

Les fonctions de transfert

Gilbert Colletaz

26 août 2019

Résumé

Les fonctions de transfert sont une extension naturelle des processus ARMA ou ARIMA puisqu'il s'agit simplement d'ajouter des variables explicatives à ces modélisations qui expliquent une variable uniquement par son passé ou le passé d'un processus d'innovation. On sait qu'avec ces processus une attention toute particulière est portée à l'étape d'identification. Cela va se répéter avec les fonctions de transfert où il va s'agir d'identifier les retards pertinents des variables explicatives ajoutées.

Table des matières

1	L'identification : blanchiment et corrélations croisées	1
1.1	l'étape de prewhitening	2
1.1.1	la justification	2
1.1.2	le prewhitening	2
1.1.3	illustration de l'intérêt du prewhitening	3
2	Forme générale d'une fonction de transfert	6
3	L'identification de la fonction de transfert	6
3.1	en présence de variables stationnaires non saisonnières	6
3.2	prise en compte d'une saisonnalité	8
3.2.1	saisonnalité additive	8
3.2.2	saisonnalité multiplicative	9
3.3	fonction de transfert sur séries intégrées non cointégrées	10
4	Processus résiduel et premier test de validation	11
5	Autres tests de validation	12
6	Les prévisions	13
6.1	On connaît les valeurs futures des explicatives	14
6.2	On ne connaît pas les valeurs futures des explicatives	14
6.3	Des combinaisons sont possibles	14
6.4	La syntaxe de la commande forecast	15

1 L'identification : blanchiment et corrélations croisées

Les fonctions de transfert sont des équations dynamiques dans lesquelles une variable est éventuellement expliquée par ses propres valeurs passées, mais surtout par les valeurs présentes ou/et passées d'une ou plusieurs variables explicatives avec un processus résiduel susceptible d'être lui-même gouverné par un processus ARMA(p,q). La spécification d'une fonction de transfert exige donc que l'on puisse déterminer les ordres des retards de la variable expliquée, des explicatives, et ceux du processus ARMA résiduel.

Pour cela la démarche usuelle consiste en une étape dite de blanchiment (*prewhitening*) et en l'utilisation d'un outil d'identification, la fonction de corrélation croisée, qui va rappeler l'emploi que l'on fait des corrélogrammes lors de la recherche d'une spécification d'un ARMA.

1.1 l'étape de prewhitening

1.1.1 la justification

On peut comprendre aisément en quoi va consister cette étape, et son utilité, au moyen d'un exemple simple : supposons une variable expliquée y_t et une explicative x , la relation entre les deux étant décrite par un modèle dynamique linéaire tel que :

$$y_t = c + a_0 x_t + a_1 x_{t-1} + a_2 x_{t-2} + \dots + a_m x_{t-m} + v_t \quad (1)$$

où v_t est indépendant des x . Supposons que x soit un bruit blanc centré. Dans ce cas, les coefficients a_i mesurent la réponse de y_t à x_{t-i} puisque $a_i = \delta y_t / \delta x_{t-i}$, et il vient alors :

$$\begin{aligned} Cov(y_t, x_t) &= E(y_t x_t) = a_0 E[x_t^2] = a_0 \sigma_x^2 & \Rightarrow cor(y_t, x_t) &= a_0 \frac{\sigma_x}{\sigma_y} \\ Cov(y_t, x_{t-1}) &= E(y_t x_{t-1}) = a_1 E[x_{t-1}^2] = a_1 \sigma_x^2 & \Rightarrow cor(y_t, x_{t-1}) &= a_1 \frac{\sigma_x}{\sigma_y} \\ Cov(y_t, x_{t-2}) &= E(y_t x_{t-2}) = a_2 E[x_{t-2}^2] = a_2 \sigma_x^2 & \Rightarrow cor(y_t, x_{t-2}) &= a_2 \frac{\sigma_x}{\sigma_y} \\ &\vdots & & \\ Cov(y_t, x_{t-k}) &= E(y_t x_{t-k}) = a_k E[x_{t-k}^2] = a_k \sigma_x^2 & \Rightarrow cor(y_t, x_{t-k}) &= a_k \frac{\sigma_x}{\sigma_y} \end{aligned}$$

On s'aperçoit que la fonction de corrélation croisée est proportionnelle à la fonction de réponse de y à x :

$$\begin{aligned} a_i > 0 &\Leftrightarrow cor(y_t, x_{t-i}) > 0 \\ a_i = 0 &\Leftrightarrow cor(y_t, x_{t-i}) = 0 \\ a_i < 0 &\Leftrightarrow cor(y_t, x_{t-i}) < 0 \end{aligned}$$

L'examen de la fonction de corrélation croisée révélerait donc les retards de l'explicative x qu'il convient d'inclure dans la fonction de transfert. Naturellement ce résultat n'est pas généralisable à tout x : il n'est valide que lorsque cette explicative est un bruit blanc. Si cette condition n'est pas remplie, on ne peut pas identifier les a_i non nuls à partir des corrélations croisées. Pour s'en persuader, on peut reprendre l'équation (1) mais en supposant maintenant que x_t est gouverné par un AR(1) : $x_t = \phi x_{t-1} + u_t$, pour lequel on sait que $cov(x_t, x_{t-i}) = \gamma_i = \phi \gamma_{i-1}$ et $\gamma_0 = \sigma_u^2 / (1 - \phi^2)$. Avec des calculs similaires aux précédents on obtient maintenant :

$$Cov(y_t, x_t) = a_0 \gamma_0 + a_1 \gamma_1 + \dots + a_m \gamma_m \neq a_0$$

et on pourrait par exemple avoir $a_0 = 0$ avec dans le même temps $cor(y_t, x_t) \neq 0$. La proportionnalité serait donc rompue. Afin de retrouver cette dernière il faut absolument que l'explicative soit transformée en un bruit blanc tout en s'assurant que la transformation laisse intacte la fonction de transfert existante entre les variables y et x initiales puisque c'est cette fonction que l'on veut identifier. C'est ce que va réaliser l'étape de prewhitening.

1.1.2 le prewhitening

Pour illustrer cette étape, on va repartir d'une écriture telle que (1) :

$$y_t = a(L)x_t + v_t \quad (2)$$

où v_t est toujours indépendant des x présent, passés et futurs, mais dans laquelle on va supposer que x est un ARMA stationnaire :

$$\phi_x(L)x_t = \theta_x(L)u_t$$

où u_t est un bruit blanc. Sous ces hypothèses,

$$u_t = \theta_x(L)^{-1} \phi_x(L)x_t.$$

En multipliant les deux membres de l'égalité (2) par le filtre $\theta_x(L)^{-1}\phi_x(L)$ on arrive à :

$$\begin{aligned} [\theta_x(L)^{-1}\phi_x(L)] y_t &= a(L) [\theta_x(L)^{-1}\phi_x(L)] x_t + [\theta_x(L)^{-1}\phi_x(L)] v_t, \text{ soit encore,} \\ \tilde{y}_t &= a(L)u_t + \tilde{v}_t \end{aligned} \quad (3)$$

Notez que si v et x sont indépendants, alors il en va de même pour toutes leurs combinaisons linéaires et donc \tilde{v}_t est indépendant de u .

