2ème année ENSAE

Projet de Série temporelles linéaires

Modélisation ARIMA d'un indice de production industrielle

Thomas Aujoux, Eléa Bordais

Second semestre 2022-2023

Sommaire

1 Les données

1.1 Présentation de la série choisie

Nous avons décidé d'étudier la production industrielle de glaces et sorbets [1]. L'Insee arrive à calculer les indices de la production industrielle à partir de données obtenues à l'aide d'enquêtes mensuelles effectuées auprès d'un échantillon d'entreprises dans la France entière. Les enquêtes de branche portent principalement sur les quantités produites ou livrées et les facturations. Les séries sont corrigées des variations saisonnières (CVS) et des effets de calendrier (CJO). L'estimation de ces effets est effectuée avec la méthode X13-Arima. Les indices publiés ont pour année de référence 2015, ce qui signifie que les indices ont pour moyenne 100 en 2015. Pour cette étude, nous utiliserons la méthodologie de Box-Jenkins qui consiste à identifier et à estimer un modèle ARIMA. Elle comporte plusieurs étapes : la stationnarisation, l'identification du modèle, l'estimation des paramètres et la vérification du modèle. La série prend des valeurs mensuelles de Janvier 2005 à Janvier 2023. Nous traçons la série initiale :

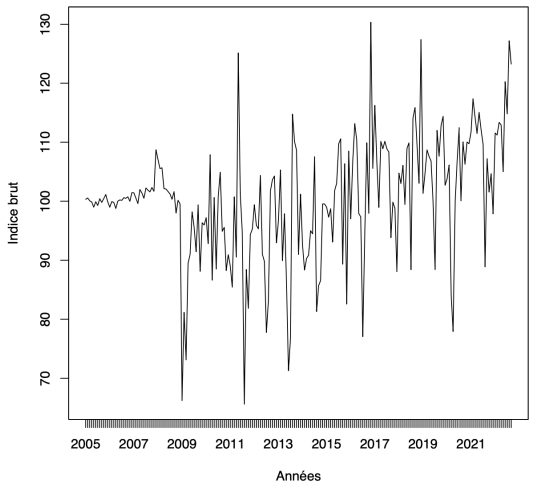

FIGURE 1: Représentation graphique de l'indice de production industrielle de glaces et sorbets

Nous remarquons que notre série n'est pas stationnaire. Pour s'en assurer, nous réaliserons des tests unitaires. Mais avant de procéder aux tests de racine unitaire, nous vérifions s'il y a une constante et/ou une tendance linéaire non nulle. Pour cela, nous régressions *indice brut* sur ses dates pour le vérifier. Le coefficient associé à la tendance linéaire (*dates*) et le coefficient associé à la constante (*intercept*) sont tous deux non-nuls et significatifs. Il faudra donc se mettre dans le cas des tests de racine unitaire avec constante et tendance non nulles.

Nous réalisons ensuite le test de Dickey-Fuller augmenté (ADF). Ce test, dans le cas avec constante et tendance, consiste en la régression suivante, pour une variable X donnée :

$$\Delta X_t = c + bt + \beta X_{t-1} + \sum_{l=1, k>0}^{k} \phi_l \Delta X_{t-l} + \epsilon_t \tag{1}$$

où $\beta+1$ est l'autocorrélation à l'ordre 1 de X et k le nombre de retards nécessaires à considérer pour rendre les résidus non autocorrélés. L'hypothèse nulle de racine unitaire $H_0 : \beta = 0$ est testée par la statistique de test $\hat{\beta}/\hat{\sigma}(\hat{\beta})$ qui suit une loi de Dicker-Fuller dépendant du nombre d'observation et du cas dans lequel on se trouve.

1. La série choisie est disponible sur le site de l'Insee en passant par le lien suivant : `https://www.insee.fr/fr/statistiques/serie/010537266`.

Cependant, avant d'interpréter le test, nous vérifions que les résidus du modèle sont bien non autocorrélés sans, sinon le test n'est pas valide. Dans notre cas, l'absence d'autocorrélation des résidus est rejetée au moins une fois, le test ADF avec aucun retard n'est donc pas valide. Nous rajoutons donc des retards jusqu'à ce que les résidus sont non corrélés (dans notre cas 7 retards). Ensuite, nous affichons les résultats du test valide maintenu.

```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 7
  STATISTIC:
    Dickey-Fuller: -2.8185
  P VALUE:
    0.2327
```

FIGURE 2: Résultat du test ADF de notre série

La racine unitaire n'est pas rejetée à un seuil de 95% pour la série en niveau. On rejette donc la stationnarité de la série.

