Université Versailles Saint-Quentin
Année : 2013-2014 2ème semestre
Niveau : MASTER IS 1ère année
Cours : Séries Chronologiques I
Enseignant : A. Illig

TP R de séries chronologiques

Ce TP a pour but d'étudier la série chronologique du *Nombre de morts accidentelles aux Etats-Unis de 1973 à 1978* au moyen des techniques décrites en cours.

Table des matières

1 Mise en place de l'environnement de travail

1.1 Aide, packages, données

Pour obtenir des renseignements sur une fonction donnée, on peut faire appel à l'aide en ligne :

```
> help(plot.ts)
```

Par ailleurs, la commande

```
> help.search("plot")
```

recherche toutes les fonctions R où le mot-clé `plot` est utilisé dans le descriptif de la fonction.

Pour une étude spécifique, il peut être nécessaire de faire appel à un *package* de R bien précis. La commande

```
> library()
```

affiche les *packages* disponibles dans la version de R que vous avez téléchargée à l'adresse http ://cran.r-project.org/. Par exemple, pour charger le *package* `survival` dédié à l'analyse de survie et afficher la liste des fonctions du *package* `survival`, il conviendrait de taper les commandes

```
> library(survival)
> library(help=survival)
```

Enfin pour obtenir la liste des *packages* utilisés, il suffit de taper la commande

```
> search()
```

et l'on se rend compte que R charge automatiquement les *packages* de base.

La liste des données disponibles dans R (augmentant avec le nombre de les packages utilisés) s'obtient en tapant la commande

```
> data()
```

Selon le package auquel appartiennent les données, elles sont éventuellement déjà disponibles dans R. Si ce n'est pas le cas, il est nécessaire de les charger. Par exemple les données `airquality` (de type *data frame*) sont chargées par la commande

```
> data(airquality)
```

et on accède maintenant aux noms des variables composant l'ensemble des données `airquality` :

```
> names(airquality)
```

Les données *Ozone* sont enregistrées dans le vecteur `airquality$Ozone` et directement accessibles dans le vecteur `Ozone` si l'on tape la commande

```
> attach(airquality)
```

La commande `search()` permet d'afficher les données attachées en sus des *packages* utilisés. Les données `airquality` se détachent en tapant

```
> detach(airquality)
```

et n'apparaissent plus si l'on fait à nouveau appel à la commande `search()`.

1.2 Installation de nouveaux packages et chargement des données utilisées dans le TP

Pour ce TP, nous aurons besoin du *package* `tseries` qui n'est pas nécessairement disponible dans votre version de R. Si tel est le cas, télécharger et installer le *package* selon la procédure suivante :

- Cliquer sur l'icône "Packages" de R (barre des tâches située en haut de la fenêtre R de Windows).
- Cliquer sur "Install package(s)".
- Choisir un site miroir pour le téléchargement (par exemple France (Paris 2)) et cliquer sur le bouton "OK".
- Choisir le package à intaller et cliquer sur le bouton "OK".

1. Installer le *package* `tseries` dédié à l'étude des séries chronologiques.
2. Télécharger l'aide consacrée au *package* `tseries` à l'adresse http ://cran.r-project.org/web/packages/tseries/tseries.pdf.
3. Charger les jeux de données `AirPassengers`, `EuStockMarkets` et `USAccDeaths`.

1.3 Exemples de représentations graphiques et sauvegardes

Nous allons représenter graphiquement les données de R citées au paragraphe précédent. Nous effectuerons une sauvegarde de chaque graphique au format *eps* ou *pdf*.

1. A titre d'exemple, taper les commandes suivantes :

```
> plot(AirPassengers,type="b",col="red",lwd=1,pch=20,
main="Nombre de voyageurs aériens en milliers")
> dev.copy2eps(file="Graph_AirPassengers.eps")
> par(bg="lightyellow")
> plot(EuStockMarkets,type="l",col=4)
> dev.copy2pdf(file="Graph_EuStockMarkets.pdf")
> windows()
> # Dispositif graphique à 4 fenêtres
> layout(matrix(1:4,2,2))
> layout.show(4)
>  plot(EuStockMarkets[,1],main="EuStockMarkets_DAX")
>  plot(EuStockMarkets[,2],main="EuStockMarkets_SMI")
>  plot(EuStockMarkets[,3],main="EuStockMarkets_CAC")
>  plot(EuStockMarkets[,4],main="EuStockMarkets_FTSE",ylab="")
> dev.off()
```

2. Tracer les données `USAccDeaths`. Sauvegarder au besoin le graphique produit.

2 Etude des données `USAccDeaths`

2.1 Représentations graphiques

Nous étudions plus particulièrement ici les données mensuelles `USAccDeaths` du nombre morts accidentelles aux Etats-Unis de 1973 à 1978 de R.