- Le premier avantage de l'étape de prewhitening est donc de permettre de retrouver la fonction de transfert existant entre les variables initiales x et y puisqu'au final, (3) est l'écriture d'une fonction de transfert entre une transformée de l'expliquée initiale et le processus d'innovation afférent à l'explicative x . On remarque aussi que cette fonction de transfert est précisément celle qui existe entre les variables y et x initiales. Enfin, comme par construction dans (3), l'explicative est un bruit blanc, les coefficients a_i de (3) et donc de (2) sont proportionnels aux corrélations croisées existantes entre la transformée de y et l'innovation de x .
- le second avantage intervient lorsque l'on veut juger de la significativité d'une corrélation croisée entre deux variables x et y . Un résultat dû à Bartlett montre que, sous l'hypothèse d'indépendance de x et y la variance asymptotique de son estimateur est de la forme :

$$Var(r_k(x, y)) \approx \frac{1}{T} \left(1 + 2 \sum_{i=1}^{\infty} \rho_i(x)\rho_i(y) \right) \quad (4)$$

où $r_k(x, y)$ est la corrélation empirique de x_t et y_{t-k} , et $\rho_k(x)$ et $\rho_k(y)$ sont les autocorrélations théoriques respectives de x et y .

Cette formule se simplifie naturellement lorsqu'au moins une des deux variables est un bruit blanc et on retrouve d'ailleurs une expression déjà rencontrée lors de l'étude des corrélogrammes des processus ARMA :

$$Var(r_k(x, y)) \approx \frac{1}{T} \quad (5)$$

Précisément l'étape de blanchiment doit transformer l'explicative x en son innovation u et donc un IC à 95% autour de zéro permettant de statuer sur la nullité de $\rho_k(x, y)$ sera simplement de la forme $\pm 2/\sqrt{T}$.

L'identification d'une fonction de transfert reliant une expliquée y à une explicative stationnaire x passe donc par trois étapes :

1. une étape d'identification du processus ARMA afférent à l'explicative, i.e. détermination de $\phi_x(L)$ et de $\theta_x(L)$,
2. une étape de blanchiment par application du filtre $\theta_x(L)^{-1}\phi_x(L)$ à l'expliquée et à l'explicative. Notez que dans cette étape, seule x est blanchie : l'application de ce filtre n'a en effet aucune raison de transformer y_t en un \tilde{y}_t qui soit un bruit blanc,
3. calcul des corrélations croisées entre les deux transformées, \tilde{y}_t d'une part, et le processus d'innovation afférent à x d'autre part, les corrélations non nulles signalant les lags de l'explicative devant être intégrés dans la fonction de transfert finale.

Compte-tenu de ce qui précède, vous devez également comprendre que la validité d'une procédure d'identification réalisée au moyen d'une étape de prewhitening impose le respect de certaines conditions :

- la stationnarité des variables sur lesquelles on estime les corrélations croisées,
- l'indépendance de l'explicative et du résidu de la fonction

Notez enfin que si d'autres variables explicatives sont présentes, alors la procédure précédente peut être appliquée successivement sur chacune d'elle à la condition que les explicatives soient indépendantes. Dans ce cas en effet, lorsqu'on examine l'une d'entre elles, les autres sont implicitement intégrées dans le résidu de l'équation alors en construction et on respecte la seconde condition énoncée ci-dessus.

1.1.3 illustration de l'intérêt du prewhitening

On va illustrer la démarche qui vient d'être énoncée au moyen de séries simulées selon les équations suivantes :

$$x_t = 0.8x_{t-1} + u_t + 0.5u_{t-1}, \quad (6)$$

$$y_t = 1.2x_{t-2} + 1.0x_{t-3} + 0.8x_{t-4} + v_t \quad (7)$$

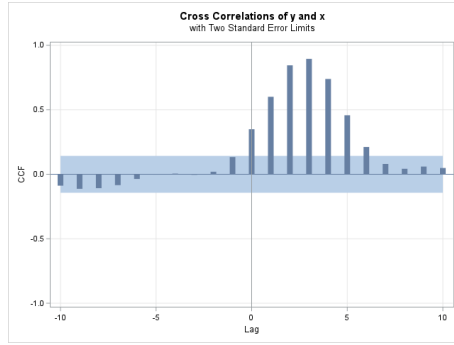


FIGURE 1 – Fonction de corrélation croisée en l’absence de prewhitening

où u et v sont des bruits blancs indépendants d’écart-types unitaires. L’expliquée y_t est donc définie sur 3 retards de x et on note la présence d’un délai de 2 périodes : ce n’est qu’à partir de ce délai que l’influence de x se fait jour. Par ailleurs l’explicative, qui obéit à un ARMA(1,1) possède une structure d’autocorrélation non nulle. En conséquence la proportionnalité de la fonction de corrélation croisée et de la fonction de transfert n’est pas assurée si on travaille avec les séries initiales.

Dans un premier temps, afin d’illustrer les remarques précédentes, nous allons utiliser la proc Arima de SAS sur les séries x et y non transformées avant de reprendre, dans un second temps, l’étude avec une étape de prewhitening.

- Au moyen de cette procédure, on peut calculer les corrélations de y_t et x_{t-k} , pour $k = -K, \dots, -1, 0, 1, \dots, K$ avec K un entier positif imposé par l’option `nlag=` . En conséquence, aux corrélations non nulles associées aux valeurs positives (resp. négatives) de k correspond une influence des x passés (resp. y passés) sur le présent de y (resp. x). Cette fonction de corrélation croisée est obtenue au moyen d’une commande `identify` de la forme :

```
identify var=expliquée crosscor(explicative);
```

Ici, il a donc suffit d’exécuter :

```
identify var=y crosscor(x) nlag=10;
```

pour obtenir le graphe 1 à partir d’une série x de 200 observations. Sur celui-ci plusieurs corrélations sont significatives en plus des 3 retards qui seuls devraient être apparents dans cette étape d’identification. On peut par exemple voir ressortir une corrélation instantanée qui pourrait laisser conclure à l’existence d’un impact immédiat de x sur y , impact qui on le sait n’existe pas.

- Dans ce second temps nous effectuons l’étape de prewhitening. En pratique, il faut donc identifier le processus ARMA afférent à x pour filtrer y et obtenir sa transformée \tilde{y} , filtrer x lui-même pour retrouver son processus d’innovation u , avant de calculer les corrélations croisées entre \tilde{y} et u . Cet ensemble de tâches est aisément réalisé dans la procédure ARIMA : si la commande `identify` réclamant les corrélations croisées qui vient d’être utilisée est précédée d’un couple `identify / estimate` alors elle filtre les variables qu’elle référence, via `var=` et `crosscor()`, au moyen du processus spécifié et estimé par ce couple de commandes. Dans la sortie la procédure rappelle que les deux séries ont été "blanchies" et redonne l’équation du filtre utilisé.

Nous savons qu’ici l’explicative x est gouvernée par un ARMA(1,1). En conséquence, il suffit d’exécuter les commandes suivantes :

```
identify var=x;
```

```
estimate p=1 q=1 method=ml;
```

```
identify var=y crosscor(x) nlag=10;
```

pour récupérer, à l’issue de l’étape de blanchiment, la fonction de corrélation croisée de la figure 2 laquelle met bien en évidence les trois seules corrélations non nulles, $cor(y_t, x_{t-2})$, $cor(y_t, x_{t-3})$ et $cor(y_t, x_{t-4})$, identifiant ainsi parfaitement, contrairement au graphe 1, la fonction de transfert de l’équation (7).

