Séries temporelles non stationnaires (compléments du Chapitre 5)

Yves Aragon*
Université Toulouse 1 Capitole

16 août 2016

5.1 Séries intégrées - Modèles ARIMA et SARIMA

Exercice 5.1 (Différenciation saisonnière)
Vérifier empiriquement l'effet d'une différenciation saisonnière sur

$$y_t = a + b\cos(2\pi t/4) + c\sin(2\pi t/4) + u_t \text{ avec } u_t = \frac{1}{1-0.9\mathrm{B}} z_t. \tag{5.1}$$

Pour cela on pourra définir les séries $\cos(2\pi t/4)$ et $\sin(2\pi t/4)$, $t = 1, \cdots, 48$ et calculer leurs différences saisonnières.
Pour des compléments théoriques on peut consulter Gourieroux et Monfort (1995), chap. 3, qui présentent les propriétés algébriques des filtres de moyenne mobile, appelés aussi filtres de moyenne glissante (*running mean*). Ladiray et Quenneville (2000) expliquent en détail l'usage de ces filtres en macro-économie.

Réponse.

```
> # serie de 48 points
> # x1 et x2 sont les fonctions periodiques
> # et on verifie que leurs differences saisonnieres sont nulles.
> temps = 1:48
> x1 = cos(2*pi*temps/4)
> diff(x1, 4)
```

```
 [1]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
 [5]  2.449213e-16  0.000000e+00 -2.021278e-15  0.000000e+00
 [9] -1.531436e-15  0.000000e+00 -2.449213e-16  0.000000e+00
[13]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
[17]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
[21]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
[25]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
[29]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
[33]  2.449213e-16  0.000000e+00 -2.449213e-16  0.000000e+00
[37]  7.350349e-15  0.000000e+00 -2.449213e-16  0.000000e+00
[41] -6.860506e-15  0.000000e+00  6.860506e-15  0.000000e+00
```

*yves.aragon@gmail.com

```
> x2 = sin(2*pi*temps/4)
> diff(x2, 4)
```

```
 [1]  0.000000e+00  2.449213e-16  0.000000e+00 -2.449213e-16
 [5]  0.000000e+00  2.449213e-16  0.000000e+00 -2.449213e-16
 [9]  0.000000e+00  2.449213e-16  0.000000e+00 -2.449213e-16
[13]  0.000000e+00  2.449213e-16  0.000000e+00 -2.449213e-16
[17]  0.000000e+00  3.797635e-15  0.000000e+00 -2.449213e-16
[21]  0.000000e+00 -6.860506e-15  0.000000e+00 -2.449213e-16
[25]  0.000000e+00  7.350349e-15  0.000000e+00 -2.449213e-16
[29]  0.000000e+00 -6.860506e-15  0.000000e+00 -2.449213e-16
[33]  0.000000e+00  7.350349e-15  0.000000e+00 -2.449213e-16
[37]  0.000000e+00 -6.860506e-15  0.000000e+00 -7.350349e-15
[41]  0.000000e+00  7.350349e-15  0.000000e+00  6.860506e-15
```

On voit que les différenciations saisonnières de `x1` et `x2` sont nulles.

Exercice 5.2 (Estimation d'un SARIMA avec dérive)
On a indiqué à la section 5.1, comment R estime les modèles intégrés. Ainsi, s'il y avait différenciation aux ordres 1 et 12, il faudrait introduire le régresseur t^2 (la constante et le régresseur t sont éliminés dans les différenciations) :

$$Y_t = c\,t^2 + e_t.$$

La différenciation aux ordres 1 et 12 donne

$$(1-\mathrm{B})(1-\mathrm{B}^{12})Y_t = 24\,c + (1-\mathrm{B})(1-\mathrm{B}^{12})e_t,$$

et R fournit donc une estimation de c et non de $24\,c$.
Vérifier cette assertion en simulant un SARIMA$(1,1,0)(0,1,1)_{12}$ puis en l'estimant.

