Université Paris I, Panthéon - Sorbonne

Première Année Master M.A.E.F. 2013 – 2014

TP de Séries Chronologiques n^0 2 : Tendance d'une série chronologique

L'objectif de ce TP est de mettre en pratique différentes procédures de régression permettant d'estimer les tendances et saisonnalités éventuelles d'une série chronologique. Cette fois-ci, et contrairement à ce que l'on avait fait avec les séries simulées du TP1, on travaille avec une série issue de données réelles et dont on ne sait rien d'une possible modélisation. Il s'agira du cours de l'or le 14/11/11 à 10h en données tick by tick.

```
data = read.csv("chemin/gld.csv", header=T, sep=";", dec=",") # Importe le fichier
head(data) # Pour voir
plot(data[,"Ask"], type='l') # Pour voir
ask = data[,"Ask"] # Affecte à l'objet 'ask' la colonne Ask du fichier
bid = data[,"Bid"] # Affecte à l'objet 'bid' la colonne Bid du fichier
dim(data) # Dimension de la matrice
T = dim(data)[1] # Nombre d'observations
```

Nous allons maintenant calculer le mid price, i.e. la moyenne du ask et du bid price à chaque instant.

```
mid = c()  # On définit l'objet mid comme étant un vecteur
for ( t in 1:T ) {
        mid[t] = (ask[t] + bid[t])/2
}
```

En servant du fait que R est un langage matriciel, on peut également écrire,

```
mid = (ask + bid)/2
```

Intérêt de la seconde méthode sur la première ?

Nous allons maintenant tracer les différents éléments pour avoir une vue d'ensemble.

```
t = 1:length(ask)
plot(t,mid, type='l')
lines(t, bid, col="red")
lines(t, ask, col="blue")
```

Si l'on regarde en détail les données, on voit que les transactions n'arrivent pas à intervalle régulier, on préfera donc,

```
temps = data[,"Min"] + data[,"Sec"]/60 + data[,"Milli"]/(60*1000)
plot(t,mid, type='l')
lines(t, bid, col="red")
lines(t, ask, col="blue")
```

Pour être plus rigoureux il aurait fallu intégrer la date et l'heure dans la série 'temps', mais comme dans notre cas nous regardons exclusivement une seule heure d'un jour donné (en l'occurrence à 10h le 14/11/11), cela n'a pas d'importance.

L'objectif est maintenant de déterminer la (les) tendance(s) du mid price. Pour commencer nous allons simplement chercher le modèle de régression linéaire,

$$y_t = \beta_0 + \beta_1 t + \xi_t$$

avec β_0 et β_1 deux constantes réelles, y le prix moyen de l'actif financier , ξ le bruit et t le temps (on voit ici l'intérêt d'avoir prit les temps d'arrivées plutôt qu'un temps espacé régulièrement).

```
# modélisation:
reg1=lm(mid~temps)
summary(reg1)

# Graphe de la tendance
plot(temps, mid, type='l')
lines(temps, reg1$fitted.values)

# pour réecrire le modèle 'à la main'
reg1$coeff
coef = reg1$coeff
model = coef[1] + coef[2]*temps
plot(temps, mid, type='l')
lines(temps, model)

# Graphe et test de normalité des résidus
plot(temps, reg1$residuals, type='l')
qqnorm(reg1$residuals)
qqline(reg1$residuals)
```

On peut maintenant essayer de tracer une tendance polynômiale,

$$y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + ... + \beta_\alpha t^\alpha + \xi_t$$

$\alpha \in \mathbb{N}$ une constante.

```
reg2 = lm(mid~cbind(temps,temps^2))
coef = reg2$coeff
lines(temps, coef[1] + coef[2]*temps^2 + coef[3]*temps^2)
reg2=lm(mid ~ poly(temps,2))
```

Comparer le résultat avec la commande suivante

```
reg2=lm(mid ~ poly(temps,2))
lines(temps, reg2$fitted.values, col=2)
```

Analyse des résidus,

```
split = matrix(c(1,2,1,3), 2, 2)
split
layout(split)
plot(reg2$residuals, type='l')
```

```
qqnorm(reg2$residuals)
qqline(reg2$residuals)
acf(reg2$residuals)
```

Essayons de voir ce qu'il ce passe si l'on augmente de manière arbitraire l'ordre du polynôme,

```
reg3=lm(mid ~ poly(temps,7))
lines(temps, reg3$fitted.values, col=3)
```

Analyse des résidus,

```
split = matrix(c(1,2,1,3), 2, 2)
split
layout(split)
plot(reg3$residuals, type='l')
qqnorm(reg3$residuals)
qqline(reg3$residuals)
acf(reg3$residuals)
```

A la vue des résidus, quel modèle choisir ? $\alpha = 1, 2$ ou 9 ? Un critère de choix pourrait être l'ordre du polynôme minimisant le plus l'erreur quadratique moyenne,

```
MSE = function(y,ytilde){
        N = length(y)
        mse=0
        for ( t in 1:N) {
                mse = mse + (y[t] - ytilde[t])^2
        }
        mse = mse/N

        return(mse)
}

MSE(mid, reg3$fitted.values)
```

Réécrire la fonction MSE en utilisant le calcul vectoriel de R afin de ne pas avoir de boucle. Pour trouver la valeur optimale du degré du polynôme,

```
ValueInit = 1000
for ( i in 1:20 ){

        TestReg=lm(mid ~ poly(temps,i))
        Value = MSE(mid, TestReg$fitted.values)

        if ( Value < ValueInit){
                iOpti = i
                ValueInit = Value
        }
```

Commenter le résultat trouvé, cela parrait il 'correct' ? Quels autres critères pouvons nous regarder ?