Pour terminer cet exemple nous allons estimer l’équation qui vient d’être identifiée. Cela s’effectue naturellement au moyen d’une commande `estimate` qui va suivre la dernière commande `identify` : celle-ci a précisé le nom de la

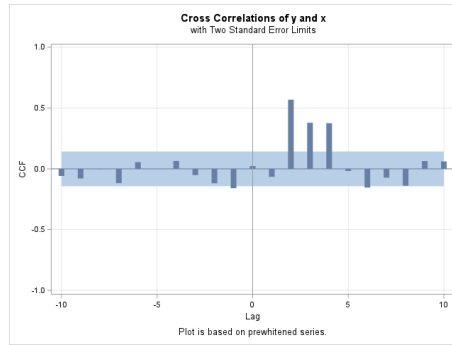


FIGURE 2 – Fonction de corrélation croisée après prewhitening

variable expliquée et celui de l'explicative, il reste à spécifier les retards de x que l'on souhaite mettre ainsi que le délai éventuel, ou *shift*, ici de 2 périodes, à partir duquel l'explicative commence à affecter l'expliquée. On dispose pour entrer ces informations de l'instruction `input=()`. Entre les parenthèses va tout d'abord apparaître la valeur du délai, suivie du caractère `$` puis, entre parenthèses, les retards de l'explicative, retards spécifiés relativement à la valeur du délai, et enfin le nom de l'explicative concernée. Dans le cas présent, nous aurons donc :

```
estimate input=(2 $(1,2)x);
```

qui correspond alors bien à : $a_2 x_{t-2}$ (délai) + $a_3 x_{t-3}$ (retard 1 à partir de 2) + $a_4 x_{t-4}$ (retard 2 à partir de 2).

***Attention** : par défaut dans la procédure Arima la valeur du délai vaut 0 et le retard de l'explicative associée à cette valeur est obligatoirement présent dans l'équation estimée. Ainsi,

```
estimate input=(0$(2,3,4)x); est équivalent à
estimate input=((2,3,4)x);
```

et ces deux commandes réclament l'estimation de $a_0 x_t + a_2 x_{t-2} + a_3 x_{t-3} + a_4 x_{t-4}$.

Dans notre exemple, le programme adéquat complet serait donc :

```
identify var=x;
estimate p=1 q=1 method=ml;
identify var=y crosscor(x) nlag=10;
estimate input=(2 $(1,2)x) method=ml;
```

Ce qui, en sortie, conduit entre autres à une dernière table rappelant l'équation ajustée et les valeurs estimées :

Input Variable	x
Shift	2
Numerator Factors	
Factor 1:	1.21263 + 0.94597 B**(1) + 0.81644 B**(2)

TABLE 1 – Résumé de la fonction de transfert estimée

Dans tout ce qui suit nous ne traiterons pas de l'estimation de ces fonctions de transfert. Celle-ci n'amène pas de difficultés particulières par rapport à l'estimation des processus ARMA que nous avons déjà étudiée. On va retrouver en particulier les trois méthodes vues alors : OLS, ULS et maximum de vraisemblance, le choix étant précisé par l'option `method=OLS|ULS|ML` qui apparaît dans la commande `estimate`, avec OLS comme choix par défaut. Nous nous intéresserons tout particulièrement à la spécification de la fonction dans la procédure Arima.

2 Forme générale d'une fonction de transfert

Dans un modèle arimax, la variable expliquée va dépendre éventuellement de ses valeurs passées et des valeurs présentes et/ou passées d'une ou plusieurs explicatives. Dans ce dernier cas, rappelez-vous cependant la condition de validité de l'identification après prewhitening énoncée à la fin de la section 1.1.2. : les explicatives doivent être indépendantes entre elles. L'écriture générale de ce modèle avec k explicatives, x_1, x_2, \dots, x_k , chacune ayant son propre délai d'action, d_1, d_2, \dots, d_k est :

$$y_t - \mu = \frac{a_1(L)}{b_1(L)}x_{1,t-d_1} + \frac{a_2(L)}{b_2(L)}x_{2,t-d_2} + \dots + \frac{a_k(L)}{b(L)}x_{k,t-d_k} + \frac{\theta(L)}{\Phi(L)}\epsilon_t \quad (8)$$

où les $a_i(L)$ et $b_i(L), i = 1, \dots, k$ sont des polynômes en L , et où le résidu est lui-même un ARMA(p,q). Par exemple, avec une seule explicative on aura :

$$y_t = \mu + \frac{(a_0 + a_1L + \dots + a_dL^d)}{(1 - b_1L - \dots - b_fL^f)}x_{t-d} + v_t \text{ avec,} \quad (9)$$

$$\phi(L)v_t = \theta(L)\epsilon_t$$

Afin de montrer la généralité de cette écriture, on peut en rappeler quelques cas particuliers bien connus :

— si dans (8) on pose que les $a_i(L)$ et $b_i(L)$ sont tous de degré 0 en L et si $p = q = 0$, on obtient le modèle de régression multiple basique :

$$y = \mu + a_1x_1 + a_2x_2 + \dots + a_kx_k + \epsilon_t$$

— Si seulement les $a_i(L)$ et $b_i(L)$ sont tous de degré 0 en L , on obtient le modèle de régression multiple à erreur ARMA :

$$y = \mu + a_1x_1 + a_2x_2 + \dots + a_kx_k + v_t$$

— le modèle à retards échelonnés fini apparaît en posant $f=0$ dans (9) :

$$y_t = \mu + a_0x_t + a_1x_{t-1} + \dots + a_kx_{t-d} + v_t$$

— un modèle à retards échelonnés infini arrive en posant $f>0$. Le cas d'une décroissance exponentielle des coefficients est obtenu, avec $d=0$ et $f=1$:

$$y_t = \mu + (1 - b_1L)^{-1}(a_0x_t) + v_t$$

$$= \mu + a_0 \sum_{i=0}^{\infty} b_1^i x_{t-i} + v_t$$

— enfin, en l'absence d'explicative x , on peut retrouver une écriture ARMA usuelle sur y .

Naturellement la difficulté majeure sera l'identification des ordres des divers polynômes.

3 L'identification de la fonction de transfert

Pour discuter de l'identification, on retiendra dans la section qui suit une fonction ne possédant qu'une seule variable explicative x et nous supposons dans un premier temps que nos variables sont stationnaires et sans composante saisonnière. Par la suite nous traiterons de la prise en compte de la saisonnalité et enfin des variables intégrées.

3.1 en présence de variables stationnaires non saisonnières

La fonction à identifier possède donc l'écriture générale :

$$y_t = \mu + \frac{(a_0 + a_1L + \dots + a_dL^d)}{(1 - b_1L - \dots - b_dL^d)}x_{t-d} + v_t \text{ avec,} \quad (10)$$

$$\phi(L)v_t = \theta(L)\epsilon_t \quad (11)$$

Deux cas types peuvent être présentés :

1. en présence d'un nombre fini et plutôt faible de retards significatifs de l'explicative.