1. A l'aide de la commande `help`, obtenir les informations disponibles dans R sur les données `USAccDeaths`.
2. Créer un vecteur `USA` numéroté de 1 à 72 à partir des données `USAccDeaths`.

3. Représenter les données sous forme de points à l'aide de la commande `plot`. Ajouter au graphique précédent les lignes brisées reliant chaque donnée grâce à la commande `lines`. Analyser le graphique obtenu.
4. Représenter, sur un autre graphique, la fonction d'autocorrélation empirique. Commentez le graphique obtenu.
5. A titre d'exemple, taper les commandes suivantes :

```
> windows()
> layout(matrix(1:2,1,2))
> layout(matrix(1:2,1,2))
> layout.show(2)
> plot(USA,type="b",pch=22,col=5,main="Données USA")
> acf(USA)
> dev.off()
> decomp=stl(USAccDeaths,s.window="periodic")
> plot(decomp)
```

2.2 Traitement de la tendance et du mouvement saisonnier

2.2.1 Méthode des moindres carrés

On se place dans le modèle suivant

$$X_t = a_0 + a_1 t + a_2 \cos\left(\frac{2\pi t}{12}\right) + a_3 \sin\left(\frac{2\pi t}{12}\right) + \epsilon_t.$$

1. Utiliser la commande `lm()` pour ajuster le modèle aux données.

```
> t=seq(1,72,1)
> cos.t=cos((pi/6)*t)
> sin.t=sin((pi/6)*t)
> reg=lm(USA~t+cos.t+sin.t)
> summary(reg)
> plot(USA,type="b",pch=22,col=5,main="Données USA")
> lines(reg$fit,lwd=2,col=5)
```

2. Tenter d'améliorer l'ajustement en modifiant la fonction périodique du modèle.
3. Enregistrer les résidus du modèle dans un vecteur `residus1`

2.2.2 Méthode "classique"

1. Effectuer une première approximation de la tendance (éliminant le mouvement saisonnier) à l'aide d'un filtre moyenne mobile de longueur à déterminer. Enregistrer la série de tendances dans un vecteur `tendance1`. Représenter simultanément les données `USA` et la tendance `tendance1`.
   Indication : Utiliser la commande `filter(USA,c(0.5,rep(1,11),0.5))`.
2. Estimer le mouvement saisonnier à partir des données corrigées de la tendance `na.omit(USA-tendance1)`. Enregister le mouvement saisonnier dans un vecteur `saison` de même taille que `USA`. Représenter graphiquement le mouvement saisonnier.
   Indication : On pourra s'aider des lignes de commandes suivantes :

```
USAcleartrend=na.omit(USA-trendance1)
w=rep(0,12)
for (i in 1:12){
j=0
while (i+j*12 <= length(USAcleartrend)){
w[i]=w[i]+USAcleartrend[i+j*12]
j=j+1}
w[i]=w[i]/j}
s=w-mean(w)
ss=rep(0,12)
for (i in 1:12){
if(i<=6){
ss[i]=s[i+6]}
if(i>=7){
ss[i]=s[i-6]}}
saison=rep(ss,6)
```

3. Calculer et enregistrer dans un vecteur `desaison` la série chronologique corrigée de ses variations saisonnières. Représenter graphiquement la chronique désaisonnalisée `desaison`.
4. Estimer la tendance de la série corrigée de ses variations saisonnières à l'aide d'un filtre moyenne mobile de longueur convenable et enregister la série dans un vecteur `tendance2`.
5. Donner une estimation `residus2` des résidus après retrait de la tendance `tendance2` et du mouvement saisonnier `saison`.

2.2.3 Méthode de différenciation

Eliminer la tendance et le mouvement saisonnier de `USA` par différenciations successives pour former un nouveau vecteur de résidus `residus3`.
Indication : Pour appliquer l'opérateur ∇_{12} aux données `USA`, taper la commande `diff(USA,lag=12,differences=1)`.

2.3 Tests des résidus

1. Tester si les résidus `residus1` sont des réalisations de variables aléatoires indépendantes à l'aide de la fonction d'autocorrélation empirique. Effectuer le test de portmanteau. Tester enfin la normalité des résidus.
   Indication : La commande `Box.test()` permet d'effectuer le test de portmanteau. Quant aux commandes `ks.test` et `shapiro.test()`, elles permettent de mettre en oeuvre les tests de Kolmogorov et de Shapiro.
2. Reprendre la question 1. avec `residus2` et `residus3`.