

SERIES TEMPORELLES

ESTIMATION PARAMETRIQUE ET NON PARAMETRIQUE AVEC LE LOGICIEL R

Mohamed BOUTAHAR ¹

4 décembre 2007

¹Département de mathématiques case 901, Faculté des Sciences de Luminy. 163 Av. de Luminy 13288 MARSEILLE Cedex 9, e-mail : boutahar@univmed.fr

Table des matières

1	Modèle de régression multiple	4
0.1	Hypothèse de normalité (conséquence)	6
0.2	Prévision	7
1	Exemples de modélisation	8
2	Processus autorégressif et à moyenne mobile (A.R.M.A.)	15
1	Processus du second ordre	15
1.1	Généralité	15
2	Processus stationnaire au second ordre	16
2.1	Définitions et notations	16
2.2	Notion de filtre de moyenne mobile	17
3	Analyse spectrale d'un processus stationnaire au second ordre	18
4	Processus A.R.M.A.	21
4.1	Processus AR(p)	21
5	Processus A.R.M.A	24
6	Estimation	25
6.1	Estimation d'un $AR(p)$	26
6.2	Estimation d'un $ARMA(p)$	27
6.3	Prévision d'un $ARMA(p,q)$	29
3	Processus non stationnaires	36
1	Modèles ARIMA.	36

1.1	MODELISATION ARIMA	37
1.2	Etapes de la modélisation	37
1.3	Test de non-stationnarité	38
1.4	Tests de stationnarité	40
2	Modèles SARIMA	47
3	Modélisation par les modèles S.A.R.I.M.A.	48
3.1	Modélisation :	49
4	PROCESSUS VECTORIELS STATIONNAIRES AU SECOND ORDRE- MODELE VARMA	54
1	Définitions	54
2	Modèle $VARMA_d(p, q)$	55
3	Estimation	55
3.1	Densité spectrale croisée	59
5	Analyse Spectrale	62
6	Estimation de la densité de probabilité d'une variable aléatoire.	66
1	Estimation par histogramme	66
1.1	Estimateur du noyau	67
7	Régression non paramétrique	72
1	Estimateur à noyau	72

Partie 1 : ESTIMATION PARAMETRIQUE

Chapitre 1

Modèle de régression multiple

Modèle

$$y_t = \sum_{j=1}^k b_j x_{tj} + u_t, \quad 1 \leq t \leq T, \quad (1.1)$$

où u_n est un bruit blanc c'est à dire est une suite de v.a.r. satisfaisant à :

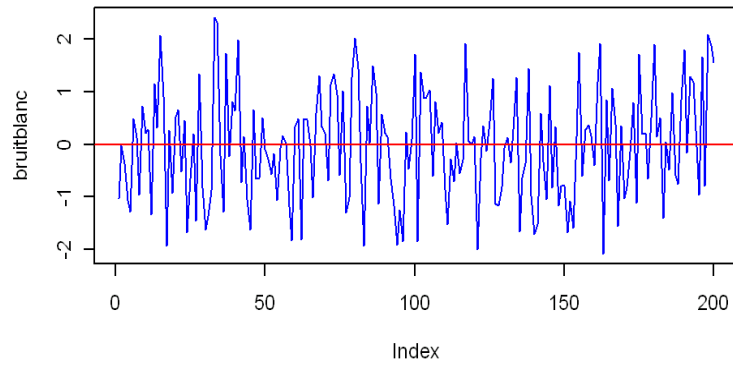
$$\left\{ \begin{array}{ll} \text{(H.1)} & E(u_t) = 0, \quad 1 \leq t \leq T. \\ \text{(H.2)} & E(u_t u_s) = \sigma^2 \delta_s^t, \quad \sigma^2 \geq 0. \end{array} \right.$$

Avec les commandes sur R :

```
bruitblanc <- rnorm(200,0,1)
```

```
plot(bruitblanc,type="l")
```

on obtient la trajectoire suivante :



Trajectoire d'un bruit blanc Gaussien $N(0, 1)$

Posons $b = {}^t(b_1, \dots, b_k)$, $y = {}^t(y_1, \dots, y_T)$, \mathbf{X} la matrice (T, k) dont la j^{ieme} colonne est $x_j = {}^t(x_{1j}, \dots, x_{Tj})$, et $u = {}^t(u_1, \dots, u_T)$, le système prend alors la forme matricielle suivante :

$$y = \mathbf{X}b + u \quad (1.2)$$

$x_j, 1 \leq j \leq k$ représentent les variables explicatives (ou facteurs exogènes) ;

y représente la variable expliquée (ou facteur endogène),

u est la perturbation,

b est le vecteur des paramètres.

	Aléatoire	Déterministe
Observable	y	$x_j, 1 \leq j \leq k$
Non observable	u	b

Estimateur des moindres carrés de b :

$$\hat{b} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'y. \quad (1.3)$$

Proposition 1

i) $E(\hat{b}) = b$ (estimateur sans biais)

ii) $Var(\hat{b}) = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}$ (matrice de covariance).

0.1 Hypothèse de normalité (conséquence)

En plus des hypothèses (H.1) et (H.2) on ajoute une hypothèse (H.3) de normalité du vecteur aléatoire u .

Proposition 2 Sous $(\mathbf{H.j})_{1 \leq j \leq 3}$, \hat{b} et $\frac{1}{T} \|\hat{u}\|^2$ sont des estimateurs de maximum de vraisemblance de b et σ^2 ; $\hat{u} = y - \hat{y}$ (vecteur des résidus), $\hat{y} = X\hat{b}$.

Proposition 3

i) $\hat{b} \sim N(b, \sigma^2 (\mathbf{X}'\mathbf{X})^{-1})$,

ii) $\frac{(T-k)\hat{\sigma}^2}{\sigma^2} \sim \chi^2(T-k)$,

iii) $\hat{b}, \hat{\sigma}^2$ sont des statistiques indépendantes.

$$\frac{\hat{b}_j - b_j}{\sqrt{\hat{\sigma}^2 a_{jj}}} \sim t(T-k), \quad 1 \leq j \leq k;$$

a_{jj} est le j^{ieme} élément diagonal de $(\mathbf{X}'\mathbf{X})^{-1}$.

Construction de tests d'hypothèses

Au seuil α :

$$H_0 : b_j = 0 \quad \xrightarrow{\text{région critique}} \quad T_j = \frac{|\hat{b}_j|}{\sqrt{\hat{\sigma}^2 a_{jj}}} \geq t_{1-\alpha/2}(T-k) \quad (1.4)$$

$$H_0 : b_j \leq 0 \quad \text{région critique} \longrightarrow \quad T_j^1 = \frac{\hat{b}_j}{\sqrt{\hat{\sigma}^2 a_{jj}}} \geq t_{1-\alpha}(T-k),$$

où $t_{1-\alpha}(T-k)$ désigne le quantile d'ordre $1-\alpha$ de la loi de Student à $(T-k)$ degrés de liberté.

(On pourrait de même construire des tests concernant σ^2).

Construction de région de confiance (au niveau α)

pour $b_j, (1 \leq j \leq k)$, c'est l'intervalle :

$$\hat{b}_j \pm \sqrt{\hat{\sigma}^2 a_{jj}} \quad t_{1-\alpha/2}(T-k)$$

0.2 Prédiction

Comment, à l'instant T , prévoir la réalisation de la v.a. réelle y_{T+1} disposant de $x_{T+1} = {}^t(x_{T+1,1}, \dots, x_{T+1,k})$, c'est à dire d'une réalisation supplémentaire de chacune des k variables explicatives. Les hypothèses sont **(H.1)**, **(H.2)** et

$$\left\{ \begin{array}{l} y_{T+1} = {}^t x_{T+1} b + u_{T+1}, \\ E(u_{T+1}) = 0, \\ E(u_{T+1} u_t) = \sigma^2 \delta_t^{T+1}, \quad 1 \leq t \leq T. \end{array} \right.$$

On doit tenir compte dans ce problème de prédiction de deux incertitudes :

- i) Celle due à u_{T+1} (perturbation aléatoire);
- ii) Celle due au fait que b est inconnu.

Prédiction optimale

$$\hat{y}_{T+1} = {}^t x_{T+1} \hat{b}.$$

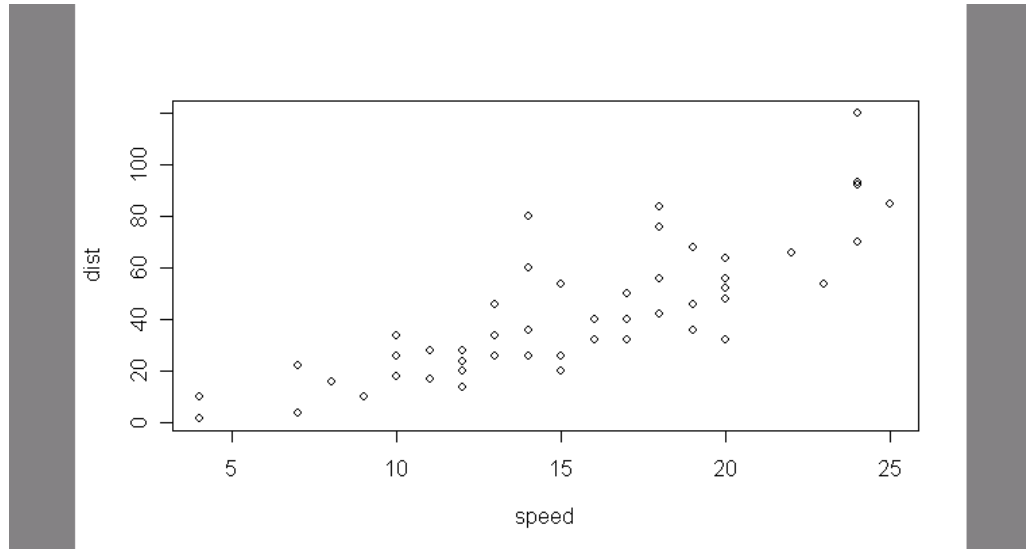
L'intervalle de confiance au niveau $(1-\alpha)$ est donné par :

$$\hat{y}_{T+1} \pm \hat{\sigma} t_{1-\alpha/2}(T-k) \sqrt{1 + {}^t x_{T+1} (\mathbf{X}'\mathbf{X})^{-1} x_{T+1}}$$

1 Exemples de modélisation

Exemple 1 : Modèle linéaire simple

Avec les commandes sur R : `data(cars)` , `plot(cars)` on obtient :



On cherche la relation qui lie la variable endogène *dist* (distance de freinage) et la variable exogène *speed* (vitesse). On va estimer le modèle $dist_t = b_1 + b_2 speed_t + u_t$,

Avec les commandes : `fit <- lm(cars$dist ~ cars$speed)`, `summary(fit)` on obtient :

Call :

`lm(formula = cars$dist ~ cars$speed)`

Residuals :

<i>Min</i>	<i>1Q</i>	<i>Median</i>	<i>3Q</i>	<i>Max</i>
-29.069	-9.525	-2.272	9.215	43.201

Coefficients :

<i>Estimate</i>	<i>Std. Error</i>	<i>t value</i>	<i>Pr(> t)</i>
(Intercept) -17.5791	6.7584	-2.601	0.0123 *
<i>cars\$speed</i> 3.9324	0.4155	9.464	1.49e-12 ***

Signif. codes : 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error : 15.38 on 48 degrees of freedom

Multiple R-Squared : 0.6511, Adjusted R-squared : 0.6438

F-statistic : 89.57 on 1 and 48 DF, p-value : 1.490e-12

Interprétation des résultats :

En premier lieu R affiche les statistiques descriptives des résidus de la régression.

Ensuite les estimations $\hat{b}_1 = -17.5791$, $\hat{b}_2 = 3.9324$.

Std. Error désigne l'écart type estimé.

t. value est la statistique de Student donnée par (1.4).

$Pr(>|t|)$ est la p-value associé à la statistique de Student *t* value, une valeur plus petite que 0.01 nous conduit au rejet de H_0 c'est à dire que le paramètre est significatif.

Pour notre modèle le paramètre b_1 n'est pas significatif alors que b_2 l'est.

$$\text{Residual standard error} = \sqrt{\frac{\|\tilde{u}\|^2}{T-k}}.$$

Multiple R-Squared est définie par :

Definition 1 Lorsqu'il y'a une constante dans le modèle de régression multiple, on appelle coefficient de détermination le scalaire

$$R^2 = \frac{\|\hat{y} - \bar{y}\delta_T\|^2}{\|y - \bar{y}\delta_T\|^2}$$

où $\bar{y} = \frac{1}{T} \sum_{n=1}^T y_n$ et $\delta_T = {}^t(1, \dots, 1)$, vecteur $(T, 1)$.

Plus R^2 est proche de 1, plus l'ajustement est meilleur.

Adjusted R-squared est définie par $R_a^2 = 1 - (1 - R^2) \frac{T-1}{T-p-1}$, où p est le nombre de paramètres sans compter la constante. R_a^2 ne croit que si la nouvelle variable explicative ajoutée améliore l'ajustement, elle peut être négative, et $R_a^2 \leq R^2$.

F-statistic désigne la statistique de Fisher qui correspond à l'hypothèse nulle (sans la constante) $H_0 : b_1 = \dots = b_k = 0$; ou $(H_0 : b_2 = \dots = b_k = 0 \text{ si } x_{t,1} = 1)$. Elle est donnée par $F = \frac{\|\tilde{u}\|^2 - \|\hat{u}\|^2}{\|\hat{u}\|^2}$ avec \tilde{u} : résidus sous H_0 et \hat{u} : résidus sous H_1 . On a sous H_0 la statistique F suit une loi de Fisher $F(k, T - k)$ ou $F(k - 1, T - k)$ s'il y a une constante dans le modèle

Avec les commandes : `fit1 <- lm(cars$dist ~ cars$speed), summary(fit1)` on obtient :

Call :

```
lm(formula = cars$dist ~ cars$speed - 1)
```

Residuals :

Min	1Q	Median	3Q	Max
-26.183	-12.637	-5.455	4.590	50.181

Coefficients :

Estimate Std. Error t value Pr(>|t|)

cars\$speed 2.9091 0.1414 20.58 <2e-16 ***

Signif. codes : 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error : 16.26 on 49 degrees of freedom

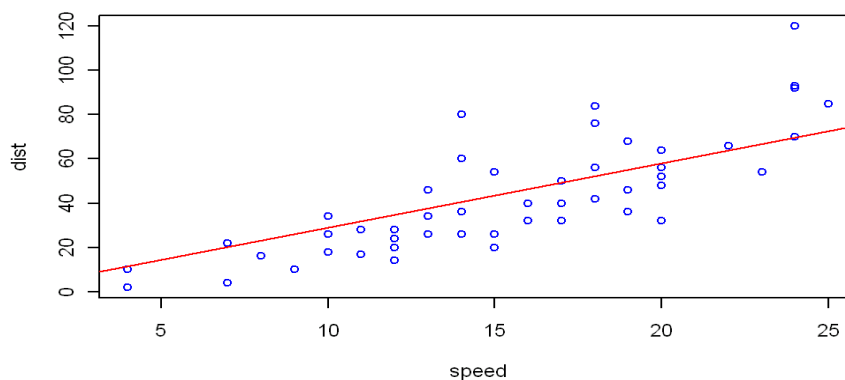
Multiple R-Squared : 0.8963, Adjusted R-squared : 0.8942

F-statistic : 423.5 on 1 and 49 DF, p-value : < 2.2e-16.

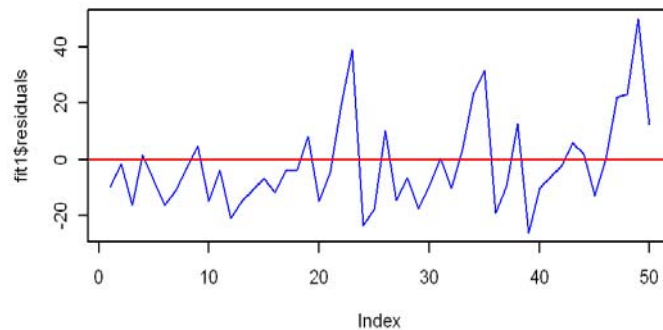
Le R^2 s'approche de 1, donc l'ajustement est meilleur.