Cette configuration correspond à l'exemple discuté précédemment pour lequel on avait seulement 3 retards pertinents avec $y_t = f(x_{t-2}, x_{t-3}, x_{t-4})$. On sait qu'il n'y a alors pas de difficulté majeure puisque l'examen des corrélations croisées sur les séries préalablement "blanchies" doit permettre de trouver les retards pertinents : ce sont ceux pour lesquels les corrélations en question sont significatives comme l'a illustré la figure 2. Dans ce cas, l'équation (10) prendra la forme suivante :

$$y_t = \mu + (a_0 + a_1L + \dots + a_dL^d)x_{t-d} + v_t \quad (12)$$

où $a(L)$ est un polynôme éventuellement troué.

2. en présence d'un nombre élevé de corrélations significatives et/ou lorsque la fonction de corrélation se caractérise par une régularité de l'évolution de ces corrélations.

Si beaucoup de corrélations sont significatives, on peut par souci de parcimonie faire précéder x d'un polynôme rationnel plutôt que d'un simple polynôme $a(L)$ de degré élevé. On sait que la division de deux polynômes de degrés mêmes faibles a généralement comme résultat un polynôme de degré infini caractérisé par des coefficients à décroissance plus ou moins rapide de type décroissance exponentielle ou avec une alternance en signe. Le jeu consiste donc à choisir les deux polynômes simples de sorte à ce que le polynôme rationnel possède des coefficients dont l'évolution soit proche de celle observée sur les corrélations croisées.

Un exemple de ce cas est donné dans la figure 3 : on peut y voir que x est un prédicteur avancé de y , avec un délai de deux périodes, et que sept ou huit coefficients sont significatifs. L'approche précédente conduirait alors à des commandes d'estimation de la forme :

```
identify var=y crosscor(x) nlag=20;
estimate input=(2 $(1,2,3,4,5,6)x);
```

Toutefois la caractéristique remarquable est la décroissance régulière des corrélations et donc des coefficients de la fonction de transfert. La façon la plus simple de reproduire celle-ci est de faire intervenir l'inverse d'un polynôme en L de degré 1 puisque :

$$\frac{a}{1-bL} = a \sum_{i=0}^{\infty} b^i L^i$$

ce qui mène à la fonction de transfert suivante :

$$y_t - \mu = \frac{a}{1-bL} x_t + v_t = ax_t + abx_{t-1} + ab^2x_{t-2} + \dots + ab^i x_{t-i} + \dots + v_t$$

La syntaxe dans la proc Arima est simple : on sépare par une barre de division / les listes des degrés en L des polynômes dividende et diviseur. Si on continue l'exemple en intégrant le délai de deux périodes de façon usuelle, la figure 3 peut conduire à la commande `estimate` suivante :

```
estimate input=(2 $ / (1) x);
```

ou à son équivalente :

```
estimate input=(2 $ (0) / (1) x);
```

Les deux décrivent la même fonction de transfert :

$$y_t - \mu = \frac{a}{1-bL} x_{t-2} + v_t = ax_{t-2} + abx_{t-3} + ab^2x_{t-4} + \dots + ab^i x_{t-2-i} + \dots + v_t$$

Pour montrer l'étendue des possibilités de cette commande `estimate`, on peut considérer deux variantes de la fonction de transfert basique qui vient d'être décrite.

1. On va par exemple supposer que la décroissance exponentielle n'est vérifiée qu'après deux retards, i.e. qu'elle décrit les coefficients de $x_{t-4}, x_{t-5}, x_{t-6}, \dots$ mais que ceux de x_{t-2} et x_{t-3} sont spécifiques. Il suffirait alors de mettre un polynôme de degré 1 au numérateur pour avoir la fonction :

$$\begin{aligned} y_t - \mu &= \frac{a_0 + a_1L}{1-bL} x_{t-2} + v_t \\ &= a_0x_{t-2} + (a_0b + a_1)x_{t-3} + (a_0b^2 + a_1b)x_{t-4} + \dots + (a_0b^i + a_1b^{i-1})x_{t-i-2} + v_t \\ &= a_0x_{t-2} + (a_0b + a_1) \sum_{i=0}^{\infty} b^i x_{t-3-i} + v_t \end{aligned}$$

dont on réclame l'estimation avec la commande :

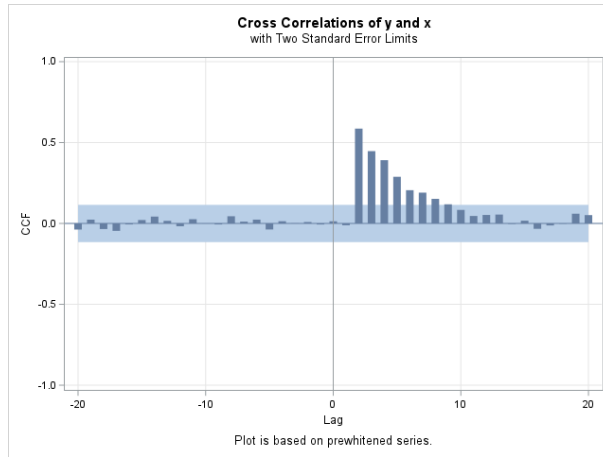


FIGURE 3 – Fonction de corrélation croisée après prewhitening

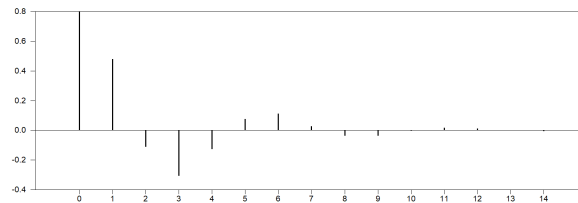


FIGURE 4 – Polynôme rationnel avec diviseur quadratique à racines complexes

```
estimate input=(2 $ (1) / (1) x);
```

- On sait que les polynômes d'ordre 2 peuvent générer des régularités de type décroissance exponentielle ou sinusoidale selon que leurs racines sont réelles ou complexes. Si de telles régularités avaient été visibles sur les corrélations croisées, on aurait pu mobiliser un polynôme quadratique avec la commande :

```
estimate input=(2 $ (1) / (1,2) x);
```

qui aurait défini la fonction de transfert :

$$y_t - \mu = \frac{a_0 + a_1 L}{1 - b_1 L - b_2 L^2} x_{t-2} + v_t$$

La figure 4 donne un exemple de coefficients d'une fonction de réponse associée à une telle équation dans laquelle le polynôme du second degré possède des racines complexes.

3.2 prise en compte d'une saisonnalité

Celle-ci peut être de type additif ou multiplicatif. Dans le premier cas il s'agit simplement d'introduire des variables indicatrices dans la fonction de transfert. Dans le second, on va faire apparaître des produits de polynômes saisonniers et non saisonniers ainsi que nous l'avons vu lors de l'étude des processus SARMA.

