = \sum I(\omega_j)$$

Rmq : dans le cas réel,  $I(\omega_j) = I(-\omega_j)$

Rmq : généralement  $x$  est normalisé (en enlevant également la tendance)

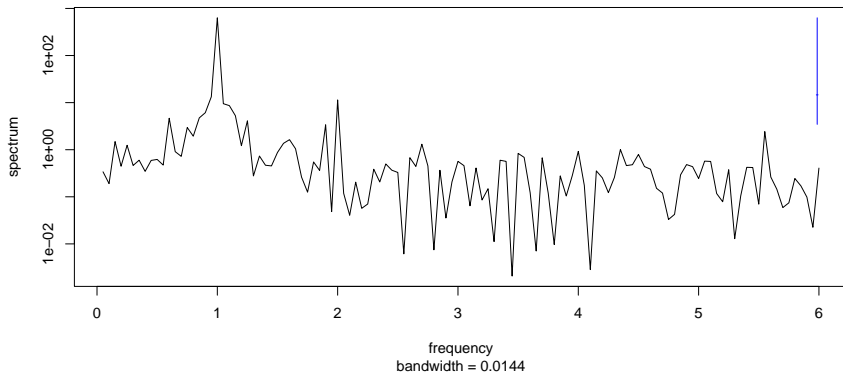
Donne les mêmes informations que les autocorrélogrammes :

$$I(\omega_j) = \sum_{|k| < n} \hat{\gamma}(k) e^{-ik\omega_j}$$

# Exemple

```
spec.pgram(nottem)
```

Series: nottem  
Raw Periodogram



# Périodogramme cumulatif Bartlett (1955)

---

Un graphique dérivé est le périodogramme cumulatif

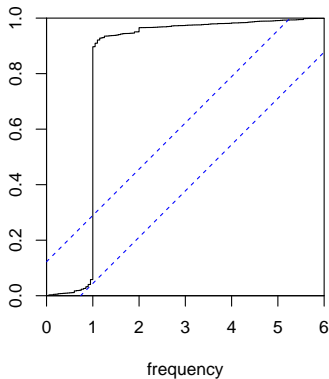
$$U(\omega) = \frac{\sum_{0 < \omega_k < \omega} I(\omega_k)}{\sum I(\omega_k)}$$

Permet de détecter des mouvements périodiques non-aléatoires : on observe un saut s'il y a un mouvement périodique et une droite si proche d'un bruit blanc

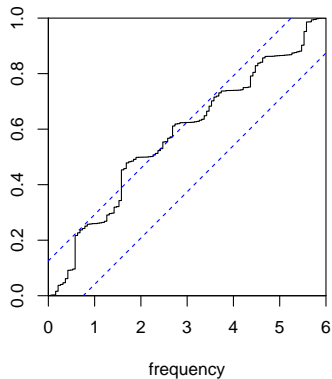
# Exemple

```
par(mfrow = c(1,2))  
cpgram(nottem)  
cpgram(diff(nottem, 12))
```

Series: nottem



Series: diff(nottem, 12)





# Spectre autorégressif

Sous **R** : `spec.ar()` ou `spectrum(method = "ar")`

Autre méthode : calculer le spectre d'un processus AR :

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \varepsilon_t$$

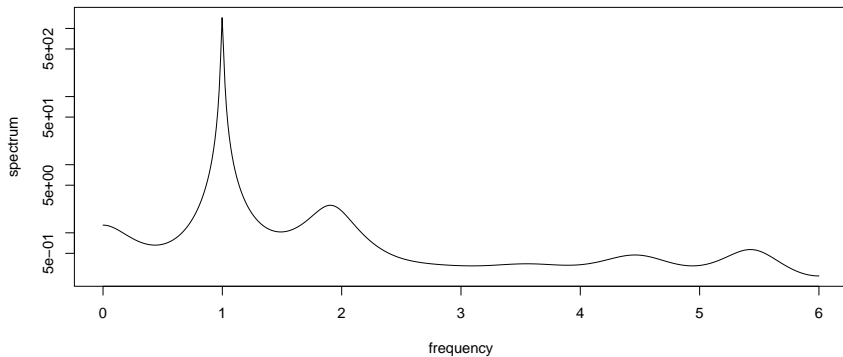
$$f(\omega) = \sigma^2 \frac{1}{2\pi \left| 1 - \sum_{j=1}^p \phi_j e^{ij\omega} \right|^2}$$

