1ère Journée d'Économétrie Appliquée

Lemna
Laboratoire d'Économie et de Management Nantes-Atlantique

Insee
Institut national de la statistique et des études économiques
Mesurer pour comprendre

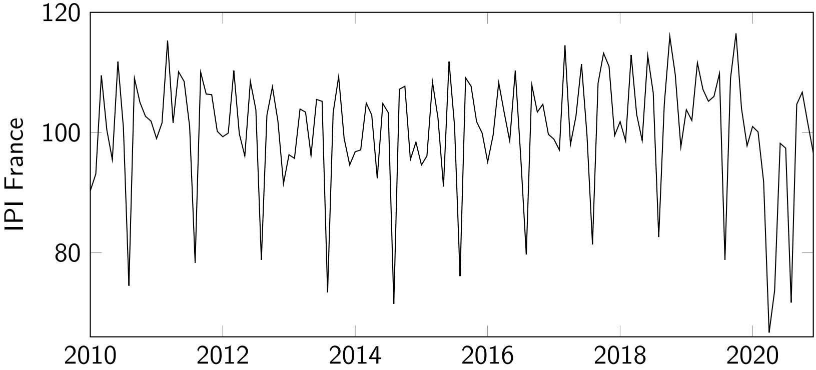
Estimation en temps réel de la tendance-cycle : Apport de l'utilisation des filtres asymétriques dans la détection des points de retournement

Alain Quartier-la-Tente
Session 27 : Séries temporelles
31/03/2022
Insee et LEMNA

Introduction (1)

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

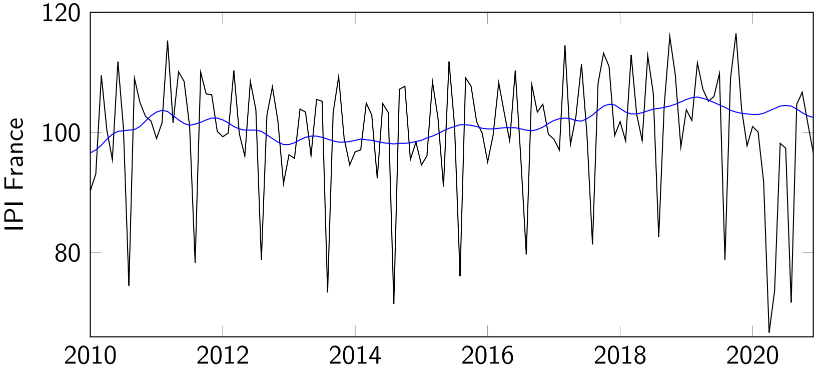
$$X_t = \qquad\qquad\qquad\qquad \text{(décomposition additive)}$$



Introduction (1)

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

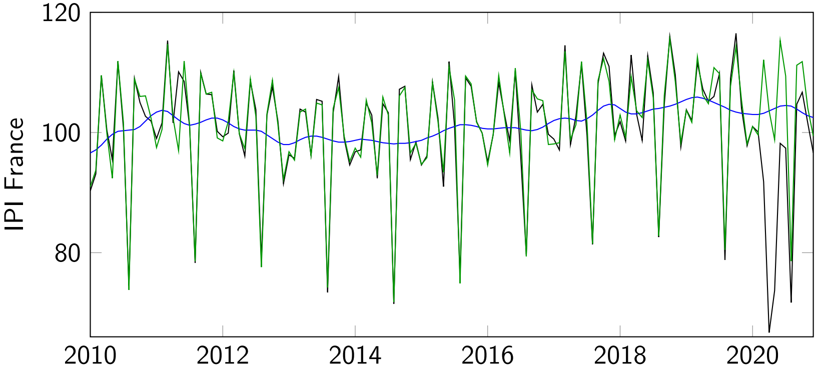
$$X_t = \underbrace{TC_t}_{\text{tendance-cycle}} \qquad \text{(décomposition additive)}$$



Introduction (1)

