

LemnaLaboratoire d'Économie et de
Management Nantes-AtlantiqueInstitut national de la statistique
et des études économiquesMesurer pour comprendre

Estimation en temps réel de la tendance-cycle : Apport de l'utilisation des filtres asymétriques dans la détection des points de retournement

ALAIN QUARTIER-LA-TENTE

08 novembre 2023

Sommaire

1. Introduction

2. Méthodes étudiées

3. Extensions

4. Comparaison des méthodes

5. Conclusion

6. Et maintenant ?

Présentation


Travaille à l'Insee depuis 2015 : aux enquêtes de conjoncture (2015-2017), en tant que méthodologue sur les CVS-CJO (2017-2019) puis en tant que chargé d'études macroéconomiques (2021-)

Ensai (2012-2015) puis Ensae (2019-2021) et inscription en thèse à temps partiel en 2021

Présentation

Travaille à l'Insee depuis 2015 : aux enquêtes de conjoncture (2015-2017), en tant que méthodologue sur les CVS-CJO (2017-2019) puis en tant que chargé d'études macroéconomiques (2021-)

Ensaï (2012-2015) puis Ensaë (2019-2021) et inscription en thèse à temps partiel en 2021

Passionné de , développe et maintient plusieurs packages autour de la désaisonnalisation

Travaux effectués

Principalement autour de l'utilisation de moyennes mobiles asymétriques pour l'estimation en temps réel de la tendance-cycle et la détection des points de retournement.

- Un document de travail Insee en cours avec code ouvert et entièrement reproductible, pour l'instant disponible ici <https://aqlt.github.io/DT-est-tr-tc/>
- Une soumission en cours au Journal of Official Statistics (JOS).
- Un package `rjd3filters` pour créer et manipuler les moyennes mobiles
- Divers présentations sur le contenu statistique et informatique.

Travaux effectués

Principalement autour de l'utilisation de moyennes mobiles asymétriques pour l'estimation en temps réel de la tendance-cycle et la détection des points de retournement.

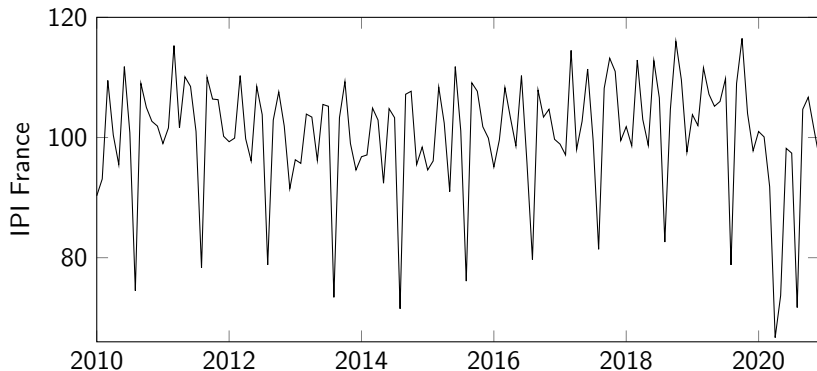
- Un document de travail Insee en cours avec code ouvert et entièrement reproductible, pour l'instant disponible ici <https://aqlt.github.io/DT-est-tr-tc/>
- Une soumission en cours au Journal of Official Statistics (JOS).
- Un package `rjd3filters` pour créer et manipuler les moyennes mobiles
- Divers présentations sur le contenu statistique et informatique.

NB : études rédigées seul

La tendance-cycle (1)

X_t (ex : IPI France) se décompose en plusieurs composantes inobservées :

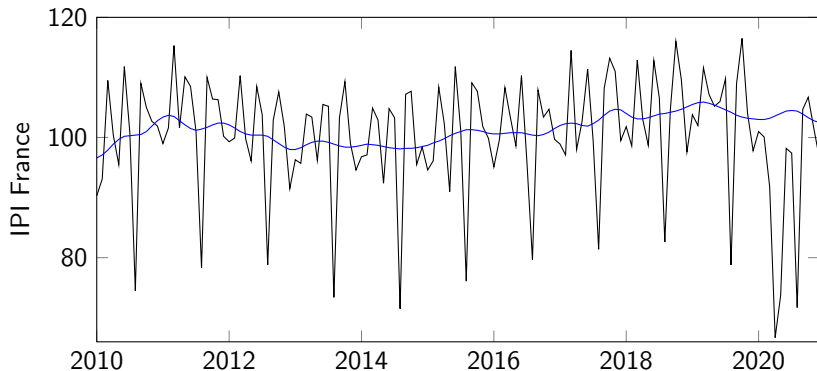
$$X_t = \quad \quad \quad (\text{décomposition additive})$$



La tendance-cycle (1)

X_t (ex : IPI France) se décompose en plusieurs composantes inobservées :

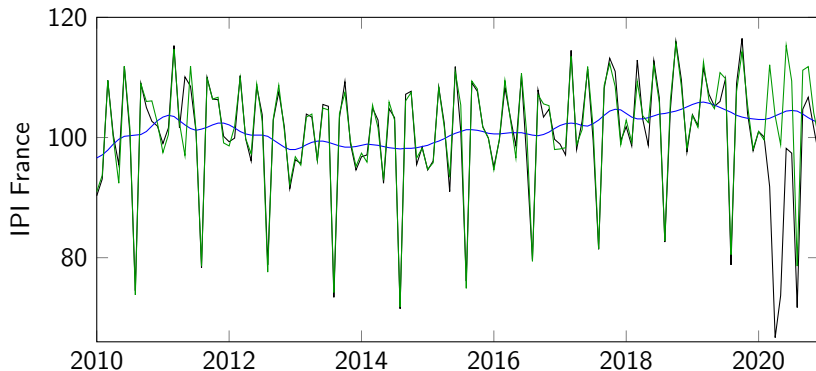
$$X_t = \underbrace{TC_t}_{\text{tendance-cycle}} \quad (\text{décomposition additive})$$



La tendance-cycle (1)

X_t (ex : IPI France) se décompose en plusieurs composantes inobservées :

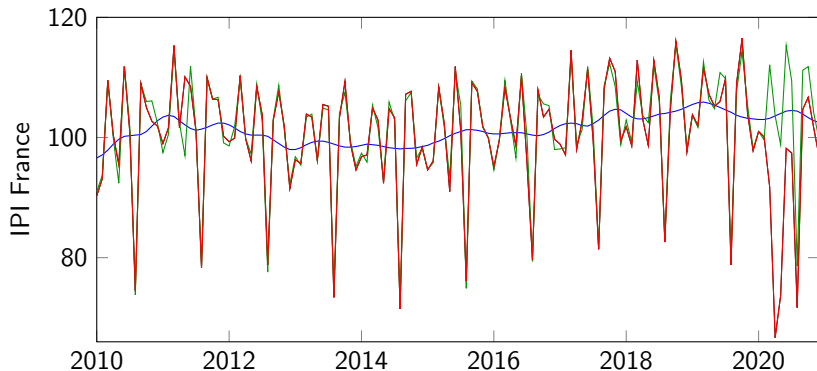
$$X_t = \underbrace{TC_t}_{\text{tendance-cycle}} + \underbrace{S_t}_{\text{saisonnalité}} \quad (\text{décomposition additive})$$



La tendance-cycle (1)

X_t (ex : IPI France) se décompose en plusieurs composantes inobservées :

$$X_t = \underbrace{TC_t}_{\text{tendance-cycle}} + \underbrace{S_t}_{\text{saisonnalité}} + \underbrace{I_t}_{\text{irrégulier}} \quad (\text{décomposition additive})$$

