POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

MTH8304 - Apprentissage non-supervisé et séries chronologiques

Département de Mathématiques et Génie Industriel
Hiver 2024
www.moodle.polymtl.ca

Devoir 5 : remise le 15 avril avant minuit

Les objectifs de ce devoir sont de se familiariser avec :

- les analyses graphiques pour mettre en évidence les composantes d'une série chronologique ;
- la décomposition STL pour analyser et faire la prévision ;
- le processus ARIMA et le lissage exponentiel pour faire la prévision.

Ce devoir noté sur 10 comporte deux parties valant respectivement 3 points et 5 points, 1 point étant attribué à la qualité de la présentation du rapport (sous forme de présentation) et 1 point pour le code à fournir en annexe.

Dans tout le devoir, assurez-vous de n'utiliser que la librairie `fpp3` qui accompagne le manuel de référence pour la partie du cours sur les séries chronologiques.

Partie A : analyses préliminaires et modèle de référence

Dans cette partie, l'évaluation porte sur

(1) l'illustration et la discussion des caractéristiques de la série chronologique à modéliser ;
(2) la justification des choix de paramétrage utilisé pour la décomposition STL ;
(3) la mise en place d'une approche de référence simple.

Préparation : lecture des données

Vous allez travailler sur des séries mensuelles d'anomalies de température en degrés Celsius. Les anomalies sont des déviations par rapport aux valeurs moyennes calculées sur la période de référence 1951-1980. Ces données sont mises à disposition via le portail web https://data.giss.nasa.gov/gistemp/. Elles représentent une valeur moyenne sur une maille de 2° par 2°. Un certain nombre de mailles se trouvant à différents endroits au Canada vous est mis à disposition avec le devoir. Il vous faut d'abord choisir aléatoirement une maille et vous créer une série d'entraînement :

```
library(fpp3)
library(ggplot2)

load("~/Data/cours/dev5_tanomaly.RData") ## adapter le chemin
set.seed(monmatricule) # remplacer par votre matricule
ind.cell <- sample(ncol(tanomaly.train)-1, 1) ## selection de maille

## pour visualiser où se trouve la maille
ggplot(lonlat.coord[ind.cell,], aes(x = lon, y = lat)) + geom_point(size = 2)+
    borders(xlim = c(-140, -50), ylim = c(40, 89))


tanomaly.series <- tsibble(
    Date = tanomaly.train[,1],
    Observation = tanomaly.train[,ind.cell+1],
    index = Date
    )
```

Créez-vous une série de test `tanomaly.series.test` en adaptant le code ci-dessus afin de sélectionner la même maille dans l'objet `tanomaly.test` qui contient une année réservée pour le test.

Élément (1) de l'évaluation

Vous allez réaliser des analyses graphiques pour mettre en évidence les composantes de la série chronologique `tanomaly.series`. Pour cela, adaptez le code ci-dessous :

```
## graphique temporel
autoplot(tanomaly.series, Observation, linewidth = 1.2)+
  labs(y = "degres C", x = "mois") + theme(text = element_text(size=20))

## deux types de graphiques saisonniers
tanomaly.series |> gg_season(Observation, labels = "both", linewidth = 1.2) +
  labs(y = "degres C",x = "mois")+ theme(text = element_text(size=20))

tanomaly.series |> gg_subseries(Observation) +  theme(text = element_text(size=20))

## graphique de lag: tester plusieurs lags en changeant la valeur de l'argument "lags"
tanomaly.series |> gg_lag(Observation, geom = "point", lags = 1) +
    theme(text = element_text(size=20),legend.title=element_blank())

## graphique d'auto-corrélation
tanomaly.series |> ACF(Observation, lag_anomaly = 24) |> autoplot()
```

Pour ces analyses graphiques, l'important est de commenter les caractéristiques (structures récurrentes ou motifs) que vous identifiez dans chacun des types de graphiques en gardant en tête que ces caractéristiques vont servir à la modélisation et à la prévision.

Élément (2) de l'évaluation

Vous allez ensuite effectuer une décomposition STL et tester les choix de paramétrages : faut-il une composante saisonnière périodique ou est-il préférable de la laisser varier ? quelles valeurs sont plus adéquates pour les paramètres de lissage pour la tendance et la composante saisonnière, si celle-ci n'est pas périodique ? Pour déterminer un choix de paramétrage adéquat, vous devez évaluer si la composante résiduelle du modèle peut être considérée ou non comme un bruit blanc en vous basant sur l'auto-corrélation empirique.

Vous pouvez vous inspirez du code ci-dessous pour faire des essais de choix de paramétrage. Dans le code, la composante saisonnière est périodique. Utilisez une composante saisonnière qui s'adapte en changeant la valeur de l'argument `window` (paramètre de lissage) de la fonction `season` pour des valeurs numériques associées au niveau de lissage. Testez aussi d'autres valeurs pour le paramètre de lissage de la tendance (argument `window` de la fonction `trend`).

```
## Décomposition STL stockée dans l'objet tanomaly.stl
tanomaly.series |> model(STL(Observation ~ trend(window = 12) + season(window = "periodic"), robust = TRUE))
     |> components() -> tanomaly.stl

## Graphique standard de la décomposition
autoplot(tanomaly.stl)

## Graphique de l'ACF de la composante résiduelle
tanomaly.stl  |> ACF(remainder, lag_anomaly = 24) |> autoplot()
```

Incluez dans votre rapport les choix de paramétrages qui sont les plus informatifs, en particulier pour évaluer si le niveau de lissage retenu est acceptable et si la fenêtre saisonnière doit être périodique.