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

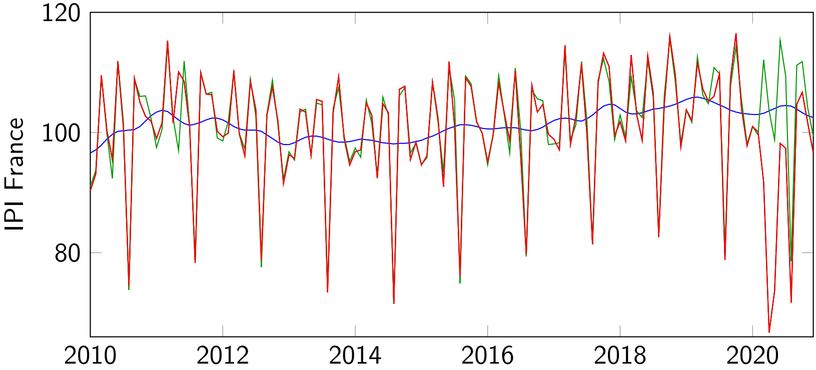
$$X_t = \underbrace{TC_t}_{\text{tendance-cycle}} + \underbrace{S_t}_{\text{saisonnalité}} \qquad \text{(décomposition additive)}$$



Introduction (1)

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

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

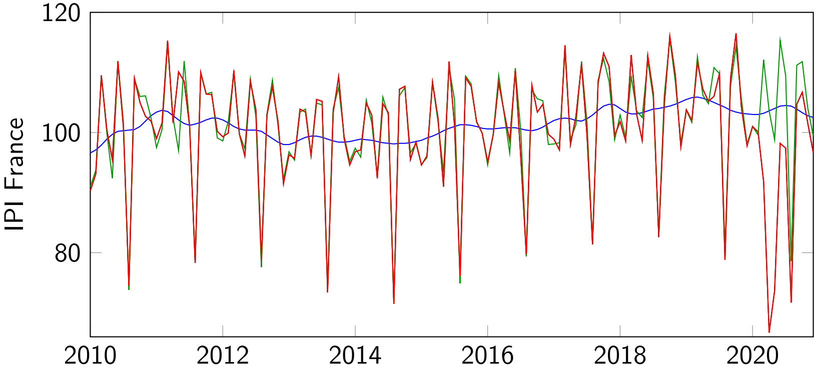


Introduction (1)

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*



Introduction (2)

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

$$X_t - S_t = TC_t + I_t$$

Introduction (2)

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

$$X_t - S_t = TC_t + I_t$$

Si bruit important, on peut utiliser un lissage supplémentaire

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

Introduction (2)

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

$$X_t - S_t = TC_t + I_t$$

Si bruit important, on peut utiliser un lissage supplémentaire

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

TC_t généralement estimée sur une série *sans* saisonnalité

Introduction (2)

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

$$X_t - S_t = TC_t + I_t$$

Si bruit important, on peut utiliser un lissage supplémentaire

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

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

Introduction (3)

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

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

Introduction (3)

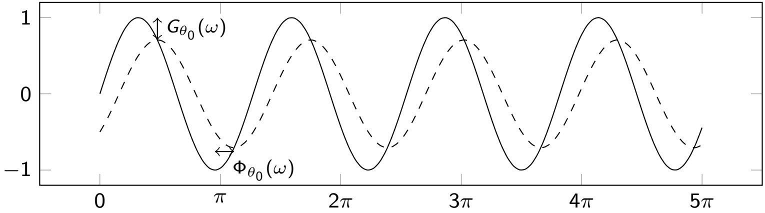
Moyennes mobiles (ou *filtres linéaires*) omniprésents dans l'extraction de la tendance-cycle et la désaisonnalisation (e.g. : X-13ARIMA) :

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

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

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

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



Introduction (4)

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

Introduction (4)

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

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

Introduction (4)

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

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

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

Introduction (4)

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

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

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

Objectifs 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)

Introduction (4)

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

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

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

Objectifs 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)
- Montrer qu'il est possible d'établir une théorie générale englobant toutes ces méthodes

Introduction (4)

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

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

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

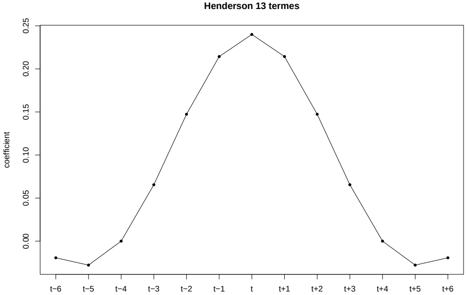
Objectifs cette étude :

- 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)
- Montrer qu'il est possible d'établir une théorie générale englobant toutes ces méthodes
- Présenter le package R `rjdfilters`