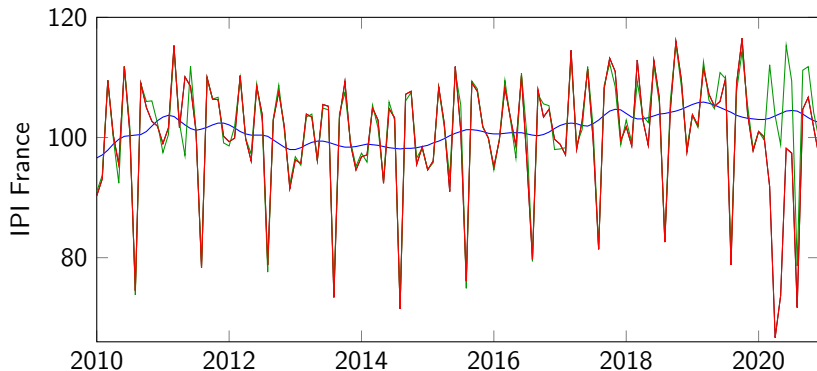


La tendance-cycle (1)

X_t (ex : IPI France) se décompose en plusieurs composantes inobservées :

$$X_t = \underbrace{TC_t}_{\text{tendance-cycle}} + \underbrace{S_t}_{\text{saisonnalité}} + \underbrace{I_t}_{\text{irrégulier}} \quad (\text{décomposition additive})$$

tendance et cycle ici estimés *simultanément*



La tendance-cycle (2)

Pour l'analyse conjoncturelle, on étudie généralement des séries désaisonnalisées

$$X_t - S_t = TC_t + I_t$$

La tendance-cycle (2)

Pour l'analyse conjoncturelle, on étudie généralement des séries désaisonnalisées

$$X_t - S_t = TC_t + I_t$$

La présence de l'irrégulier peut rendre l'interprétation difficile, un lissage supplémentaire peut être utilisé :

$$(X_t - S_t) - I_t = TC_t$$

La tendance-cycle (2)

Pour l'analyse conjoncturelle, on étudie généralement des séries désaisonnalisées

$$X_t - S_t = TC_t + I_t$$

La présence de l'irrégulier peut rendre l'interprétation difficile, un lissage supplémentaire peut être utilisé :

$$(X_t - S_t) - I_t = TC_t$$

Tendance-cycle publiée par peu d'instituts (ONS, Statistics Canada, ABS) mais volonté de faire des bonnes pratiques au niveau européen (Destatis). Utile pour analyser le cycle des affaires.

La tendance-cycle (2)

Pour l'analyse conjoncturelle, on étudie généralement des séries désaisonnalisées

$$X_t - S_t = TC_t + I_t$$

La présence de l'irrégulier peut rendre l'interprétation difficile, un lissage supplémentaire peut être utilisé :

$$(X_t - S_t) - I_t = TC_t$$

Tendance-cycle publiée par peu d'instituts (ONS, Statistics Canada, ABS) mais volonté de faire des bonnes pratiques au niveau européen (Destatis). Utile pour analyser le cycle des affaires.

Critères importants de la tendance-cycle :

1. minimiser les révisions
2. minimiser le nombre de faux points de retournement
3. détecter correctement et *rapidement* les (bons) points de retournement

Estimations de la TC et moyennes mobiles (1)



Objectif différent de celui des méthodes d'analyse des cycles économiques

Estimations de la TC et moyennes mobiles (1)



Objectif différent de celui des méthodes d'analyse des cycles économiques
 TC_t généralement estimée sur une série *sans* saisonnalité

Estimations de la TC et moyennes mobiles (1)



Objectif différent de celui des méthodes d'analyse des cycles économiques

TC_t généralement estimée sur une série *sans* saisonnalité

Méthode de décomposition X-13ARIMA une des plus utilisées : études de méthodes non-paramétriques pour estimer TC_t

Estimations de la TC et moyennes mobiles (1)



Objectif différent de celui des méthodes d'analyse des cycles économiques

TC_t généralement estimée sur une série *sans* saisonnalité

Méthode de décomposition X-13ARIMA une des plus utilisées : études de méthodes non-paramétriques pour estimer TC_t

Moyennes mobiles (ou *filtres linéaires*) omniprésents dans l'extraction de la tendance-cycle et la désaisonnalisation (e.g. : X-13ARIMA) :

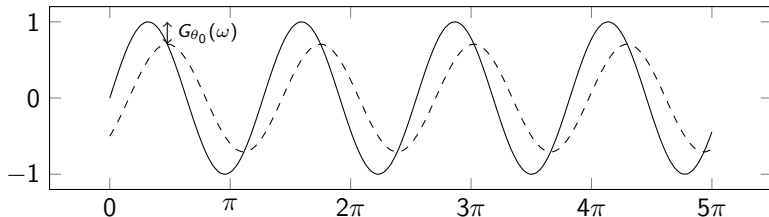
$$M_{\theta}(X_t) = \sum_{k=-p}^{+f} \theta_k X_{t+k}$$

Estimations de la TC et moyennes mobiles (1)

Appliquer M_θ sur $X_t = e^{-i\omega t}$ va avoir deux effets :

$$M_\theta X_t = \sum_{k=-p}^{+f} \theta_k e^{-i\omega(t+k)} = \left(\sum_{k=-p}^{+f} \theta_k e^{-i\omega k} \right) \cdot X_t = G_\theta(\omega) e^{-i\Phi_\theta(\omega)} X_t$$

1. Multiplier le niveau par $G_\theta(\omega)$ (*gain*)

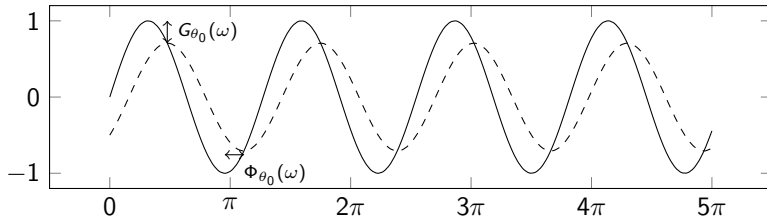


Estimations de la TC et moyennes mobiles (1)

Appliquer M_θ sur $X_t = e^{-i\omega t}$ va avoir deux effets :

$$M_\theta X_t = \sum_{k=-p}^{+f} \theta_k e^{-i\omega(t+k)} = \left(\sum_{k=-p}^{+f} \theta_k e^{-i\omega k} \right) \cdot X_t = G_\theta(\omega) e^{-i\Phi_\theta(\omega)} X_t$$

1. Multiplier le niveau par $G_\theta(\omega)$ (*gain*)
2. Créer un *déphasage* $\Phi_\theta(\omega)/\omega$: affecte détection des points de retournement



Estimations de la TC et moyennes mobiles (2)

➡ Généralement, utilisation de filtres *symétriques* ($p = f$ et $\theta_{-i} = \theta_i$)

$$M_{\theta}(X_t) = \sum_{k=-p}^{+p} \theta_k X_{t+k}, \quad \text{avec } \theta_{-i} = \theta_i$$

Estimations de la TC et moyennes mobiles (2)

➡ Généralement, utilisation de filtres *symétriques* ($p = f$ et $\theta_{-i} = \theta_i$)

$$M_{\theta}(X_t) = \sum_{k=-p}^{+p} \theta_k X_{t+k}, \quad \text{avec } \theta_{-i} = \theta_i$$

➡ Pour l'estimation en **temps réel**, utilisation de filtres *asymétriques* ($f < p$) \implies révision et détection avec retard des points de retournement (*déphasage*)

$$\text{ex : } M_{\theta}(X_t) = \sum_{k=-p}^0 \theta_k X_{t+k}$$

Estimations de la TC et moyennes mobiles (2)

➡ Généralement, utilisation de filtres *symétriques* ($p = f$ et $\theta_{-i} = \theta_i$)

$$M_{\theta}(X_t) = \sum_{k=-p}^{+p} \theta_k X_{t+k}, \quad \text{avec } \theta_{-i} = \theta_i$$