Avec le code : `plot(cars,col="blue")` `z <- lm(dist ~ speed, data = cars)`, `abline(z,col="red")`

on obtient



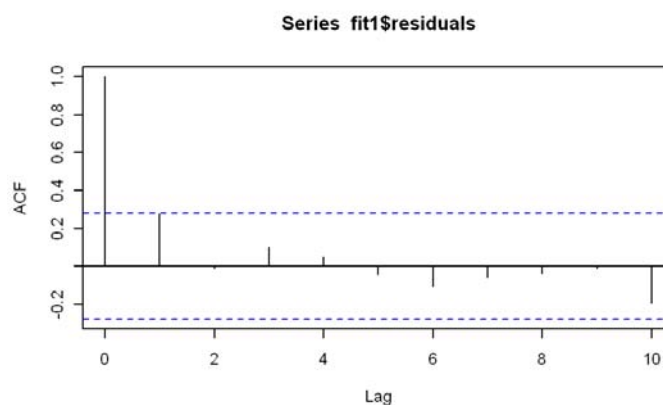
Analyse des résidus de la régression



Trajectoire des résidus de la régression

1. Absence d'autocorrélation

Avec la commande : `acf(fit1$residuals,10,"correlation")` on obtient



La fonction `acf` définit la fonction d'autocorrélation

Exmple 2. Régression multiple : On considère l'évolution de l'ozone en fonction de la radiation solaire, le vent et la température

Avec la commande

```
fit3 <- lm(air$ozone ~ air$radiation + air$temperature + air$wind - 1)
```

on obtient

Call :

```
lm(formula = air$ozone ~ air$radiation + air$temperature + air$wind - 1)
```

Coefficients :

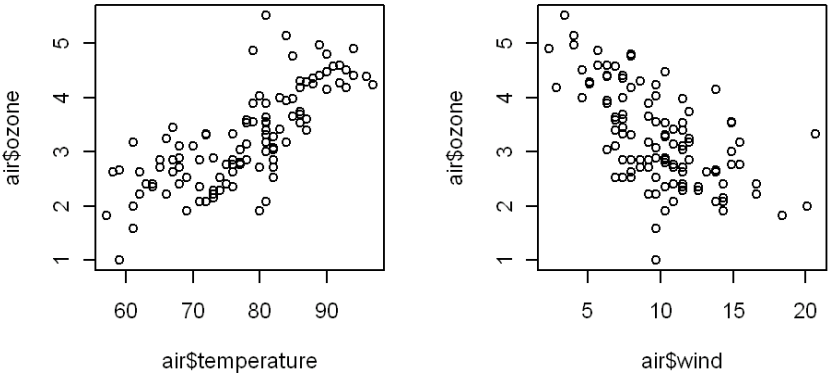


FIG. 1.1 –

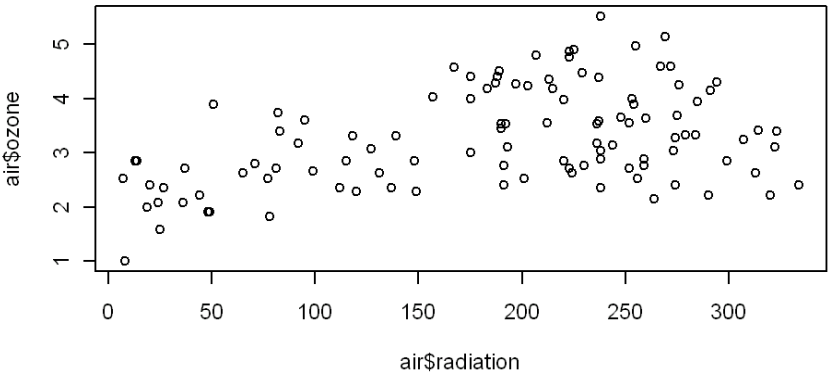


FIG. 1.2 –

<i>air\$radiation</i>	<i>air\$temperature</i>	<i>air\$wind</i>
0.002220	0.046962	-0.081866

et *summary(fit3)* donne

Call :

lm(formula = air\$ozone ~ air\$radiation + air\$temperature + air\$wind - 1)

Residuals :

Min 1Q Median 3Q Max

-1.12247 -0.38863 -0.04382 0.32710 1.46376

Coefficients :

	<i>Estimate</i>	<i>Std. Error</i>	<i>t value</i>	<i>Pr(> t)</i>
<i>air\$radiation</i>	0.0022205	0.0005559	3.994	0.000119 ***
<i>air\$temperature</i>	0.0469625	0.0020346	23.082	< 2e-16 ***
<i>air\$wind</i>	-0.0818660	0.0113257	-7.228	7.34e-11 ***

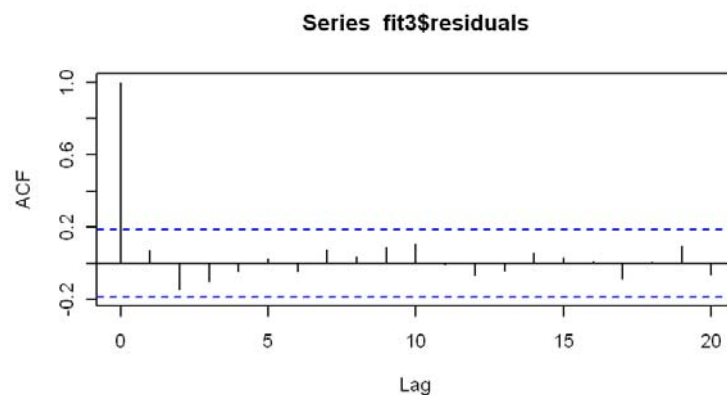
*Signif. codes : 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

Residual standard error : 0.5085 on 108 degrees of freedom

Multiple R-Squared : 0.9778, Adjusted R-squared : 0.9772

F-statistic : 1586 on 3 and 108 DF, p-value : < 2.2e-16

Les résidus de la régression sont donnés par :



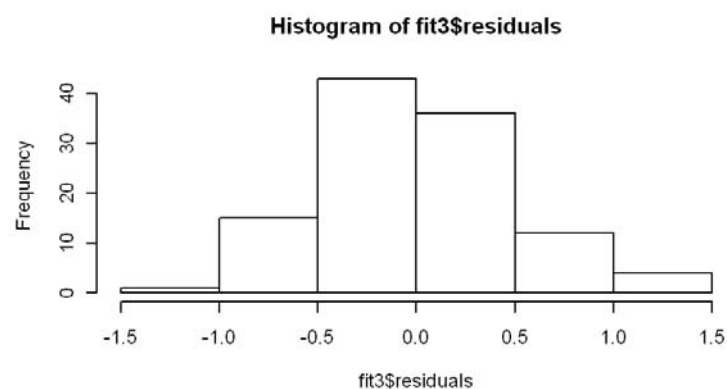
Avec la commande : `Box.test(fit3$residuals, lag = 20, type = "Box-Pierce")` on obtient :

Box-Ljung test

data : fit3\$residuals

X-squared = 12.2498, df = 20, p-value = 0.9072

Les résidus ne sont pas corrélés.



Conclusion : L'ajustement est de très bonne qualité et l'hypothèse de normalité des données est satisfaite.

Chapitre 2

Processus autorégressif et à moyenne mobile (A.R.M.A.)

1 Processus du second ordre

1.1 Généralité

Definition 2 On appelle *processus aléatoire du second ordre* à temps discret, réel toute suite $X_t : \Omega \longrightarrow \mathbb{R}$ définie sur un espace de probabilité (Ω, \mathcal{Q}, P) à valeurs dans $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, indexé par $n \in \mathbb{N}$ ou \mathbb{Z} telle que :

$$E(|X_t|^2) < \infty, \text{ pour tout } t.$$

Pour un tel processus on peut définir les fonctions de moyenne et d'auto-covariance :

- i) Moyenne : $\mu : \mathbb{Z} \longrightarrow \mathbb{R}, \mu(t) = E(X_t), \forall t \in \mathbb{Z}.$
- ii) Auto-covariance : $K : \mathbb{Z} \times \mathbb{Z} \longrightarrow \mathbb{R}, K(t, s) = E((X_t - \mu(t))(X_s - \mu(s))), \forall t, s \in \mathbb{Z}.$

Ces fonctions s'expriment encore à l'aide du produit scalaire :

$$\langle X, Y \rangle = E(XY)$$

défini sur l'espace de probabilité $L^2(\Omega, \mathcal{Q}, P)$ sous jacent au processus ; alors

$$\mu(t) = \langle X_t, 1 \rangle, \quad K(t, s) = \langle X_t, X_s \rangle - \mu(t)\mu(s).$$

Le processus est dit centré si $\mu = 0$.

Proposition 4 *La fonction d'auto-covariance est symétrique de type positif, i.e.*

$$K(t, s) = K(s, t); \quad \sum_t \sum_s a(t)K(t, s)a(s) \geq 0$$

pour toute fonction $a : \mathbb{Z} \longrightarrow \mathbb{R}$ à support fini (quels que soient les entiers $(t_j)_{1 \leq j \leq k}$, la matrice $[K(t_i, t_j)]_{1 \leq i, j \leq k}$ est symétrique positive).

2 Processus stationnaire au second ordre

2.1 Définitions et notations

Soit X un processus; on dit que X est strictement (ou fortement) stationnaire si pour toute suite finie d'instant t_1, \dots, t_k élément de \mathbb{Z} et tout entier $r \in \mathbb{Z}$, les lois jointes de $(X_{t_1}, \dots, X_{t_k})$ et de $(X_{r+t_1}, \dots, X_{r+t_k})$ sont les mêmes (lois jointes invariantes par translation dans le temps).

En effet dans bien des cas concrets, il est impossible de recommencer des séries de mesures (ex : température quotidienne en un lieu géographique précis, toujours à la même heure pendant 3 ans), on peut donc chercher à n'étudier que les modèles aléatoires dont le comportement dans le temps est assez stable.

On s'intéressera, dans la suite, à une stabilité plus faible.

Un processus au second ordre est stationnaire au second ordre (on dit aussi faiblement stationnaire) si moyenne et auto-covariance sont invariantes par translation dans le temps, i.e. :

$$\left\{ \begin{array}{l} E(X_t) = \mu, \forall t \in \mathbb{Z}, \\ \text{et } E(X_t - \mu)(X_s - \mu) = E(X_{t+r} - \mu)(X_{s+r} - \mu), \quad \forall (t, s, r) \in \mathbb{Z}^3. \end{array} \right.$$

où μ est une constante et $K(t, s)$ ne dépend plus que de $t - s$, donc il existe une fonction $\gamma : \mathbb{Z} \longrightarrow \mathbb{R}$, telle que : $K(t, s) = \gamma(t - s)$. Alors pour un processus faiblement

stationnaire, on a :

$$\left\{ \begin{array}{l} \text{var}(X_t) = \gamma(0), \forall t \in \mathbb{Z}, \\ |\gamma(k)| \leq \gamma(0), \forall k \in \mathbb{Z}, \text{ (par l'inégalité de Shwartz)} \\ \gamma(j) = \gamma(-j), \forall k \in \mathbb{Z}, \\ \rho(j) = \frac{\gamma(j)}{\gamma(0)}, \forall j \in \mathbb{Z}, \text{ fonction de corrélation.} \\ r_p(j) = \frac{\text{cov}(X_1 - \mathbf{P}_{[X_2, \dots, X_j]}(X_1), X_{j+1} - \mathbf{P}_{[X_2, \dots, X_j]}(X_{j+1}))}{\text{var}(X_1 - \mathbf{P}_{[X_2, \dots, X_j]}(X_1))} \\ \text{(fonction de corrélation partielle) .} \end{array} \right.$$

De plus γ est de type positif au sens où :

$$\sum_t \sum_s a(t) \gamma(t-s) a(s) \geq 0,$$

pour toute fonction réelle a définie sur \mathbb{Z} et à support fini.

2.2 Notion de filtre de moyenne mobile

L'opération de convolution

$$Y_t = \sum_{j=0}^p a_j X_{t-j} \quad (2.1)$$

effectuée par le filtre moyenne mobile $(a_j)_{0 \leq j \leq p}$ sur le processus X au cours du temps se représente commodément à l'aide de l'opérateur retard B ($BY_t = Y_{t-1}$) à

$$Y_t = \sum_{j=0}^p a_j B^j X_t = A(B)X_t,$$

où A désigne le polynôme $A(B) = \sum_{j=0}^p a_j B^j$.

Notons $U = [-\pi, \pi[$; la fonction $\mathcal{R} : U \longrightarrow \mathbb{C}$, définie par :

$$\mathcal{R}(\lambda) = \sum_{j=0}^p a_j e^{-ij\lambda} = A(e^{-i\lambda})$$

est appelé fonction réponse du filtre moyenne mobile $(a_j)_{0 \leq j \leq p}$. De plus si Y traverse un autre filtre moyenne mobile de polynôme $C(B) = \sum_{j=0}^{p'} c_j B^j$, le processus image Z défini par :

$$Z_t = \sum_{j=0}^{p'} c_j Y_{t-j} = C(B)Y_t,$$

s'exprime à partir de X comme suit :

$$Z_t = (A.C)(B)X_t, \quad (2.2)$$

à l'aide du produit des deux polynômes A et B .

3 Analyse spectrale d'un processus stationnaire au second ordre

Theorem 5 (Herglotz-Fourier) Soit $X = (X_t)_{t \in \mathbb{Z}}$ un processus stationnaire au second ordre, de fonction d'auto-covariance $\gamma(k)$ et de densité spectrale $f(\lambda)$. Alors

$$\gamma(k) = \int_{-\pi}^{\pi} e^{ik\lambda} f(\lambda) d\lambda \quad \text{pour tout } k \in \mathbb{Z}$$

$$f(\lambda) = \frac{1}{2\pi} \sum_{k \in \mathbb{Z}} e^{-ik\lambda} \gamma(k)$$

Exemples

i) **Bruit blanc** : $\varepsilon = (\varepsilon_t)_{t \in \mathbb{Z}}$ un processus centré tel que $E(\varepsilon_t \varepsilon_s) = \sigma^2 \delta_t^s$. Alors $f(\lambda) = \frac{\sigma^2}{2\pi}$.

voir la figure ??.

ii) **Moyenne mobile** : Soit $(\varepsilon_t)_{t \in \mathbb{Z}}$ un bruit blanc, un processus $MA(q)$ est défini par

$$X_t = \sum_{k=0}^q c_k \varepsilon_{t-k}, \quad c_0 = 1.$$

$$\begin{aligned}
\gamma(k) &= E(X_{t+k}X_t) \\
&= \sum_{l,m} c_l c_m E(\varepsilon_{t+k-l}\varepsilon_{t-m}), \quad |k| \leq q, \\
&= \sigma^2 \sum_{l=0}^{q-k} c_{k+l} c_l, \quad |k| \leq q,
\end{aligned}$$

$$\gamma(k) = 0, \quad \text{si } |k| > q;$$

donc un tel processus admet une densité spectrale (continue)

$$\begin{aligned}
f(\lambda) &= \frac{1}{2\pi} \sum_k \gamma(k) e^{-ik\lambda}, \\
&= \frac{\sigma^2}{2\pi} \sum_{|k| \leq q} \sum_{l=0}^{q-k} c_{k+l} c_l e^{-ik\lambda}
\end{aligned}$$

posons $k + l = m$,

$$\begin{aligned}
f(\lambda) &= \frac{\sigma^2}{2\pi} \sum_{m=0}^q \sum_{l=0}^q c_m c_l e^{-i(m-l)\lambda} \\
&= \frac{\sigma^2}{2\pi} \left| \sum_{l=0}^q c_l e^{-il\lambda} \right|^2
\end{aligned}$$

Proposition 6 Soit X un processus stationnaire au second ordre, $(c_j)_{0 \leq j \leq q}$ un filtre de moyenne mobile et Y le processus défini par

$$Y_t = \sum_{j=0}^q c_j X_{t-j}.$$

Alors la densité spectrale de Y , est

$$f_Y(\lambda) = |\mathcal{R}(\lambda)|^2 f_X(\lambda) = |C(e^{-i\lambda})|^2 f_X(\lambda),$$

où $\mathcal{R}(\lambda)$ est la fonction réponse du filtre $(c_j)_{0 \leq j \leq q}$ et f_X la densité spectrale de X .

Avec les commandes : `library(stats), ts.ma <- arima.sim(list(order = c(0,0,1), ma = 0.7), n = 200), ts.plot(ts.ma, type = "l"), acf(ts.ma, 10, "correlation")` on obtient :

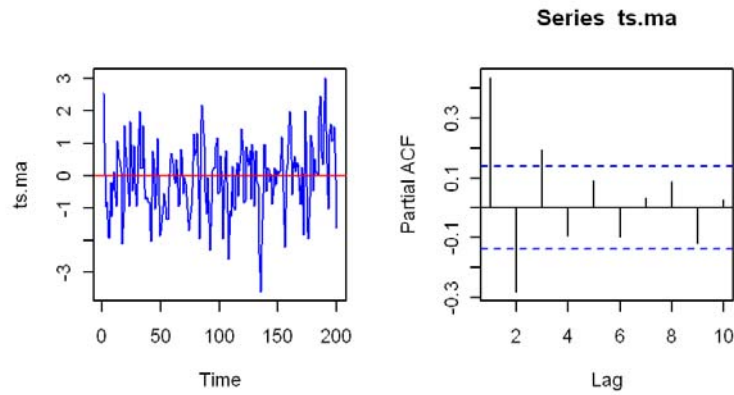
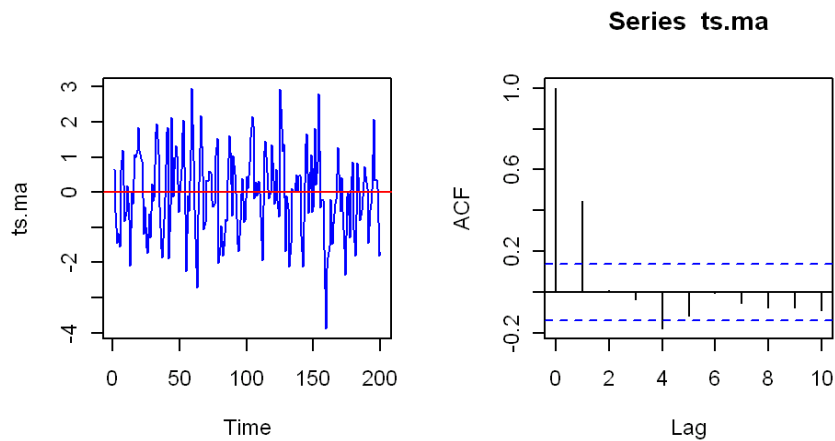


FIG. 2.1 – Trajectoire du processus $MA(1) : X_t = \varepsilon_t + 0.7\varepsilon_{t-1}$ et sa fonction d'autocorrélation partielle.



Trajectoire du processus $MA(1) : X_t = \varepsilon_t + 0.7\varepsilon_{t-1}$ et sa fonction d'autocorrélation.

Avec le code : `ts.ma <- arima.sim(list(order = c(0,0,1), ma = 0.7), n = 200)`,
`ts.plot(ts.ma,type="l",col="blue"),abline(0,0,col="red")`, `acf(ts.ma,10,"partial")` on ob-
tient

4 Processus A.R.M.A.

Definition 3 soit X un processus stationnaire au second ordre et centré.

Nous dirons que X est un processus A.R.M.A(p, q), s'il existe un bruit blanc ε défini sur le même (Ω, \mathcal{Q}, P) et des nombres réels $a_0(=1), a_1, \dots, a_p, c_0(=1), c_1, \dots, c_q$ tels que

$$\sum_{k=0}^p a_k X_{t-k} = \sum_{l=0}^q c_l \varepsilon_{t-l} \quad (2.3)$$

en particulier :

X est dit autorégressif d'ordre p , ou encore $AR(p)$, s'il vérifie (2.3) avec $q = 0$ i.e. si

$$\sum_{k=0}^p a_k X_{t-k} = \varepsilon_t \quad (2.4)$$

X est dit moyenne mobile d'ordre q , ou encore $MA(q)$, s'il vérifie (2.3) avec $p = 0$ i.e. si

$$X_t = \sum_{l=0}^q c_l \varepsilon_{t-l} \quad (2.5)$$

Nous allons caractériser rigoureusement les processus $AR(p)$ et donner, sans démonstration, quelques résultats sur les processus $ARMA(p, q)$.