3.2.1 saisonnalité additive

Avec un span de valeur s , on sait que l'on peut introduire s indicatrices lorsque l'option `noint`, ou `noconstant`, n'est pas présente dans la commande `estimate`, et $s - 1$ si elle l'est. Il semble que le passage par une étape `data` soit obligatoire

pour créer au moins une de ces s ou $s - 1$ indicatrices. Supposons ainsi, qu'avec des données trimestrielles 4 variables muettes, $trim_1, \dots, trim_4$ soient présentes dans la table de données telles que pour $i = 1, 2, 3$:

$$trim_{i,t} = \begin{cases} 1 & \text{si en } t \text{ correspond une observation du } i^{\text{ème}} \text{ trimestre,} \\ 0 & \text{sinon} \end{cases}$$

Supposons qu'on introduise une saisonnalité additive dans la dernière fonction de transfert considérée, soit donc la nouvelle fonction, dans laquelle la base d'interprétation des coefficients f_i est le quatrième trimestre :

$$y_t = \frac{a_0 + a_1 L}{1 - \phi_1 L - \phi_2 L^2} x_{t-2} + f_1 trim_{1,t} + f_2 trim_{2,t} + f_3 trim_{3,t} + \mu + v_t$$

On pourra la définir et l'estimer au moyen des commandes suivantes :

```
identify var=y crosscorr=(x trim1 trim2 trim3)
estimate input=(2 $ (1) /(1,2) x , trim1 , trim2 , trim3)
```

3.2.2 saisonnalité multiplicative

On peut combiner un impact saisonnier et un impact non saisonnier de x sur y simplement en multipliant deux polynômes adaptés, *i.e.* l'un étant un polynôme en L et l'autre en L^s . Par exemple, avec des données mensuelles on pourrait avoir

$$y_t = a_0(1 + a_1 L)(1 + a_2 L^{12})x_t + v_t, \text{ soit encore} \quad (13)$$

$$y_t = (a_0 + a_3 L)(1 + a_2 L^{12})x_t + v_t, \text{ ou} \quad (14)$$

$$y_t = (b_0 + b_1 L + b_2 L^{12} + b_3 L^{13})x_t + v_t \quad (15)$$

Ces trois équations sont équivalentes. L'équation (15) rappelle simplement que le produit de deux polynômes, l'un saisonnier et l'autre non tel qu'en (13), définit souvent un polynôme troué. Définir la fonction de transfert comme produit de type (13) plutôt que comme polynôme troué tel qu'en (15) est plus parcimonieux puisque l'on a trois coefficients à estimer (a_0, a_1 et a_2) dans la première écriture au lieu de quatre (b_0, b_1, b_2 et b_3) dans la seconde. Dans la procédure Arima, l'équation (13) correspond à la syntaxe :

```
estimate input=((1)(12)x) / noint;
alors que (15) est associée à :
estimate input=((1,12,13)x) / noint;
```

L'équation (14) est également équivalente à (13) : dans cette dernière équation, on a simplement mis en facteur la constante a_0 de sorte que le premier polynôme soit de la forme $(1 + \dots)$, *i.e.* dans (14) on a $a_3 = a_0 a_1$. La seule différence entre ces deux équations est de forcer ou de ne pas forcer le premier polynôme à avoir une écriture commençant par une valeur égale à 1 : il s'agit simplement d'une question de présentation qui est sans effet sur la qualité de l'ajustement. Ce forçage est obtenu en faisant apparaître l'option `altparm` dans la commande `estimate`. Ainsi, avec

```
estimate input=((1)(12)x) altparm;
```

les résultats seront présentés conformément à l'écriture (13) avec la valeur de l'estimation de a_0 référencée sous l'intitulé "*Overall Regression Factor*". Sans cette option, ils seront calés sur l'écriture (14).

En pratique, il faut se rappeler que le choix des polynômes doit être effectué de sorte à pouvoir reproduire l'évolution des coefficients repérée sur les corrélations croisées. On doit également comprendre que les polynômes troués, ou les produits de polynômes saisonniers et non saisonniers peuvent être envisagés lorsque le nombre de ces coefficients significativement non nuls est relativement réduit et que l'on observe des pics aux environs des multiples du span. Lorsqu'un nombre élevé de coefficients paraissent significatifs et/ou que des évolutions assez complexes se font jour, alors il peut être utile de faire intervenir des polynômes rationnels, selon une logique déjà discutée dans le cas de fonctions de réponse infinies.

Pour une illustration d'emploi de polynômes rationnels avec saisonnalité, supposons que l'on observe une décroissance de type exponentielle des corrélations croisées avec toutefois une remontée de leurs valeurs à des multiples du span : un mouvement de x à la date t a un impact sur le niveau de y qui s'atténue régulièrement au fur et à mesure que l'on s'éloigne de cette date t , avec toutefois pour des données mensuelles par exemple, une remontée de son influence aux

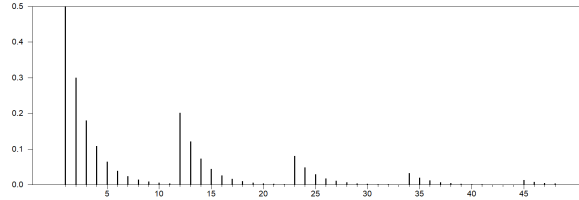


FIGURE 5 – Fonction de transfert avec décroissances saisonnière et non saisonnière des coefficients

dates $t + 12, t + 24, \dots$, remontée qui elle-même finit par s'estomper. La figure 5 correspond à ces combinaisons d'impacts saisonniers et non saisonniers du passé de x sur le présent de y . D'après des raisonnements déjà tenus, on sait que la décroissance non saisonnière peut être saisie par la prise en compte de l'inverse d'un polynôme de degré 1 en L . En poursuivant cette logique, vous devez comprendre que la décroissance aux multiples du span doit pouvoir être représentée par l'inverse d'un polynôme en L^s . Ainsi, sur données mensuelles, une fonction de transfert qui pourrait s'accorder avec cette figure 5 pourrait être :

$$y_t = \frac{a_0}{(1 - b_1 L)(1 - b_2 L^{12})} x_t + v_t \quad (16)$$

qui peut être estimée avec la commande :

```
estimate input=(/)(1)(12)x;
```

3.3 fonction de transfert sur séries intégrées non cointégrées

Dans cette section nous traiterons de la construction de fonction de transfert sur des variables non stationnaires particulières. On ne retiendra en effet que des variables à trend stochastique dont nous savons qu'elles peuvent être stationnarisées par des passages aux différences premières, secondes, etc... Pour mémoire, on rappelle que $x_t \sim I(d) \Leftrightarrow (1 - L)^d x_t = \Delta^d x_t \sim I(0)$. De plus, nous exigeons que ces variables ne soient pas cointégrées.

Cette exclusion des variables non stationnaires cointégrées se comprend aisément : d'une part, étant non stationnaires, la structure de leurs corrélations est susceptible de se modifier avec t , ce qui rend invalide tout travail d'identification de fonction stable à partir de corrélogrammes. D'autre part, si pour éviter cet écueil on les différencie pour les stationnariser alors ce faisant, on élimine dans la série résultante leur trend de long terme qui précisément en cas de cointégration est le facteur dominant de leurs interactions. En d'autres termes, pour utiliser les techniques d'identification vues ici, il est nécessaire de différencier les séries intégrées, mais ce faisant, on sur-différencie les variables cointégrées et on perd l'essentiel de leurs dépendances.