➡ Pour l'estimation en *temps réel*, utilisation de filtres *asymétriques* ($f < p$) \implies révision et détection avec retard des points de retournement (*déphasage*)

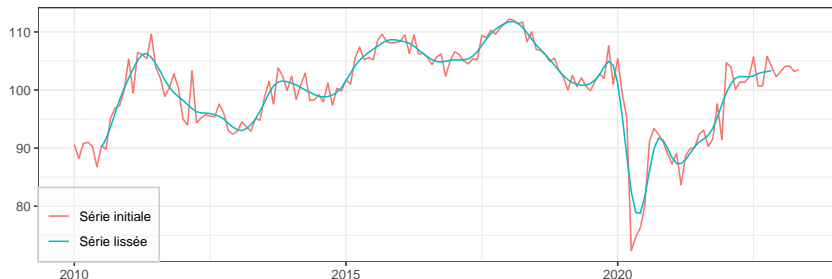
$$\text{ex : } M_{\theta}(X_t) = \sum_{k=-p}^0 \theta_k X_{t+k}$$

Solution classique : prolonger la série par prévision et utiliser filtre symétrique

➡ revient à utiliser des filtres asymétriques optimisés avec certains critères

➡ sous-optimal pour séries très variables

Illustration avec climats des affaires dans les matériels de transport



<https://www.insee.fr/fr/statistiques/6522175>

Évaluer la qualité des estimations avec prévisions implicites

Fonction `rjd3filters::implicit_forecast`

$$\forall q, \underbrace{\sum_{i=-h}^0 v_i y_i + \sum_{i=1}^h v_i y_i^*}_{\text{lissage par } v \text{ de la série prolongée}} = \underbrace{\sum_{i=-h}^0 w_i^q y_i + \sum_{i=1}^h w_i^q y_i^*}_{\text{lissage par } w^q \text{ de la série prolongée}} \quad \text{avec } \forall i > q, w_i^q = 0$$

Ce qui est équivalent à :

$$\forall q, \sum_{i=1}^h (v_i - w_i^q) y_i^* = \sum_{i=-h}^0 (w_i^q - v_i) y_i.$$

Matriciellement :

$$\begin{pmatrix} v_1 & v_2 & \cdots & v_h \\ v_1 - w_1^1 & v_2 & \cdots & v_h \\ \vdots & \vdots & \ddots & \vdots \\ v_1 - w_1^{h-1} & v_2 - w_2^{h-1} & \cdots & v_h \end{pmatrix} \begin{pmatrix} y_1^* \\ \vdots \\ y_h^* \end{pmatrix} = \begin{pmatrix} w_{-h}^0 - v_{-h} & w_{-(h-1)}^0 - v_{-(h-1)} & \cdots & w_0^0 - v_0 \\ w_{-h}^1 - v_{-h} & w_{-(h-1)}^1 - v_{-(h-1)} & \cdots & w_0^1 - v_0 \\ \vdots & \vdots & \ddots & \vdots \\ w_{-h}^{h-1} - v_{-h} & w_{-(h-1)}^{h-1} - v_{-(h-1)} & \cdots & w_0^{h-1} - v_0 \end{pmatrix} \begin{pmatrix} y_{-h} \\ \vdots \\ y_0 \end{pmatrix}.$$

Objectifs

Objectifs de cette étude :

- Étudier et comparer des approches récentes pour l'extraction de la tendance-cycle en temps réel : Régression polynomiale locale (Proietti et Luati 2008) ; RKHS (Dagum et Bianconcini 2016) ; Optimisation sous contrainte d'une somme pondérée de critères (Grun-Rehomme *et ali* 2018, Wildi et McElroy, 2019)
- On se concentre uniquement sur les moyennes mobiles asymétriques et pas sur l'estimation finale


Objectifs

Objectifs de cette étude :

- Étudier et comparer des approches récentes pour l'extraction de la tendance-cycle en temps réel : Régression polynomiale locale (Proietti et Luati 2008) ; RKHS (Dagum et Bianconcini 2016) ; Optimisation sous contrainte d'une somme pondérée de critères (Grun-Rehomme *et ali* 2018, Wildi et McElroy, 2019)
- On se concentre uniquement sur les moyennes mobiles asymétriques et pas sur l'estimation finale
- Montrer qu'il est possible d'établir une théorie générale englobant toutes ces méthodes et expliquer les liens entre les méthodes.
- Permet d'avoir une revue de la littérature de l'existant.

Objectifs

Objectifs de cette étude :

- Étudier et comparer des approches récentes pour l'extraction de la tendance-cycle en temps réel : Régression polynomiale locale (Proietti et Luati 2008) ; RKHS (Dagum et Bianconcini 2016) ; Optimisation sous contrainte d'une somme pondérée de critères (Grun-Rehomme *et ali* 2018, Wildi et McElroy, 2019)
- On se concentre uniquement sur les moyennes mobiles asymétriques et pas sur l'estimation finale
- Montrer qu'il est possible d'établir une théorie générale englobant toutes ces méthodes et expliquer les liens entre les méthodes.
- Permet d'avoir une revue de la littérature de l'existant.
- Présenter le package  `rjd3filters`
<https://github.com/rjdemetra/rjd3filters>

Sommaire

1. Introduction

2. Méthodes étudiées

2.1 Filtre symétrique

2.2 Polynômes Locaux

2.3 Filtres et Reproducing Kernel Hilbert Space (RKHS)

2.4 Minimisation sous contrainte : FST et ATS

3. Extensions

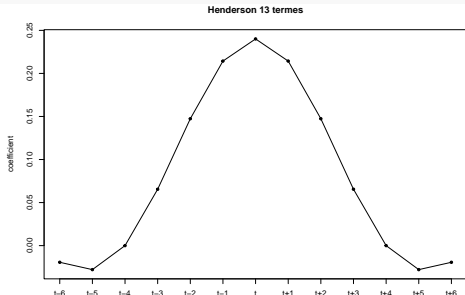
4. Comparaison des méthodes

5. Conclusion

6. Et maintenant ?

Moyenne mobile symétrique d'Henderson

```
library(rjd3filters)
f <- lp_filter(6, kernel = "Henderson")
plot_coef(f, q = 6, legend = FALSE,
          main="Henderson 13 termes")
```



MM Henderson (utilisé dans X-13ARIMA) largement répandue pour estimer TC_t

MM Henderson préserve les tendances polynomiales de degré 3 et minimise le critère de "lissage" ($\sum(\nabla^3\theta_i)^2$)

Sur séries mensuelles : MM de 13 termes généralement

Polynômes Locaux : `rjd3filters::lp_filter()`

Hypothèse : $y_t = \mu_t + \varepsilon_t$ avec $\varepsilon_t \stackrel{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

μ_t localement approchée par un polynôme de degré d :

$$\forall j \in \llbracket -h, h \rrbracket : y_{t+j} = m_{t+j} + \varepsilon_{t+j}, \quad m_{t+j} = \sum_{i=0}^d \beta_i j^i$$

Polynômes Locaux : `rjd3filters::lp_filter()`

Hypothèse : $y_t = \mu_t + \varepsilon_t$ avec $\varepsilon_t \stackrel{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

μ_t localement approchée par un polynôme de degré d :

$$\forall j \in \llbracket -h, h \rrbracket : y_{t+j} = m_{t+j} + \varepsilon_{t+j}, \quad m_{t+j} = \sum_{i=0}^d \beta_i j^i$$

Estimation en utilisant les WLS avec *noyaux* : $\hat{\beta} = (X' K X)^{-1} X' K y$ et

$$\hat{m}_t = \hat{\beta}_0 = w' y = \sum_{j=-h}^h w_j y_{t-j} \Rightarrow \text{équivalent à une moyenne mobile symétrique}$$

➡ Filtre de Henderson avec $d = 3$ et noyau spécifique.

