Université Paris I, Panthéon - Sorbonne

Master M.A.E.F. 2016 – 2017

TP de Séries Temporelles n^0 1 : Manipulation, simulation et estimation pour une série temporelle

L'objectif de ce TP est de vous familiariser avec les objets et les commandes du logiciels R, notamment de tout ce qui a trait aux séries chronologiques. Dans la suite, sont écrites à gauche des commandes à taper, et à droite des commentaires sur ces commandes ou des questions relatives. Vous pouvez donc ainsi apprendre les commandes du logiciel par la pratique.

Une façon intéressante d'utiliser le logiciel est d'ouvrir des "Scripts" dans lesquels vous allez pouvoir écrire l'ensemble des commandes. Il sera alors possible d'exécuter une partie des commandes tapées. Ce sera donc très pratique plutôt que de retaper les commandes à chaque fois. Il faut donc aller dans la colonne *Fichier*, et cliquer sur nouveau *script*. Puis, après avoir taper les commandes choisies, avec la souris vous sélectionner la ou les commandes particulières que vous désirer faire tourner, vous allez ensuite dans la colonne *Edition* et vous cliquer sur *Executer une selection*.

Manipulations

x=c(4,3,7,1.7,3,8.4,5,12,9,12)	Génère un vecteur de 10 données (c comme collector).
is.ts(x)	x est-il une "time series" (série chronologique)?
y=as.ts(x)	On transforme x en un objet Splus time series y.
y	Changements apparents?
time(y)	Permet de consulter le descriptif de cette série temporelle.
is.ts(y)	Vérification.
	Plus généralement, le logiciel R fonctionne avec différents types d'objets :
	les vecteurs (commandes *is.vector* et *as.vector*);
	les matrices (commandes *is.matrix* et *as.matrix*);
	les tableaux de données (commandes $is.data.frame$ et $as.data.frame$);
	les séries chronologiques (commandes *is.ts* et *as.ts*);
	les listes (commandes *is.list* et *as.list*);
t=tsp(y)	Associe à la série y une série temporelle t (par défaut de 1 en 1, fréquence 1).
plot.ts(y)	Tracé de la série chronologique (relie les données).
plot.ts(y,type="b")	Marque les différents points de la série.
z=ts(y,freq=4)	Divise la série en trimestre (marche aussi en divisant en mois donc $freq = 12$ ...).
z	Vérification.
time(z)	Temps associé?
z=ts(x,freq=4,1991+1/4, 1993)	On renouvelle la série des temps, avec une autre année de début et de fin.
time(z)	
plot.ts(z)	Nouveau tracé.
frequency(z)	Détermine la fréquence de la série des temps.
length(z)	Longueur de la série z.
m=tapply(z,cycle(z),mean)	Permet de calculer la moyenne par cycle.
m	Vérification. Dans le cas présent, est-ce une évaluation d'un éventuel saisonnier?

Simulation d'une série chronologique

	Aller dans l'aide et taper le mot-clé "distributions".
	le préfixe r est utilisé pour la génération de nombre aléatoire, d pour la densité,
	p la fonction de répartition et q pour le quantile.
qnorm(0.95)	Quantile à 95% pour la gaussienne centrée réduite.
pnorm(2)	Fonction de répartition en 2.
	Générer 100 variables gaussiennes indépendantes de loi $\mathcal{N}(-1,3)$.
	Calculer la moyenne, l'écart-type et tracer un histogramme.
x=rnorm(30)	On génère un bruit blanc.
par(mfrow=c(2,2))	On découpe la fenêtre graphique en 4.
plot.ts(x)	Représentation du bruit blanc.
hist(x,nclass=6)	On précise le nombre de classes.
x=rbinom(30,10,.3)	Génération de variables binomiales. Observer les caractéristiques de x.
	Centrer et normaliser cette série. On note y la nouvelle série.
y=as.ts(x)	Transformer y en série chronologique.
y=sort(y)	On ordonne les éléments de y. Tracer la série ordonnée.
plot.ts(x,y)	Pour tracer dans l'ordre les y en fonction des x.
rep(c(2,-3,1),10)	Permet de répéter une séquence quelconque.

Exercice

Simuler une série $X_2 = (X_2(k))_k$ trimestrielle commençant le deuxième trimestre 1980 et finissant le troisième trimestre 2000, telle que $X_2(k) = a(k) + S(k) + \varepsilon_k$ (k étant le numéro du trimestre), où:

- La tendance est $a(k) = 0.02 \times k + 3$.
- Le saisonnier S est une fonction périodique de période 1 an, telle que $S = 3$ au premier trimestre, $S = 5$ au second, $S = -1$ au troisième et $S = -7$ au quatrième.
- ε est un bruit blanc gaussien de variance 1.5.

Quelle est la partie déterministe de X_2 ? Les variables $X_2(k)$ sont-elles indépendantes?

Estimation de la tendance d'une série temporelle

L'objectif est de mettre en pratique différentes procédures de régression permettant d'estimer la tendance éventuelle d'une série chronologique. On commence par simuler une série chronologique avec tendance (connue) puis on estime cette tendance comme si on ne la connaisser pas...