Sous cette restriction, la première étape d'un travail sur séries non stationnaires consiste donc à trouver leur ordre d'intégration qui va préciser la différenciation à appliquer. En pratique, on commencera donc le travail par un test de Dickey-Fuller sur les variables x et y , sachant que l'on ne rencontre généralement pas d'ordre supérieur à deux pour des séries non saisonnières, la majorité étant $I(1)$. Pour les séries saisonnières, plutôt que de se fonder sur des tests de racine unitaire saisonnière, on peut encore préférer, conformément aux recommandations de Box et Jenkins, examiner les corrélations croisées obtenues après application des filtres $(1 - L)$, $(1 - L^s)$ et $(1 - L)(1 - L^s)$. On retiendra alors la transformation permettant l'identification la plus aisée et la plus parcimonieuse possible. Comme lors de l'identification des processus arima(p,d,q), plusieurs modèles concurrents peuvent être sélectionnés à cette étape pour être ensuite soumis à des tests de validation. Enfin, également selon la syntaxe vue alors, c'est avec la commande `identify` que l'on va préciser les différenciations à appliquer aux variables avant qu'elles ne soient transmises à l'étape d'estimation via la commande `estimate`. La table 2 donne quelque exemples de syntaxe de ces commandes et les fonctions de transfert correspondantes.

commandes	fonction correspondante	soit encore
<code>identify var=y(1) crosscorr=x(1); estimate input=((1)x) noint</code>	$(1-L)y_t = (1-\phi_1L)(1-L)x_t + v_t$	$y_t = (1-\phi_1L)x_t + \frac{1}{(1-L)}v_t$
<code>identify var=y(1) crosscorr=x; estimate input=((1)x)noint</code>	$(1-L)y_t = (1-\phi_1L)x_t + v_t$	$y_t = \frac{(1-\phi_1L)}{(1-L)}x_t + \frac{1}{(1-L)}v_t$
<code>identify var=y(1)(4) crosscorr=x(1)(4); estimate input=/(1)(4)x) noint</code>	$(1-L)(1-L^4)y_t = \frac{(1-L)(1-L^4)}{(1-\phi_1L)(1-\phi_2L^4)}x_t + v_t$	$y_t = \frac{1}{(1-\phi_1L)(1-\phi_2L^4)}x_t + \frac{1}{(1-L)(1-L^4)}v_t$
<code>identify var=y(1)(4) crosscorr=x(1)(4); estimate input=((1 4)x) noint</code>	$(1-L)(1-L^4)y_t = (1-\phi_1L-\phi_2L^4)(1-L)(1-L^4)x_t + v_t$	$y_t = (1-\phi_1L-\phi_2L^4)x_t + \frac{1}{(1-L)(1-L^4)}v_t$

TABLE 2 – Exemples de fonction de transfert avec variables en différences

4 Processus résiduel et premier test de validation

La détermination du processus gouvernant le résidu de la fonction de transfert constitue la dernière étape de la spécification complète de celle-ci. On sait en effet qu'elle s'écrit

$$\Delta^{(y)} \Delta_s^{(y)} y_t = \mu + \frac{a(L)c(L^s)}{b(L)e(L^s)} \Delta^{(x)} \Delta_s^{(x)} x_{t-d} + v_t \quad \text{avec,} \quad (17)$$

$$\phi(L)v_t = \theta(L)\epsilon_t \quad (18)$$

où Δ et Δ_s sont les opérateurs de passage aux différences saisonnières et non saisonnières permettant de stationnariser les variables y et x que l'on veut modéliser, chacune pouvant avoir des degrés d'intégration qui lui sont propres. Par exemple, si ces ordres saisonniers et non saisonniers pour y sont respectivement d et d_s , alors $\Delta^{(y)} = (1-L)^d$ et $\Delta_s^{(y)} = (1-L^s)^{d_s}$.

Les sections précédentes s'intéressaient à l'identification de l'équation (17). Nous allons maintenant nous intéresser à (18). Cette équation pose que le résidu v_t peut obéir à un processus de type ARMA(p,q) stationnaire, de sorte que :

$$v_t = \frac{\theta(L)}{\phi(L)} \epsilon_t \quad (19)$$

où ϵ_t est un processus en bruit blanc. Naturellement on ne connaît pas la trajectoire de v mais seulement, une fois l'estimation de (17) réalisée, celle des résidus empiriques \hat{v} . C'est donc sur cette série que vont être calculés les outils usuels d'aide à l'identification des ARMA, à savoir autocorrélations, autocorrélations partielles et autocorrélations inverses.

Une fois ce processus identifié, ces caractéristiques sont passées dans la commande `estimate` via $p =$ pour la spécification de $\delta(L)$ et $q =$ pour celle de $\theta(L)$. Les quelques exemples ci-dessous rappellent la syntaxe de ces deux options :

$$\begin{aligned} p = 4 & \Rightarrow \phi(L) = (1 - \phi_1L - \phi_2L^2 - \phi_3L^3 - \phi_4L^4) \\ p = (1\ 4) & \Rightarrow \phi(L) = (1 - \phi_1L - \phi_4L^4) \\ p = 2 \quad q = (4) & \Rightarrow (1 - \phi_1L - \phi_2L^2)v_t = (1 - \theta_4)\epsilon_t \end{aligned}$$

Toujours pour illustration nous présentons dans la figure 6 les résultats obtenus sur les résidus empiriques de la fonction de transfert suivante,

$$y = \frac{a_0(1-a_1L)}{1-b_1L}x_{t-2} + v_t \quad (20)$$

elle est spécifiée et estimée avec les commandes :

```
identify var=x;
estimate p=1 q=1 noconstant method=ml;
identify var=y crosscor=x nlag=20;
estimate input=(2$(1)/(1)x) noconstant method=ml altparm;
```

Pour rappel, la mise en facteur de a_0 est liée à la présence de l'option `altparm`. On observe dans cette figure 6 une décroissance régulière des autocorrélations, une seule autocorrélation partielle et inverse significatives au retard 1 : tout plaide pour la prise en compte d'un processus AR(1) sur la série résiduelle. La fonction à estimer s'écrit alors :

$$y = \frac{a_0(1-a_1L)}{1-b_1L}x_{t-2} + (1-\phi L)v_t \quad (21)$$

En conséquence, la dernière commande `estimate` est modifiée et devient :

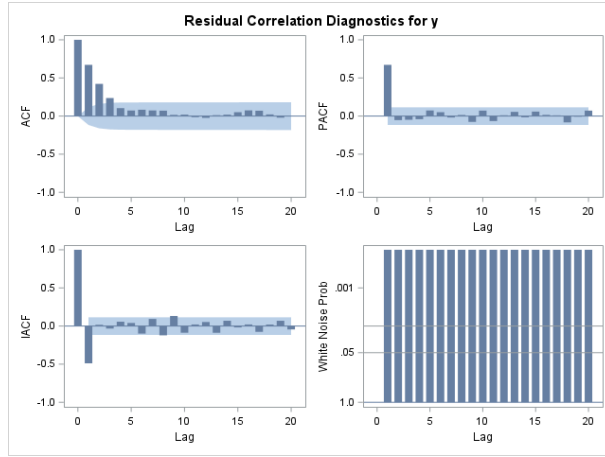


FIGURE 6 – Autocorrélations et test de Ljung-Box sur les résidus empiriques de (20)