Sommaire

Moyenne mobile symétrique d'Henderson

```
library(rjdfilters)
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 : `rjdfilters::lp_filter()`

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

μ_t localement approchée par un polynôme de degré d :

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

Polynômes Locaux : `rjdfilters::lp_filter()`

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

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* : $\hat{\beta} = (X'KX)^1 X'Ky$ et

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

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

Filtres asymétriques : `rjdfilters::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 : `rjdfilters::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 : `rjdfilters::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 `rjdfilters::lp_filter()`

Filtres asymétriques : `rjdfilters::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 `rjdfilters::lp_filter()`

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

Filtres RKHS : `rjdfilters::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 :

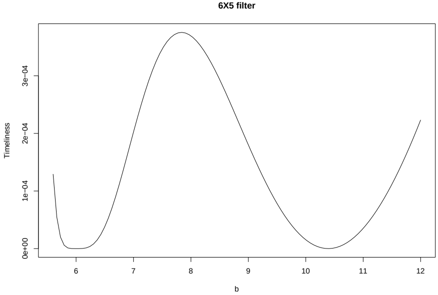
$$\forall j \in \llbracket -h, q \rrbracket : w_{a,j} = \frac{K_p(j/b)}{\sum_{i=-h}^{q} K_p(i/b)}$$

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

Filtres asymétriques

Plusieurs extremum

```
library(rjdfilters)
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

Sommaire

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*

Méthodologie

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

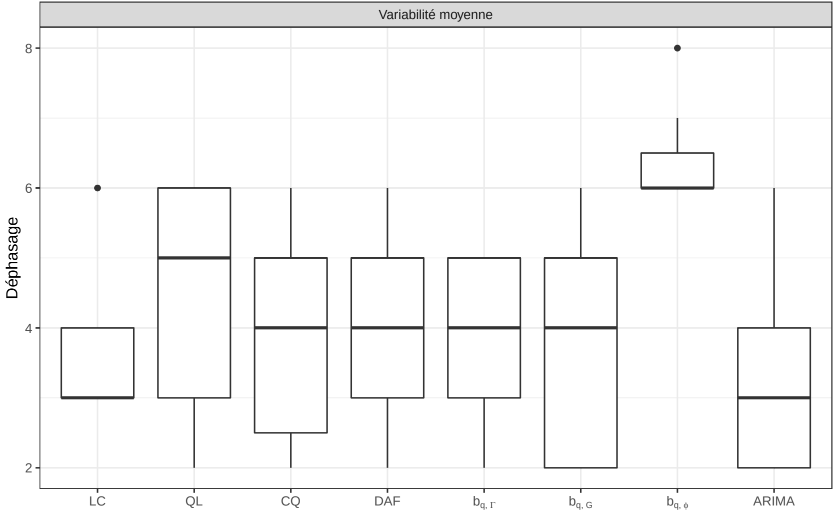
1. Estimation de la tendance-cycle à chaque date en utilisant les différentes méthodes et un filtre symétrique de 13 termes
2. À chaque date, estimation des points de retournement :
   - redressements : $y_{t-3} \geq y_{t-2} \geq y_{t-1} < y_t \leq y_{t+1}$
   - ralentissements : $y_{t-3} \leq y_{t-2}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

Déphasage = temps nécessaire pour détecter le bon point de retournement *sans révision*

3. Calcul des révisions avec deux critères :

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

Résultats sur le déphasage (séries simulées)



Médiane des révisions (séries simulées)

Pour les séries à variabilité moyenne :

Méthode	$q=0$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
MAE entre q^e et la dernière estimation						
LC	0,21	0,10	0,03	0,03	0,03	0,01
QL (rel)	1,6	1,0	1,3	1,5	1,3	1,1
CQ (rel)	2,2	1,3	4,2	3,3	2,1	1,6
DAF (rel)	2,3	1,5	4,9	3,5	2,2	1,5
$b_{q,\Gamma}$ (rel)	3,1	2,3	1,1	3,6	3,5	3,9
$b_{q,G}$ (rel)	4,1	4,0	1,1	3,6	3,5	4,0
$b_{q,\varphi}$ (rel)	1,5	1,1	1,0	1,8	2,7	8,7
ARIMA (rel)	1,0	1,0	1,1	1,1	1,1	1,0
MAE entre q^e et la $q+1^e$ estimation						
LC	0,19	0,10	0,02	0,01	0,07	0,01
QL (rel)	1,6	43,2	0,1	3,1	0,9	1,1
CQ (rel)	2,3	0,2	4,3	7,3	1,4	1,6
DAF (rel)	3,5	2,6	4,6	12,9	1,3	1,5
$b_{q,\Gamma}$ (rel)	2,1	2,9	3,7	0,3	16,2	3,9
$b_{q,G}$ (rel)	3,7	4,7	4,3	0,5	17,0	4,0
$b_{q,\varphi}$ (rel)	1,2	1,4	3,5	5,3	0,7	8,7
ARIMA (rel)	1,1	1,3	0,8	1,8	2,3	1,0