Filtres asymétriques : `rjd3filters::lp_filter()`

1. Même méthode mais moins de données (DAF) \iff minimiser les révisions sous mêmes contraintes polynomiales
 - ➔ **sans biais** mais **beaucoup de variance**
 - ➔ utilisé dans STL

Filtres asymétriques : `rjd3filters::lp_filter()`

1. Même méthode mais moins de données (DAF) \iff minimiser les révisions sous mêmes contraintes polynomiales
 - ➔ **sans biais** mais **beaucoup de variance**
 - ➔ utilisé dans STL
 2. Minimisation des révisions sous contraintes polynomiales :
 - 2.1 *Linear-Constant* (LC) : y_t linéaire and v reproduit les constantes (*Musgrave*)
 - 2.2 *Quadratic-Linear* (QL) : y_t quadratique et v reproduit droites
 - 2.3 *Cubic-Quadratic* (CQ) : y_t cubique et v reproduit tendances quadratiques
- ➔ Filtres asymétriques v dépendent de “IC-Ratio”

Filtres asymétriques : `rjd3filters::lp_filter()`

1. Même méthode mais moins de données (DAF) \iff minimiser les révisions sous mêmes contraintes polynomiales

➡ **sans biais** mais **beaucoup de variance**

➡ utilisé dans STL

2. Minimisation des révisions sous contraintes polynomiales :

2.1 *Linear-Constant* (LC) : y_t linéaire and v reproduit les constantes (*Musgrave*)

2.2 *Quadratic-Linear* (QL) : y_t quadratique et v reproduit droites

2.3 *Cubic-Quadratic* (CQ) : y_t cubique et v reproduit tendances quadratiques

➡ Filtres asymétriques v dépendent de “IC-Ratio”



modèles simples facilement interprétables



Déphasage non contrôlé ➡ méthode étendue dans `rjd3filters::lp_filter()`

Filtres asymétriques : `rjd3filters::lp_filter()`

1. Même méthode mais moins de données (DAF) \iff minimiser les révisions sous mêmes contraintes polynomiales

➡ **sans biais** mais **beaucoup de variance**

➡ utilisé dans STL

2. Minimisation des révisions sous contraintes polynomiales :

2.1 *Linear-Constant* (LC) : y_t linéaire and v reproduit les constantes (*Musgrave*)

2.2 *Quadratic-Linear* (QL) : y_t quadratique et v reproduit droites

2.3 *Cubic-Quadratic* (CQ) : y_t cubique et v reproduit tendances quadratiques

➡ Filtres asymétriques v dépendent de “IC-Ratio”



modèles simples facilement interprétables



Déphasage non contrôlé ➡ méthode étendue dans `rjd3filters::lp_filter()`

🖥 Visualisation <https://aqlt.shinyapps.io/FiltersProperties/>

Coefficients

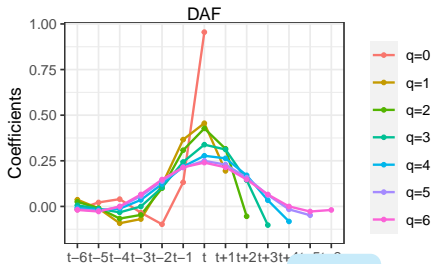
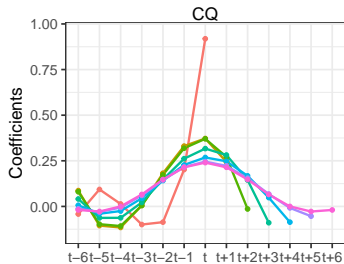
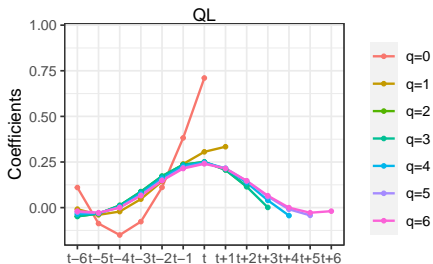
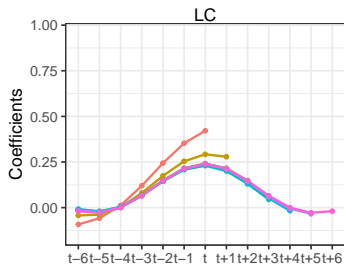
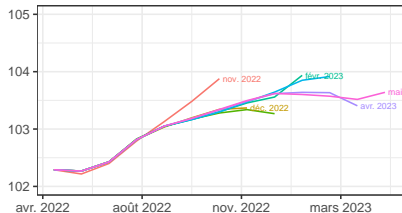
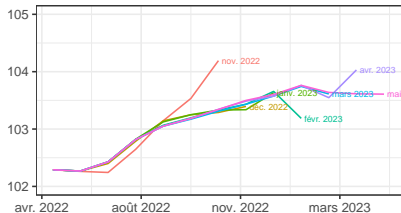


Illustration (1)

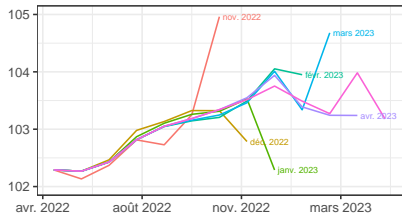
Tendance-cycle avec LC



Tendance-cycle avec QL



Tendance-cycle avec CQ



Tendance-cycle avec DAF

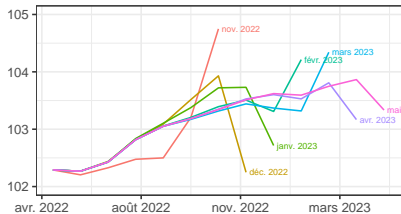
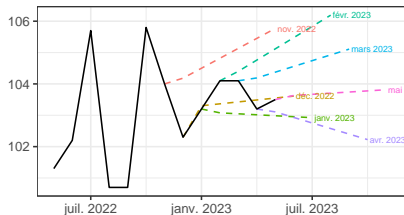
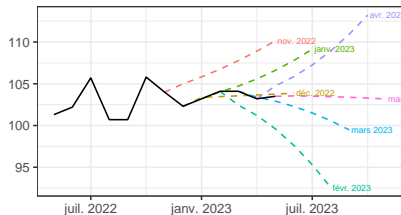


Illustration (2)

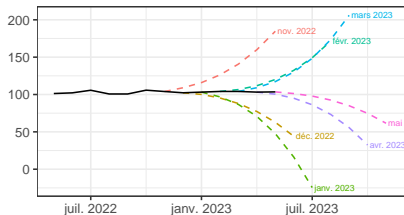
Prévisions implicites avec LC



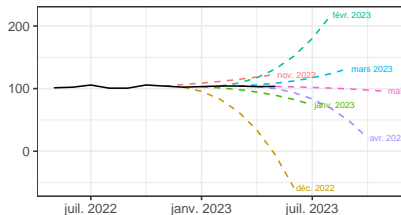
Prévisions implicites avec QL



Prévisions implicites avec CQ



Prévisions implicites avec DAF



Filtres RKHS : `rjd3filters::rkhs_filter()`

- Utilisation de la théorie des RKHS pour approcher le filtre d'Henderson
- Avec K_p une **fonction de noyau** définie sur $[-1, 1]$, le filtre symétrique :

$$\forall j \in \llbracket -h, h \rrbracket : w_j = \frac{K_p(j/b)}{\sum_{i=-h}^h K_p(i/b)}$$

Filtres RKHS : `rjd3filters::rkhs_filter()`

- Utilisation de la théorie des RKHS pour approcher le filtre d'Henderson
- Avec K_p une **fonction de noyau** définie sur $[-1, 1]$, le filtre symétrique :

$$\forall j \in \llbracket -h, h \rrbracket : w_j = \frac{K_p(j/b)}{\sum_{i=-h}^h K_p(i/b)}$$