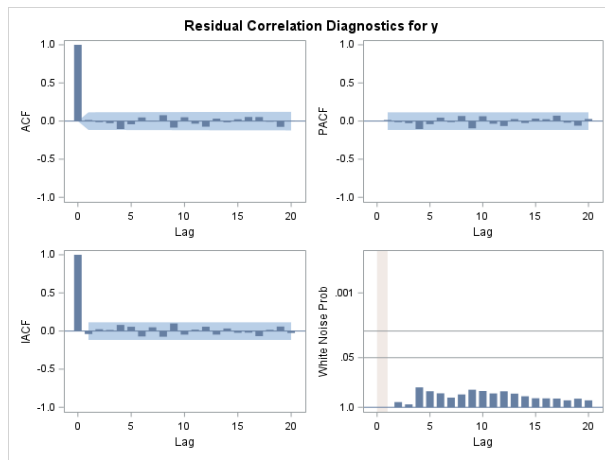


FIGURE 7 – Autocorrélations et test de Ljung-Box sur les résidus empiriques \hat{v}_t de (21)

```
estimate p=1 input=(2$(1)/(1)x) noconstant method=ml altparm;
```

On obtient alors, sur les résidus de (21), la figure 7 permettant de conclure que la prise en compte de l'AR(1) a parfaitement rempli son office.

Naturellement, le fait de retrouver un processus en bruit blanc pour l'innovation ϵ_t de (7) est favorable à la spécification estimée tout comme cela était le cas déjà pour les processus ARMA. Ainsi, en plus des graphiques qui viennent d'être présentés, on va trouver dans les sorties de la procédure un tableau, dont un exemple est donné ici dans la table 3, rappelant la valeur des autocorrélations obtenues sur $\hat{\epsilon}_t$ ainsi que les tests de Ljung-Box de nullité de ces autocorrélations.

5 Autres tests de validation

On sait que l'une des conditions nécessaire à la validité de l'étape d'identification par prewhitening est l'indépendance des résidus et des explicatives présentes dans la fonction de transfert. La procédure Arima permet de regarder ce point en présentant les corrélations croisées et des tests de Ljung-Box calculés entre l'innovation $\hat{\epsilon}_t$ de la fonction de transfert et celle afférente à l'explicative, i.e. le résidu \hat{u}_t du processus ARMA que l'on a défini et estimé sur cette explicative, et qui a été utilisé dans l'étape de blanchiment. Des corrélations significatives plaident pour l'apport de modifications dans la fonction initiale. L'intuition étant que si la partie dite inexplicquée de y_t , à savoir ϵ_t , est dépendante de x cela signifie qu'il subsiste encore dans la série x de l'information utile pour y que la fonction de transfert initiale ne capte pas.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	1.63	4	0.8034	-0.001	0.002	0.015	-0.036	0.009	0.061
12	8.15	10	0.6139	-0.070	0.055	-0.076	0.066	0.025	-0.046
18	11.54	16	0.7751	0.029	-0.025	-0.031	0.072	0.025	0.049
24	17.47	22	0.7371	-0.059	-0.074	0.082	0.017	-0.037	0.031
30	28.69	28	0.4283	-0.168	0.067	0.016	0.005	-0.020	0.021
36	41.27	34	0.1828	-0.063	0.034	0.005	-0.108	-0.053	-0.131
42	44.79	40	0.2777	-0.041	-0.002	0.044	0.067	0.041	-0.020
48	48.23	46	0.3829	0.052	0.001	0.043	-0.063	-0.020	0.027

TABLE 3 – Autocorrélations et tests de Ljung-Box sur le processus d’innovation $\hat{\epsilon}_t = (1 - L)\hat{v}_t$ de (21)

Crosscorrelation Check of Residuals with Input x									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	3.85	3	0.2777	0.021	-0.025	0.043	-0.016	-0.083	0.055
11	9.11	9	0.4275	-0.003	0.090	0.007	0.031	0.076	0.054
17	15.49	15	0.4168	-0.000	-0.034	-0.111	-0.019	-0.083	0.027
23	20.29	21	0.5029	0.051	-0.084	-0.040	-0.008	-0.033	0.061
29	30.69	27	0.2839	0.073	-0.118	0.050	-0.062	0.029	0.094
35	32.70	33	0.4820	-0.027	-0.045	0.013	0.053	0.031	-0.011
41	38.00	39	0.5151	-0.038	0.027	-0.066	0.022	0.067	-0.079
47	43.10	45	0.5525	0.042	-0.069	0.061	-0.016	-0.082	-0.006

TABLE 4 – Corrélations croisées et tests de Ljung-Box entre le processus d’innovation de la fonction de transfert, $\hat{\epsilon}_t$, et celui afférent à l’explicative, \hat{u}_t .

Un exemple de présentation de ces corrélations croisées et des tests d’orthogonalité associés est fourni par la table 4. Celle-ci est automatiquement générée par la proc `arma` dès lors qu’une commande `input` est présente dans le programme exécuté. Dans cet exemple, on n’a pas de raison de remettre en cause la spécification de la fonction ajustée. Dans le cas contraire, on peut par exemple augmenter les degrés des polynômes et donc augmenter le nombre de retard et/ou modifier la structure des polynômes rationnels présents dans la fonction de transfert initiale.

Naturellement les autres tests connus peuvent aussi être regardés. Ainsi, les coefficients doivent de préférence être significatifs. On peut également appliquer la technique dite d’overfitting déjà vue avec les processus ARMA. Enfin, pour choisir entre deux formulations il est toujours possible de recourir aux critères de sélection du type Akaike ou Schwarz dont les valeurs sont d’ailleurs indiquées dans les sorties de la proc `arma`.

6 Les prévisions

Techniquement, le calcul des prévisions une fois la fonction de transfert estimée n’amène pas de difficulté particulière par rapport à la construction de prévisions au moyen d’un processus ARMA estimé. La seule difficulté nouvelle provient de la présence de variables explicatives dans la fonction de transfert : si la période d’observation s’étend de 1 à T et si on dispose des observations de l’expliquée et des explicatives sur cette période, il n’est pas rare que le calcul des prévisions au-delà de la date T pour la variable expliquée fasse appel aux observations des explicatives également postérieures à T . Deux cas de figure sont possibles :

- on connaît ces observations et celles-ci sont présentes dans la table de données utilisée, ou
- on ne connaît pas ces observations et il faut alors prévoir les valeurs futures de x afin de les intégrer dans le calcul des valeurs futures de y .