1.2 Transformation de la série

Nous allons maintenant transformer la série afin de la rendre stationnaire. Pour cela, nous différencions notre série à l'ordre 1, nous obtenons ainsi la représentation suivante :

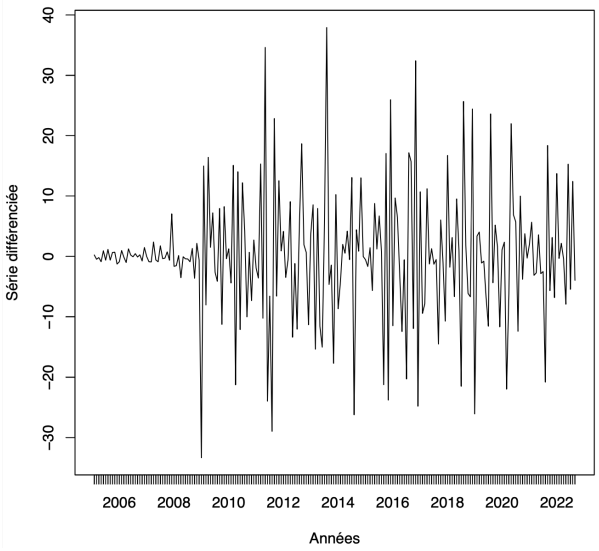

FIGURE 3: Représentation graphique de la série différenciée à l'ordre 1

Ainsi, la série différenciée d'ordre 1 semble stationnaire et évoluer autour d'une même moyenne proche de 0. Pour s'en assurer, nous utilisons la même méthodologie que nous venons d'utiliser pour valider la stationnarité de la série (régression sur *dates*, calcul du nombre de retard, test ADF). La p-valeur obtenue du test ADF est de 0.01, donc le test rejette la racine unitaire, donc on dira que la série différenciée à l'ordre 1 est stationnaire. Il n'est donc pas nécessaire de calculer une différentiation supplémentaire. De plus, nous n'observons pas de saisonnalité puisque elle a déjà été corrigée dans les données.

2 Modèles ARMA

2.1 Choix du modèle ARMA et validation

Nous cherchons ensuite le modèle ARMA (p, q) qui correspond le plus à notre série transformée. Afin d'identifier les paramètres de notre modèle ARMA, nous utiliserons la fonction d'auto-corrélation (ACF) pour trouver l'ordre q maximal et la fonction d'auto-corrélation partielle (PACF) pour trouver l'ordre p maximal.