- Pour les filtres asymétriques :

$$\forall j \in \llbracket -h, q \rrbracket : w_{a,j} = \frac{K_p(j/b)}{\sum_{i=-h}^q K_p(i/b)}$$

Filtres RKHS : `rjd3filters::rkhs_filter()`

- Utilisation de la théorie des RKHS pour approcher le filtre d'Henderson
- Avec K_p une **fonction de noyau** définie sur $[-1, 1]$, le filtre symétrique :

$$\forall j \in \llbracket -h, h \rrbracket : w_j = \frac{K_p(j/b)}{\sum_{i=-h}^h K_p(i/b)}$$

➡ avec $b = h + 1$ et K_p spécifique on retrouve le filtre d'Henderson

- Pour les filtres asymétriques :

$$\forall j \in \llbracket -h, q \rrbracket : w_{a,j} = \frac{K_p(j/b)}{\sum_{i=-h}^q K_p(i/b)}$$

Filtres RKHS : `rjd3filters::rkhs_filter()`

- Utilisation de la théorie des RKHS pour approcher le filtre d'Henderson
- Avec K_p une **fonction de noyau** définie sur $[-1, 1]$, le filtre symétrique :

$$\forall j \in \llbracket -h, h \rrbracket : w_j = \frac{K_p(j/b)}{\sum_{i=-h}^h K_p(i/b)}$$

➔ avec $b = h + 1$ et K_p spécifique on retrouve le filtre d'Henderson

- Pour les filtres asymétriques :

$$\forall j \in \llbracket -h, q \rrbracket : w_{a,j} = \frac{K_p(j/b)}{\sum_{i=-h}^q K_p(i/b)}$$

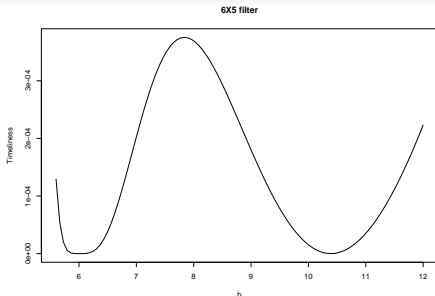
➔ b choisit par optimisation, e.g. minimisant les révisions $(b_{q,r})$, les révisions liées à la fonction de gain $(b_{q,g})$ et celles liées au déphasage $(b_{q,\varphi})$

Filtres asymétriques



Plusieurs extremum

```
library(rjd3filters)
fun <- rkhs_optimization_fun(horizon = 6,
                             leads = 5, degree = 3,
                             asymmetricCriterion = "Timeliness")
plot(fun, 5.6, 12, xlab = "b",
      ylab = "Timeliness", main = "6X5 filter")
```



```
rkhs_optimal_bw()
```

##	q=0	q=1	q=2	q=3	q=4	q=5
##	6.0000	6.0000	6.3875	8.1500	9.3500	6.0000



Méthode
généralisable à des
filtres avec fréquences
irrégulières

Coefficients

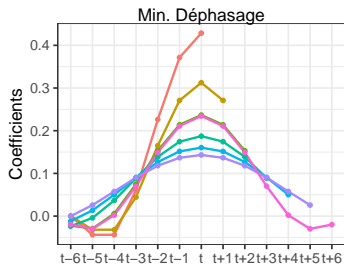
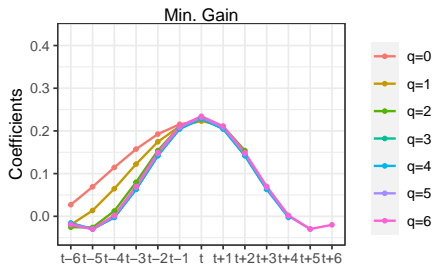
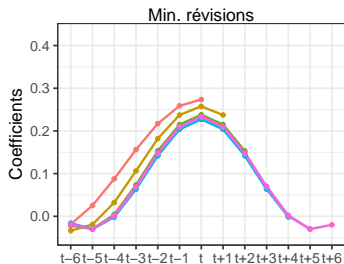
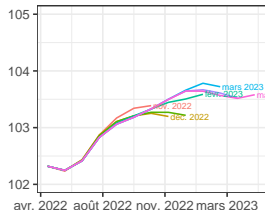
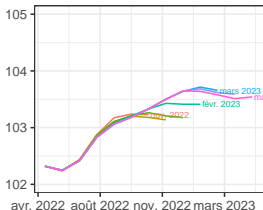
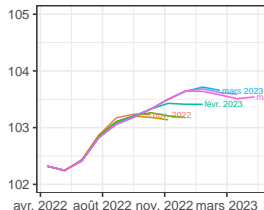
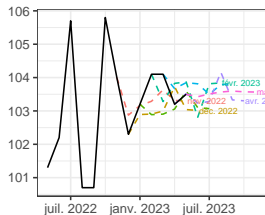
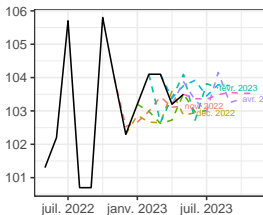
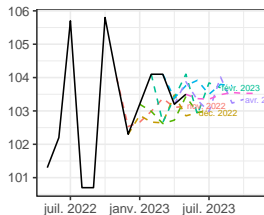


Illustration (1)

Tendance-cycle avec $b_{q,\Gamma}$ Tendance-cycle avec $b_{q,G}$ Tendance-cycle avec $b_{q,\phi}$ Prévisions implicites avec b_q Prévisions implicites avec $b_{q,G}$ Prévisions implicites avec $b_{q,\epsilon}$ 

Approche FST : `rjd3filters::fst_filter()`

Minimisation sous contrainte d'une somme pondérée de 3 critères :

$$\begin{cases} \min_{\theta} & J(\theta) = \alpha F_g(\theta) + \beta S_g(\theta) + \gamma T_g(\theta) \\ s.c. & C\theta = a \end{cases}$$

F_g fidélité (*fidelity*, réduction de variance $\sum_{k=-p}^{+f} \theta_k^2$), S_g lissage (*smoothness*, critère d'Henderson $\sum_j (\nabla^3 \theta_j)^2$), T_g temporalité (*timeliness*, déphasage $\int_0^{2\pi/12} \rho_\theta(\omega) \sin(\varphi_\theta(\omega))^2 d\omega$)

Approche FST : `rjd3filters::fst_filter()`

Minimisation sous contrainte d'une somme pondérée de 3 critères :

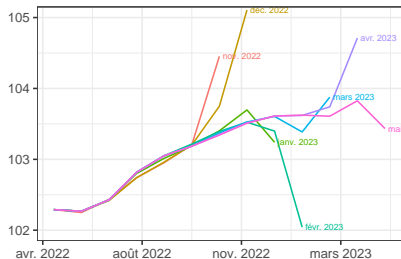
$$\begin{cases} \min_{\theta} & J(\theta) = \alpha F_g(\theta) + \beta S_g(\theta) + \gamma T_g(\theta) \\ \text{s.c.} & C\theta = a \end{cases}$$

F_g fidélité (*fidelity*, réduction de variance $\sum_{k=-p}^{+f} \theta_k^2$), S_g lissage (*smoothness*, critère d'Henderson $\sum_j (\nabla^3 \theta_j)^2$), T_g temporalité (*timeliness*, déphasage $\int_0^{2\pi/12} \rho_{\theta}(\omega) \sin(\varphi_{\theta}(\omega))^2 d\omega$)

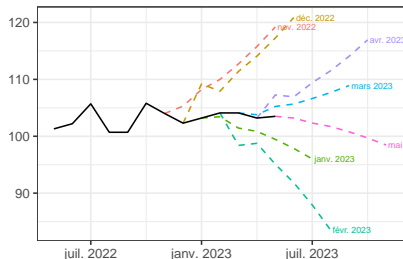
- 😊 Solution unique
- 😊 Filtres asymétriques indépendants des données et du filtre symétrique
- 😞 Poids non normalisés