Pour illustration, considérons la fonction de transfert particulièrement simple : $y_t = ax_t + v_t$ où v est un bruit blanc. On va supposer que les observations de y s’arrêtent en décembre 2018 et que l’on veuille prévoir y sur les 12 mois de 2019. Une fois la fonction estimée nous devons évidemment connaître les 12 valeurs prévues pour x sur 2019 puisqu’ici, si $t = 12/2018$,

$${}_t y_{t+h} = E_t[y_{t+h}] = \hat{a}E_t[x_{t+h}], \quad h = 1, \dots, 12 \quad (22)$$

6.1 On connaît les valeurs futures des explicatives

Dans ce premier cas, la table de données contient les 12 valeurs x_{t+h} , $h = 1, \dots, 12$, et évidemment on va poser que les prévisions pour x en 2019 sont simplement ces valeurs connues, *i.e.* que dans (22) : $E_t[x_{t+h}] = x_{t+h}$. Remarquez que ces valeurs ne sont pas nécessairement les vraies valeurs que prendra x : elles peuvent par exemple correspondre à une trajectoire de x dont on veut étudier l'impact sur y . Il s'agit alors de décrire l'évolution anticipée de y si x évolue de telle ou telle façon en 2019. Dans ce cas, lorsqu'on va réclamer des prévisions sur y au moyen de la commande `forecast`, la proc `arima` va prendre les valeurs de x présentes dans la table pour faire le calcul décrit en (22). Comme on sait, le coefficient \hat{a} est alors considéré comme étant la vraie valeur de la pente a : la seule incertitude provient du résidu v , et les intervalles de confiance construits sur les prévisions de y seront fondés sur σ_v^2 . En d'autres termes, dans cette configuration, les valeurs futures de x sont considérées comme certaines. Ce dernier aspect va être une des différences importantes entre les deux cas que nous considérons ici.

6.2 On ne connaît pas les valeurs futures des explicatives

Dans ce second cas, toutes les données de la table s'arrêtent en décembre 2018 : on ne dispose pas des observations de x sur l'année 2019. La prévision de y au moyen de (22) exige donc que l'on commence par prévoir la variable x , *i.e.* que l'on construise les valeurs $E_t[x_{t+h}]$, $h = 1, \dots, 12$. Pour cela, la proc `arima` va automatiquement utiliser le processus sélectionné et estimé sur x à l'étape de prewhitening : elle va construire les prévisions de x au moyen de ce filtre puis injecter les valeurs obtenues dans l'équation de construction des prévisions de y . Pour illustration, si on reprend l'exemple de la fonction de transfert décrite par l'équation (20), on aura ainsi :

```
identify var=x;
estimate p=1 q=1 noconstant method=ml;
identify var=y crosscor=x nlag=20;
estimate input=(2$(1)/(1)x) noconstant method=ml altparm;
forecast lead=12;
```

Au moyen du processus ARMA(1,1) estimé sur x , la procédure va construire les 12 valeurs prévues de cette variable, valeurs qui seront utilisées pour calculer les prévisions de y conformément à la fonction de transfert spécifiée.

Contrairement au cas de figure précédent, la procédure est alors en mesure d'apprécier l'incertitude qui affecte les prévisions des x futurs et qui est notamment fonction de σ_u^2 . Dans la proc `arima`, cette incertitude est utilisée, avec celle émanant de σ_v , pour calculer des intervalles de confiance construits autour des prévisions. En d'autres termes, dans le cas précédent l'incertitude avait une source unique : la variance du processus d'innovation afférent à la fonction de transfert, σ_v^2 . Dans le second cas elle a deux sources puisqu'à σ_v^2 elle ajoute la variance du processus d'innovation afférente à l'explicative, σ_u^2 . La conséquence immédiate est que, dans le second cas, les largeurs des intervalles de confiance sont obligatoirement plus grandes que dans le premier ¹.

6.3 Des combinaisons sont possibles

En ce qui concerne l'intégration de prévisions pour les explicatives d'une fonction de transfert, l'utilisation de la procédure ARIMA est assez souple. La règle essentielle à retenir est que si des valeurs d'explicatives sont présentes dans la table de données sur la période de prévision alors ces observations sont utilisées en priorité. De ce fait diverses combinaisons de modes de construction des prévisions sont possibles comme le montre l'exemple suivant.

On va supposer une fonction de transfert de la forme $y_t = a_1x1_t + a_2x2_t + a_3x3_t + u_t$ et que, sur la période de prévision allant de T à $T + 5$, la table de données se présente selon la table 5. On pourrait alors avoir des lignes de commandes de la forme :

```
identify var=x2;
estimate p=1 q=1 noconstant method=ml;
identify var=x3;
estimate p=1 method=ml;
```

1. Avec notre dernier exemple, imaginez que vous calculiez des prévisions pour x au moyen d'un ARMA(1,1). Si vous rentrez ces valeurs prévues dans la table de données alors les prévisions de y seront les mêmes que celles obtenues avec les lignes de commandes données dans le texte ci-dessus, mais les intervalles de confiance seront plus petits.

obs	y	$x1$	$x2$	$x3$
T+1	.	1	8	.
T+2	.	1	6	.
T+3	.	1	.	.
T+4	.	1	.	.
T+5	.	1	.	.

TABLE 5 – Exemple de combinaisons de modalités de construction de prévisions sur les explicatives

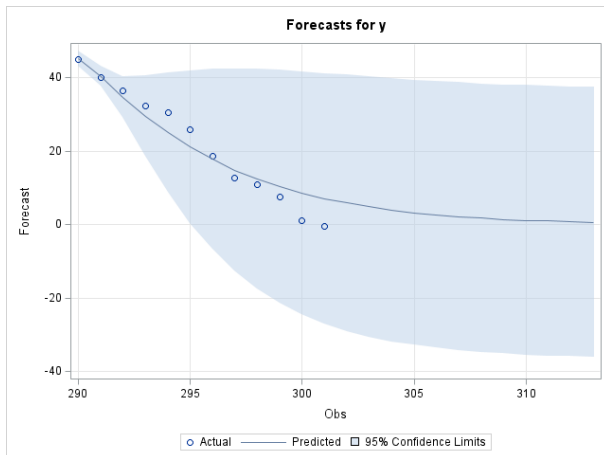


FIGURE 8 – Prévisions sur 2 ans avec comparaison des prévisions et des réalisations sur les 12 premiers mois

```

identify var=y crosscor=(x1 x2 x3);
estimate input=(x1 x2 x3) method=ml;
forecast lead=12;

```

Ici, les prévisions de y seraient obtenues au moyen :

- de prévisions pour $x1$ constantes et égales à 1,
- de prévisions pour $x2$ égales à 8 pour $T + 1$ et à 6 pour $T + 2$ puis générées par un processus estimé de type ARMA(1,1) pour les dates $T + 3$ à $T + 5$,
- de prévisions tirées d'un AR(1) sur toute la période de prévision pour $x3$.

6.4 La syntaxe de la commande forecast

La commande `forecast` dispose des options habituelles déjà vues lors de l'étude des processus ARMA avec notamment les options `ID=`, `alpha=`, ... La figure 8 est un exemple de graphe obtenu avec

```
forecast lead=24 back=12;
```

Pour mémoire, si la date de fin de la fenêtre d'observation est en décembre 2018, cette commande demande la construction en décembre 2017 de 24 prévisions couvrant les années 2018 et 2019. On rappelle également que les 12 dernières observations connues de l'année 2018 ne sont pas utilisées pour construire ces 24 prévisions. De cette façon on peut apprécier la capacité de la fonction estimée à reproduire l'évolution connue de y au cours de l'année 2018, ce qui peut contribuer à réduire ou au contraire à augmenter les doutes que l'on peut avoir vis à vis des valeurs prévues pour 2019. Dans le présent exemple, l'évolution réalisée en 2018 aurait été assez bien anticipée en décembre 2017 par la fonction de transfert estimée ce qui accroît la confiance que l'on peut avoir pour les valeurs qu'elle produit sur 2019.