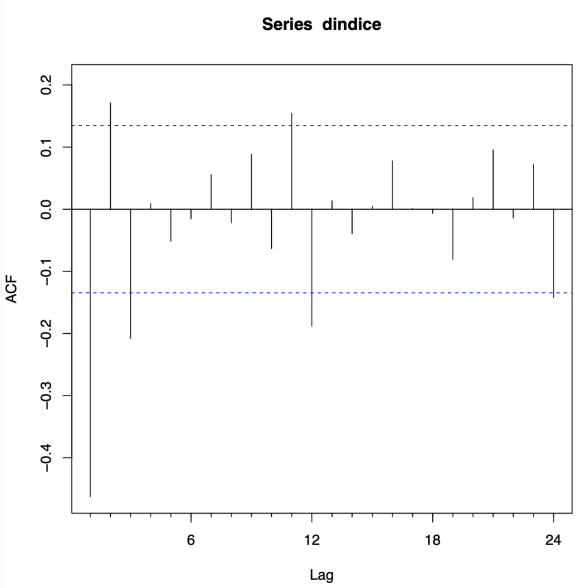

FIGURE 4: Fonction d'auto-corrélation

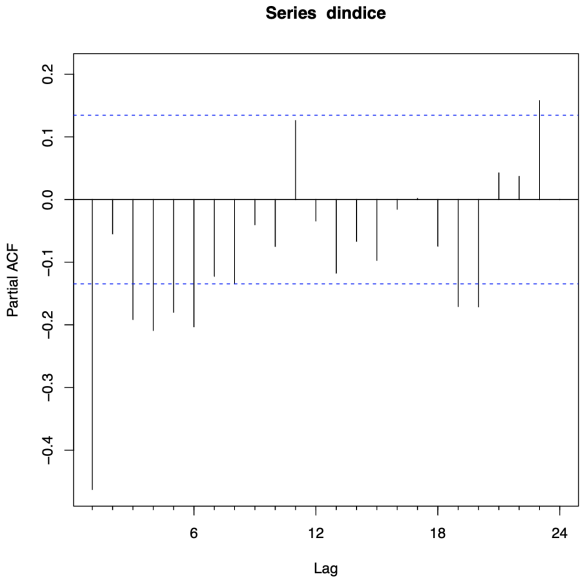

FIGURE 5: Fonction d'auto-corrélation partielle

Nous remarquons que pour l'ACF (Figure 4), les 3 premiers retards présente une auto-corrélation totale significative. On prendra donc $q_{max} = 3$. D'autre part, pour le PACF (Figure 5), les y premiers retards présentent une auto-corrélation partielle significative. On prendra donc $p_{max} = 6$.

Les modèles probables sont donc l'ensemble des modèles tel que $p \leq 6$ et $q \leq 3$. On doit donc maintenant décider du meilleur modèle. Pour cela, nous allons nous servir de deux tests :

- le test de nullité des coefficients : si le test est validé, nous dirons que le modèle est bien ajusté
- le test d'absence d'auto-corrélation des résidus ou test du portemanteau : si le test est validé, nous dirons que le modèle est valide (ici, nous souhaitons accepter H_0 donc nous vérifions que les p-valeurs sont supérieures à $0,1$)

Ainsi, seuls les modèles ARMA$(4,1)$, ARMA$(5,2)$ et ARMA$(1,3)$ sont valides et ajustés. Pour choisir entre ces 3 modèles, nous nous intéressons au modèle qui minimise les deux critères d'informations à savoir l'AIC et le BIC, dont on trouvera les valeurs dans la table 1.

	ARMA(4,1)	**ARMA(5,2)**	**ARMA(1,3)**
AIC	1529.172	1533.061	1530.024
BIC	1552.668	1563.270	1550.163

TABLE 1: Valeurs des critères d'informations AIC et BIC pour les trois modèles retenus

On remarque que le modèle ARMA(4,1) minimise le critère AIC tandis que le modèle ARMA(1,3) minimise le critère BIC. Nous allons donc utiliser le calcul du R^2 ajusté pour choisir entre ces deux modèles. Nous obtenons les résultats suivants :

$$\begin{cases} R^2_{adjusted,ARMA(4,1)} = 0.3573 \\ R^2_{adjusted,ARMA(1,3)} = 0.3515 \end{cases}$$

Le modèle ARMA(4,1) a le plus grand R^2 ajusté, nous retiendrons donc ce modèle.

2.2 Modèle ARIMA

Dans cette partie, il s'agit de montrer que le modèle ARMA(4,0,1) que l'on a pour la série différenciée est bien causal. Or un ARMA est causal si le polynôme associé n'a pas de racine à l'intérieur du cercle unité. On extrait les racines à l'aide de la fonction *polyroot*, puis on utilise la fonction *plot.Arima* qui produit un graphique des inverses des racines AR et MA. Comme vu sur la figure 6, les inverses des racines sont dans le cercle unité, donc les racines seraient en dehors du cercle. Le modèle ARMA pour la série différenciée est causal. Par définition d'un ARIMA, la série non transformée suit un modèle ARIMA(4,1,1).

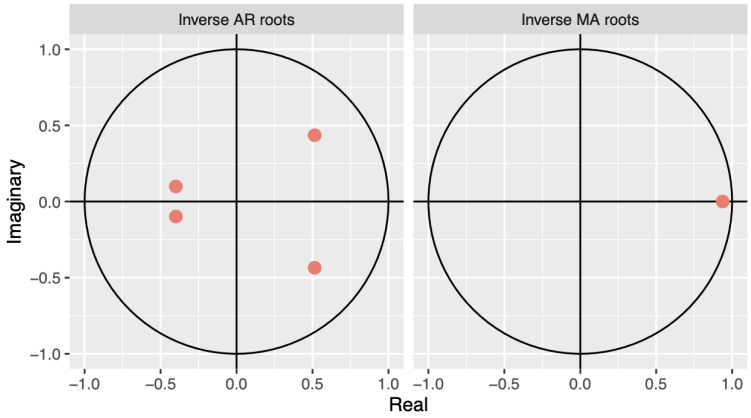

FIGURE 6: Représentation des inverses des racines sur le cercle unité

3 Prévision

3.1 Hypothèses de région de confiance

Le calcul de la région de confiance de niveau α pour les valeurs futures repose sur deux grandes hypothèses :

- Les innovations ϵ_t sont des bruits blancs forts et gaussiens : $\epsilon \sim \mathcal{N}\left(0, {\sigma_\epsilon}^2\right)$ (iid) avec $\sigma^2_\epsilon > 0$
- Le modèle théorique est identifié, c'est à dire que les coefficients théoriques sont identiques aux coefficients estimés. La variance estimée $\hat{\sigma^2_\epsilon}$ est égale à la variance théorique σ^2.