Illustration (1)

Tendance–cycle avec FST



Prévisions implicites avec FST



Approche ATS `rjd3filters::dfa_filter()` (1)

Décomposition de l'EQM :

$$\begin{aligned}\mathbb{E} [(y_t - \hat{y}_t)^2] &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |\Gamma_s(\omega) - \Gamma_\theta(\omega)|^2 h(\omega) d\omega \\ &= \frac{1}{2\pi} \times 2 \times \int_0^{\pi} |\Gamma_s(\omega) - \Gamma_\theta(\omega)|^2 h(\omega) d\omega\end{aligned}$$

et

$$\begin{aligned}|\Gamma_s(\omega) - \Gamma_\theta(\omega)|^2 &= \rho_s(\omega)^2 + \rho_\theta(\omega)^2 + \\ &\quad 2\rho_s(\omega)\rho_\theta(\omega)(1 - \cos(\varphi_s(\omega) - \varphi_\theta(\omega))) \\ &= (\rho_s(\omega) - \rho_\theta(\omega))^2 + \\ &\quad 4\rho_s(\omega)\rho_\theta(\omega)\sin^2\left(\frac{\varphi_s(\omega) - \varphi_\theta(\omega)}{2}\right)\end{aligned}$$

Approche ATS `rjd3filters::dfa_filter()` (2)

Ce qui conduit à

$$A_w = 2 \int_0^{\omega_1} (\rho_s(\omega) - \rho_\theta(\omega))^2 h(\omega) d\omega$$

$$T_w = 8 \int_0^{\omega_1} \rho_s(\lambda) \rho_\theta(\lambda) \sin^2 \left(\frac{\varphi_\theta(\omega)}{2} \right) h(\omega) d\omega$$

$$S_w = 2 \int_{\omega_1}^{\pi} (\rho_s(\omega)^2 - \rho_\theta(\omega))^2 h(\omega) d\omega$$




$$R_w = 8 \int_{\omega_1}^{\pi} \rho_s(\lambda) \rho_\theta(\lambda) \sin^2 \left(\frac{\varphi_\theta(\omega)}{2} \right) h(\omega) d\omega$$

Minimisation d'une somme pondérée de 3 critères :

$$\mathcal{M}(\vartheta_1, \vartheta_2) = \vartheta_1 T_w(\theta) + \vartheta_2 S_w(\theta) + (1 - \vartheta_1 - \vartheta_2) A_w(\theta)$$

\implies minimisation sous contraintes linéaires avec $h(\omega) = 1$

Approche ATS `rjd3filters::dfa_filter()` (3)

-  Poids ont un sens
-  Résidus pas toujours négligeables
-  Pas unicité de la solution

Difficulté du choix des poids

Comment choisir les poids dans FST et AST (DFA) ?

- Minimiser que la *timeliness*? introduit trop de variance

Difficulté du choix des poids

Comment choisir les poids dans FST et AST (DFA) ?

- Minimiser que la *timeliness*? introduit trop de variance
- Minimiser les révisions? on néglige le déphasage

Difficulté du choix des poids

Comment choisir les poids dans FST et AST (DFA) ?

- Minimiser que la *timeliness*? introduit trop de variance
- Minimiser les révisions? on néglige le déphasage
- Faire quadrillage du plan et une analyse empirique du déphasage? C'est ce qui est fait avec FST en prenant les poids qui minimisent le déphasage sur les séries simulées (avec différents niveaux de variabilité). Toujours du filtre préservant les polynômes de degré 2 avec $\alpha=0,00$ (fidelity), $\beta=0,05$ (smoothness) et $\gamma=0,95$ (timeliness).

Sommaire

1. Introduction

2. Méthodes étudiées

3. Extensions

3.1 Choix de la fenêtre

3.2 Paramétrisation locale des méthodes polynomiales

4. Comparaison des méthodes

5. Conclusion

6. Et maintenant ?

Combien de termes utiliser les MM asymétriques ? (1)

Actuellement on utilise toujours autant de points dans le passé (6) que la MM symétriques pour les estimations intermédiaires : hypothèse raisonnable ? Faudrait-il utiliser plus ou moins de points dans le passé ?

Critères classiques : validation croisée, CP-Mallow, AIC, Rice-T :

$$CV(\hat{\mu}) = \frac{1}{n-2h} \sum_{t=h+1}^{n-h} \frac{(y_t - \hat{\mu}_t)^2}{(1-w_0)^2}$$

$$CP(\hat{\mu}) = \frac{1}{\sigma^2} \sum_{t=h+1}^{n-h} (y_t - \hat{\mu}_t)^2 - (n-2h)(1-2w_0)$$

Mais en général leur minimisation ne donne pas de bon résultats (critères peu discriminants)

Pistes à explorer :

Combien de termes utiliser les MM asymétriques ? (2)

1. Méthodes plus complexes de sélection de la fenêtre (e.g. Fan et Gijbels 1992)
2. Méthode des plus proches voisins : utiliser toujours le même nombre de points (e.g. toujours 13 points)

Estimation de la pente (1)

Régression non paramétrique : $y_i = \mu(x_i) + \varepsilon_i$ avec ε_i un terme d'erreur.

Avec Taylor, pour tout point x_0 , si μ est différentiable d fois, alors :

$$\forall x : \mu(x) = \mu(x_0) + \mu'(x_0)(x - x_0) + \cdots + \frac{\mu^{(d)}(x_0)}{d!}(x - x_0)^d + R_d(x),$$

Régression polynomiale

Hypothèse : $y_t = \mu_t + \varepsilon_t$ avec $\varepsilon_t \stackrel{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$

μ_t localement approchée par un polynôme de degré d :

$$\forall j \in \llbracket -h, h \rrbracket : y_{t+j} = \sum_{i=0}^d \beta_i j^i + \varepsilon_{t+j}$$

Estimation de la pente (2)

Estimation en utilisant les WLS avec *noyaux* : $\hat{\beta} = (X'KX)^{-1}X'Ky$ et

$$\hat{m}_t = \hat{\beta}_0 = e_1' \hat{\beta} = w' y = \sum_{j=-h}^h w_j y_{t-j} \text{ avec } e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Et de la même façon :

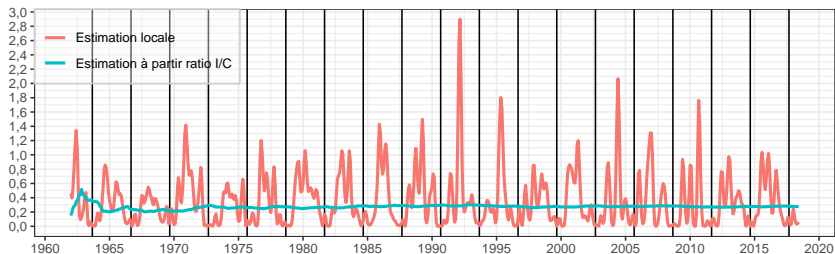
$$\begin{cases} \hat{\beta}_1 = \widehat{\mu'(t)} = e_2' \hat{\beta} & (\neq \widehat{\mu(t)})' \\ \hat{\beta}_2 = \widehat{\mu''(t)} = e_3' \hat{\beta} \end{cases}$$

Dans la méthode LC, en fin de période on suppose :

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t \text{ avec } \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

Estimation de la pente (3)