Médiane des révisions (séries simulées)

Pour les séries à variabilité moyenne :

Méthode	$q=0$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
MAE entre q^e et la dernière estimation						
LC	0,21	0,10	0,03	0,03	0,03	0,01
QL (rel)	1,6	1,0	1,3	1,5	1,3	1,1
CQ (rel)	2,2	1,3	4,2	3,3	2,1	1,6
DAF (rel)	2,3	1,5	4,9	3,5	2,2	1,5
$b_{q,\Gamma}$ (rel)	3,1	2,3	1,1	3,6	3,5	3,9
$b_{q,G}$ (rel)	4,1	4,0	1,1	3,6	3,5	4,0
$b_{q,\varphi}$ (rel)	1,5	1,1	1,0	1,8	2,7	8,7
ARIMA (rel)	1,0	1,0	1,1	1,1	1,1	1,0
MAE entre q^e et la $q+1^e$ estimation						
LC	0,19	0,10	0,02	0,01	0,07	0,01
QL (rel)	1,6	43,2	0,1	3,1	0,9	1,1
CQ (rel)	2,3	0,2	4,3	7,3	1,4	1,6
DAF (rel)	3,5	2,6	4,6	12,9	1,3	1,5
$b_{q,\Gamma}$ (rel)	2,1	2,9	3,7	0,3	16,2	3,9
$b_{q,G}$ (rel)	3,7	4,7	4,3	0,5	17,0	4,0
$b_{q,\varphi}$ (rel)	1,2	1,4	3,5	5,3	0,7	8,7
ARIMA (rel)	1,1	1,3	0,8	1,8	2,3	1,0

Médiane des révisions (séries simulées)

Pour les séries à variabilité moyenne :

Méthode	$q=0$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
MAE entre q^e et la dernière estimation						
LC	0,21	0,10	0,03	0,03	0,03	0,01
QL (rel)	1,6	1,0	1,3	1,5	1,3	1,1
CQ (rel)	2,2	1,3	4,2	3,3	2,1	1,6
DAF (rel)	2,3	1,5	4,9	3,5	2,2	1,5
$b_{q,\Gamma}$ (rel)	3,1	2,3	1,1	3,6	3,5	3,9
$b_{q,G}$ (rel)	4,1	4,0	1,1	3,6	3,5	4,0
$b_{q,\varphi}$ (rel)	1,5	1,1	1,0	1,8	2,7	8,7
ARIMA (rel)	1,0	1,0	1,1	1,1	1,1	1,0
MAE entre q^e et la $q+1^e$ estimation						
LC	0,19	0,10	0,02	0,01	0,07	0,01
QL (rel)	1,6	43,2	0,1	3,1	0,9	1,1
CQ (rel)	2,3	0,2	4,3	7,3	1,4	1,6
DAF (rel)	3,5	2,6	4,6	12,9	1,3	1,5
$b_{q,\Gamma}$ (rel)	2,1	2,9	3,7	0,3	16,2	3,9
$b_{q,G}$ (rel)	3,7	4,7	4,3	0,5	17,0	4,0
$b_{q,\varphi}$ (rel)	1,2	1,4	3,5	5,3	0,7	8,7
ARIMA (rel)	1,1	1,3	0,8	1,8	2,3	1,0

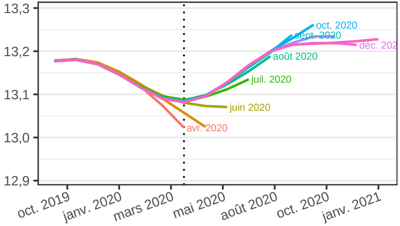
Médiane des révisions (séries simulées)