Réponse. Il faut évidemment simuler un SARIMA avec dérive. Nous allons faire cette simulation de deux façons. D'abord, nous choisissons un modèle, un SARIMA$(1,1,0)(0,1,1)_{12}$ avec dérive :

$$(1-\mathrm{B})(1-\mathrm{B}^{12})y_t = \mathtt{moy} + \frac{1+0.5\,\mathrm{B}^{12}}{1+0.8\,\mathrm{B}} z_t,\; z_t \sim \mathrm{BBN}(0,1)$$

`moy` est la moyenne du processus différencié aux ordres 1 et 12 et donne une dérive sur le processus y_t. Nous écrivons les différents polynômes qui seront utilisés.

```
> require("polynom")
> nobs = 200
> set.seed(234)
> ar.1 = c(1, .8)
> masaiso = polynomial(c(1, rep(0, 11), .5))
> ar.13 =  polynomial(c(1, rep(0, 11), -1))*polynomial(ar.1)
> ar.2 = polynomial(c(1, -1))*polynomial(ar.1)
> moy = 5
```

— **Simulation par** `arima.sim`
  On commence par simuler $(1-\mathrm{B})(1-\mathrm{B}^{12})y_t$:

```
> ya = arima.sim(n = nobs, list(order = c(1, 0, 12), ar = -.8,
+  ma = c(rep(0, 11), .5))) + moy
> mean(ya)
```

```
[1] 4.965789
```

```
> require("forecast")
> Arima(ya, order = c(1, 0, 0), seasonal = list(order = c(0, 0, 1), p
+  include.mean = TRUE)
```

```
Series: ya
ARIMA(1,0,0)(0,0,1)[12] with non-zero mean

Coefficients:
          ar1    sma1  intercept
      -0.7996  0.4544     4.9632
s.e.   0.0415  0.0711     0.0528

sigma^2 estimated as 0.8949:  log likelihood=-273.1
AIC=554.21   AICc=554.41   BIC=567.4
```

Comme vérification, nous avons imprimé la moyenne de la série simulée et estimé le modèle attendu. On intègre maintenant la série en deux étapes et à chacune, on estime le modèle de la série intégrée. On procède de deux façons, ou bien en utilisant l'option `include.drift=TRUE` ou bien en régressant sur la série $1, 2, \cdots$

```
> # I(1)
> yd0 = diffinv(ya, 1)[-1]
> x1 = as.matrix(seq(1, length(yd0)))
> Arima(yd0, order = c(1, 1, 0),
+  seasonal = list(order = c(0, 0, 1), period = 12), xreg = x1)
```

```
Series: yd0
ARIMA(1,1,0)(0,0,1)[12]

Coefficients:
          ar1    sma1      x1
      -0.7999  0.4550  4.9608
s.e.   0.0416  0.0709  0.0530

sigma^2 estimated as 0.8974:  log likelihood=-272.02
AIC=552.04   AICc=552.25   BIC=565.22
```

```
> Arima(yd0, order = c(1, 1, 0),
+ seasonal = list(order = c(0, 0, 1), period = 12),
+  include.drift = TRUE)
```

```
Series: yd0
ARIMA(1,1,0)(0,0,1)[12] with drift

Coefficients:
```

```
          ar1     sma1    drift
      -0.7999   0.4550   4.9608
s.e.   0.0416   0.0709   0.0530

sigma^2 estimated as 0.8974:  log likelihood=-272.02
AIC=552.04   AICc=552.25   BIC=565.22
```

On obtient les mêmes résultats par les deux méthodes. Enfin pour simuler la série y_t, on intègre saisonnièrement yd0

```
> # I(12)
> yd2 = diffinv(yd0, 12)[-(1:12)]
> x2 = as.matrix(seq(1, length(yd0))^2)
> (m1 = Arima(yd2, order = c(1, 1, 0), seasonal = list(order = c(0, 
+           frequency = 12), xreg = x2))
```

```
Series: yd2
ARIMA(1,1,0)

Coefficients:
          ar1     sma1       x2
      -0.9024   -1.000   0.2065
s.e.   0.0293    0.013   0.0010

sigma^2 estimated as 9.652:  log likelihood=-508.01
AIC=1024.02   AICc=1024.23   BIC=1037.18
```

```
> (m1b = Arima(yd2, order = c(1, 1, 0), seasonal = list(order = c(0,
+         frequency = 12), include.drift = TRUE))
```

```
Series: yd2
ARIMA(1,1,0)

Coefficients:
          ar1      sma1
      -0.9023   -0.5186
s.e.   0.0299    0.0457

sigma^2 estimated as 16.61:  log likelihood=-559.53
AIC=1125.05   AICc=1125.18   BIC=1134.92
```

m1 estime le coefficient de x2 et la dérive 24 fois ce coefficient. Dans l'estimation utilisant l'option include.drift=TRUE on note que la dérive n'est pas significative, alors qu'elle vaut 5 dans le modèle simulé.