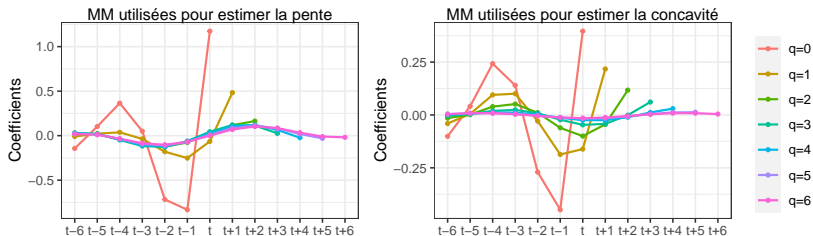
Filtres asymétriques dépendent du ratio $|\beta_1/\sigma|$ qui est toujours supposé constant : peu de sens au niveau global, notamment dans les périodes de points de retournement ($\beta_1 \simeq 0$)



Idée : paramétrisation locale

$$\begin{cases} \hat{\sigma}^2 = \frac{1}{n-2h} \sum_{t=h+1}^{n-h} \frac{(y_t - \hat{\mu}_t)^2}{1 - 2w_0^2 + \sum w_i^2} \\ \beta_1 \text{ et } \beta_2 \text{ estimés par MM (DAF par simplification)} \end{cases}$$

Estimation de la pente (4)

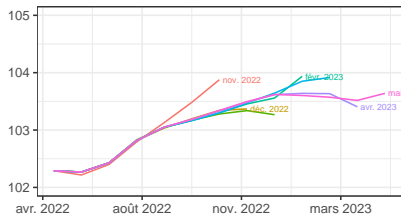


Rmq : il y a (encore) de fortes révisions entre la première et deuxième estimation, on pourrait utiliser méthode QL pour avoir les estimateurs de la pente

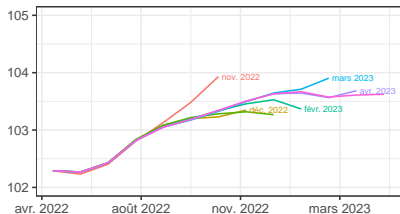
Spoil : marche plutôt bien

Illustration (1)

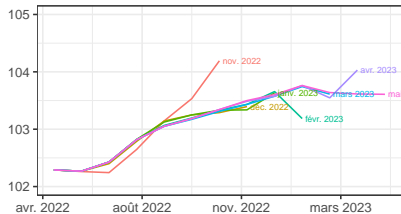
Tendance-cycle avec LC



Tendance-cycle avec LC param. locale



Tendance-cycle avec QL



Tendance-cycle avec QL param. locale

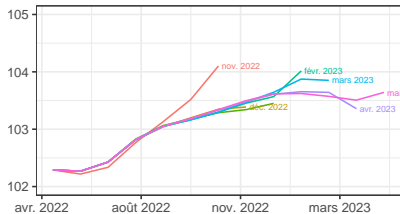
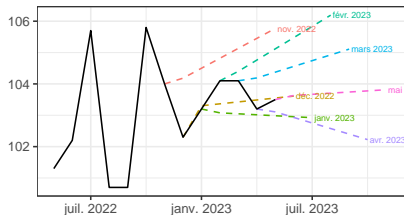
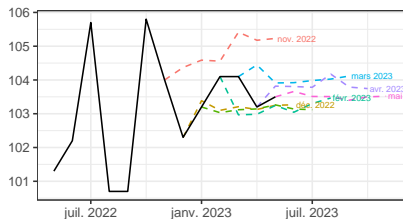


Illustration (2)

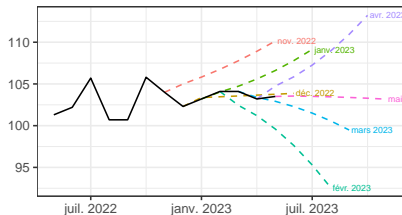
Prévisions implicites avec LC



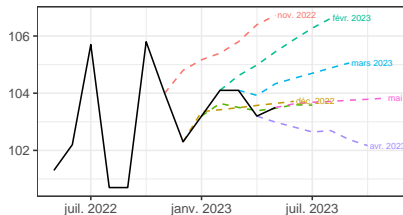
Prévisions implicites avec LC param. locale



Prévisions implicites avec QL



Prévisions implicites avec QL param. locale



Sommaire

1. Introduction

2. Méthodes étudiées

3. Extensions

4. Comparaison des méthodes

4.1 Méthodologie

4.2 Application

5. Conclusion

6. Et maintenant ?

Méthodologie

Comparaison des différentes méthodes sur séries simulées (avec 3 niveaux de variabilité) et séries réelles :

1. Estimation de la tendance-cycle à chaque date en utilisant les différentes méthodes et un filtre symétrique de 13 termes

Méthodologie

Comparaison des différentes méthodes sur séries simulées (avec 3 niveaux de variabilité) et séries réelles :

1. Estimation de la tendance-cycle à chaque date en utilisant les différentes méthodes et un filtre symétrique de 13 termes
2. À chaque date, estimation des points de retournement :
 - redressements : $y_{t-3} \geq y_{t-2} \geq y_{t-1} < y_t \leq y_{t+1}$
 - ralentissements : $y_{t-3} \leq y_{t-2} \leq y_{t-1} > y_t \geq y_{t+1}$

Déphasage = temps nécessaire pour détecter le bon point de retournement *sans révision* (\neq des papiers classiques)

Méthodologie

Comparaison des différentes méthodes sur séries simulées (avec 3 niveaux de variabilité) et séries réelles :

1. Estimation de la tendance-cycle à chaque date en utilisant les différentes méthodes et un filtre symétrique de 13 termes
2. À chaque date, estimation des points de retournement :
 - redressements : $y_{t-3} \geq y_{t-2} \geq y_{t-1} < y_t \leq y_{t+1}$
 - ralentissements : $y_{t-3} \leq y_{t-2} \leq y_{t-1} > y_t \geq y_{t+1}$

Déphasage = temps nécessaire pour détecter le bon point de retournement *sans révision* (\neq des papiers classiques)

3. Calcul des révisions avec deux critères :

$$\mathbb{E} \left[\left| \frac{y_{t|t+q} - y_{t|last}}{y_{t|last}} \right| \right] \quad \text{et} \quad \mathbb{E} \left[\left| \frac{y_{t|t+q} - y_{t|t+q+1}}{y_{t|t+q+1}} \right| \right]$$

Séries simulées (1)

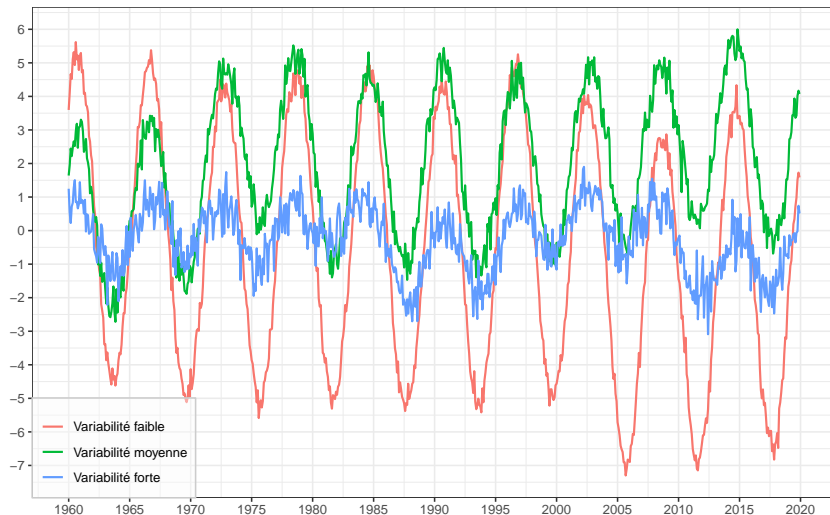
De façon similaire à Darne et Dagum (2009), on simule $y_t = C_t + T_t + I_t$ entre janvier 1960 et décembre 2020 :