Rmq : sous R  $f(\omega) = \sigma^2 \frac{1}{\text{freq} \left| 1 - \sum_{j=1}^p \phi_j e^{ij\omega} \right|^2}$

# Exemple

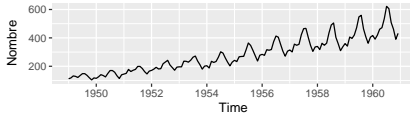
```
spec.ar(nottem)
```

Series: nottem  
AR (13) spectrum

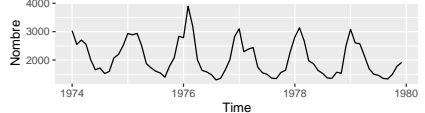


# Exercice

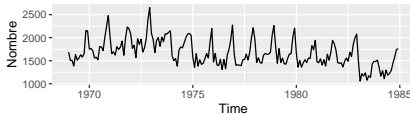
1. Nombre de passagers aériens mensuel



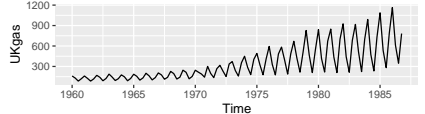
2. Nombre de décès de maladies pulmonaires au Royaume-Uni



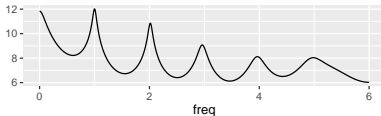
3. Nombre de conducteurs de voitures tués au RU



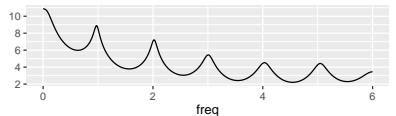
4. Consommation trimestrielle de gaz au Royaume-Uni



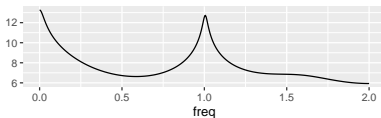
A.



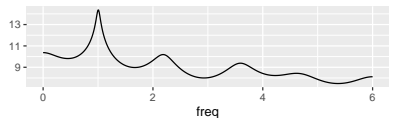
B.



C.



D.



# Solution

```

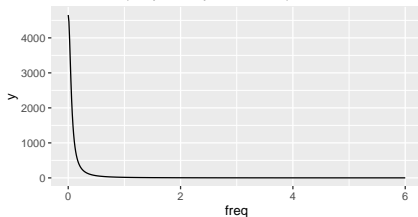
gg_spec.ar <- function(x, log = "yes", ...){
  spec = spec.ar(x, plot = FALSE, ...)
  y = spec$spec
  y = switch(log,
             yes = log(y),
             no = y,
             10 * log10(y))
  spec.matrix <- cbind.data.frame(freq = spec$freq, y = y)
  ggplot(spec.matrix, aes(x=freq,y=y)) + geom_line()
}

p1 = autoplot(AirPassengers) +
  labs(title = "1. Nombre de passagers aériens mensuel", y = "Nombre")
spec_b = gg_spec.ar(AirPassengers) + labs(title = "B.", y = NULL)
p2 = autoplot(ldeaths) +
  labs(title = "2. Nombre de décès de maladies pulmonaires au Royaume-Uni",
       y = "Nombre")
spec_d = gg_spec.ar(ldeaths) + labs(title = "D.", y = NULL)
p3 = autoplot(UKDriverDeaths) +
  labs(title = "3. Nombre de conducteurs de voitures tués au RU", y = "Nombre")
spec_a = gg_spec.ar(UKDriverDeaths) + labs(title = "A.", y = NULL)
p4 = autoplot(UKgas) +
  labs(title = "4. Consommation trimestrielle de gaz au Royaume-Uni")
spec_c = gg_spec.ar(UKgas) + labs(title = "C.", y = NULL)
patchwork::wrap_plots(p1, p2, p3, p4, spec_a, spec_b, spec_c, spec_d,
                      ncol = 2)

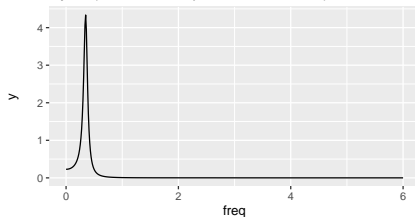
```