3.2 Région de confiance

On note X_t notre série différenciée. Soit T sa longueur. X_t suit un modèle ARMA(4,1) donc elle vérifie :

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \phi_3 X_{t-3} - \phi_4 X_{t-4} = \epsilon_t - \psi_1 \epsilon_{t-1} \tag{2}$$

La meilleure prévision linéaire de X_{t+1} est ${}_t\widehat{X}_{t+1}$ telle que :

$$\begin{aligned} {}_t\widehat{X}_{t+1} &= EL(X_{t+1}|X_t, X_{t-1}, ..., X_1) \\ &= EL(\phi_1 X_t + \phi_2 X_{t-1} + \phi_3 X_{t-2} + \phi_4 X_{t-3} + \epsilon_{t+1} - \psi_1 \epsilon_t | X_t, X_{t-1}, ..., X_1) \\ &= \phi_1 X_t + \phi_2 X_{t-1} + \phi_3 X_{t-2} + \phi_4 X_{t-3} - \psi_1 \epsilon_t \end{aligned}$$

car $EL(\epsilon_{t+1}|X_t, X_{t-1}, ..., X_1) = 0$ car ϵ_t est une innovation et $EL(\epsilon_t|X_t, X_{t-1}, ..., X_1) = \epsilon_t$

De même, la meilleure prévision linéaire de X_{t+2} est ${}_t\widehat{X}_{t+2}$ telle que :

$$\begin{aligned} {}_t\widehat{X}_{t+2} &= EL(X_{t+2}|X_t, X_{t-1}, ..., X_1) \\ &= EL(\phi_1 X_{t+1} + \phi_2 X_t + \phi_3 X_{t-1} + \phi_4 X_{t-2} + \epsilon_{t+2} - \psi_1 \epsilon_{t+1} | X_t, X_{t-1}, ..., X_1) \\ &= \phi_1 {}_t\widehat{X}_{t+1} + \phi_2 X_t + \phi_3 X_{t-1} + \phi_4 X_{t-2} \end{aligned}$$

Calculons maintenant les erreurs de prédiction, définies comme ceci : $e_{t+h} = X_{t+h} - {}_t\widehat{X}_{t+h}$

$e_{t+1} = X_{t+1} - {}_t\widehat{X}_{t+1} = \epsilon_{t+1}$

$e_{t+2} = \phi_1(X_{t+1} - {}_t\widehat{X}_{t+1}) + \epsilon_{t+2} - \psi_1 \epsilon_{t+1} = \phi_1 \epsilon_{t+1} + \epsilon_{t+2} - \psi_1 \epsilon_{t+1} = (\phi_1 - \psi_1)\epsilon_{t+1} + \epsilon_{t+2}$

Les résidus sont gaussiens et décorrélés, on en déduit donc la variance et la covariance des erreurs de prédiction :

$$\begin{cases} \mathbb{V}(e_{t+1}) = \mathbb{V}(\epsilon_{t+1}) = {\sigma_\epsilon}^2 \\ \mathbb{V}(e_{t+2}) = \mathbb{V}((\phi_1 - \psi_1)\epsilon_{t+1}) + \mathbb{V}(\epsilon_{t+2}) + 2Cov((\phi_1 - \psi_1)\epsilon_{t+1}, \epsilon_{t+2}) = (1 + (\phi_1 - \psi_1)^2){\sigma_\epsilon}^2 \\ Cov(e_{t+1}, e_{t+2}) = Cov(\epsilon_{t+1}, (\phi_1 - \psi_1)\epsilon_{t+1} + \epsilon_{t+2}) = (\phi_1 - \psi_1){\sigma_\epsilon}^2 \end{cases}$$