4.1 Processus $AR(p)$

Ces processus jouent un rôle important dans les applications car ce sont ceux qui tout instant n peuvent être extrapolés linéairement à partir de p valeurs précédents X_{n-1}, \dots, X_{n-p} à un bruit blanc près. Si A désigne le polynôme unitaire ($A(0) = 1$) de degré p

$$A(t) = 1 + \sum_{j=0}^p a_j t^j, \quad (2.6)$$

l'équation (2.4) peut alors s'écrire, à l'aide de l'opérateur retard B , sous la forme

$$A(B)X_t = \varepsilon_t, \quad \forall t \in \mathbb{Z}. \quad (2.7)$$

Proposition 7 *La fonction d'autocorrélation partielle d'un processus $AR(p)$ vérifie*
 $r_p(j) = 0 \quad \forall j > p.$

Proposition 8 *Un processus autorégressif d'ordre p , possède une densité spectrale définie par*

$$f(\lambda) = \frac{\sigma^2}{2\pi |A(e^{-i\lambda})|^2} \quad (2.8)$$

(σ^2 est la variance de ε), et le polynôme A ne peut donc posséder de zéros de module 1.

Réciproquement si un processus centré stationnaire au second ordre X possède une densité spectrale de la forme (2.8) pour un nombre réel $\sigma > 0$ et un polynôme unitaire A de degré p (nécessairement sans zéro de module 1) ce processus est autorégressif d'ordre p les $(a_j)_{1 \leq j \leq p}$ étant donnés par (2.6).

Proposition 9 *La densité spectrale d'un processus $AR(p)$ s'écrit d'une seule manière sous la forme*

$$f(\lambda) = \frac{\sigma^2}{2\pi |A(e^{-i\lambda})|^2},$$

pour un réel $\sigma > 0$, et un polynôme unitaire A de degré p dont toutes les racines sont de module > 1 .

La représentation par (2.4) du processus X est alors appelée représentation canonique de X .

Avec les commandes : `library(stats), ts.ar <- arima.sim(list(order = c(1,0,0), ar = 0.7), n = 200), ts.plot(ts.ar, type="l"), acf(ts.ar, 10, "correlation")` on obtient :

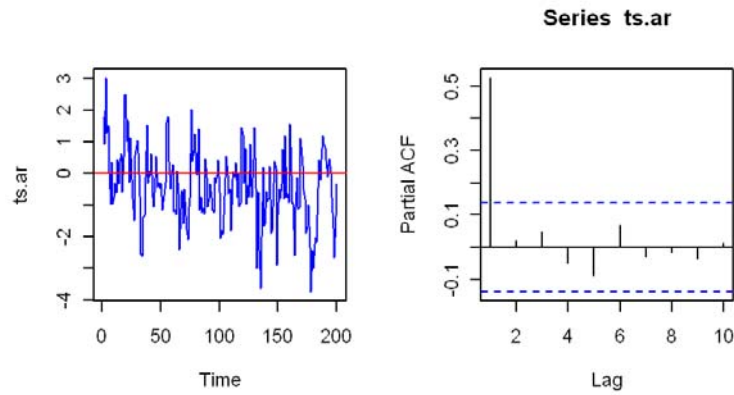
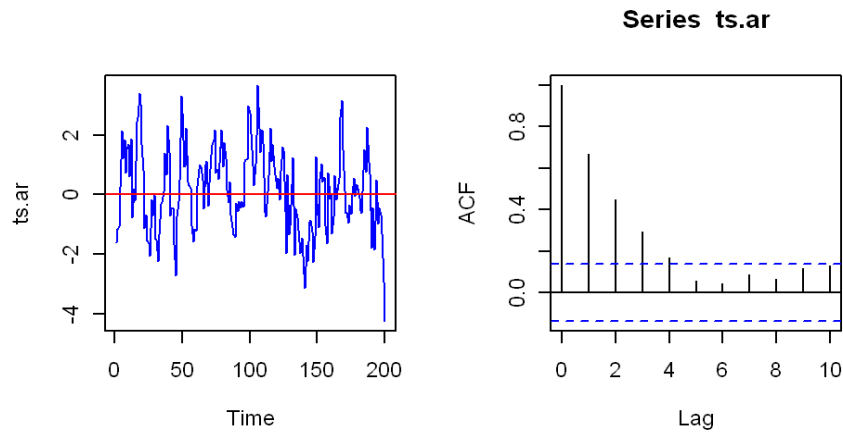


FIG. 2.2 –



Trajectoire du processus $AR(1)$: $X_t = 0.7X_{t-1} + \varepsilon_t$
et sa fonction d'autocorrélation.

Avec le code : `ts.ar <- arima.sim(list(order = c(1,0,0), ar = 0.7), n = 200)`,
`ts.plot(ts.ar,type="l",col="blue"), abline(0,0,col="red")`, `acf(ts.ar,10,"partial")` on ob-
tient

5 Processus A.R.M.A

Definition 4 Soit (X_t) le processus A.R.M.A défini par l'équation (2.3). On dit que (X_t) est causal si les racines de $A(z)$ sont toutes strictement à l'extérieur du disque unité.

Definition 5 On dit que (X_t) est inversible si les racines de $C(z)$ sont toutes strictement à l'extérieur du disque unité.

Proposition 10 (représentation canonique) Soit (X_t) un processus A.R.M.A stationnaire au second ordre ; i) Sa densité spectrale est donnée par :

$$f(\lambda) = \frac{\sigma^2 |C(e^{-i\lambda})|^2}{2\pi |A(e^{-i\lambda})|^2} \quad (2.9)$$

ii) Il existe deux polynômes $\tilde{A}(z)$ n'ayant que des racines strictement à l'extérieur du disque unité et $\tilde{C}(z)$ n'ayant que des racines à l'extérieur du disque unité, et un bruit blanc (η_t) tel que :

$$\tilde{A}(B)X_t = \tilde{C}(B)\eta_t,$$

cette représentation est unique et dit canonique, plus précisément si

$$A(z) = \prod_{j=1}^p (1 - \alpha_j z); \quad C(z) = \prod_{j=1}^q (1 - \beta_j z);$$

avec

$$|\alpha_j| < 1, \text{ si } 1 \leq j \leq r; \quad |\alpha_j| > 1, \text{ si } r+1 \leq j \leq p;$$

$$|\beta_j| < 1, \text{ si } 1 \leq j \leq s; \quad |\beta_j| > 1, \text{ si } s+1 \leq j \leq q;$$

alors

$$\tilde{A}(z) = \prod_{j=1}^r (1 - \alpha_j z) \prod_{j=r+1}^p \left(1 - \frac{1}{\alpha_j} z\right); \quad \tilde{C}(z) = \prod_{j=1}^s (1 - \beta_j z) \prod_{j=s+1}^q \left(1 - \frac{1}{\beta_j} z\right);$$

et le bruit blanc a pour variance

$$\sigma_\eta^2 = \sigma_\varepsilon^2 \frac{\prod_{j=s+1}^q |\beta_j|^2}{\prod_{j=r+1}^p |\alpha_j|^2}.$$

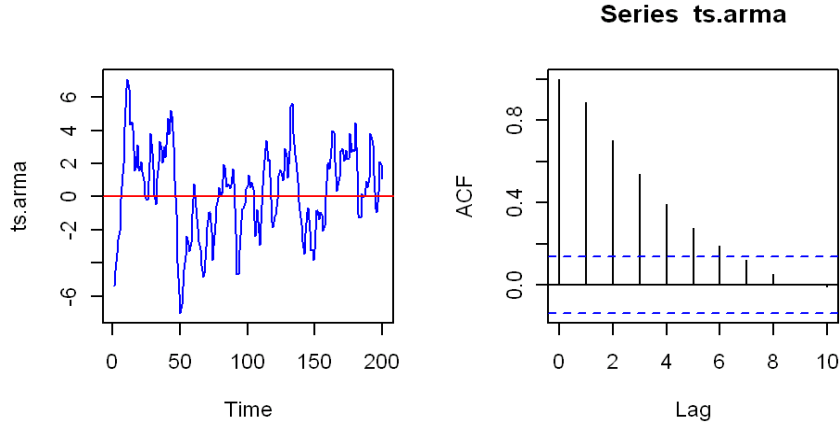


FIG. 2.3 – Trajectoire du processus $ARMA(1)$: $X_t = 0.8X_{t-1} + 0.6\varepsilon_{t-1} + \varepsilon_t$ et sa fonction d'autocorrélation

Avec les commandes : `library(stats)`, `ts.arma<- arima.sim(list(order = c(1,0,1), ar = 0.8,ma=0.6), n = 200)`, `ts.plot(ts.arma,type="l")`, `acf(ts.arma,10,"correlation")` on obtient :

6 Estimation

L'estimation des paramètres d'un processus $ARMA(p,q)$ peut se faire par plusieurs méthodes.

Critères d'information pour un $ARMA(p,q)$

$$\mathbf{Akaike} \text{ (AIC)} : AIC(p, q) = \ln(\hat{\sigma}^2) + 2(p + q)/T,$$

$$\mathbf{Schwarz} \text{ (BIC)} : BIC(p, q) = \ln(\hat{\sigma}^2) + 2(p + q) \ln(T)/T,$$

$\hat{\sigma}^2$ est un estimateur de $\sigma^2 = \text{var}(u_t)$.

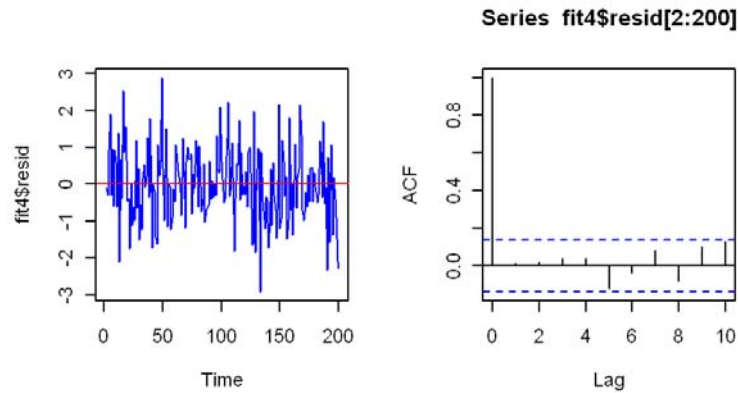


FIG. 2.4 –

6.1 Estimation d'un $AR(p)$

Sur R il existe quatre méthode d'estimation utilisant le critère d'information AIC pour déterminer un choix optimal pour le retard p .

Usage : `ar(x, aic = TRUE, order.max = NULL, method=c("yule-walker", "burg", "ols", "mle", "yw"))`

Avec la commande : `fit4 <- ar(ts.ar,aic=TRUE,"burg")` on obtient

Call :

`ar(x = ts.ar, aic = TRUE, method = "burg")`

Coefficients :

1

0.6872

Order selected 1 sigma^2 estimated as 1.030

Interprétation :

Le modèle optimal sélectionné est un $AR(1)$ avec $\hat{a}_1 = 0.6872$.

Analyse des résidus

Avec le code :

`ts.plot(fit4$resid,type="l",col="blue"), abline(0,0,col="red"),
acf(fit4$resid[2 :200],10, "correlation")` on obtient

Avec la commande : `Box.test(fit4$resid, lag = 10, type = "Box-Pierce")` on obtient :

Box-Pierce test

data : fit4\$resid

X-squared = 11.9391, df = 10, p-value = 0.2892

Donc les résidus ne sont pas autocorrélés.

Avec les commandes : `library(nortest)`, `ad.test(fit4$resid)` on obtient

Anderson-Darling normality test

data : fit4\$resid

A = 0.2462, p-value = 0.754

Les résidus sont normaux.

La procédure `ar.burg` fournit 2 méthodes pour estimer la variance des innovations et donc le critère AIC. La méthode 1 utilise la mise à jours récursive Levinson-Durbin (Brockwell et Davis, 1991, (8.2.6), page 242). La méthode 2 calcule la moyenne de la somme des carrés des erreurs de prévisions en avant et en arrière (comme dans Brockwell et Davis, 1996, page 145).

6.2 Estimation d'un $ARMA(p)$

Usage : `library(fArma)`, `armaFit(formula, data, method = c("mle", "ols"), include.mean = TRUE, fixed = NULL, title = NULL, description = NULL, ...)`

Exemple : Simulation et estimation d'un $ARMA(2,1)$

Avec le code : `library(fArma)`, `x = armaSim(model = list(ar = c(0.5, -0.5), ma = 0.1), n = 1000)`, `fit5 = armaFit(~arma(2, 1), data = x)`, `print(fit5)`, `par(mfrow = c(2, 2), cex = 0.7)`, `summary(fit5, which = "all")` on obtient

Title :

ARIMA Modelling

Call :

armaFit(formula = ~arma(2, 1), data = x)

Model :

ARIMA(2,0,1) with method : CSS-ML

Coefficient(s) :

<i>ar1</i>	<i>ar2</i>	<i>ma1</i>	<i>intercept</i>
0.5173281	-0.5283857	0.0838513	0.0003677

Residuals :

<i>Min</i>	<i>1Q</i>	<i>Median</i>	<i>3Q</i>	<i>Max</i>
-2.833856	-0.668508	-0.001849	0.724785	2.923971

Moments :

<i>Skewness</i>	<i>Kurtosis</i>
-0.06628	-0.24356

Coefficient(s) :

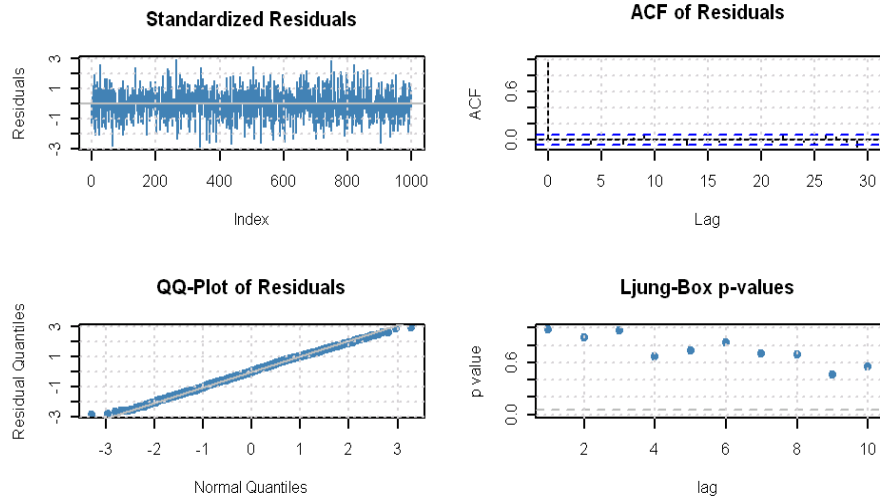
	<i>Estimate</i>	<i>Std. Error</i>	<i>t value</i>	<i>Pr(> t)</i>
<i>ar1</i>	0.5173281	0.0503452	10.276	<2e-16 ***
<i>ar2</i>	-0.5283857	0.0320337	-16.495	<2e-16 ***
<i>ma1</i>	0.0838513	0.0602138	1.393	0.164
<i>intercept</i>	0.0003677	0.0339829	0.011	0.991

*Signif. codes : 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

sigma^2 estimated as : 1.004

log likelihood : -1421.37

AIC Criterion : 2852.75



Le test QQ-PLOT de normalité est une méthode graphique : Le nuage de point est formé par (quantiles de $N(0,1)$, quantiles empiriques réduits de \hat{u}_t), sous H_0 le nuage est rectiligne sur la droite $y=x$.

Skewness mesure l'asymétrie de la distribution du processus et *Kurtosis* mesure l'excès du Kurtosis : il est égal à $3 - \widehat{KU}$ et

$$\widehat{SK} = \frac{\frac{1}{T} \sum_{t=1}^T (\hat{u}_t - \bar{u})^3}{\left(\frac{1}{T} \sum_{t=1}^T (\hat{u}_t - \bar{u})^2 \right)^{3/2}}, \text{ estimateur du Skewness .}$$

$$\widehat{KU} = \frac{\frac{1}{T} \sum_{t=1}^T (\hat{u}_t - \bar{u})^4}{\left(\frac{1}{T} \sum_{t=1}^T (\hat{u}_t - \bar{u})^2 \right)^2}, \text{ estimateur du Kurtosis,}$$

6.3 Prédiction d'un ARMA(p,q)

6.3.1 Algorithme de Durbin-Levinson pour le calcul récursif de la prédiction

Soit $(X_n)_{n \in \mathbb{Z}}$, un processus au second ordre réel centré dont la fonction d'autocovariance est noté $K(i, j) = E(X_i X_j)$. Pour tout $n \geq 1$, on note $H_n = [X_1, \dots, X_n]$ le sous espace vectoriel de $L^2(\Omega, \mathcal{Q}, \mathbb{P})$, engendré par $(X_j)_{1 \leq j \leq n}$; on pose $\hat{X} = 0$, et

$\hat{X}_j = P_{H_{j-1}}(X_j)$; on suppose de plus que la matrice $(K(i, j))_{1 \leq i, j \leq n}$ est définie positive.