Pour les séries à variabilité moyenne :

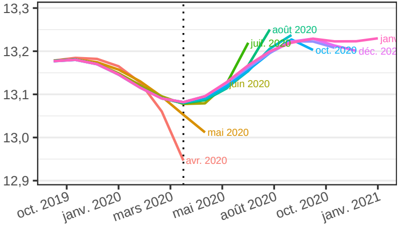
Méthode	$q=0$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
MAE entre q^e et la dernière estimation						
LC	0,21	0,10	0,03	0,03	0,03	0,01
QL (rel)	1,6	1,0	1,3	1,5	1,3	1,1
CQ (rel)	2,2	1,3	4,2	3,3	2,1	1,6
DAF (rel)	2,3	1,5	4,9	3,5	2,2	1,5
$b_{q,\Gamma}$ (rel)	3,1	2,3	1,1	3,6	3,5	3,9
$b_{q,G}$ (rel)	4,1	4,0	1,1	3,6	3,5	4,0
$b_{q,\varphi}$ (rel)	1,5	1,1	1,0	1,8	2,7	8,7
ARIMA (rel)	1,0	1,0	1,1	1,1	1,1	1,0
MAE entre q^e et la $q+1^e$ estimation						
LC	0,19	0,10	0,02	0,01	0,07	0,01
QL (rel)	1,6	43,2	0,1	3,1	0,9	1,1
CQ (rel)	2,3	0,2	4,3	7,3	1,4	1,6
DAF (rel)	3,5	2,6	4,6	12,9	1,3	1,5
$b_{q,\Gamma}$ (rel)	2,1	2,9	3,7	0,3	16,2	3,9
$b_{q,G}$ (rel)	3,7	4,7	4,3	0,5	17,0	4,0
$b_{q,\varphi}$ (rel)	1,2	1,4	3,5	5,3	0,7	8,7
ARIMA (rel)	1,1	1,3	0,8	1,8	2,3	1,0

Estimations successives de la tendance-cycle (1)

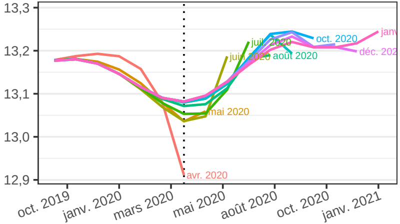
Linear – Constant (LC)
Déphasage de 3 mois



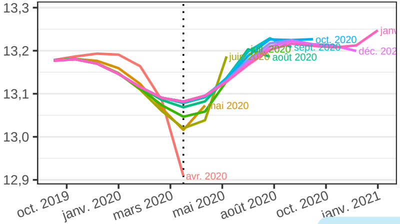
Quadratic – Linear (QL)
Déphasage de 2 mois



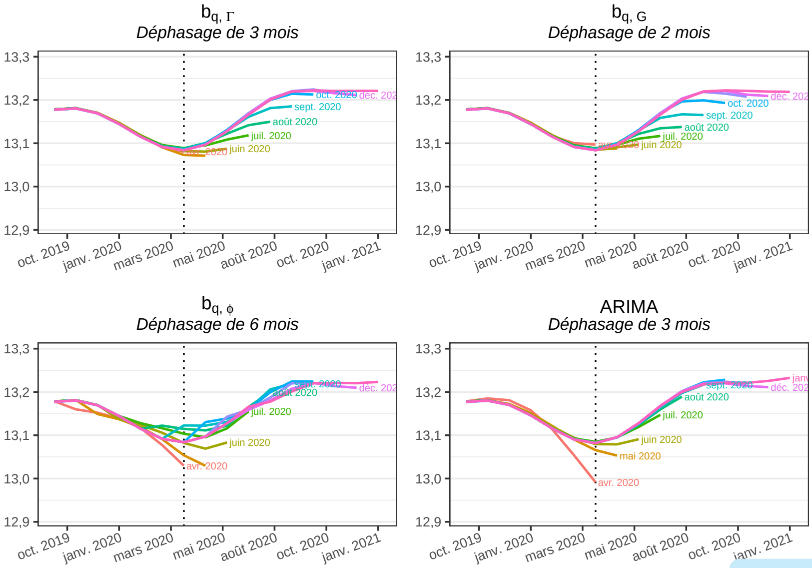
Cubic – quadratic (CQ)
Déphasage de 2 mois