Ainsi d'après les hypothèses faites à la question précédente, on a :

$$\begin{pmatrix} e_{t+1} \\ e_{t+2} \end{pmatrix} = \begin{pmatrix} \epsilon_{t+1} \\ (\phi_1 - \psi_1)\epsilon_{t+1} + \epsilon_{t+2} \end{pmatrix} \sim \mathcal{N}(0, \Sigma) \text{ où } \Sigma = \begin{pmatrix} {\sigma_\epsilon}^2 & (\phi_1 - \psi_1){\sigma_\epsilon}^2 \\ (\phi_1 - \psi_1){\sigma_\epsilon}^2 & (1 + (\phi_1 - \psi_1)^2){\sigma_\epsilon}^2 \end{pmatrix}$$

Donc $\text{Det}(\Sigma) = {\sigma_\epsilon}^4 > 0$ d'après l'hypothèse faite, donc Σ est une matrice inversible.

On note $X := \begin{pmatrix} X_{t+1} \\ X_{t+2} \end{pmatrix}$ et $\hat{X} := \begin{pmatrix} {}_t\widehat{X}_{t+1} \\ {}_t\widehat{X}_{t+2} \end{pmatrix}$

La région de confiance bivariée de niveau α s'écrit :

$$R_{1-\alpha} = \left\{ x \quad | \quad (x - \hat{X})' \Sigma^{-1} (x - \hat{X}) \leq q_{\chi^2(2)}(1 - \alpha) \right\}$$

3.3 Représentation graphique de la région de confiance

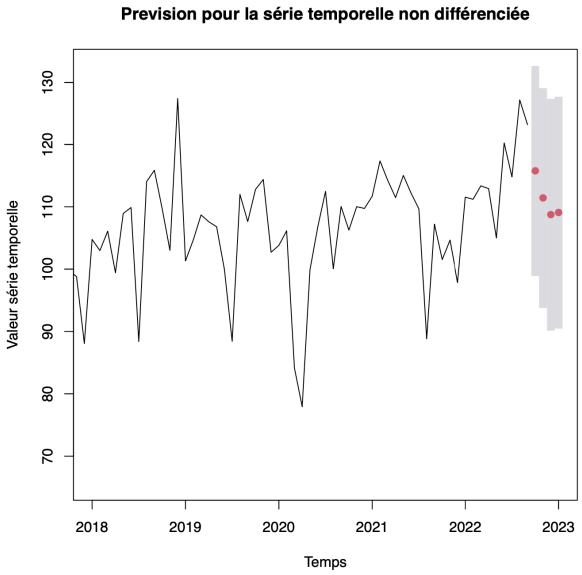

FIGURE 7: Prévision série non-différenciée

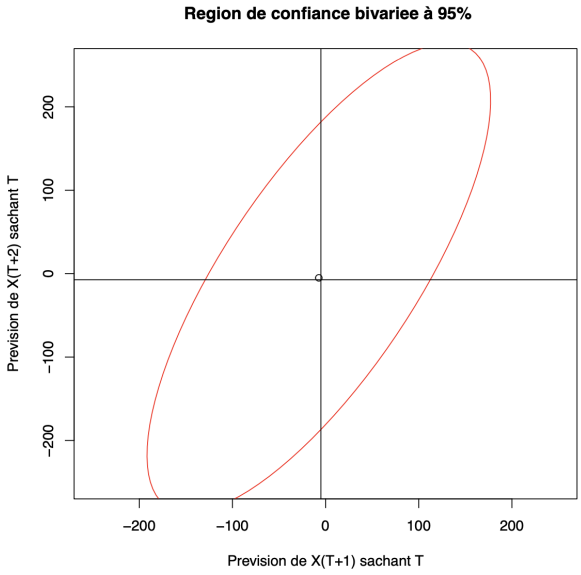

FIGURE 8: Région de confiance bivariée à 95%

La figure 7 représente les prévisions pour les 4 prochaines valeurs de notre série non-différenciée. Nous avons obtenu à l'aide de la fonction *forecast* que nous avons appliqué à notre modèle ARIMA (4,1,1). La figure 8 représente la région de confiance bivariée au niveau 95% de X_{t+1} et X_{t+2} (notre série différenciée).

Nous remarquons que notre ellipse est très grande, cela est peut-être du au fait que nous avons utilisé une hypothèse selon laquelle nos résidus sont gaussiens, hypothèse que nous n'avons pas pu vérifier en pratique. Vous trouverez en annexe (A) la démarche que nous avons suivi pour valider ou non l'hypothèse faite sur les résidus.