1) On montre que $H_n = [(X_1 - \hat{X}_1), \dots, (X_n - \hat{X}_n)]$; et on déduit que :

$$\hat{X}_{n+1} = \sum_{j=1}^n \theta_{n,j} (X_{n+1-j} - \hat{X}_{n+1-j}) \quad (\text{écriture unique}), n \geq 1.$$

2) Si on pose $v_k = E \left(X_{k+1} - \hat{X}_{k+1} \right)^2$; on peut calculer récursivement $(\theta_{n,j}, j = 1, 2, \dots, n; v_n)$.

a) On a

$$\langle \hat{X}_{n+1}, X_{k+1} - \hat{X}_{k+1} \rangle = \theta_{n,n-k} v_k, \quad 0 \leq k < n.$$

b) On déduit l'algorithme :

$$\begin{cases} v_0 = K(1, 1), \\ \theta_{n,n-k} = v_k^{-1} \left(K(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j} \theta_{n,n-j} v_j \right), k = 0, 1, \dots, n-1, \\ v_n = K(n+1, n+1) - \sum_{j=0}^{n-1} \theta_{n,n-j}^2 v_j. \end{cases}$$

Intervalle de confiance pour la prévision d'horizon h

On écrit (X_t) sous forme d'une moyenne mobile infinie :

$$X_t = \varepsilon_t + \sum_{j=1}^{\infty} b_j \varepsilon_{t-j},$$

On déduit alors l'intervalle de confiance au niveau $\alpha = 0.05$ pour X_{t+h} ;

$$X_{t+h} \in [\hat{X}_{t+h/t} - 1.96\sigma_\varepsilon \left(\sum_{j=0}^{h-1} b_j^2 \right)^{1/2}, \hat{X}_{t+h/t} + 1.96\sigma_\varepsilon \left(\sum_{j=0}^{h-1} b_j^2 \right)^{1/2}].$$

Avec le code : `predict(fit5, 10)` on obtient

`$pred`

`Time Series :`

Start = 1001

End = 1010

Frequency = 1

[1] -0.0861930481 -0.0130914091 0.0391423367 0.0275384597 -0.0060641168

[6] -0.0173163510 -0.0053823268 0.0067369991 0.0067008990 0.0002785449

\$se

Time Series :

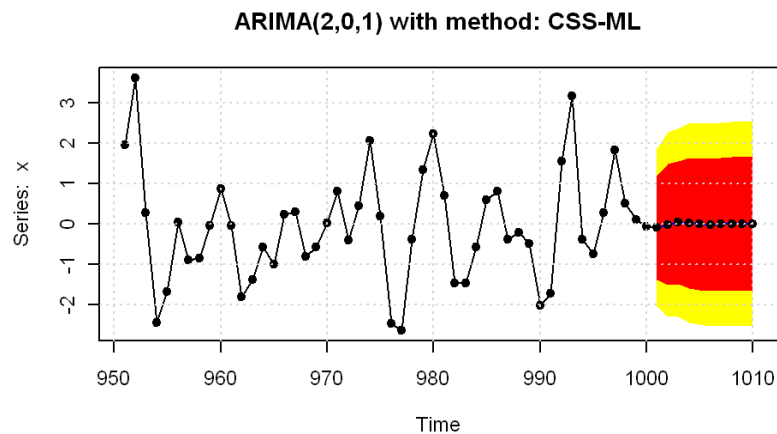
Start = 1001

End = 1010

Frequency = 1

[1] 1.001998 1.169129 1.189246 1.264928 1.269518 1.281106 1.289390 1.289482

[9] 1.292283 1.292782



Prévisions et intervalle de confiance à 80% en rouge et 90% en jaune.

6.3.2 Mesure de la qualité de prévision

Dans l'échantillon initial, (X_1, \dots, X_T) on considère seulement, $T_1 = [(1 - \varepsilon)T]$ observations avec $\varepsilon > 0$. Les $L = T - [(1 - \varepsilon)T]$ seront à prévoir par le modèle. On peut alors considérer plusieurs critères :

1. Mean Absolute Pourcentage Error

$$MAPE = \frac{1}{L} \sum_{r=1}^L \left| \frac{X_{T_1+r} - \hat{X}_{T_1+r/T_1}}{X_{T_1+r}} \right|$$

2. Mean Square Error

$$MSE = \left(\sum_{r=1}^L \frac{(X_{T_1+r} - \hat{X}_{T_1+r/T_1})^2}{L} \right)^{1/2}.$$

Exemple : Simulation et estimation d'un ARMA(1,1)

Avec le code : `library(fArma)`, `x = armaSim(model = list(ar = 0.5, ma = 0.7), n = 1000)`,

`Eaprenti <- x[1 :980]`, `Etemoin <- x[981 :1000]`, `fit6 = armaFit(~arma(2, 1), data = Eaprenti)`, `summary(fit6, which = "all")` on obtient

Title :

ARIMA Modelling

Call :

`armaFit(formula = ~arma(1, 1), data = Eaprenti)`

Model :

ARIMA(1,0,1) with method : CSS-ML

Coefficient(s) :

ar1 ma1 intercept

0.4746 0.6914 0.1483

Residuals :

<i>Min</i>	<i>1Q</i>	<i>Median</i>	<i>3Q</i>	<i>Max</i>
<i>-4.06563</i>	<i>-0.64837</i>	<i>0.01261</i>	<i>0.65807</i>	<i>2.79039</i>

Moments :

Skewness Kurtosis

-0.100923 -0.007662

Coefficient(s) :

<i>Estimate</i>	<i>Std. Error</i>	<i>t value</i>	<i>Pr(> t)</i>
-----------------	-------------------	----------------	--------------------

```

ar1      0.47457      0.03209      14.788      <2e-16 ***
ma1      0.69141      0.02665      25.943      <2e-16 ***
intercept 0.14829      0.10235      1.449      0.147

```

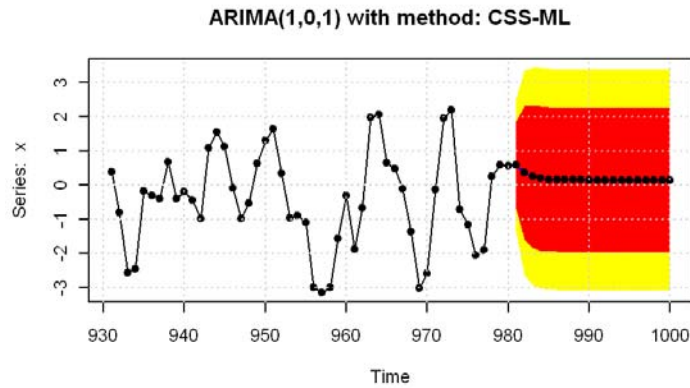
Signif. codes : 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

σ^2 estimated as : 0.9933

log likelihood : -1388.02

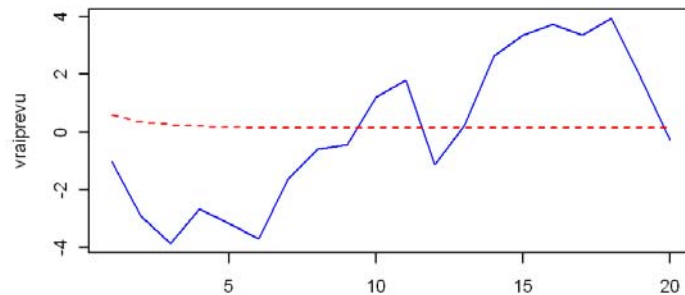
AIC Criterion : 2784.04

Avec le code : `prev <- predict(fit6, n.ahead = 20, n.back = 50, conf = c(80, 95))` on obtient



Prévisions et intervalle de confiance à 80% en rouge et 95% en jaune.

Avec le code : `vraiprevu <- matrix(rep(0,40),20,2)`, `vraiprevu[,1] <- Etemoin`, `vraiprevu[,2],`
`prev$pred`, `matplot(vraiprevu,type="l", col=c("blue","red"))` on obtient



Les vraies observations en bleu et leurs prévisions en rouge.

Avec le code : $map = Etemoin\text{-}prev\$pred,$

$map = map * (1/Etemoin),$

$MAPE = sum(abs(map))/20$

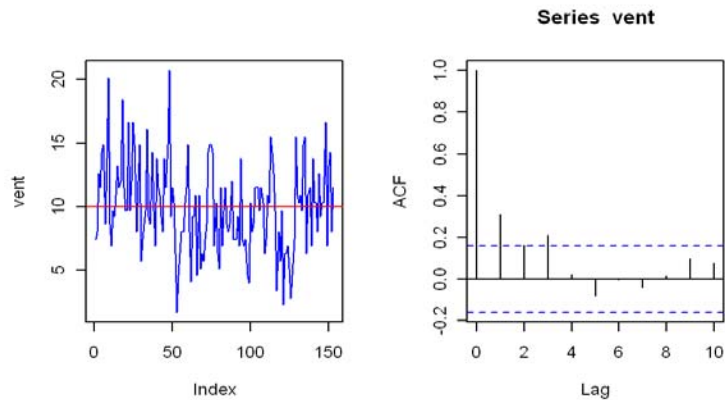
on obtient MAPE=1.057

Avec le code : $ms = Etemoin\text{-}prev\$pred, MSE = sqrt(sum(ms^2)/20)$

on obtient MSE=2.56.

Pratiquement les ordres p et q sont à estimer, on peut se baser sur les critères d'information pour sélectionner un couple (p,q) optimal qui minimise le critère.

Exemple : On considère la série du vent dans les données (airquality) :



On peut donc modéliser la série par une MA(3) mais aussi un AR ou un ARMA.

En prenant $pmax=4, qmax=4$ et en utilisant le critère AIC on obtient le modèle optimal suivant

Call :

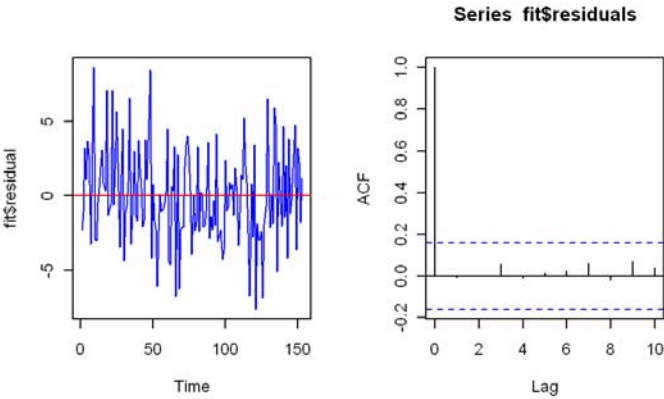
$arima0(x = x, order = c(p, 0, q), method = "ML")$

Coefficients :

<i>ma1</i>	<i>ma2</i>	<i>ma3</i>	<i>intercept</i>
0.2585	0.1317	0.2229	9.9572
s.e.	0.0789	0.0753	0.0789 0.4248

σ^2 estimated as 10.71 : log likelihood = -398.59, aic = 807.18

Les résidus ainsi que leur fonction d'autocorrélation sont donnés par



Chapitre 3

Processus non stationnaires

Il existe plusieurs processus non stationnaires au second ordre, nous nous contentons de présenter deux structures de modèles qui sont plus proche des modèles A.R.M.A.

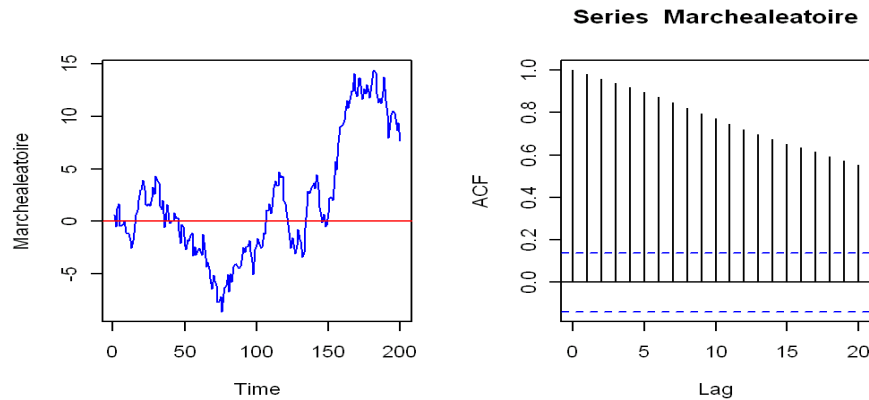
1 Modèles ARIMA.

Remarque : ARIMA signifie : Integrated-AutoRegressive-Moving-Average.

Definition 6 Nous dirons que $X = (X_t)$ est un processus ARIMA(p, d, q), s'il existe un entier naturel d tel que le processus $Y_t = (1 - B)^d X_t$ est un ARMA(p, q). Autrement dit il existe deux polynômes $A(z)$ et $C(z)$ de degré p et q respectivement et un bruit blanc (ε_n) défini sur le même (Ω, \mathcal{Q}, P) tels que

$$(1 - B)^d A(B)X_t = C(B)\varepsilon_t \quad (3.1)$$

Avec le code : `e <- rnorm(200)`, `Marchealeatoire <- cumsum(e)`,
`ts.plot(Marchealeatoire, type="l", col="blue")`, `abline(0,0,col="red")`,
`acf(Marchealeatoire, 20, "correlation")` on obtient :



Trajectoire de $X_t = X_{t-1} + \varepsilon_t$ et sa fonction d'autocorrélation.

1.1 MODELISATION ARIMA

1.2 Etapes de la modélisation

1.2.1 Stationnarisation par différentiation.

- L'opérateur de différentiation $\Delta = 1 - B$, (ou Δ^d) élimine les tendances, d est estimé en effectuant des tests de stationnarité sur la série brute puis sur les séries résiduelles. Un estimateur de d est le nombre total de fois où on rejette la stationnarité.

La série résiduelle supposée stationnaire sera modélisée par un $\text{ARMA}(p, q)$.

1.2.2 Estimation

- R offre la possibilité d'utiliser deux méthodes : 1) la méthode de maximum de vraisemblance "ML" et la méthode des moindres carrés conditionnels "CSS"

1.2.3 Validation

- Le modèle retenu est-il compatible? : on effectue les tests d'adéquation sur les résidus : absence d'autocorrélation, normalité, homoscédasticité,...

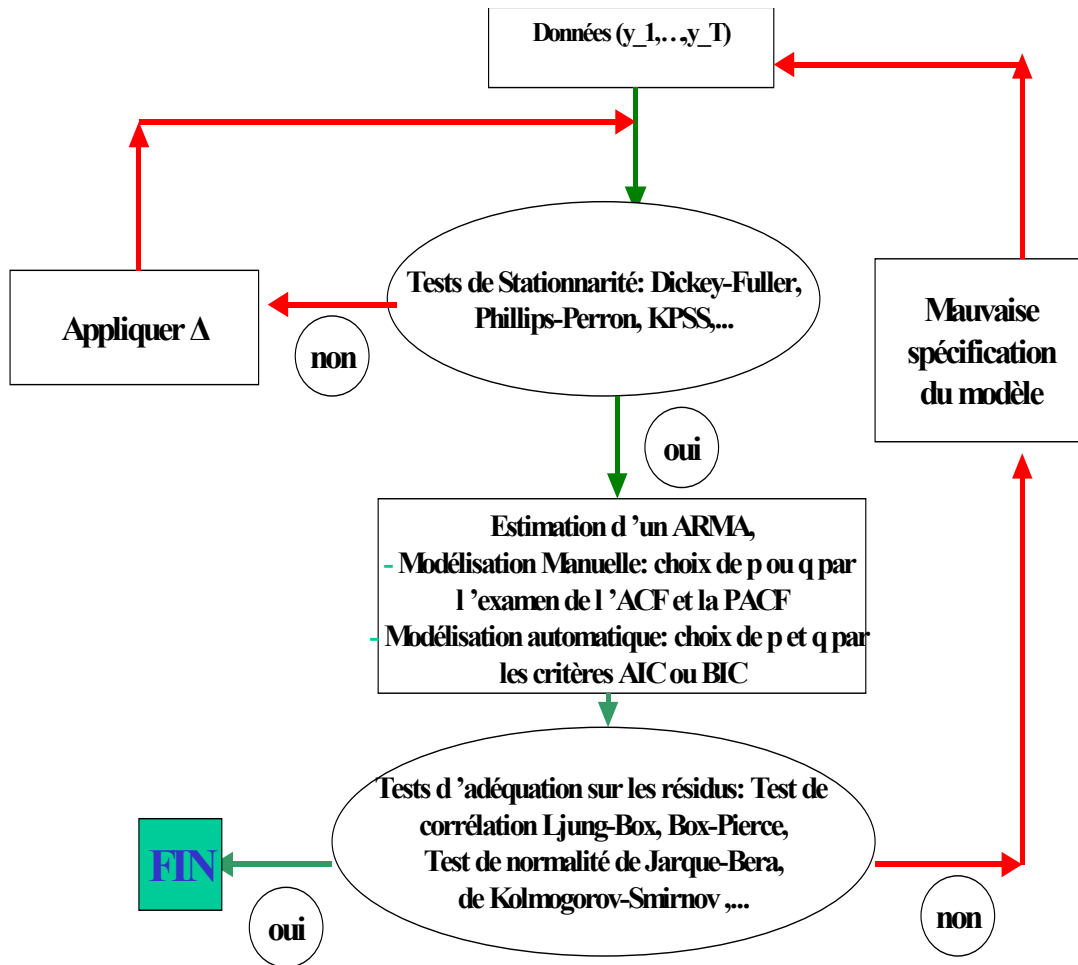


Schéma général de la modélisation d'une série temporelle par un modèle ARIMA.