— **Simulation par** simulate

Il nous faut développer le polynôme d'autorégression dans

$$(1-\mathrm{B})(1-\mathrm{B}^{12})(1+0.8\,\mathrm{B})y_t = (1+0.8\,\mathrm{B})\mathtt{moy} + (1+0.5\,\mathrm{B}^{12})z_t$$

On définit donc les trois polynômes qui définissent l'autorégression

```
> (ar.1 = polynomial(c(1, .8)))
```

```
1 + 0.8*x
> (ar.2 = polynomial(c(1, -1))*ar.1)
1 - 0.2*x - 0.8*x^2
> # terme autoregressif complet jusqu'au retard 14
> (ar.14 = polynomial(c(1, rep(0, 11), -1))*ar.2)
1 - 0.2*x - 0.8*x^2 - x^12 + 0.2*x^13 + 0.8*x^14
> moy = 5
> (masaiso = polynomial(c(1, rep(0, 11), .5)))
1 + 0.5*x^12
> (MA2 = array(masaiso, c(13, 1, 1)))
, , 1

      [,1]
 [1,]  1.0
 [2,]  0.0
 [3,]  0.0
 [4,]  0.0
 [5,]  0.0
 [6,]  0.0
 [7,]  0.0
 [8,]  0.0
 [9,]  0.0
[10,]  0.0
[11,]  0.0
[12,]  0.0
[13,]  0.5
> #
> AR2 = array(ar.14, c(15, 1, 1))
> MA2= array(masaiso, c(13, 1, 1))
> require("dse")
> modsarima = ARMA(A = AR2, B = MA2, C = 1)
> derive = moy
> cte = derive*predict(polynomial(ar.1), 1)
> u.t = cte*matrix(rep(1, nobs))
> y2 = simulate(modsarima, input = u.t, sampleT = nobs)$output
```

Pour vérifier, on calcule la moyenne de la série différenciée une fois simplement et une fois saisonnièrement :

```
> mean(diff(diff(y2, 1), 12))
[1] 4.920148
```

puis estimons le modèle dont on a simulé une trajectoire :

```
> x1 <- as.matrix(seq(1, length(y2)))
> x2 <- as.matrix(seq(1, length(y2))^2)
```

```
> (m1 = Arima(y2, order = c(1, 1, 0),
+ seasonal = list(order = c(0, 1, 1), period = 12),
+  xreg = x1))
```

```
Series: y2
ARIMA(1,1,0)(0,1,1)[12]

Coefficients:
         ar1    sma1         x1
      0.5131  0.6824    33.5463
s.e.  0.0676  0.0557  3316.4340

sigma^2 estimated as 9.258:  log likelihood=-475.8
AIC=959.6   AICc=959.82   BIC=972.53
```

```
> (m2 = Arima(y2, order = c(1, 1, 0),
+  seasonal = list(order = c(0, 1, 1), period = 12),
+  include.drift = TRUE))
```

```
Series: y2
ARIMA(1,1,0)(0,1,1)[12]

Coefficients:
         ar1    sma1
      0.5132  0.6824
s.e.  0.0676  0.0556

sigma^2 estimated as 9.205:  log likelihood=-475.79
AIC=957.59   AICc=957.72   BIC=967.28
```

```
> (m3 = Arima(y2, order = c(1, 1, 0),
+  seasonal = list(order = c(0, 1, 1), period = 12),
+  xreg = x2))
```

```
Series: y2
ARIMA(1,1,0)(0,1,1)[12]

Coefficients:
          ar1    sma1      x2
      -0.8181  0.5796  0.2055
s.e.   0.0408  0.0701  0.0027

sigma^2 estimated as 1.084:  log likelihood=-274.4
AIC=556.8   AICc=557.01   BIC=569.72
```

On peut observer que l'estimation n'a pas fonctionné correctement aussi bien en introduisant le régresseur $1, 2, \cdots$ qu'une dérive. On peut enfin vérifier que $24 \times$ `cf3["x2"]` $= 4.93$ qui est très proche de la moyenne empirique de la série différenciée deux fois.

Exercice 5.3 (Lag plot d'une série avec dérive)
Simuler des séries de 200 points suivant

$$y_t = c + y_{t-4} + u_t, \quad \text{avec } u_t = \frac{1}{1-0.9\mathrm{B}} z_t$$

avec $\sigma_z^2 = 1$ et les valeurs initiales égales à 0. D'abord avec $c = -.2$ puis $c = .2$. Dessiner les lag plots de ces séries jusqu'au retard 6 et observer la dérive sur ces graphes. Commenter.

