Modélisation des séries temporelles

Séance 2

Remarque : de manière générale, il est important de bien étudier l'aide R pour comprendre ce que l'on fait et le sens des paramètres estimés. Attention aux différences de notations ou de méthodes avec des logiciels comme SAS (ou d'autres que vous pourriez utiliser). Dans chaque exercice, il faut être capable de justifier le choix d'un modèle ARMA et de l'écrire à l'aide des polynômes symboliques de l'opérateur retard.

Rappel : Pour installer un package (ici `forecast`) (à faire une seule fois, il faut choisir un dépôt) :

```
install.packages("forecast")
```

Pour activer un package (à faire à chaque nouvelle session) :

```
library(forecast)
```

A priori, les packages sont déjà installés sur les machines de l'école.

Notions abordées : méthode de Box-Jenkins (ACF, PACF, validation statistique des hypothèses).

1 Chroniques simulées

L'objectif est d'illustrer les différentes étapes de la méthode de Box et Jenkins sur des séries temporelles issues de données simulées. Bien entendu, lorsqu'on cherche à modéliser une série réelle, on ne connaît pas le modèle a priori .

1.1 MA(2)

On simule 200 valeurs d'une série temporelle X_t suivant le modèle moyenne mobile d'ordre 2 suivant :

$$X_t = \varepsilon_t + 0.4\varepsilon_{t-1} + 0.7\varepsilon_{t-2}$$

où (ε_t) est un bruit blanc gaussien d'écart-type 0.1, à l'aide de l'instruction suivante :

```
X <- arima.sim(n=200,list(ma=c(0.4,0.7)),sd=.1)
plot(X, ylab="X", type="o", pch=20)
```

Vous pouvez lancer plusieurs fois la simulation pour constater que les valeurs changent d'une exécution à l'autre.

À l'aide de la méthode de Box et Jenkins, identifiez un modèle MA(2) sur les données générées, validez ce modèle, et estimez les paramètres.

Vous aurez besoin des fonctions R suivantes, que vous exécuterez successivement :

1. `Acf(X)`
   `Pacf(X)`
   `tsdisplay(X)  # que l'on privilégiera`
   tracent ACF et PACF de la chronique X. On utilise `Acf` et `Pacf` du package `forecast` (notez la majuscule), plutôt que `acf` / `pacf` dans le package `stats` (pour forcer l'utilisation d'une fonction dans un package, on peut faire : `stats::acf(X)`).
   Remarquez que ces ACF et PACF sont caractéristiques d'un processus MA(2).

2. `output_X <- Arima(X,order=c(0,0,2))`
   `summary(output_X)`
   pour estimer les paramètres d'un processus MA(2) (`Arima` avec majuscule dans le package `forecast`, plutôt que `arima` dans `stats`).
   Notez les estimations des coefficients du modèle, de la variance des résidus (`sigma^2`) et de l'écart-type RMSE, de la log-vraisemblance, et des critères AIC / AICc / BIC (que l'on expliquera en séance 3).

3. `plot(output_X$residuals, ylab="residus")`
   `abline(0,0)`
   `grid()`
   permet de visualiser les résidus. Vérifiez qu'ils ont l'aspect d'un bruit blanc gaussien.

4. `tsdiag(output_X)`
   pour l'étude des résidus du modèle (attention, il s'agit d'une fonction de `stats`, donc l'ACF commence au décalage de 0...).
   Remarquez qu'il n'y a pas de structure apparente dans le graphe des résidus, pas d'autocorrélation significative, et que les tests de Portmanteau (*Ljung Box statistics*) ne permettent pas de rejeter l'hypothèse de résidus correspondant à un bruit blanc gaussien.

5. ```
   hist(residuals(output_X), freq = F, col = "grey")
   curve(dnorm(x, mean = mean(residuals(output_X)),
         sd = sd(residuals(output_X))), col = 2, add = TRUE)
   qqnorm(residuals(output_X))
   qqline(residuals(output_X),col=2)
   ```
   pour histogramme et QQ-plot des résidus : la répartition est cohérente avec une répartition gaussienne.

6. `coeftest(output_X)`
   `coefci(output_X)`
   pour tester la significativité des coefficients et afficher les intervalles de confiance, une fois la gaussianité du modèle établie (ces fonctions sont dans le package `lmtest`, qu'il faut donc installer).

7. Comme la constante du modèle n'est pas significative, on va tester un modèle MA(2) dans lequel la constante est forcée à 0 :

   `output_X <- Arima(X,order=c(0,0,2),include.mean=FALSE)`

   et on recommence (rapidement, il n'y a pas de surprise à attendre) la validation au point 2.

8. Écrivez le modèle final que vous avez trouvé sous la forme :

$$X_t = \varepsilon_t + \alpha_1 \varepsilon_{t-1} + \alpha_2 \varepsilon_{t-2}$$

   ou de manière équivalente, avec le polynôme symbolique en l'opérateur retard B (comme *backward*) :

$$X_t = (1 + \alpha_1 B + \alpha_2 B^2)\varepsilon_t$$
   ```

1.2 AR(1)

On considère une série suivant un modèle autorégressif d'ordre 1 :

$$(1 - 0.8B)(X_t - 2) = \varepsilon_t$$

générée par :

```
X <- 2+arima.sim(n=200,list(ar=c(.8)),sd=.5)
```

À l'aide de la méthode de Box et Jenkins, identifiez un modèle ARMA.
(même démarche que pour la chronique précédente : n'hésitez pas à faire des copier/coller)

1.3 AR(2) particulier

On considère une série suivant un modèle autorégressif d'ordre 2 :

$$(1 - 0.8B^2)(X_t + 5) = \varepsilon_t$$

générée par :

```
X <- -5 + arima.sim(n=200,list(ar=c(0,.8)),sd=.8)
```

À l'aide de la méthode de Box et Jenkins, identifiez un modèle ARMA.

2 GNP

Le fichier `gnp.txt` donne la série trimestrielle du produit national brut des USA du premier trimestre 1947 au premier trimestre 1991. Il s'agit de données corrigées des variations saisonnières.

On crée la série temporelle correspondante par :

```
gnp <- read.table("gnp.txt",header=T)
gnp <- ts(gnp,frequency=4,start=c(1947,1))
```

À l'aide de la méthode de Box et Jenkins, identifiez un modèle ARIMA.

3 Internet *(pour les plus rapides)*

Le fichier `internet.txt` donne le nombre d'utilisateurs connectés sur internet chaque minute sur une séquence de 100 minutes.

```
internet <- read.table("internet.txt",header=T)
nlogs <- ts(internet[,1])
```

Identifiez un modèle ARIMA. On constate que la méthode de Box-Jenkins nous guide vers un modèle ARIMA(3,1,0).