
TP 1 : Se familiariser avec les bases/notations

Exercice 1 (La gestion des dates et séries temporelles). Puisque ce cours concerne les séries temporelles, il s'agit déjà de savoir manipuler les dates avec R et définir une série temporelle (à pas de temps régulier).

1. Familiarisez vous avec la fonction `ts` de R. A titre illustratif voici un exemple d'utilisation typique.

```
## Creation d'une serie temporelle basique
y <- ts(cumsum(rnorm(19)), c(2019, 12), frequency = 365)
plot(y)
```

L'argument `frequency` définit la « fréquence » de notre série temporelle, ici journalière puisque 365 jours dans l'année. Le deuxième argument spécifie la date de début de la série temporelle au format (année, nombre de fréquence). Ici notre série temporelle contient donc les valeurs pour 19 jours consécutifs et débute le 12 janvier 2019.

- a) Définissez une série hebdomadaire débutant le 5 mars 2018 et finissant le 1er avril 2018.
 - b) Définissez une série temporelle mensuelle commençant le 1er janvier 2019 et durant une année.
2. Retrouver les caractéristiques de la série temporelle. Essayez de comprendre le rôle des fonctions suivantes
 - a) `time`;
 - b) `frequency`;
 - c) `deltat`.
 3. Gestion des dates :
 - a) Pour convertir une chaîne de caractère en date reconnue par R, il suffit d'utiliser la fonction

```
as.Date("2019/12/01", "%Y/%d/%m")
## [1] "2019-01-12"
```

Convertissez les chaînes de caractères suivantes

```
c("12/04/2018", "2jan2019", "01/01/2019 12:44")
## [1] "12/04/2018"      "2jan2019"        "01/01/2019 12:44"
```

Conseil : Allez voir l'aide de la fonction `strptime` et plus particulièrement la section « Détails » pour voir tous les formats disponibles.

- b) Nous pouvons également récupérer les jours, mois, années assez facilement. Pour cela, il est nécessaire cependant de charger (et d'installer si ce n'est pas déjà fait) le package `chron`. Apprenez à utiliser les fonctions `years`, `days`, `months`, `quarters`, `weekdays`, `julian`.



Exercice 2 (Différentiation d'une série). Vous avez probablement vu en cours que l'on pouvait différencier une série temporelle afin de supprimer une tendance. La fonction `diff` permet de faire la différenciation.

1. Lire l'aide de la fonction `diff`.
2. Essayez (enfin réussissez) de comprendre les lignes de code suivantes.

```
x <- ts((1:10)^2)
diff(x)
diff(x, lag = 2)
diff(x, differences = 2)
diff(x, 2, 2)
```

3. Considérons la série temporelle suivante

$$X_t = 10 \cos(2\pi t/4) + \epsilon_t, \quad \epsilon_1, \epsilon_2, \dots \stackrel{\text{iid}}{\sim} N(0, 1).$$

- a) Générez une réalisation de longueur $T = 40$ de cette série temporelle.
- b) Visualisez cette série temporelle ainsi que celle donnée par $(I - B^4)X_t$. Que constatez vous ?



Exercice 3 (Analyse harmonique de la saisonnalité). Petit rappel je l'espère inutile, la transformée de Fourier (discète) de la série temporelle X_1, \dots, X_T a pour coefficients

$$a_k = \sum_{t=0}^{T-1} X_{t+1} \exp\left(-\frac{2\pi ikt}{T}\right), \quad k = 0, \dots, T-1.$$

Ces coefficients coefficients sont à valeurs dans \mathbb{C} et un fort coefficient (en module) indique que la fréquence associée est importante dans le signal, i.e., si $|a_k|$ est grand alors la fréquence k/T est significative au sein de la série temporelle.

Attention : Il n'existe pas une unique paramétrisation de la transformée de Fourier, des fois le 2π est présent, des fois non. Des fois il y a la normalisation en T des fois non. Bref, il faut faire attention.

1. La fonction `spectrum` permet de tracer le spectre d'un signal / série temporelle. Par défaut, elle représente le spectre avec une échelle logarithmique sur les ordonnées. Je n'aime pas trop, mais cela permet d'avoir un intervalle de confiance de représenté. Faites donc comme bon vous semble. Etudiez la sortie des lignes de codes suivantes :

```
x <- ts(cos(2 * pi * 0:365 / 4), frequency = 4)
y <- ts(cos(2 * pi * 0:365 / 4), frequency = 12)
par(mfrow = c(2, 2))
spectrum(x)
spectrum(x, log = "no")

spectrum(y)
spectrum(y, log = "no")
```

2. On s'intéresse à la série temporelle suivante

$$X_t = 2 \cos(0.5236t) - 3 \sin(2.618t) + \epsilon_t, \quad \epsilon_t \stackrel{\text{iid}}{\sim} N(0, 1), \quad t \in \mathbb{N}.$$

- a) Générez une réalisation de cette série temporelle de taille $T = 500$ et visualisez cette série temporelle. On supposera que ce sont des données mensuelles.
 - b) Faites vous une image mentale de ce à quoi devrait ressembler le spectre de cette série temporelle.
 - c) Vérifiez si vous aviez juste via la fonction `spectrum`.
3. On s'intéresse désormais à la série temporelle suivante

$$X_t = 0.01t^\kappa + \epsilon_t, \quad \epsilon_t \stackrel{\text{iid}}{\sim} N(0, 1), \quad t \in \mathbb{N}.$$

- (a) Générez une réalisation de cette série temporelle de taille $T = 500$ avec $\kappa = 1$ et visualisez cette série temporelle. On supposera que ce sont des données mensuelles.
- (b) Visualisez le spectre de cette série temporelle.
- (c) Recommencez pour $\kappa = 2, 5, 10$. Commentez.



Exercice 4 (Traffic aérien mondial). Le jeu de données `AirPassengers` contenu de base dans R recense le nombre de voyageurs mondial (en milliers) prenant l'avion chaque mois de 1949 à 1960.

1. Visualisez la série temporelle et commentez.
2. Faites un `monthplot` et commentez.
3. Visualisez le spectre de la série temporelle.
4. Utilisez la fonction `stl` ou `decompose` pour décomposer cette série temporelle.



Exercice 5 (Varicelle aux USA).

1. Importez les données du fichier `varicelle.dat`
2. Visualisez cette série temporelle. Que constatez vous ?
3. Utilisez la fonction `monthplot` et analysez.
4. Faites une analyse harmonique de la série temporelle et identifiez la saisonnalité.



Exercice 6 (Visualisation du réchauffement climatique). Rendez vous sur la page du cours et téléchargez le jeu de données nommé `temperatures_globe.txt`. Ce dernier recense les anomalies de températures sur le globe (moyenne sur le monde entier, période de référence 1951-1980) depuis 1880 jusqu'à nos jours.

1. Lisez les quelques lignes d'information en début de fichier puis importez les sous R.
2. Utilisez les fonctions `frequency`, `cycle` and `co` pour voir si vous obtenez les sorties attendues.
3. Faites un graphe de la série temporelle. Que constatez vous ?
4. Lancez le code suivant (on suppose ici que vos données sont stockées sous le nom `data`)

```
d <- stl(data, "periodic")
plot(d)
```

Que fait cette fonction ?