Réponse. Le plus simple est de simuler l'ARMA de la série différenciée puis de l'intégrer.

```
> set.seed(51)
> c0 = -.2
> y0 = arima.sim(list(order = c(1, 0, 0), ar = .9), n = 200) + c0
> y1 = diffinv(y0, 4)[-(1:4)]
> lag.plot(rev(y1), 4, layout = c(2, 2), do.lines = FALSE,
+    main = "Derive negative", diag.col = "red")
> c0 = .2
> y0 = arima.sim(list(order = c(1, 0, 0), ar = .9), n = 200) + c0
> y1 = diffinv(y0, 4)[-(1:4)]
> lag.plot(rev(y1), 4, layout = c(2, 2), ask = NA, do.lines = FALSE,
+          main = "Derive positive", diag.col = "red")
```

On observe dans la plupart des simulations qu'au retard 4 il y a plus de points au-dessus de la droite $y = x$ quand la dérive est positive et moins de points au-dessus de la droite $y = x$ quand elle est négative. L'écart vertical à la diagonale est de l'ordre de c.

Exercice 5.4 (Régression d'une marche aléatoire)
Simuler deux marches aléatoires x et y indépendantes, par exemple à l'aide du code :

```
> set.seed(514); nobs = 300
> y0 = 2 + rnorm(nobs)
> y = diffinv(y0)
> x0 = 2 - 1.5*rnorm(nobs)
> x = diffinv(x0)
```

1. Dessiner le diagramme de dispersion de (x,y).
2. Superposer les chronogrammes des deux séries.
3. Que suggèrent ces graphiques ?
4. Effectuer la régression linéaire de y sur x.
5. Etudier la stationnarité du résidu et conclure sur la pertinence de cette régression.

Réponse

1. `> plot(x, y, type = "l")`

2. `> plot.ts(cbind(x, y), plot.type = "single", lty = 1:2)`
   `> legend("topleft", c("x", "y"), lty = 1:2)`

   Le diagramme de dispersion (fig. 1 haut) suggère contre tout bon sens une forte corrélation, et les chronogrammes (fig. 1 bas) évoluent parallèlement si on néglige le décrochement entre les deux séries.

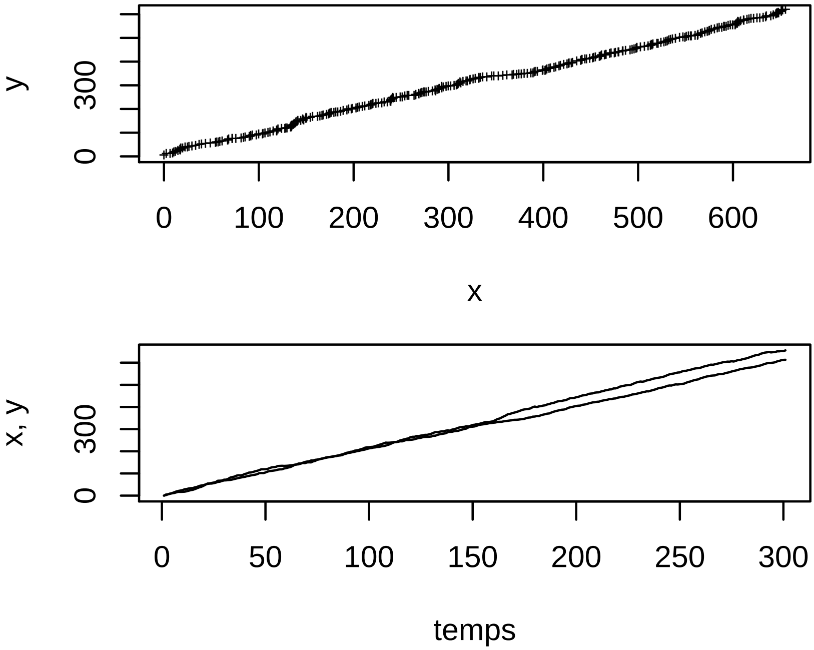

Fig. 1 – Diagramme de dispersion (haut) et chronogrammes (bas) de deux marches aléatoires, `x` et `y`, indépendantes.