1.3 Test de non-stationnarité

1.3.1 Test de Dickey et Fuller augmenté :

Robuste à l'autocorrélation par rapport au test de Dickey-Fuller

$$\begin{cases} H_0 : y_t \sim I(1) \\ H_1 : y_t \text{ n'est pas } I(1) \end{cases}$$

On exécute la régression

$$\Delta y_t = \beta' D_t + \pi y_{t-1} + \sum_{j=1}^p \psi_j \Delta y_{t_j} + u_t,$$

$$ADF = T\hat{\pi}/(1 - \hat{\psi}_1 - \dots - \hat{\psi}_p).$$

1.3.2 Test de Phillips-Perron

Robuste à l'hétéroscédasticité

On effectue la régression :

$$\Delta y_t = \beta' D_t + \pi y_{t-1} + u_t,$$

$$PP = T\hat{\pi} - \frac{1}{2}T^2 \frac{SE(\hat{\pi})}{\hat{\sigma}^2} (\hat{\lambda}^2 - \hat{\sigma}^2),$$

$$\hat{\sigma}^2 = \frac{1}{T-k} \sum_{t=1}^T \hat{u}_t^2 : \text{résidus de la régression},$$

$$SE(\hat{\pi}) = \text{écart-type de } \hat{\pi},$$

$$\hat{\lambda}^2 = \hat{\sigma}^2 + 2 \sum_{j=1}^q (1 - \frac{j}{q+1}) \hat{\gamma}(j), (Newey - West),$$

$$\hat{\gamma}(j) = \frac{1}{T} \sum_{t=j+1}^T \hat{u}_t \hat{u}_{t-j}.$$

Valeurs critiques de ADF et PP : $P(ADF < c) = \alpha$

• $D_t = 0$ (processus centré)

α	T	25	50	100	250	500	∞
0.025		-9.3	-9.9	-10.2	-10.3	-10.4	-10.5
0.05		-7.3	-7.7	-7.9	-8	-8	-8.1
0.95		1.40	1.35	1.31	1.28	1.28	1.28
0.975		1.79	1.70	1.65	1.62	1.61	1.60

• $D_t = 1$ (processus avec drift)

α	T	25	50	100	250	500	∞
0.025		-14.6	-15.7	-16.3	-16.6	-16.8	-16.9
0.05		-12.5	-13.3	-13.7	-14	-14	-14.1
0.95		0.01	-0.07	-0.10	-0.12	-0.13	-0.13
0.975		0.65	0.53	0.47	0.43	0.42	0.41

• $D_t = (1, t)'$ (processus avec tendance)

α	T	25	50	100	250	500	∞
0.025		-19.9	-22.4	-23.6	-24.6	-24.8	-25.1
0.05		-17.9	-19.8	-20.7	-21.3	-21.5	-21.8
0.95		-2.51	-2.60	-2.62	-2.64	-2.65	-2.66
0.975		-1.53	-1.66	-1.73	-1.78	-1.78	-1.79

1.4 Tests de stationnarité

1.4.1 Test de KPSS

Modèle

$$y_t = \beta' D_t + \mu_t + u_t, \mu_t = \mu_{t-1} + \varepsilon_t, \text{var}(\varepsilon_t) = \sigma_\varepsilon^2.$$

$$\begin{cases} H_0 : \sigma_\varepsilon^2 = 0 \\ H_1 : \sigma_\varepsilon^2 \neq 0 \end{cases}$$

$$KPSS = \left(T^{-2} \sum_{t=1}^T \hat{S}_t^2 \right) / \hat{\lambda}^2.$$

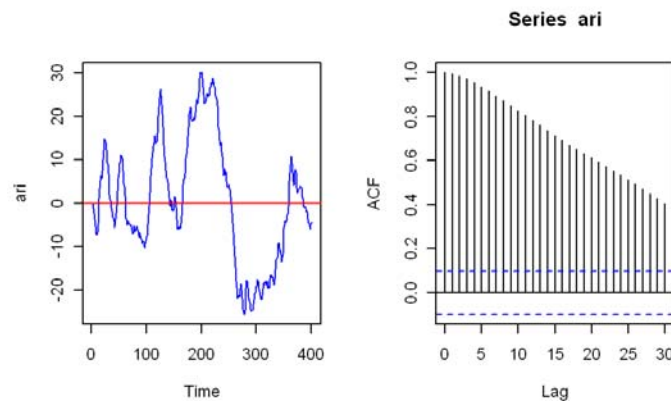
$\hat{S}_t = \sum_{j=1}^t \hat{u}_j, \hat{u}_t$: résidus de la régression de y_t sur D_t .

Valeurs critiques du KPSS : $P(KPSS > c) = \alpha$

D_t	α	0.10	0.05	0.025	0.01
$D_t = 1$		0.347	0.436	0.574	0.739
$D_t = (1, t)'$		0.119	0.146	0.176	0.216

Exemple : Simulation et estimation d'un ARIMA :

1. **Simulation** : `ari <- arima.sim(list(order = c(1,1,0), ar = 0.7), n = 400)`.



2. Tests de Stationnarité :

- ADF : avec le code : `library(tseries), adf.test(ari)` on obtient :

Augmented Dickey-Fuller Test

data : ari

Dickey-Fuller = -2.4035, Lag order = 7, p-value = 0.4068

alternative hypothesis : stationary

On ne rejette pas l'hypothèse nulle d'une racine unitaire.

- PP : avec le code : `library(tseries), pp.test(ari)` on obtient :

Phillips-Perron Unit Root Test

data : ari

Dickey-Fuller Z(alpha) = -6.1984, Truncation lag parameter = 5, p-value = 0.7634

alternative hypothesis : stationary

La série contient une racine unitaire.

- KPSS : avec le code : `library(tseries), kpss.test(ari)` on obtient :

KPSS Test for Level Stationarity

data : ari

KPSS Level = 1.3669, Truncation lag parameter = 4, p-value = 0.01

Warning message :

In kpss.test(ari) : p-value smaller than printed p-value

On rejette l'hypothèse nulle de stationnarité.

Conclusion : Tous les tests confirment l'existence d'une racine unitaire :

3. Stationnarisation : dari <- diff(ari).

4. Tests de Stationnarité de la série différenciée :

●ADF : avec le code : *library(tseries), adf.test(dari)* on obtient :

Augmented Dickey-Fuller Test

data : dari

Dickey-Fuller = -5.8272, Lag order = 7, p-value = 0.01

alternative hypothesis : stationary

Warning message :

In adf.test(dari) : p-value smaller than printed p-value

donc la série différenciée est stationnaire.

●KPSS : avec le code : *library(tseries), kpss.test(ari)* on obtient :

KPSS Test for Level Stationarity

data : dari

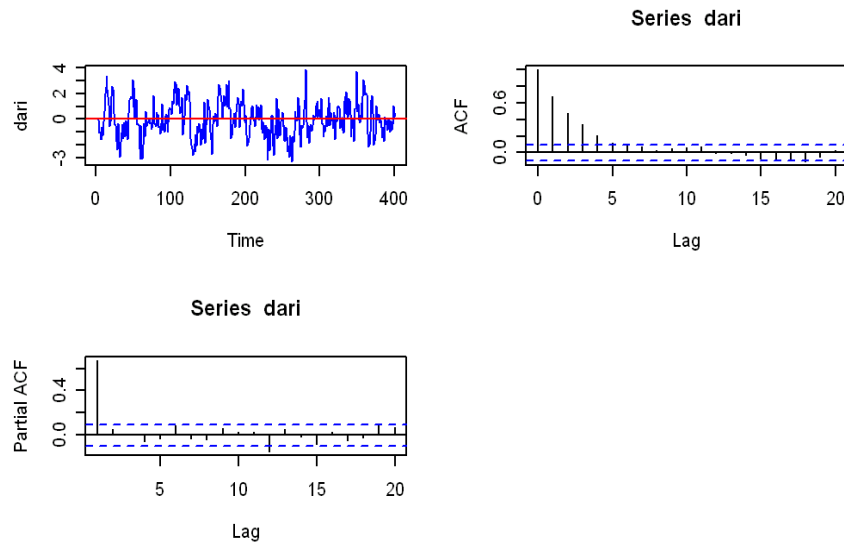
KPSS Level = 0.0976, Truncation lag parameter = 4, p-value = 0.1

Warning message :

In kpss.test(dari) : p-value greater than printed p-value

On ne rejette pas l'hypothèse nulle de stationnarité.

La série différenciée est stationnaire, on l'a modélise alors par un ARMA.



L'examen de la corrélation partielle nous indique bien que la série peut être modélisable par un AR(1).

5. Estimation :

Avec le code : `fit7 <- arima0(ari, order = c(1, 1, 0), method = "ML")`, `fit7` on obtient

Call :

`arima0(x = ari, order = c(1, 1, 0), method = "ML")`

Coefficients :

`ar1`

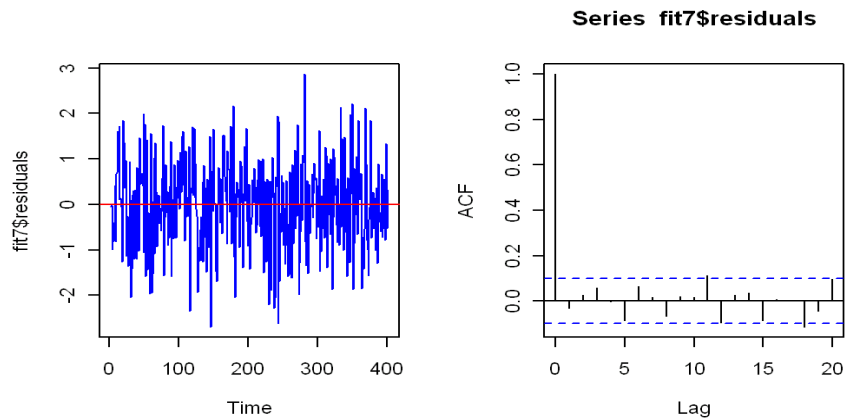
`0.6714`

s.e. 0.0368

sigma^2 estimated as 0.9678 : log likelihood = -561.32, aic = 1126.64

Remarque : L'estimation du modèle paraît de bonne qualité, car l'estimateur de a_1 est très proche de la vraie valeur.

6. Tests d'adéquations :



Avec le code : `Box.test(fit7$residuals, lag = 10, type = "Box-Pierce")` on obtient :
Box-Pierce test

data : fit7\$residuals

X-squared = 8.728, df = 10, p-value = 0.5581

La p-value est grande donc les résidus ne sont pas corrélés.

Avec le code : `library(nortest), lillie.test(fit7$residuals)` on obtient

Lilliefors (Kolmogorov-Smirnov) normality test

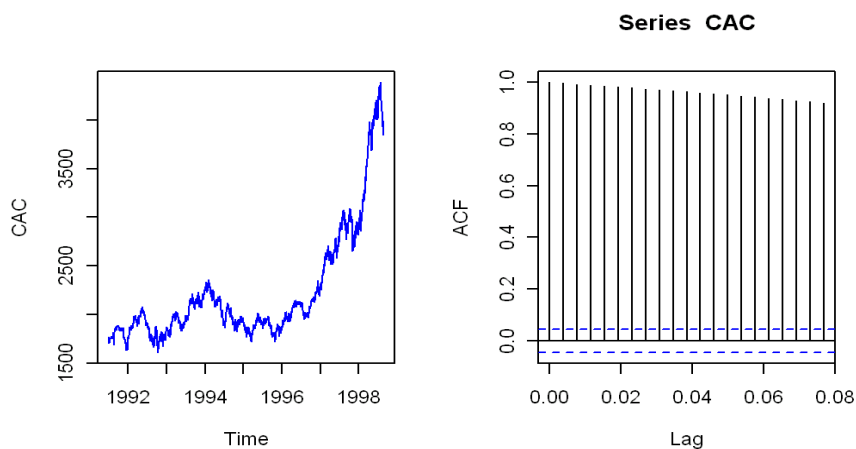
data : fit7\$residuals

$D = 0.0407$, p-value = 0.1114

Donc les données suivent une loi de Gauss.

Exemple 2 : Modélisation des indices boursiers (bourse de Paris). On considère les prix de clôture de l'indice boursier CAC40 sur la période 1991-1998 (1860 observations).

Avec le code `data(EuStockMarkets), CAC <- EuStockMarkets[,3], ts.plot(CAC, type="l", col="blue"), acf(CAC,20,"correlation")` on obtient



2. Tests de Stationnarité :

●ADF : avec le code : `library(tseries), adf.test(CAC)` on obtient :

Augmented Dickey-Fuller Test

data : CAC

Dickey-Fuller = -0.249, Lag order = 12, p-value = 0.99

alternative hypothesis : stationary

Warning message :

In adf.test(CAC) : p-value greater than printed p-value

On ne rejette pas l'hypothèse nulle d'une racine unitaire.

3. Stationnarisation : `dCAC<- diff(CAC)`.

4. Tests de Stationnarité de la série différenciée :

●ADF : avec le code : `library(tseries), adf.test(dCAC)` on obtient :

Augmented Dickey-Fuller Test

data : dCAC

Dickey-Fuller = -11.4471, Lag order = 12, p-value = 0.01

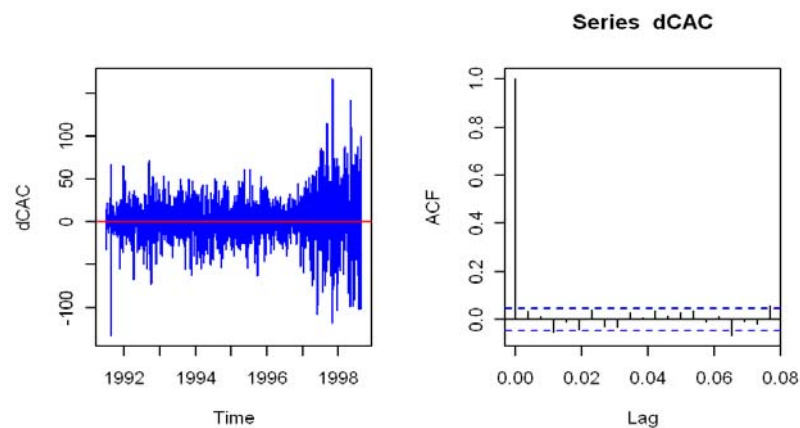
alternative hypothesis : stationary

Warning message :

In adf.test(dCAC) : p-value smaller than printed p-value

La série différenciée est donc stationnaire :

5. Estimation



La fonction d'autocorrélation nous indique que la série différenciée est un bruit blanc.

Test d'autocorrélation :

Avec le code : `Box.test(dCAC, lag = 10, type = "Box-Pierce")` on obtient :

Box-Pierce test

data : dCAC

X-squared = 19.9932, df = 10, p-value = 0.02932

Avec 1% de risque on ne rejette pas l'hypothèse que les accroissements du CAC40 est un bruit blanc.

Test de normalité :

Avec le code : `library(nortest), pearson.test(dCAC)` on obtient

Pearson chi-square normality test

data : dCAC

$P = 293.1802$, $p\text{-value} < 2.2e-16$

La normalité est donc rejetée

Conclusion : Le CAC40 suit une marche aléatoire ce qui se concorde avec la théorie de l'efféience des marchés, cependant les accroissements ne sont pas normaux avec des régimes sur la variance (on constate des périodes calmes et d'autres agitées), ceci peut se résoudre de deux manières : i) utilisation des distributions avec des queues

épaisses (Student, GED,...) ii) utilisation de l'approche d'Engel (1982) en modélisant les accroissement par un modèle ARCH (modèles Auto Régressifs Conditionnellement Hétéroscédastiques).

2 Modèles SARIMA

Les modèles SARIMA " Seasonal-Integrated-AutoRegressive-Moving-Average" sont utilisés pour modéliser les données saisonnières (mensuelles, trimestrielles) ou cycliques, les portions de trajectoires obtenues par la translation $Y_n \longrightarrow Y_{n+s}$ ont de fortes similarités.

Exemple :

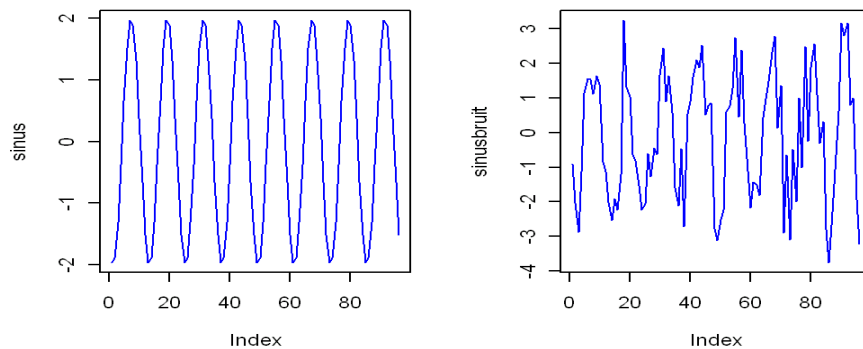
1) Soit le processus (X_t) définit par :

$$X_t = a \sin\left(\frac{\pi t}{6} + 4\right) + \varepsilon_t,$$

où (ε_t) est un bruit blanc. On a

$$E(X_t) = a \sin\left(\frac{\pi t}{6} + 4\right)$$

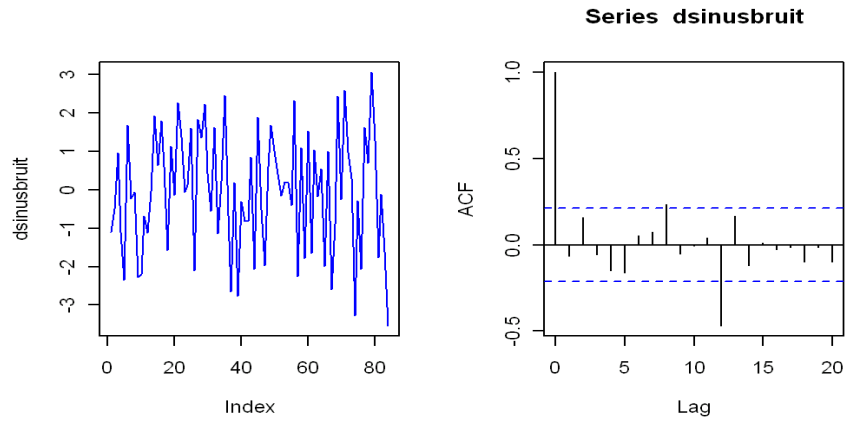
(X_t) n'est pas stationnaire en moyenne, la partie déterministe est périodique de période 12.



Le processus (Y_t) défini par $Y_t = (1 - B^{12})X_t = X_t - X_{t-12}$ vérifie : $Y_t = \varepsilon_t - \varepsilon_{t-12}$ c'est un MA(12), donc stationnaire au second ordre.

Avec le code :

```
dsinusbruit <- diff(sinusbruit, lag=12),
plot( dsinusbruit),
acf(dsinusbruit,20,"correlation")
```



Definition 7 Nous dirons que $X = (X_n)$ est un processus S.A.R.I.M.A. (p,d,s,q) , s'il vérifie l'équation :

$$(1 - B)^d (1 - B^s) A(B)X_n = C(B)\varepsilon_n \quad (3.2)$$

3 Modélisation par les modèles S.A.R.I.M.A.

Souvent on utilise une structure plus générale, on suppose que la série (ou le processus la modélisant) satisfait l'équation :

$$(1 - B)^d (1 - B^s)^D \phi(B)\Phi(B^s)X_t = \theta(B)\Theta(B^s)\varepsilon_t \quad (3.3)$$

p, P, d, s, D, q, Q sont des entiers naturels inconnus ; les polynômes $\phi(z), \Phi(z), \theta(z)$ et $\Theta(z)$ sont à coefficients inconnus.

$$\Phi(B^s) = 1 + \Phi_1 B^s + \dots + \Phi_P B^{sP}, \Theta(B^s) = 1 + \Theta_1 B^s + \dots + \Theta_Q B^{sQ}$$

Dans la structure (3.3), on appelle :

d : Différence,

s : saison,

D : différence saisonnière,

$\phi(z)$ la partie autorégressive, $\Phi(z)$ la partie autorégressive saisonnière.