# Spectres théoriques (voir TP)

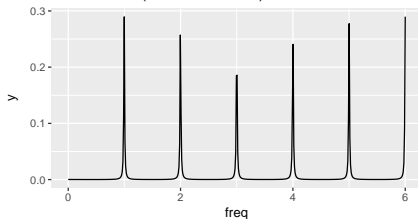
Tendance (fréquence proche de 0)



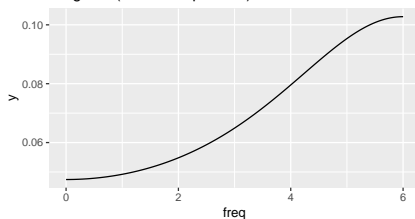
Cycle (entre 0 et fréquence saisonnière)



Saisonnalité (nombres entiers)



Irrégulier (autres fréquences)



# Sommaire

---

1. Analyse graphique

2. Transformation de la série

3. Conclusion

# Stabilisation de variance

---

Lorsque la variance de la série varie avec le niveau (signe hétéroscédasticité) il peut être utile de transformer la série.

# Stabilisation de variance

Lorsque la variance de la série varie avec le niveau (signe hétéroscédasticité) il peut être utile de transformer la série.

Une famille de transformation possible est la transformation de Box-Cox (`forecast::BoxCox()` ou `fabletools::box_cox()`):

$$f_{\lambda}(x) = \begin{cases} \log(x) & \text{si } \lambda = 0 \\ \frac{\text{sign}(x)|x|^{\lambda}-1}{\lambda} & \text{si } \lambda \neq 0 \end{cases}$$

Rmq :  $\text{sign}(x)$  permet de prendre en compte les séries négatives contrairement à la transformation initiale (voir Bickel, P. J., & Doksum, K. A. (1981))



# Stabilisation de variance

Lorsque la variance de la série varie avec le niveau (signe hétéroscédasticité) il peut être utile de transformer la série.

Une famille de transformation possible est la transformation de Box-Cox (`forecast::BoxCox()` ou `fabletools::box_cox()`):

$$f_{\lambda}(x) = \begin{cases} \log(x) & \text{si } \lambda = 0 \\ \frac{\text{sign}(x)|x|^{\lambda}-1}{\lambda} & \text{si } \lambda \neq 0 \end{cases}$$

Rmq :  $\text{sign}(x)$  permet de prendre en compte les séries négatives contrairement à la transformation initiale (voir Bickel, P. J., & Doksum, K. A. (1981))

Transformation en log plus interprétable (analyse en évolution)

# Stabilisation de variance

Lorsque la variance de la série varie avec le niveau (signe hétéroscédasticité) il peut être utile de transformer la série.

Une famille de transformation possible est la transformation de Box-Cox (`forecast::BoxCox()` ou `fabletools::box_cox()`):

$$f_{\lambda}(x) = \begin{cases} \log(x) & \text{si } \lambda = 0 \\ \frac{\text{sign}(x)|x|^{\lambda}-1}{\lambda} & \text{si } \lambda \neq 0 \end{cases}$$

Rmq :  $\text{sign}(x)$  permet de prendre en compte les séries négatives contrairement à la transformation initiale (voir Bickel, P. J., & Doksum, K. A. (1981))

Transformation en log plus interprétable (analyse en évolution)

Il existe des méthodes de sélection automatique de  $\lambda$  :

`forecast::BoxCox.lambda()` ou `feasts::guerrero()`.

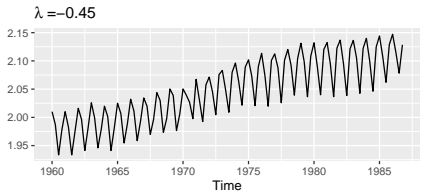
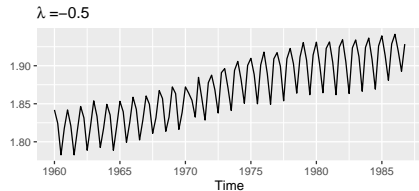
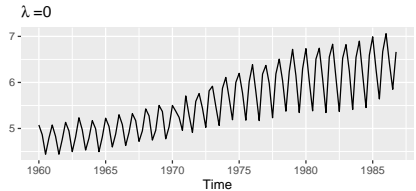
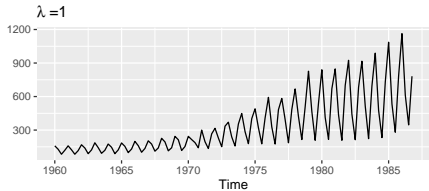
# Exemple (1)

