

Feuille d'exercices no 1 (durée : 3h)

Préliminaires

Créer un fichier texte dans lequel vous répondrez clairement aux questions ci-dessous, en incluant vos codes R, les résultats obtenus sous R (graphique y compris), vos interprétations, remarques ... Une fois ce TP fini, vous mettrez en forme votre compte-rendu et l'exporterez au format pdf (c'est ce qui sera demandé au partiel).

1 Données de varicelle

Récupérer le fichier contenant le nombre de cas de varicelle relevés à New-York de janvier 1931 à juin 1972 (<http://math.unice.fr/~rubentha/enseignement/varicelle.dat>).

1. Créer un objet de type série temporelle contenant cette série. Représenter graphiquement la série. (Voir appendice pour les instructions utiles en R.)
2. Analyser qualitativement cette série, c'est-à-dire repérer d'éventuelles tendances et/ou saisonnalités (changer d'échelle si besoin).
3. Quel est le nombre de cas de varicelles mensuel moyen ?
4. Tracer les 25 premières auto-corrélations. Interpréter ces résultats.
5. Tracer sur un même graphique, les évolutions mensuelles du nombre de cas de varicelle pour chaque année (une courbe pour chaque année, ce qui nous donnera un certain nombre de courbes superposées).
6. Tracer sur un graphique l'évolution annuelle du nombre de cas de varicelle.
7. Ces deux dernières questions vous permettent-elles d'améliorer vos conclusions de la question 2 ?

2 Simulations de séries temporelles

On appelle bruit blanc gaussien une suite de variables aléatoires indépendantes et identiquement distribuées $(\epsilon_t)_{t \in \mathbb{N}}$ de loi normale centrée réduite.

1. Quelle est la fonction d'auto-corrélation d'un bruit blanc ?
2. Simuler un bruit blanc gaussien de taille 100, et représenter le graphiquement.
3. Tracer la fonction d'auto-corrélation.
4. Recommencer les deux questions précédentes et observer la variabilité des résultats. Jouer sur la longueur de la série.
5. Simuler maintenant la série temporelle $X(t) = 0.5t + 2\epsilon_t$ avec $\epsilon_t \sim \mathcal{N}(0, 1)$ (taille 100).
6. Représenter graphiquement la série et interpréter-la qualitativement.
7. Faites de même pour $X(t) = 0.5t + \epsilon_t + 3 \cos(t\pi)$ avec $\epsilon_t \sim \mathcal{N}(0, 1)$.

Appendice : mise en œuvre sous R.

Quelques fonctions R utiles à l'étude des séries temporelles :

- Lire un fichier de données en sautant les k premières lignes : `data=scan(file='donnee.dat',skip=k)`. Définir le répertoire courant : `setwd("~/Documents/")` (ici on veut être dans `~/Documents`), savoir quel est le répertoire courant : `getwd()`.
- Créer un objet de type série temporelle : `serie <- ts (data,start,end,frequency)`. `data` contient le vecteur des données (un fichier contenant les données peut être mentionné en remplaçant `data` par `file='donnees.dat'`), `start` et `end` mentionne les dates de début et de fin de la série (ex : `start=c(1990,1)` et `end=c(1999,6)` pour des données allant de janvier 90 à juin 99), et enfin `frequency` mentionne le nombre de données par unité de temps (par exemple, si les dates de début et de fin sont des années, et que les données sont mensuelles, il faudra indiquer `frequency=12`). Découper une série temporelle de `c(i,j)` à `c(k,1)` : `xd<-window(x,c(i,j),c(k,1))`. Transformer une série temporelle en vecteur de nombres : `v=as.numeric(cp)`.
- Représenter graphiquement un objet de type série temporelle : `plot.ts(serie)`.
- La fonction `acf(x, lag.max = 10, type = c("correlation", "covariance"), plot = TRUE)` calcule (et trace si l'option `plot` est à `TRUE`) les `lag.max` premières auto-corrélations ou auto-covariances (choisir corrélation OU covariance).

Quelques conseils utiles pour les graphiques en R :

- Pour représenter plusieurs courbes sur le même graphique, tracer la première à l'aide de la commande `plot` qui crée la fenêtre graphique et y insère la courbe, puis tracer les autres courbes à l'aide de la commande `lines` qui trace une courbe sur une fenêtre graphique existante.
- Pour partager la fenêtre graphique en plusieurs ($n \times p$) sous-graphes, utiliser la commande `par(mfrow=c(n,p))`.
- Préciser les limites des axes des graphiques : `plot(...,xlim=c(0,10),ylim=c(-10,10))`.
- Pour exporter les graphiques en *jpeg* (idem pour *bmp*, *png*), il faut lui procéder de la sorte
 1. `jpeg(filename='nomfichier%d.jpeg')`,
 2. réaliser le graphique,
 3. la commande `dev.off()` permet enfin de rediriger le dernier graphique tracé vers le fichier `nomfichier1.jpeg`, et ainsi de suite après chaque graphique. Le nom de fichier sera automatiquement incrémenté.

Simulation de variables normales indépendantes : `x<-rnorm(n,mean=0,sd=1)` renvoie une suite de variables normales indépendantes de moyenne 0 et d'écart-type 1.