$\theta(z)$ la partie moyenne-mobile., $\theta(z)$ la partie moyenne-mobile saisonnière.

Usage sur R : *arima0*(x , $order = c(p, d, q)$, $seasonal = list(order = c(P, D, Q), period = s)$, $xreg = NULL$, $include.mean = TRUE$, $delta = 0.01$, $transform.pars = TRUE$, $fixed = NULL$, $init = NULL$, $method = c("ML", "CSS")$, $n.cond$, $optim.control = list()$)

On utilise souvent $\Theta(B^s)$ si on applique aux données le filtre $(1 - B^s)^D$.

3.1 Modélisation :

Identique à celle de la modélisation ARIMA sauf la première étape de stationnarisation qui est remplacée par la suivante :

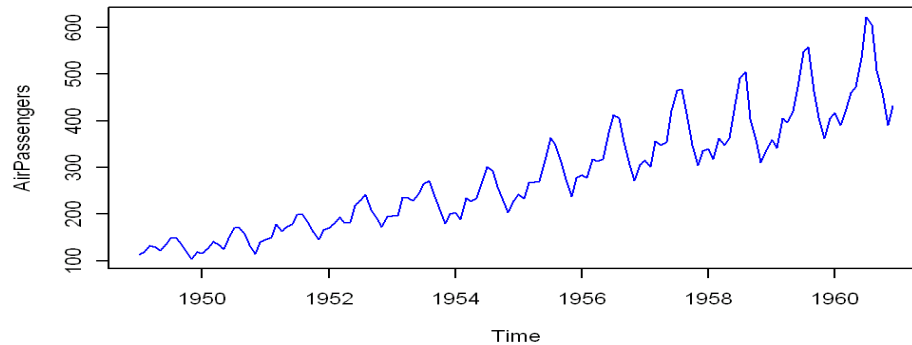
3.1.1 Stationnarisation par filtrage

- L'opérateur $(1 - T^s)^D$, élimine la périodicité (désaisonnalisation).

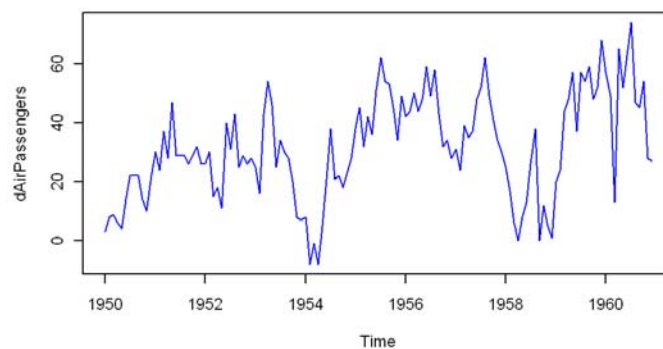
- L'opérateur de différentiation $\Delta = 1 - B$, (ou Δ^d) élimine les tendances, d est estimé en effectuant des tests de stationnarité sur la série brute puis sur les séries résiduelles. Un estimateur de d est le nombre total de fois où on rejette la stationnarité.

Ses deux opérations rendent la série stationnaire et donne une estimation des entiers d, s, D .

Exemple : Modélisation de la série des voyageurs :



Avec le code : `dAirPassengers <- diff(AirPassengers, lag=12)` on obtient



L'effet saisonnier semble disparaître.

Avec le code : `library(tseries), adf.test(dCAC)` on obtient :

Augmented Dickey-Fuller Test

data : dAirPassengers

Dickey-Fuller = -3.1519, Lag order = 5, p-value = 0.09899

alternative hypothesis : stationary

Donc il y a une racine unitaire.

Avec le code : `fit8 <- arima0(log(AirPassengers),c(0,1,1),seasonal = list(order=c(0,1,1), period=12))` on obtient

Call :

`arima0(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))`

Coefficients :

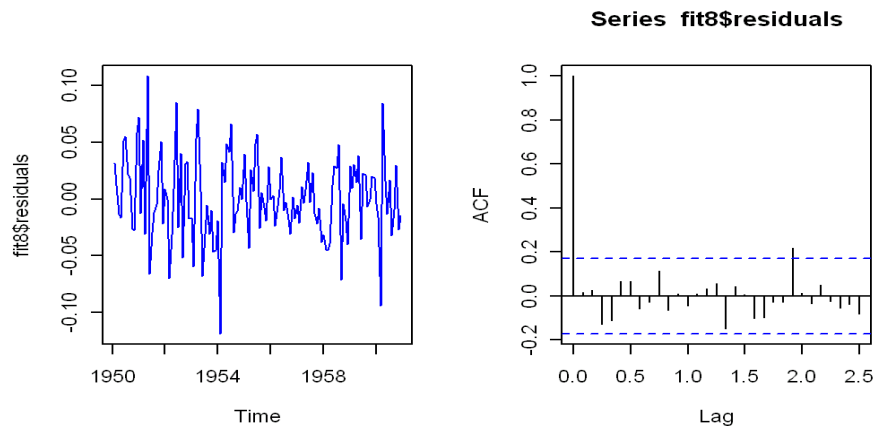
<i>ma1</i>	<i>sma1</i>
-0.4013	-0.5544
<i>s.e.</i>	0.0897 0.0713

σ^2 estimated as 0.00135 : log likelihood = 244.69, aic = -483.37

C'est à dire que le modèle proposé pour la série est donné par

$$(1 - B)(1 - B^{12}) \log(\text{AirPassengers}_t) = (1 - 0.4013B)(1 - 0.65544B^{12})\varepsilon_t$$

Les résidus sont donnés par



Tests d'adéquation :

1. Test d'autocorrélation :

Avec le code : `Box.test(fit8$residuals, lag = 10, type = "Box-Pierce")` on obtient :

Box-Pierce test

data : fit8\$residuals

X-squared = 7.8203, df = 10, p-value = 0.6464

Les résidus sont non corrélés

Test de normalité :

Avec le code : `library(nortest), sf.test(fit8$residuals)` on obtient

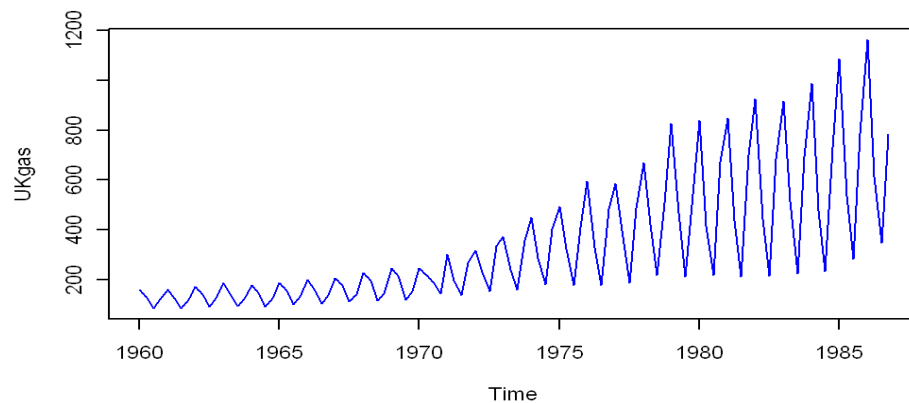
Shapiro-Francia normality test

data : fit8\$residuals

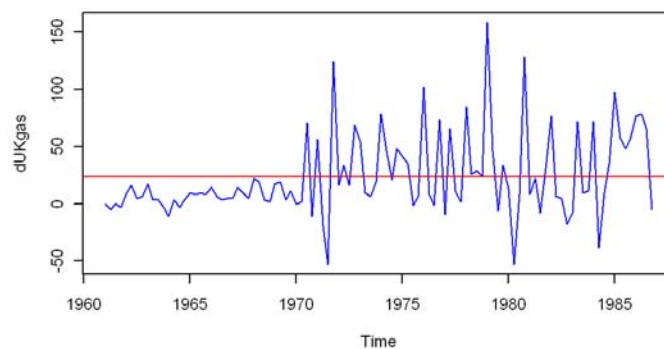
$W = 0.9886$, $p\text{-value} = 0.30036$

La normalité n'est pas rejetée.

Exemple 2 : On considère la consommation trimestrielle du gaz suivante :



Avec le code : `dUKgas <- diff(UKgas, lag=4)` on obtient



L'effet saisonnier semble disparaître.

Avec le code : `library(tseries), adf.test(dUKgas)` on obtient :

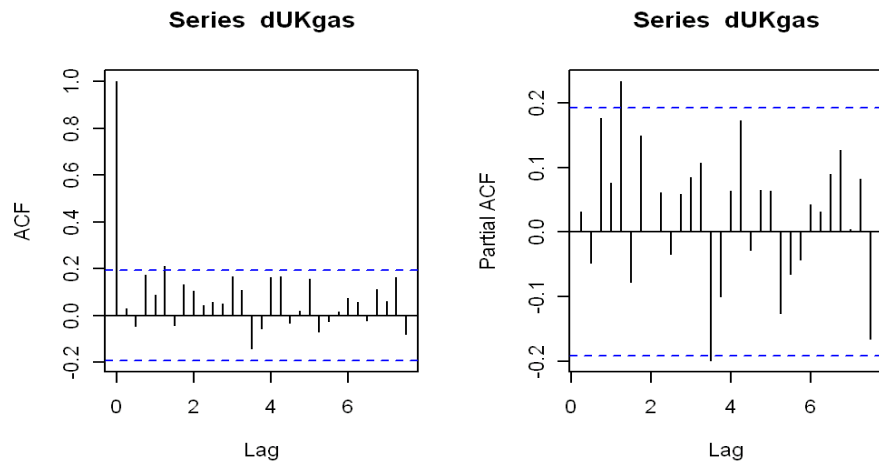
Augmented Dickey-Fuller Test

data : dUKgas

Dickey-Fuller = -3.7419, Lag order = 4, p-value = 0.02438

alternative hypothesis : stationary

Avec un risque de 5% on rejette la présence de racine unitaire.



Avec le code : `Box.test(dUKgas, lag = 24, type = "Box-Pierce")` on obtient :

1. Test d'autocorrélation :

Box-Pierce test

data : dUKgas

X-squared = 28.7167, df = 24, p-value = 0.2310

Les résidus sont non corrélés

2. Test de normalité :

Avec le code : `library(nortest), lillie.test(dUKgas)` on obtient

Lilliefors (Kolmogorov-Smirnov) normality test

data : dUKgas

D = 0.1886, p-value = 9.11e-10

La normalité est fortement rejetée. Donc le modèle pour la consommation du gaz est le suivant

$$(1 - B^4) UKgas_t = 23.19712 + \varepsilon_t$$

Chapitre 4

PROCESSUS VECTORIELS STATIONNAIRES AU SECOND ORDRE-MODELE VARMA

1 Définitions

Soit $X_t = (X_t^1, \dots, X_t^d)$ un processus vectoriel.

Definition 8 *On dit que (X_t) est au second ordre si*

$$E(||X_t||^2) < \infty \quad \forall t. \quad (4.1)$$

Definition 9 *On dit que (X_t) est stationnaire si*

- $E(X_t) = \mu \quad \forall t,$
- $\Gamma(j) = E(X_t - \mu)(X_{t-j} - \mu)' \quad \forall t.$

2 Modèle $VARMA_d(p, q)$

Le modèle AutoRégressif-Moyenne mobile vectoriel est donné par

$$X_t + A_1 X_{t-1} + \dots + A_p X_{t-p} = \varepsilon_t + \Theta_1 \varepsilon_{t-1} + \dots + \Theta_q \varepsilon_{t-q}. \quad (4.2)$$

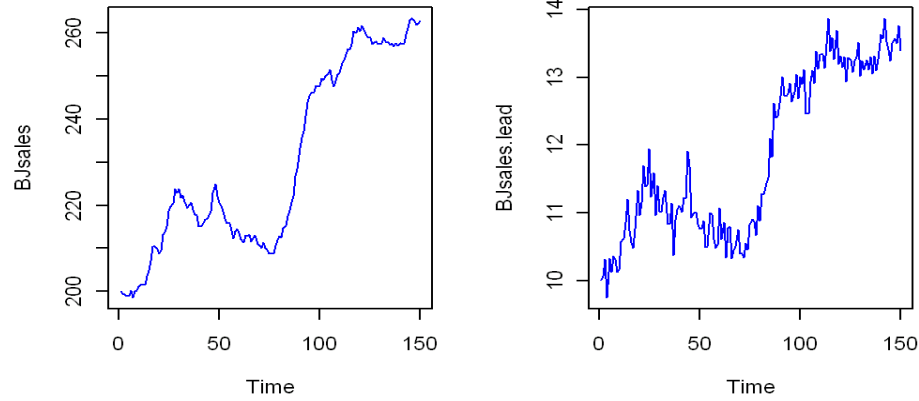
(ε_t) est un bruit blanc centré de variance Σ ($E(\varepsilon_t) = 0, E\varepsilon_t \varepsilon_s' = \delta_t^s \Sigma$), $A_i, 1 \leq i \leq p$, $\Theta_j, 1 \leq j \leq q$, sont des matrices (d, d) . Pour un modèle $MA_d(q)$ on a

$$\Gamma(j) = \begin{cases} \sum_{k=j}^q \Theta_k \Sigma \Theta_k' & \text{si } 0 \leq j \leq q, (\Theta_0 = \mathbf{I}_d) \\ \Gamma'(-j) & \text{si } -q \leq j \leq 0, \\ 0 & \text{si } |j| > q. \end{cases}$$

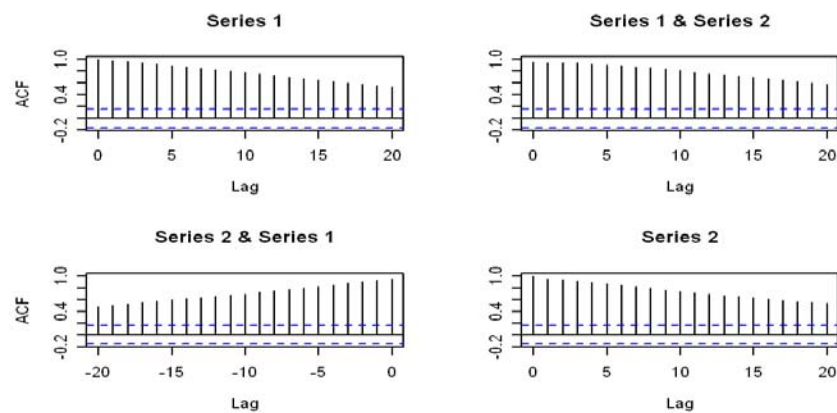
3 Estimation

Sur on peut utiliser la fonction *ar* basée sur le critère AIC,

Exemple :



```
z <- matrix(rep(0,1),150,2), z[,1] <- BJsales z[,2] <- BJsales.lead, acf(z,20,"correlation")
```

Les deux séries semblent être non stationnaires :

Avec le code : `adf.test(z[,1])`, `adf.test(z[,2])`

Augmented Dickey-Fuller Test

data : `z[, 1]`

Dickey-Fuller = -2.1109, Lag order = 5, p-value = 0.5302

alternative hypothesis : stationary

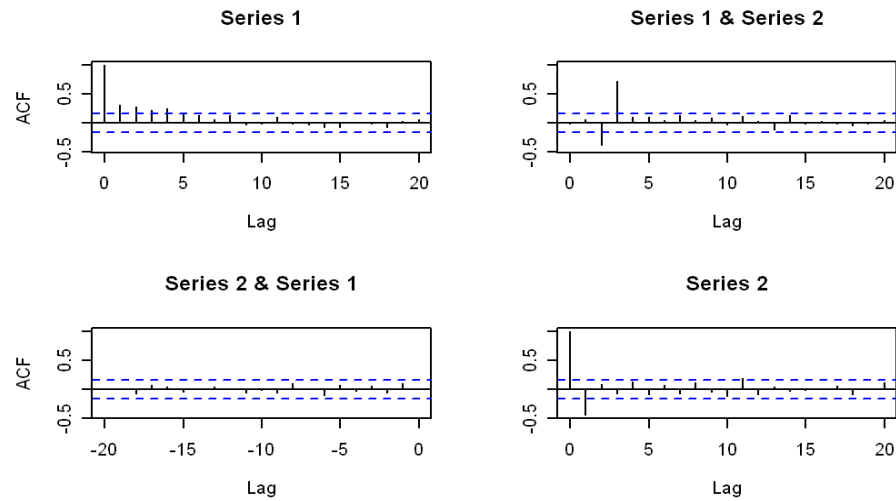
Augmented Dickey-Fuller Test

data : `z[, 2]`

Dickey-Fuller = -1.7237, Lag order = 5, p-value = 0.6915

alternative hypothesis : stationary

La présence de racine unitaire est confirmé par le test ADF.