---

```
(autoplot(BoxCox(UKgas, 1)) +
  labs(y = NULL, title = expression(lambda~"=1"))) +
autoplot(BoxCox(UKgas, 0)) +
labs(y = NULL, title = expression(lambda~"=0")) ) / (
  autoplot(BoxCox(UKgas, -0.5)) +
    labs(y = NULL, title = expression(lambda~"=-0.5")) +
    autoplot(BoxCox(UKgas, BoxCox.lambda(UKgas))) +
    labs(y = NULL, title = expression(lambda~"=-0.45"))
) + plot_annotation(
  title = "Consommation trimestrielle de gaz au Royaume-Uni"
)
```

# Exemple (2)

Consommation trimestrielle de gaz au Royaume-Uni



## En prévision

Si on utilise la transformation de Box-Cox en prévision, il faudra alors utiliser la transformation inverse (`forecast::InvBoxCox()` ou `fabletools::inv_box_cox()`) pour calculer les prévisions de la série initiale :

$$f_{\lambda}^{-1}(x) = \begin{cases} \exp(x) & \text{si } \lambda = 0 \\ \text{sign}(\lambda x + 1) |\lambda x + 1|^{1/\lambda} & \text{si } \lambda \neq 0 \end{cases}$$

## En prévision

Si on utilise la transformation de Box-Cox en prévision, il faudra alors utiliser la transformation inverse (`forecast::InvBoxCox()` ou `fabletools::inv_box_cox()`) pour calculer les prévisions de la série initiale :

$$f_{\lambda}^{-1}(x) = \begin{cases} \exp(x) & \text{si } \lambda = 0 \\ \text{sign}(\lambda x + 1) |\lambda x + 1|^{1/\lambda} & \text{si } \lambda \neq 0 \end{cases}$$



Mais l'inverse de la prévision ne sera pas égale à la moyenne de la distribution des prévisions (mais généralement à la médiane)

## En prévision

Si on utilise la transformation de Box-Cox en prévision, il faudra alors utiliser la transformation inverse (`forecast::InvBoxCox()` ou `fabletools::inv_box_cox()`) pour calculer les prévisions de la série initiale :

$$f_{\lambda}^{-1}(x) = \begin{cases} \exp(x) & \text{si } \lambda = 0 \\ \text{sign}(\lambda x + 1) |\lambda x + 1|^{1/\lambda} & \text{si } \lambda \neq 0 \end{cases}$$



Mais l'inverse de la prévision ne sera pas égale à la moyenne de la distribution des prévisions (mais généralement à la médiane)

Si médiane non adaptée (par exemple agrégation de prévisions), il faut faire une correction du biais :

$$\widetilde{f_{\lambda}^{-1}}(x) = \begin{cases} \exp(x) \left[ 1 + \frac{\sigma_h^2}{2} \right] & \text{si } \lambda = 0 \\ \text{sign}(\lambda x + 1) |\lambda x + 1|^{1/\lambda} \left[ 1 + \frac{\sigma_h^2(1-\lambda)}{2(\lambda x + 1)^{2\lambda}} \right] & \text{si } \lambda \neq 0 \end{cases}$$

Voir <https://aqlt.github.io/formation/formation.2025.ast/backcasttransform>

# Sommaire

---

1. Analyse graphique
2. Transformation de la série
3. Conclusion



# Analyse graphique indispensable

---

Avant d'appliquer tout modèle l'analyse graphique est indispensable ! Permet d'identifier :

- les différentes composantes (saisonnalité ? tendance ?)
- les points atypiques
- les signes d'hétéroscédasticité

De nombreux types de graphiques vous permettent d'étudier différemment la série

Pour stabiliser la variance, une transformation de la série peut être nécessaire de transformer la série (log ou Box-Cox)

# Bibliographie

---

Aragon, Y. (2011), *Séries temporelles avec R. Méthodes et cas*, Springer.

Brockwell, P. J. and Davis, R. A. (1991) *Time Series: Theory and Methods*. Second edition. Springer.

Avec `ts()` :

Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. [OTexts.com/fpp2](https://otexts.com/fpp2). Accessed on june 2025.

Sur `tsibble` :

Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. [OTexts.com/fpp3](https://otexts.com/fpp3). Accessed on june 2025.