Élément (3) de l'évaluation

Vous allez mettre en place une approche de référence simple : la méthode saisonnière naïve. Utilisez le code ci-dessous :

```
## Méthode naïve saisonnière stocké dans l'objet tanomaly.benchmark
tanomaly.benchmark <- tanomaly.series  |> model('Naive saison.' = SNAIVE(Observation))

## Prévision sur 12 mois avec le modèle naïf
tanomaly.benchmark.fc <- tanomaly.benchmark |> forecast(h = 12)

## Graphique avec intervalle de prévision 95%
tanomaly.benchmark.fc |> autoplot(tanomaly.series, level = 95, linewidth = 1.2) +
  autolayer(tanomaly.series.test, Observation,linewith = 1.2, linetype = 2)+
  autolayer(tanomaly.series, Observation, linewidth = 1.2)+
  labs(y = "degres C", x = "mois") + theme(text = element_text(size=20))
```

Ce modèle vous servira de référence en termes de comparaison avec les approches développées dans la partie B.

Partie B : modélisation, prévision et comparaison

Dans cette partie, l'évaluation porte sur

(1) l'évaluation de l'adéquation d'un modèle en analysant les résidus du modèle ;

(2) l'évaluation et la comparaison de plusieurs méthodes de prévision ;

(3) le calcul détaillé de la prévision par un modèle ARIMA.

Élément (1) de l'évaluation

Pour modéliser votre série chronologique, vous allez utiliser la décomposition STL de la partie A. Stockez dans l'objet `tanomaly.stl` la décomposition STL de la partie A qui vous semble la plus pertinente en termes de simplicité et des propriétés de la composante résiduelle, puis :

```
## L'objet tanomaly.stl.fc va servir pour la suite de la modélisation:
tanomaly.stl.fc <- tanomaly.stl |> select(-.model)
```

Dans la décomposition STL, la composante saisonnière sera estimée par la méthode naïve et vous allez tester trois méthodes pour la prévision de la série dé-saisonnalisée (tendance + composante résiduelle). Les trois méthodes (ou modèles) sont : la méthode naïve, la méthode de Holt ou lissage exponentielle avec tendance et le modèle ARIMA. Pour chacune de ces trois méthodes, vous devez d'abord déterminer si les résidus ont les propriétés attendus. Inspirez-vous de la démarche ci-dessous appliquée à la méthode naïve :

```
## Méthode naïve: analyse graphique des résidus
tanomaly.stl.fc |> model(NAIVE(season_adjust)) |> gg_tsresiduals()

## Création d'un objet contenant les résidus
tanomaly.sadj.naive <-  tanomaly.stl.fc |> model(NAIVE(season_adjust)) |> augment()

## Test portmanteau de Ljung-Box avec 10 lags
tanomaly.sadj.naive |> features(.innov, ljung_box, lag = 10)
```

Pour adapter la démarche ci-dessus aux autres modèles, changez l'argument de la fonction `model` dans le code par :

```
## Méthode de Holt
AAN=ETS(season_adjust ~ error("A") + trend("A") + season("N"))

## Modèle ARIMA
ARIMA(season_adjust)
```

À partir de ces analyses (graphique et test statistique de portmanteau), vous devez établir quels modèles, parmi les trois considérés, fournissent des résidus ayant les propriétés adéquates pour la modélisation. Ce seront les modèles à retenir pour la prochaine étape.

Élément (2) de l'évaluation

Vous allez comparez en termes de performance de prévision sur l'ensemble des tests les modèles retenus à l'étape précédente. Parmi les informations à prendre en compte, inspirez-vous du code ci-dessous :

```
## Méthode naïve: prévision de la série dé-saisonnalisée
tanomaly.stl.fc |> model(NAIVE(season_adjust)) |> forecast(h = 12) |> autoplot(tanomaly.stl.fc)

## Création d'un objet contenant le modèle de prévision basé sur la décomposition STL
fit.tanomaly.stl <- tanomaly.series |>
  model(stlf = decomposition_model(
    STL(Observation ~ trend(window = 12) + season(), robust = TRUE),
    NAIVE(season_adjust)))

## Pour afficher les paramètres du modèle
report(fit.tanomaly.stl)

## Prévision de la série complète avec le modèle basé sur la décomposition STL
fit.tanomaly.stl |> forecast(h = 12) |> autoplot(tanomaly.series, level = 95)

## Calcul des mesures de perfomance sur la série de test
fit.tanomaly.stl |> forecast(h = 12) |> accuracy(tanomaly.series.test)
```

Comme lors des analyses des résidus de la première étape, remplacez dans le code ci-dessus la méthode naïve `NAIVE(season_adjust)` par le code adapté pour les modèles que vous avez retenus (par exemple, `ETS` pour le modèle de Holt et `ARIMA` pour le modèle ARIMA). Dans votre rapport, rassemblez les mesures de performances pertinentes pour les différents modèles considérés dans un tableau et présentez des graphiques mettant en évidence les propriétés des modèles. N'oubliez pas d'inclure la modèle de référence de la partie A, c'est-à-dire la méthode naïve saisonnière. Déterminez le modèle qui vous paraît le plus approprié pour la prévision et justifiez.

Élément (3) de l'évaluation

Répondez à l'exercice # 16 de la section 9.11 du livre fpp3.