La procédure ar donne un VAR(5) mais les résidus restent autocorrélés, en utilisant le code : `library(fAr) fit9 <- mAr.est(dz,11)` on obtient

\$SBC

[1] -2.111513

\$wHat

[1] 0.06205346 0.03740171

\$AHat

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] -0.55378253 -0.07236847 -0.30504871 -0.004809931 -0.20244218 4.73793144

[2,] 0.05922097 -0.44261244 -0.08387237 -0.161179807 -0.01015259 -0.06203615

[,7] [,8] [,9] [,10] [,11] [,12] [,13]

[1,] -0.14579083 6.1011179 0.08998048 5.7384537 0.1700602 5.1396008 0.0646940

[2,] -0.02939429 -0.1888549 -0.05528138 0.2052097 -0.1311495 0.2550947 0.1327873

[,14] [,15] [,16] [,17] [,18] [,19]

[1,] 4.3466291 0.13235399 2.8625170 -0.02347302 1.2685788 0.0005868455

[2,] 0.2764883 -0.04612804 0.5582377 0.04634413 0.9112916 -0.0011330641

[,20] [,21] [,22]

[1,] 0.56238921 -0.002787284 -0.1720188

[2,] -0.04332397 -0.010953376 0.3518515

\$CHat

[,1] [,2]

[1,] 0.04943240 -0.00257888

[2,] -0.00257888 0.08274237

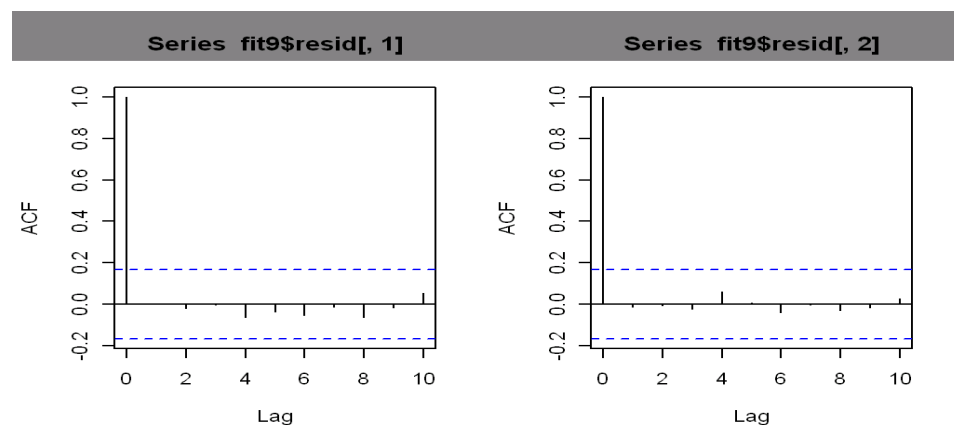
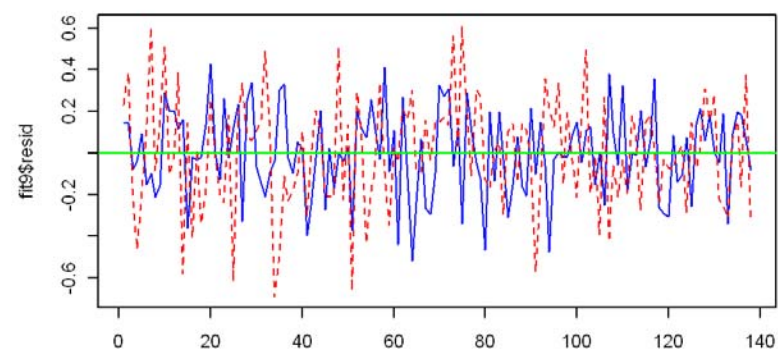
Interprétation

SBC : Schwartz Bayesian Criterion

wHat : vecteur des constantes

AHat : les matrices estimées.

Les résidus sont donnés par :



1. Test d'autocorrélation :

Avec le code : `Box.test(fit9$resid[,1], lag = 10, type = "Box-Pierce")` on obtient :

Box-Pierce test

data : fit9\$resid[, 1]

X-squared = 2.2772, df = 10, p-value = 0.9937

Avec le code : *Box.test (fit9\$resid[,2], lag = 10, type = "Box-Pierce")* on obtient :

Box-Pierce test

data : fit9\$resid[, 2]

X-squared = 1.0999, df = 10, p-value = 0.9997

Les résidus sont non corrélés

2. Test de normalité :

Avec le code : *library(nortest), lillie.test(fit9\$resid[,1])* on obtient

Lilliefors (Kolmogorov-Smirnov) normality test

data : fit9\$resid[, 1]

D = 0.0403, p-value = 0.8443

Avec le code : *library(nortest), lillie.test(fit9\$resid[,2])* on obtient

Lilliefors (Kolmogorov-Smirnov) normality test

data : fit9\$resid[, 2]

D = 0.0605, p-value = 0.2489

La normalité des résidus n'est pas rejetée

3.1 Densité spectrale croisée

Le spectre croisé $f_{x,y}(\lambda)$ est la transformée de Fourier de la fonction de covariance croisée, $\gamma_{x,y}(k)$, où

$$\gamma_{x,y}(k) = E(X_t - EX_t)(Y_{t+k} - EY_{t+k}).$$

Le cospectre est $c(\lambda) = \text{Re}(f_{x,y}(\lambda))$,

Le spectre quadratique est $q(\lambda) = \text{Im}(f_{x,y}(\lambda))$,

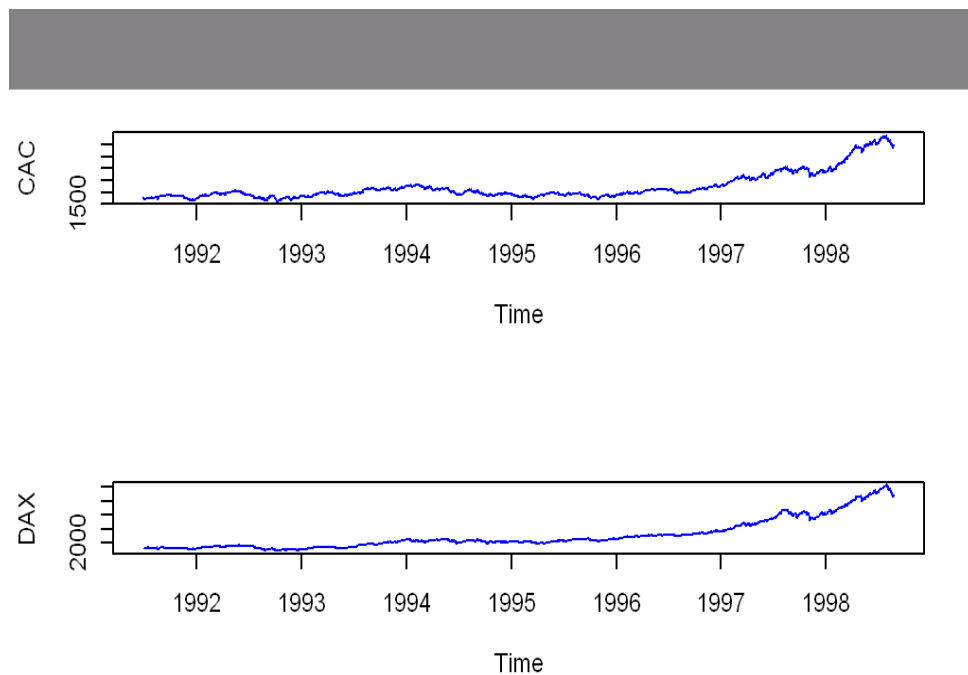
L'amplitude du spectre croisé est donnée par : $A_{x,y}(\lambda) = |f_{x,y}(\lambda)|$.

Le gain du spectre croisé est $G_{x,y} = A_{x,y}(\lambda)/f_{x,x}(\lambda)$.

La phase du spectre croisé est $\Phi_{x,y}(\lambda) = \text{Arctan}(q(\lambda)/c(\lambda))$.

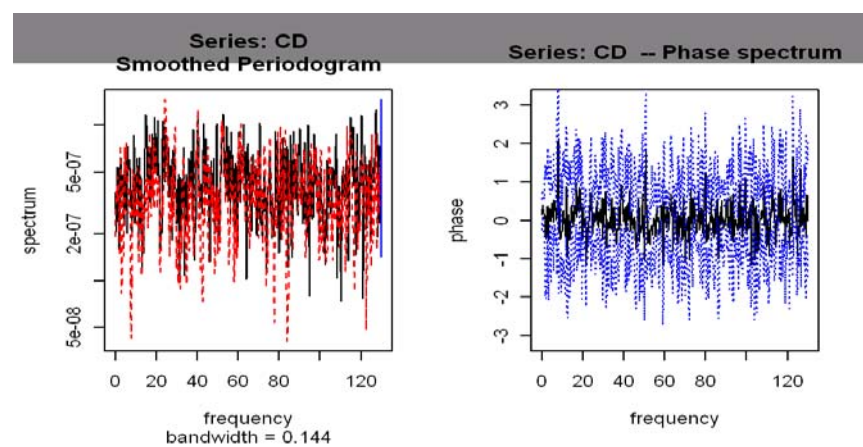
La cohérence est donnée par $K_{x,y} = |f_{x,y}(\lambda)| / (f_{x,x}(\lambda)f_{y,y}(\lambda))^{1/2}$.

Exemple : Phase et cohérence des indices européens : On considère les prix de des indices boursiers CAC40 (France) et DAX (Allemagne) sur la période 1991-1998 (1860 observations).

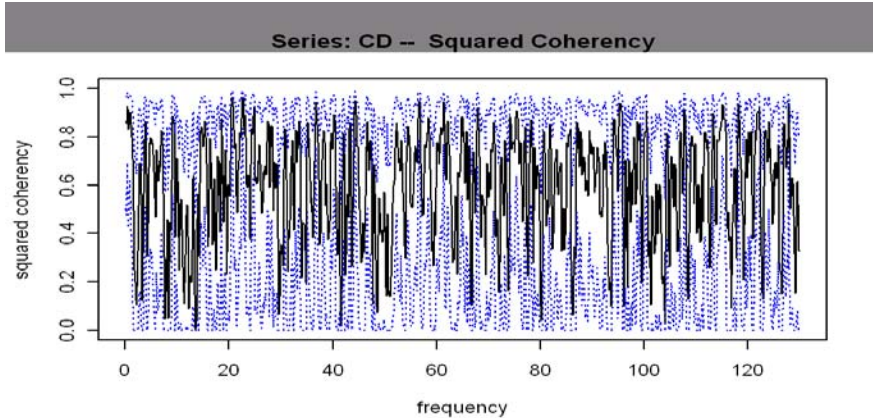


Avec le code :

```
data(EuStockMarkets), CAC <- EuStockMarkets[,3],
DAX <- EuStockMarkets[,1], CD <- cbind( diff(log(CAC)), diff(log(DAX)) )
fit10 <- spec.pgram(CD,spans=c(3,3))
plot(fit10,plot.type="phase")
```



Phase entre les deux indices



Cohérence entre les deux indices

Chapitre 5

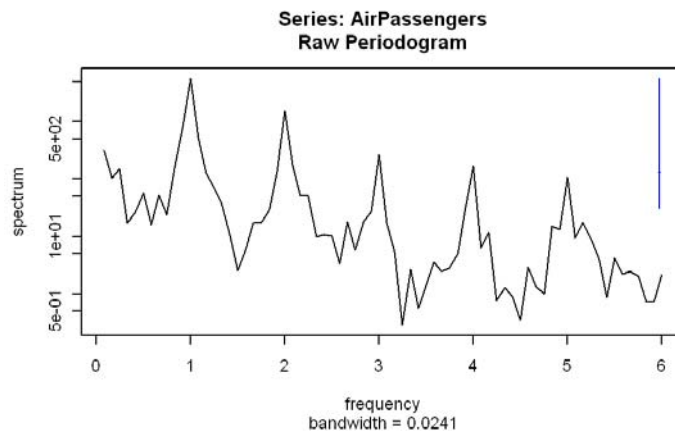
Analyse Spectrale

L'analyse spectrale a plusieurs application dans le domaine des séries chronologiques, une application intéressante est celle de l'indentification des cycles dans les données.

Pour les données saisonnières avec une saison s la densité spectrale a un pic dominant à la fréquence de Fourier $\omega_j = \frac{js}{T}$ où T est la taille de l'échantillon le cycle correspondant est donné par $cycle = \frac{T}{j}$

Exemple : On considère la série des voyageurs AirPassengers.

Avec le code : `fit11 <- spec.pgram(AirPassengers, plot=TRUE)` on obtient



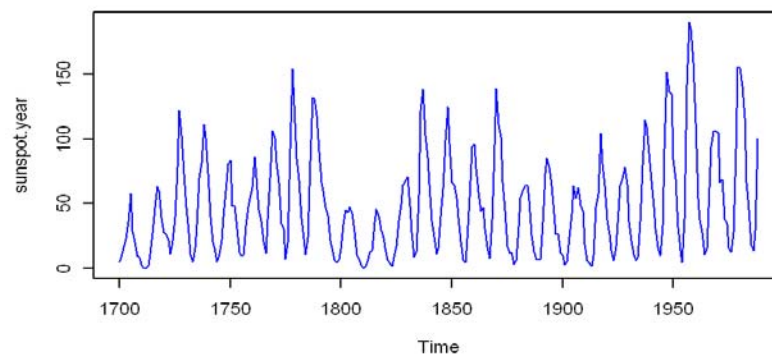
On remarque que le pic dominant est à la fréquence $\omega_j = 1$ donc $j = \frac{T}{s}$, et $cycle = \frac{T}{T/s} = s = 12$.

D'une manière général on détermine le cycle d'une série x par la formule $cycle = \frac{frequency(x)}{\omega_j}$ où

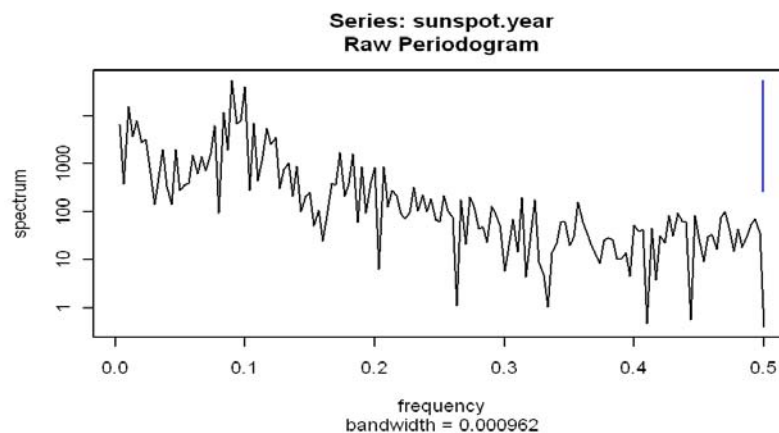
$$frequency(x) = \begin{cases} 12 & \text{si les données sont mensuelles} \\ 4 & \text{si les données sont trimestrielles} \\ 1 & \text{si les données sont quotidiennes ou annuelles} \\ \dots & \end{cases}$$

$frequency(x)$ est la fréquence des données. La fonction suivante calcule le cycle d'une série temporelle : $cycle <- function(x) \{ fit <- spec.pgram(x) \ m <- which.max(fit\$spec) \ cyc <- frequency(x)/fit\$freq[m] \ cyc \}$

Exemple : On considère la série annuelle des tâches solaires :



Le graphique suggère la présence d'un cycle



Il est difficile de localiser exactement le pic sur le graphique. La fonction cycle donne un cycle de 11.11 ans.

Partie 2 : ESTIMATION NONPARAMETRIQUE

Chapitre 6

Estimation de la densité de probabilité d'une variable aléatoire.

Problème : Comment faire une inférence statistique quand on ne connaît pas la loi des observations X_1, \dots, X_T ?

1 Estimation par histogramme

Une première solution est de reproduire la représentation en escalier de la fonction de répartition empirique F_n pour f

$$\hat{f}(x) = \sum_{i=1}^k \omega_i \mathbf{1}_{[a_i, a_{i+1}[}(x), a_1 < \dots < a_{k+1}. \quad (6.1)$$

En choisissant les a_i telles que $\sum_{i=1}^k \omega_i (a_{i+1} - a_i) = 1$ et $\omega_i (a_{i+1} - a_i) = \hat{P}_F(X \in [a_i, a_{i+1}[)$,

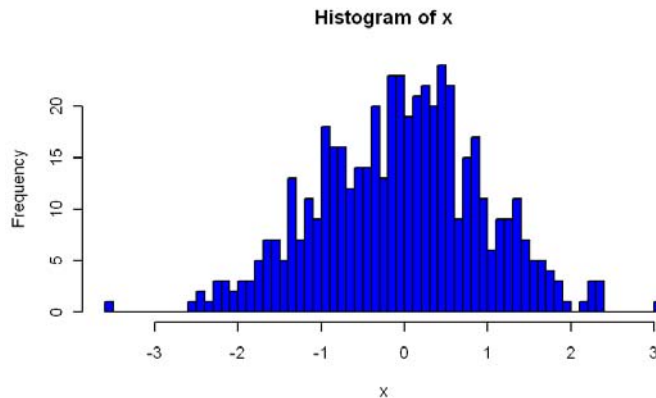
Par exemple,

$$\omega_i (a_{i+1} - a_i) = \frac{1}{T} \sum_{j=1}^T \mathbf{1}_{[a_i, a_{i+1}[}(X_j)$$

est un estimateur convergent de $P_F(X \in [a_i, a_{i+1}[)$.

le code : `hist(x)$density` donne les valeurs des ω_i et `hist(x)$breaks` les valeurs des a_i .

le code : `x <- rnorm(500), hist(x,breaks=50,col="blue")` donne



Histogramme d'une suite normale avec $k = 50, T = 500$.

Inconvénients :

- L'estimateur dépend du choix de la partition (a_i) , souvent construite en fonction des données (comme dans R)
- Problème des extrémités a_1 et a_{k+1} : ils ne peuvent pas être infinis mais doivent suffisamment approcher le support de f .
- k et (a_i) doivent dépendre de T pour que \hat{f} converge vers f mais... $a_{i+1} - a_i$ ne doit pas décroître trop vite vers 0 pour que l'estimation soit convergente : il faut suffisamment d'observations par intervalle $[a_i, a_{i+1}[$
- L'histogramme est une fonction discontinue

1.1 Estimateur du noyau

Au lieu de considérer une approximation uniforme autour de chaque X_i , on peut utiliser une fonction plus lisse :

$$\hat{f}(x) = \frac{1}{Th} \sum_{i=1}^T K\left(\frac{x - X_i}{h}\right), \quad (6.2)$$

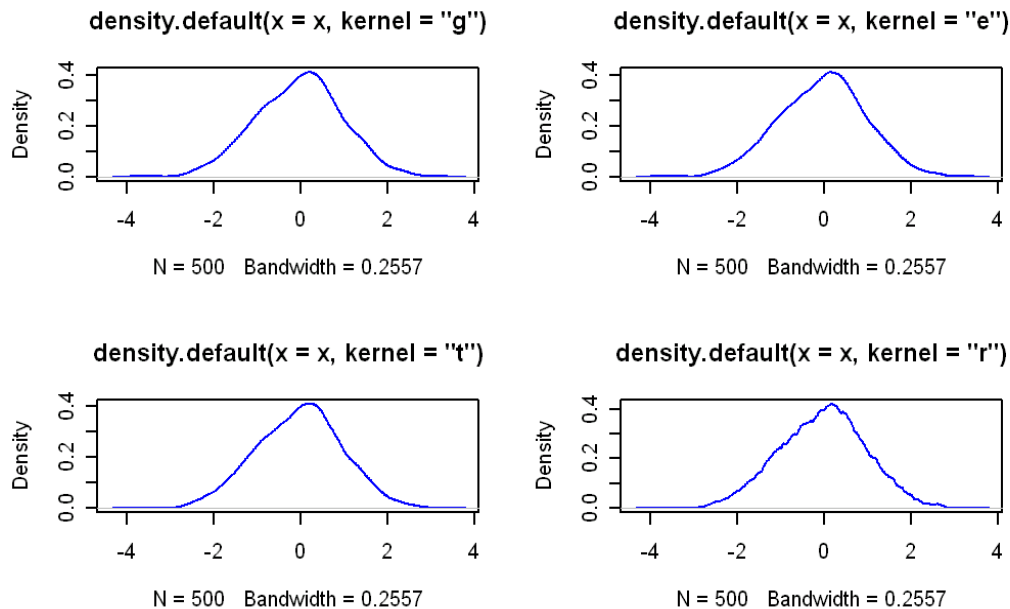
où K est un noyau (par exemple une densité de probabilité et h un facteur d'échelle.