3. Les graphiques suggèrent une évolution parallèle des deux séries.
4. Nous régressons `y` sur `x`.

```
> mod4 = lm(y~x)
> (aa = summary(mod4))

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-14.6515  -7.2223  -0.4608   7.2864  18.6121

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 9.015690   0.988905   9.117   <2e-16 ***
x           0.901712   0.002575 350.235   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 8.582 on 299 degrees of freedom
Multiple R-squared:  0.9976,       Adjusted R-squared:  0.9976
F-statistic: 1.227e+05 on 1 and 299 DF,  p-value: < 2.2e-16
```

Nous obtenons R2 $= 0.9976$, valeur élevée, et la régression semble très significative. Pour aller plus loin, nous dessinons le chronogramme du résidu et son ACF (fig. ??). Le chronogramme du résidu montre de longues séries de valeurs de même signe ; typiquement, ce résidu n'est pas stationnaire. L'accumulation de valeurs consécutives de même signe entraîne une forte valeur de l'autocorrélation d'ordre 1. Evidemment, cette autocorrélation est purement numérique et n'est pas l'estimation d'une autocorrélation théorique, non définie dans cet exemple.

5. `> acf(resid)`

```
> op <- par(mfrow = c(2, 1), mar = c(4, 3, 1, 0), oma = c(0, 0, 0, 0
> plot.ts(mod4$residuals, xlab = "temps", ylab = "residu MCO")
> abline(h = 0)
> acf(mod4$residuals, main = "")
> par(op)
```

Résumons. Cet exemple nous a conduit à effectuer une régression ayant une significativité illusoire : un R2 élevé et une régression apparemment très significative, le diagramme de dispersion des deux séries est trompeur mais les chronogrammes montrent l'absence de lien entre elles.
Pour compléter le traitement de cet exemple, voyons ce que suggère un test de racine unité. On sait que le résidu est de moyenne nulle, d'où, `type=none`, indiqué dans l'appel de `ur.df()`, ci-dessous. Des essais avec différentes valeurs de `lags` nous conduisent à `lags=0`

```
require(urca)
ur.mod4=ur.df(mod4$residuals, type = "none",lags = 0)
summary(ur.mod4)
```

On obtient une p-value assez élevée, proche de 10%. On conserve donc l'hypothèse nulle : la série du résidu est non stationnaire.

Bibliographie

Gourieroux C. et Monfort A. (1995). Séries temporelles et modèles dynamiques. Economica, 2 edn..

Ladiray D. et Quenneville B. (2000). Désaisonnaliser avec la méthode x-11. Tech. Rep.. http://www.census.gov/ts/papers/x11french.pdf.

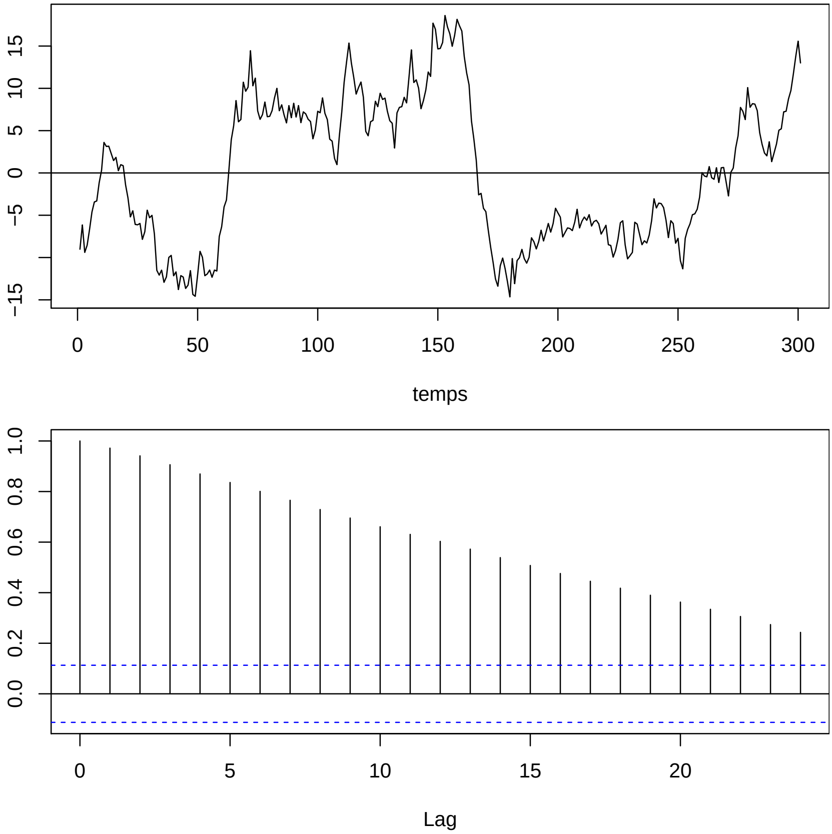

Fig. 2 – Résidu de la régression de y sur x - Chronogramme et ACF.