- $C_t = \rho[\cos(2\pi t/\lambda) + \sin(2\pi t/\lambda)]$, $\lambda = 72$ (cycles de 6 ans, 19 points de retournement détectables)
- $T_t = T_{t-1} + \nu_t$ avec $\nu_t \sim \mathcal{N}(0, \sigma_\nu^2)$, $\sigma_\nu = 0,08$
- $I_t = e_t$ avec $e_t \sim \mathcal{N}(0, \sigma_e^2)$

Niveau de variabilité :

- variabilité faible (rapport signal/bruit fort) : $\sigma_e^2 = 0,2$ et $\rho = 3,0, 3,5$ ou $4,0$ ($0,9 \geq \text{I-C ratio} \geq 0,7$)
- variabilité moyenne (rapport signal/bruit moyen) : $\sigma_e^2 = 0,3$ et $\rho = 1,5, 2,0$ ou $3,0$ ($2,3 \geq \text{I-C ratio} \geq 1,4$)
- variabilité forte (rapport signal/bruit élevé) : $\sigma_e^2 = 0,4$ et $\rho = 0,5, 0,7$ ou $1,0$ ($8,9 \geq \text{I-C ratio} \geq 5,2$)

Séries simulées (2)



Résultats sur séries simulées

Voir <https://aqlt.github.io/DT-est-tr-tc/sec-comparison.html#comparaison>

Résultats sur des séries réelles

Voir <https://aqlt.github.io/DT-est-tr-tc/sec-comparison.html#s%C3%A9rie-r%C3%A9elle>

Nouvelle Bibliographique

Estela Bee Dagum & Silvia Bianconcini (June 2023) : Monitoring the direction of the short-term trend of economic indicators

- étudient le filtre cascade avec une approximation via les RKHS en utilisant noyau triangulaire (coefficients non retrouvé avec `rjd3filters`)
- proposent deux tests statistiques pour comparer les méthodes en termes de révisions et de point de retournement
- Comparent les méthodes en étudiant deux séries de la FRED

Sommaire

1. Introduction

2. Méthodes étudiées

3. Extensions

4. Comparaison des méthodes

5. Conclusion

5.1 Conclusion

6. Et maintenant ?

Conclusion

- Dans la construction des filtres asymétriques :
1. on peut se restreindre à ceux qui conservent les polynômes de degré au plus 1 (et exclure les filtres QL, CQ et DAF)

Conclusion

- Dans la construction des filtres asymétriques :
 1. on peut se restreindre à ceux qui conservent les polynômes de degré au plus 1 (et exclure les filtres QL, CQ et DAF)
 2. on peut utiliser le filtre LC pour les estimations proches de l'estimation finale

Conclusion

- Dans la construction des filtres asymétriques :
 1. on peut se restreindre à ceux qui conservent les polynômes de degré au plus 1 (et exclure les filtres QL, CQ et DAF)
 2. on peut utiliser le filtre LC pour les estimations proches de l'estimation finale
- Dans certains cas des méthodes alternatives à la prévision ARIMA peuvent être utilisées ➡ `rjd3filters` peut aider à comparer les résultats (`rjd3filters::x11()` pour les intégrer dans X-11)

What next?

- Etudes sur d'autres méthodes comme Vasyechko et Grun-Rehomme (2014) ou Feng et Schäfer (2021) ou l'extension des méthodes polynomiales avec T_g

What next?

- Etudes sur d'autres méthodes comme Vasyechko et Grun-Rehorme (2014) ou Feng et Schäfer (2021) ou l'extension des méthodes polynomiales avec T_g
- Impact des points atypiques ? quid des méthodes robustes ?

rjd3filters

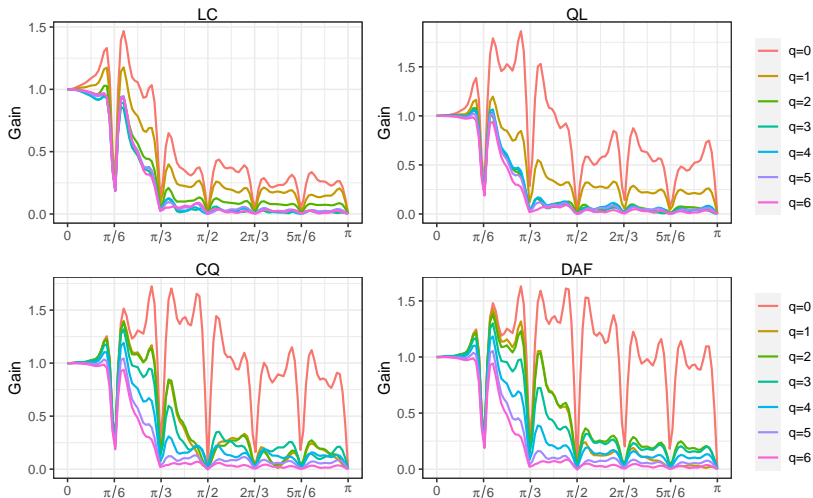
Permet de générer toutes les moyennes mobiles de X-11 (y compris asymétriques) et de les combiner pour en étudier les propriétés.

Permet de refaire toutes les étapes de X-11 (y compris correction des points atypiques), voir :

<https://github.com/rjdemetra/rjd3filters/blob/develop/vignettes/X11.Rmd>

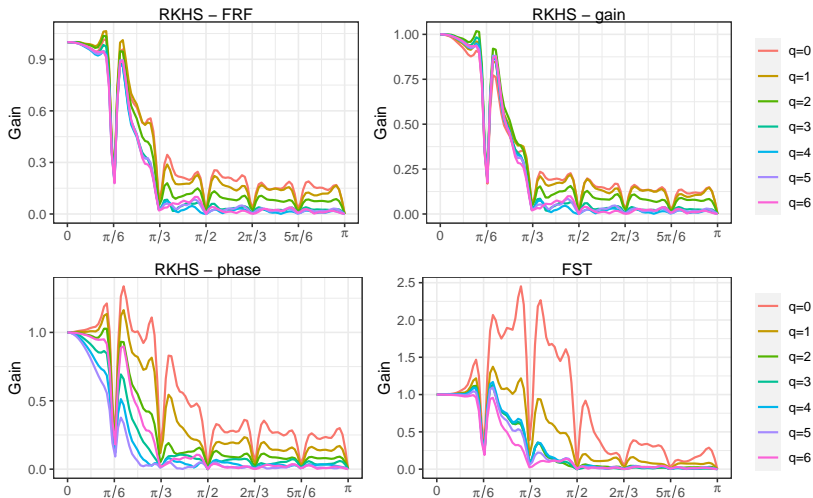
ex rjd3filters : filtres X-11

```
knitr::include_graphics("img/gain_lp.pdf")
```



ex rjd3filters : filtres X-11

```
knitr::include_graphics("img/gain_autres.pdf")
```



Sommaire

1. Introduction

2. Méthodes étudiées

3. Extensions

4. Comparaison des méthodes

5. Conclusion

6. Et maintenant ?

Suite de l'étude actuelle

Document de travail en cours de finalisation

Soumission au JOS autour de la paramétrisation locale

Objectif d'une soumission d'un article "informatique" au Journal of Statistical Software

Suite de l'étude actuelle

Document de travail en cours de finalisation

Soumission au JOS autour de la paramétrisation locale

Objectif d'une soumission d'un article "informatique" au Journal of Statistical Software

Reste une autre étude. Projet de recherche :

Le second objectif de ce projet sera d'étudier l'impact de points atypiques sur les différentes méthodes d'extraction de cycle et sur la détection des points de retournement.

Cet objectif d'étude d'impact des points atypiques amènera à également s'intéresser à l'utilisation de méthodes robustes pour l'estimation de la tendance-cycle, par exemple par l'utilisation de médianes mobiles (Tukey, 1971), mais aussi dans les autres étapes de la désaisonnalisation (pré-ajustement, estimation de la composante saisonnière, etc.).

Merci pour votre attention