	Simulation d'une série avec tendance
t=1:120	Le temps.
a=10*sqrt(t+200/log(t+2))	La tendance.
eps=10*rnorm(120)	Le bruit.
X=a+eps	La série chronologique simulée
plot.ts(X)	Pour voir...

Régressions polynomiales

On commence par estimer l'éventuelle tendance de la série chronologique par des régressions linéaires de degrés de plus en plus élevé.

	Régression linéaire simple
reg1=lm(X ∼ t)	Régression linéaire de X par t.
abline(reg1)	Droite de régression s'ajoutant à la série chronologique.
plot(reg1)	Pour voir les diagnostics graphiques classiques de la régression.
names(reg1)	Différents composants associés à res.
reg1$coef	Coefficients de res.
summary(reg1)	Résumé des résultats de la régression.
segments(t,reg1$fit,t,X)	Visualisation des résidus.
reg1$res	Valeurs des résidus.
win.graph()	On crée une nouvelle fenêtre graphique.
par(mfrow = c(2,2))	On divise cette fenêtre graphique en $(2, 2)$.
plot(reg1, las = 1)	Trace sur la même fenêtre les 4 graphiques d'analyse.
	Régression polynomiale (degré 2)
reg2=lm(X ∼ poly(t,2))	Ajustement polynomial (degré 2).
plot.ts(X)	Graphique de la série chronologique.
lines(t,reg2$fit)	Graphe du polynôme associé.
summary(reg2)	Analyse détaillée de la régression.
segments(t,reg2$fit,t,X)	Visualisation des résidus.
	Régression polynomiale (degré 9)
reg9=lm(X ∼ poly(t,9))	On utilise maintenant un ajustement polynomial de degré 9 Refaire les mêmes démarches que précédemment.

	Choix du degré du polynôme avec les critère AIC et BIC
library(MASS)	On charge le package MASS
long=length(t)	
kmax=10	Degré maximum du polynôme
Z=matrix(nrow=long,ncol=kmax+1)	On définit une matrice Z
for (i in 1:long)	On va définir la matrice Z élément par élément
for (j in 1:(kmax+1))	
Z[i,j]=i^(j-1)	
ZZ=as.data.frame(Z)	On transforme Z en objet "data.frame"
x.lm=lm(X∼.,data=ZZ)	
x.AIC=stepAIC(x.lm,k=2)	On utilise une commande qui minimise le critère AIC
x.BIC=stepAIC(x.lm,k=log(long),trace=FALSE)	Même chose mais avec le critère BIC Commentaires sur les résultats?

Régressions non-paramétriques pour estimer la tendance

Pour obtenir une meilleure adaptation de la régression aux variations locales de la série, on peut procéder avec des régressions locales (loess et lowess : "lo" comme local, "ss" comme scatterplot smoothing) qui fonctionnent avec une fenêtre (dont on peut directement choisir la longueur dans le cas lowess), "coulissant" successivement le long des temps de la série. Dans chaque fenêtre, on fait une régression polynômiale de degré en général 0, 1 ou 2, par moindres carrés pondérés, sur les données de la série, et on obtient ainsi un point lissé pris en général au milieu de la fenêtre. Les poids de la pondération sont obtenus suivant la règle suivante: les poids sont plus élévés plus on est proche du point à lisser. Dans le cas de la régression lowess, on opère de manière itérative, une nouvelle pondération (d'où le "w" pour weighted), en fonction des écarts aux valeurs initiales.

Régression loess

regl1=loess(X $\sim$ t)	régression “locale” standarde.
names(regll)	Différents résultats de la régression loess.
ts.plot(X)	On trace à nouveau les variations de la série.
lines(t,regl1$fit)	Graphique de la régression locale associée.
points(t,X)	Représentation des points.
title("Résultats de loess(X $\sim$ t)”)	On met une légende au graphique.
R2=1-var(regl1$res)/var(X)	Calcul direct du R2. Conclusion?

Régression par lissage lowess

regl2=lowess(t,X,f=0.5)	Lissage par lowess de la série X, avec un paramètre de lissage fixé
names(regl2)	Différents résultats de la régression lowess.
	Noter que lowess n’a pas la syntaxe et les options identiques
	aux autres régressions.
low2=ts(regl2$y)	On forme à nouveau une série chronologique.
ts.plot(X,low2)	Tracé de la série et de la série lissée.
regl3=lowess(t,X,f=0.1)	Lissage par lowess de la série X, avec un autre paramètre de lissage fixé
low3=ts(regl3$y)	On forme la série chronologique.
ts.plot(X,low3)	Tracé de la série X et de la série lissée.
ts.plot(low2,low3)	Pour comparer les deux lissages; expliquer la différence entre les deux lissages.
	Quelle est le paramètre que vous choisiriez ?

Régression par l’estimateur à noyaux

ks1=ksmooth(t,X, "normal", bandwidth=2)	On commence par choisir la taille de fenêtre de l’estimateur.
ks1	Pour voir...
plot.ts(X)	
lines(ks1,col=2)	A comparer avec la série
ks2=ksmooth(t,X, "normal", bandwidth=10)	Autre choix.
lines(ks2,col=3)	A comparer avec la série et l’autre régression
library(lokern)	On installe un nouveau package
kerada=glkerns(t,X)	Regression par noyau avec choix adaptatif de la taille de fenêtre
plot(kerada)	Pour voir...