DAF
Déphasage de 2 mois



Estimations successives de la tendance-cycle (2)



Prévisions implicites

Fonction `rjdfilters::implicit_forecast`

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

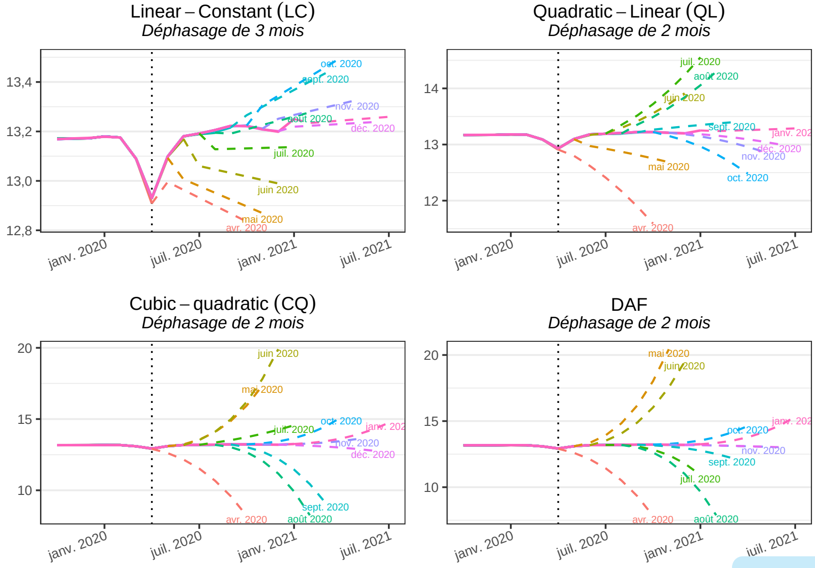
Ce qui est équivalent à :

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

Matriciellement :

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

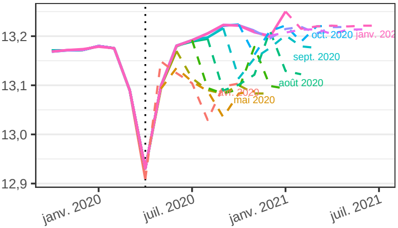
Prévisions implicites (1)



Prévisions implicites (2)

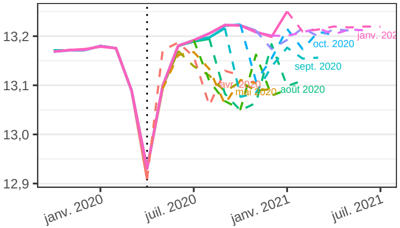
$b_{q,\Gamma}$

Déphasage de 3 mois



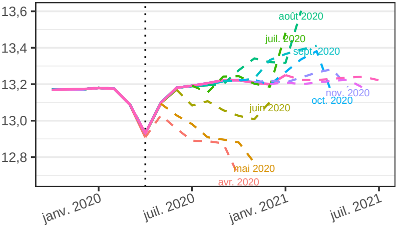
$b_{q,G}$

Déphasage de 2 mois



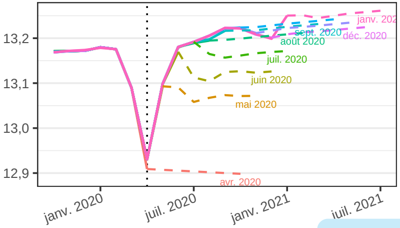
$b_{q,\phi}$

Déphasage de 6 mois



Modèle ARIMA

Déphasage 3 mois



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 ➔ `rjdfilters` peut aider à comparer les résultats (`rjdfilters::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)

What next?

- Etudes sur d'autres méthodes comme Vasyechko et Grun-Rehomme (2014) ou Feng et Schäfer (2021)

- Utiliser des paramètres différents en fin de période ? Impact de la longueur du filtre ?

What next?

- Etudes sur d'autres méthodes comme Vasyechko et Grun-Rehomme (2014) ou Feng et Schäfer (2021)
- Utiliser des paramètres différents en fin de période ? Impact de la longueur du filtre ?
- Impact des points atypiques ? quid des méthodes robustes ?

Merci pour votre attention

Package R :

palatej/rjdfilters

Version en développement AQLT/rjdfilters

Codes : https://github.com/AQLT/articles