Sur R on peut utiliser les noyaux suivants

- 1) Le noyau normal [kernel="gaussian" ou "g"]
- 2) Le noyau d'Epanechnikov [kernel="epanechnikov" ou "e"] $K(y) = C(1 - y^2)^2 \mathbf{1}_{[-1,1]}(y)$
- 3) Le noyau triangulaire [kernel="triangular" ou "t"]
 $K(y) = (1 + y)\mathbf{1}_{[-1,0]}(y) + (1 - y)\mathbf{1}_{[0,1]}(y)$.
- 4) Les noyaux "rectangular", "biweight", "cosine", "optcosine".

L'estimation est très influencée par le choix du noyau

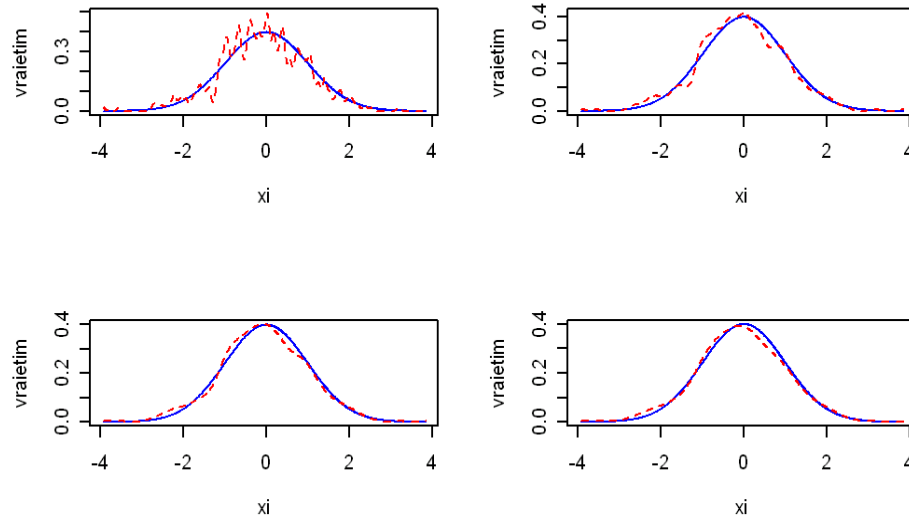
Exemple : Pour la série simulée précédemment on obtient



Le noyau rectangulaire donne un estimateur moins lisse.

Le choix de la fenêtre h est crucial,

Si h grand, un grand nombre des X_i contribuent à l'estimation de $f(x)$,
on obtient un estimateur avec un biais très grand et une variance très petite.
Si h petit, peu de X_i contribuent à l'estimation de $f(x)$,
on obtient un estimateur avec un biais très petit et une variance très grande



Estimateurs à noyau en fonction du choix de la fenêtre h_T .

La vraie densité en bleu et l'estimateur en rouge.

Quatre choix pour h_T : $T^{-1/2}$, $T^{-1/3}$, $T^{-1/4}$ et $T^{-1/5}$

Le choix $h_T = T^{-1/5}$ (en bas à droite) semble être meilleur

Fenêtre optimale

En étudiant l'erreur moyenne intégrée

$$d(f, \hat{f}) = E \int \left(f(x) - \hat{f}(x) \right)^2 dx,$$

on peut trouver un choix optimal pour la fenêtre h .

De la décomposition

$$d(f, \hat{f}) = \int \left(f(x) - E\hat{f}(x) \right)^2 dx + \int \text{var}\hat{f}(x)dx \text{ (biais}^2 + \text{var)}$$

et les approximations

$$f(x) - E\hat{f}(x) \simeq \frac{f''(x)}{2} h_T^2$$

$$E \left[\exp \left(-\frac{(X_i - x)^2}{2h_T^2} \right) \right] \simeq f(x) \sqrt{2\pi} h_T,$$

on en déduit que le biais est de l'ordre de

$$\int \left(\frac{f''(x)}{2} \right)^2 dx h_T^4,$$

et que le terme de variance est approximativement

$$\frac{1}{Th_T \sqrt{2\pi}}$$

Par conséquent, l'erreur tend vers 0 quand T tend vers ∞ si

■ h_T tend vers 0 et

■ Th_T tend vers l'infini.

La fenêtre optimale est donnée par

$$h_{opt} = \left(T \sqrt{2\pi} \int (f''(x))^2 dx \right)^{-1/5}.$$

h_{opt} dépend de la dérivée seconde qui elle-même est inconnue. Pour résoudre ce problème il existe plusieurs méthodes :

La fenêtre optimale basée sur "rule of thumb", avec le noyau de Gauss, a la forme

$$h_{opt} = \frac{0.9 \min(\hat{\sigma}, \hat{q}_{75} - \hat{q}_{25})}{1.34 T^{1/5}},$$

où σ est l'écart-type estimé et \hat{q}_{25} et \hat{q}_{75} sont les quantiles à 25% et à 75% estimés (Silverman (1986, page 48, eqn (3.31)).

Sur R il y a les méthodes suivantes :

-bw.nrd0 implémente la "rule-of-thumb" de Silverman.

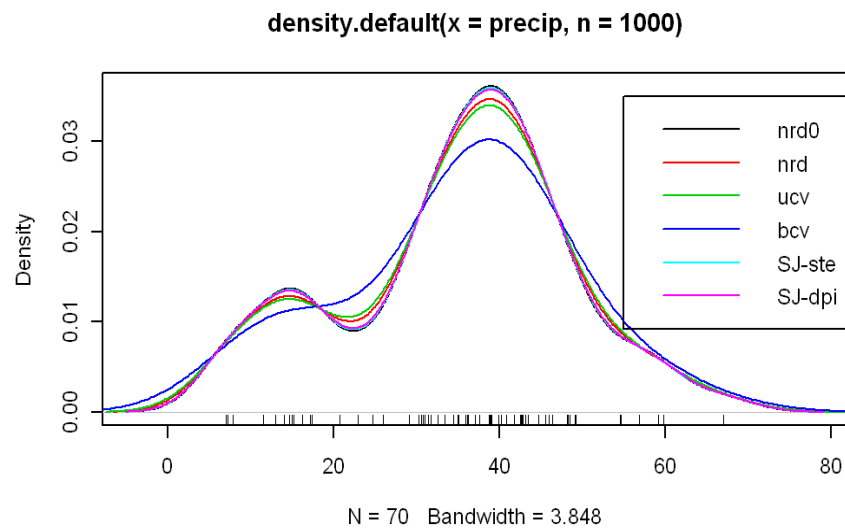
-bw.nrd une variation de la précédente Scott (1992), utilisant le 1.06.

-bw.ucv et bw.bcv utilisent la validation croisée non biaisée et biaisée respectivement.

-bw.SJ implémente la méthode de Sheather & Jones (1991).

Exemple : On considère la série des précipitations annuelles "precip", la densité estimée

avec les 6 choix possibles est donnée par



Chapitre 7

Régression non paramétrique

Le modèle (1.1) exprime le variable y_t comme une combinaison linéaire en fonction des variables explicatives x_{tj} . De tel modèle est incapable de décrire une relation non linéaire entre y_t et x_{tj} si celle-ci existe. Pour résoudre ce problème et celui du choix de la relation entre y_t et x_{tj} l'estimation non paramétrique offre l'avantage d'être plus flexible : ce sont les données qui déterminent la relation fonctionnelle entre y_t et x_{tj} . D'une manière générale on cherche une fonction r telle que

$$y_t = r(x_{t1}, \dots, x_{tk}) + u_t, \quad 1 \leq t \leq T, \quad (7.1)$$

où u_t est un bruit blanc.

La fonction r peut être estimée par plusieurs méthodes : 1) le réésoqramme 2) la méthode des k voisins les plus proches 3) la méthode basée sur les splines 4) la méthode du noyau entre autre. On se contente ici de présenter la méthode du noyau.

1 Estimateur à noyau

Posons $x_i = (x_{1i}, \dots, x_{ki})$. On dispose d'un échantillon $(x_i, y_i)_{1 \leq i \leq T}$ et on cherche à identifier la fonction r telle que

$$y_t = r(x_t) + u_t, \quad 1 \leq t \leq T, \quad (7.2)$$

Watson (1964) et Nadaraya (1964) ont proposé, indépendamment et simultanément, l'estimateur

$$\hat{r}(x) = \begin{cases} \frac{\sum_{i=1}^T y_i K((x-x_i)/h_T)}{\sum_{i=1}^T K((x-x_i)/h_T)} & \text{si } \sum_{i=1}^T K((x-x_i)/h_T) \neq 0 \\ 0 & \text{sinon} \end{cases}$$

Collomb (1976 ou 1977a) donne une évaluation asymptotique du biais et de la variance :

$$f(x) - E\hat{f}(x) \simeq \frac{h_T^2}{2} a(x)$$

et

$$E\left((f(x) - E\hat{f}(x))^2\right) \simeq \frac{1}{Th_T^k} b(x)$$

pour des fonctions $a(x)$ et $b(x)$. Ces formules permettent de montrer que

$$\min_{h_T \in \mathbb{R}^+} E\left(\hat{f}(x) - f(x)\right)^2 \simeq cT^{-4/(k+4)}.$$

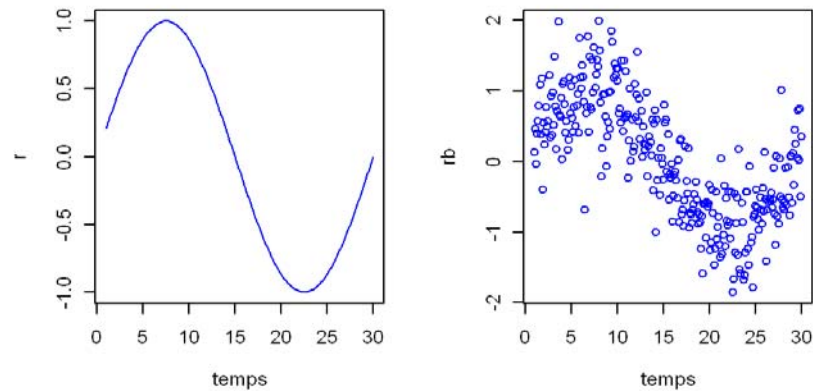
Comme pour l'estimation de la densité le choix de la fenêtre h_T est crucial.

Sur R la fonction *glkerns* permet l'estimation de la fonction r et ses dérivées avec un choix adaptative pour la fenêtre h_T .

Exemple 1. Simulation et estimation d'une fonction nonlinéaire

Simulation de la trajectoire

$$y_t = \sin\left(\frac{2\pi t}{20}\right) + u_t, t = 1, 1.1, \dots, 30, u_t \sim N(0, 0.5)$$



En expliquant le nuage de deux façons :

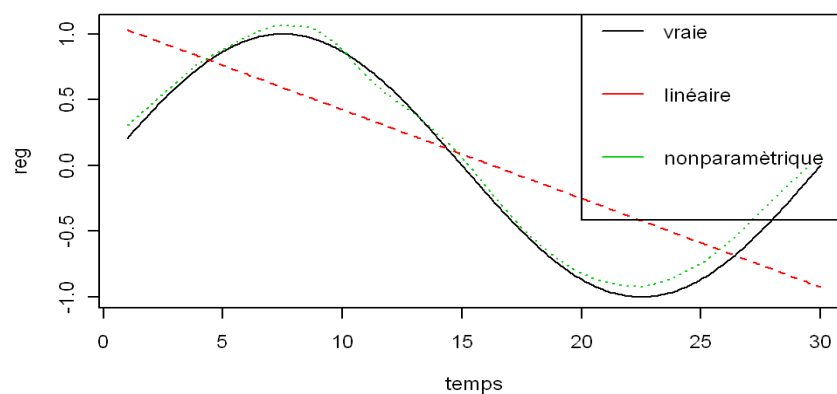
1. Linéaire

$$y_t = b_1 + b_2 t + u_t$$

2. Nonparamétrique :

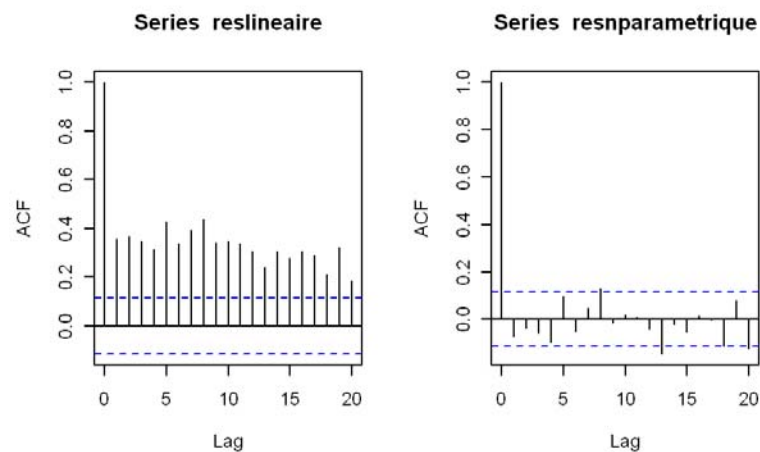
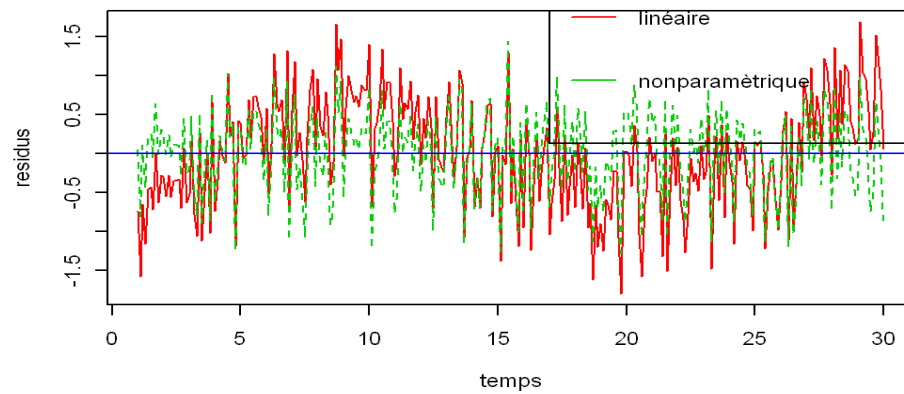
$$y_t = r(t) + u_t,$$

on obtient les estimateurs suivants :



La figure montre clairement la supériorité de l'estimateur nonparamétrique par rapport à l'estimateur de la régression linéaire.

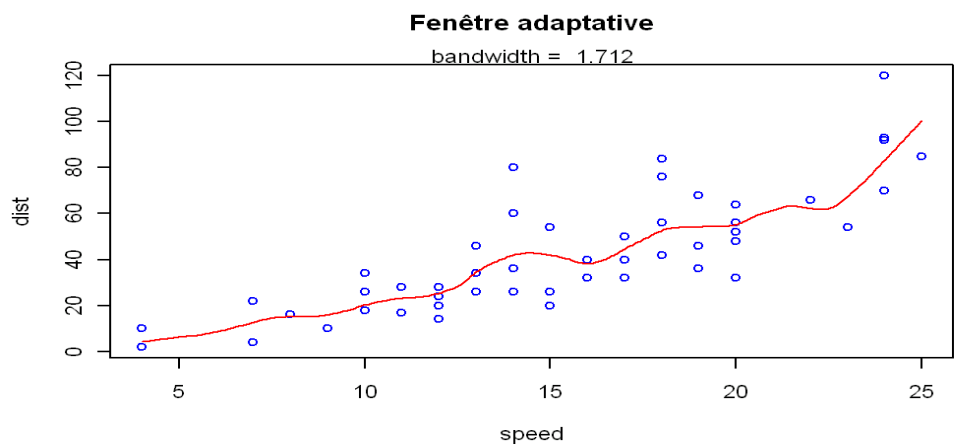
Analyse des résidus :



Exemple 2 : On reprend les données sur la distance de freinage :

Avec le code :`library(lokern), data(cars),plot(dist ~speed, data = cars, main = "Fenêtre adaptative"), fit13<- glkerns(cars$speed, cars$dist), lines(fit13$x.out, fit13$est, col=2)`

on obtient :



Bibliographie

- [1] R. Azencott, D. Dacunha. Castelle : *Séries d'observations irrégulières*, Masson 1984.
- [2] P.J. Brockwell. R.A. Davis : *Introduction to Time Series and Forcasting*, 1998.
- [3] G. Box, G. Jenkins : *Time series analysis*, Holden Day. 1976.
- [4] W.A. Fuller : *Introduction to statistical time series*, JOHN WILEY & SONS 1976.
- [5] C. Gouriéroux, A. Montfort : *Cours de séries temporelles*, Economica. 1983.
- [6] C. Chatfield : *The analysis of time series* , Chapman-Hall. 1975.
- [7] Silverman, B. W. (1986) *Density Estimation*. London : Chapman and Hall.