3.4 Question ouverte

Si l'on suppose que Y_{T+1} est disponible plus rapidement que X_{T+1}, alors on peut utiliser l'information de Y_{T+1} afin d'améliorer la prédiction de X_{T+1} si et seulement si la variable (Y_t) cause instantanément au sens de Granger la variable (X_t).

Dans notre cas, Y_t cause X_t au sens de Granger si et seulement si :

$$\hat{X}_{T+1|\{X_u,Y_u,u\leq T\}\cup\{Y_{T+1}\}} \neq \hat{X}_{T+1|\{X_u,Y_u,u\leq T\}} \tag{3}$$

Afin de vérifier cette hypothèse, nous pouvons utiliser un test de Granger. Le test compare les prédictions de X_t avec et sans l'information supplémentaire, ici Y_{T+1}.

4 Annexes

A Hypothèse sur les résidus

Nous avons voulu vérifier l'hypothèse faites dans la partie 3.1, celle selon laquelle les résidus sont gaussiens.

Nous avons commencé par tracer sur un même graphique la densité des résidus et la densité d'une loi normale en prenant la moyenne et l'écart type empirique. On remarque sur le graphique 10 que la densité de nos résidus n'est pas une densité d'une loi normale. Nous avons également tracé le graphique Q-Q Plot (quantile-quantile plot) 9. Ce graphique est un "nuage de points" qui vise à confronter les quantiles de la distribution empirique et les quantiles d'une distribution normale, de moyenne et d'écart type estimés sur les valeurs observées. Si la distribution est compatible avec la loi normale, les points forment une droite. On remarque que notre nuage de points ne forment pas complètement une droite. À partir de ces deux observations, nous pouvons avoir des doutes sur l'hypothèse de normalité des résidus.

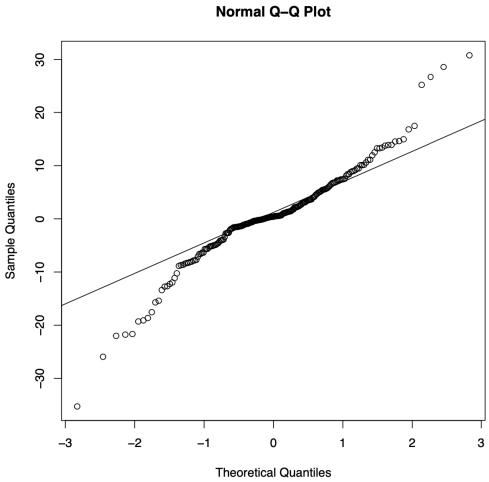

FIGURE 9: Q-Q PLOT

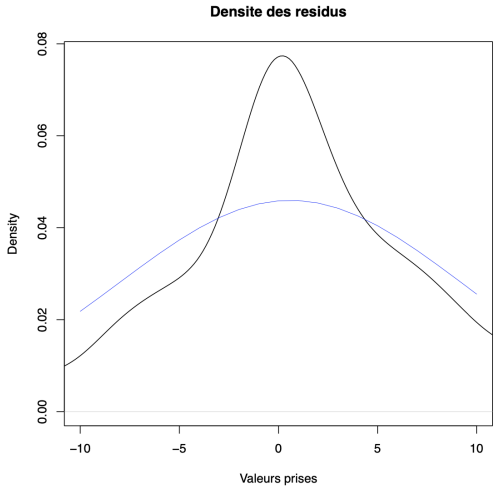

FIGURE 10: Densité des résidus

Pour vérifier nos intuitions qualitatives, nous avons réalisé un test statistique de JARQUE ET BERA (1980). C'est un test d'hypothèse qui cherche à déterminer si des données suivent une distribution normale. Il est basé sur la statistique JB qui est donné par :

$$JB = \frac{N-k}{6}\left(S^2 + \frac{(K-3)^2}{4}\right) \tag{4}$$

- S est le coefficient de Sknewness qui représente une mesure d'asymétrie de la distribution autour de la moyenne
- K est le coeffient de Kurtosis qui mesure l'aplatissement de la distribution de la variable
- N est le nombre d'observations
- k représente le nombre paramètres estimés

L'hypothèse à tester est H_0 contre H_1 :

$$\begin{cases} H_0 = \text{les données suivent une loi normale} \\ H_1 = \text{les données ne suivent pas une loi normale} \end{cases}$$

Le test nous indique que notre statistique de test de 63.056 et que la p-valeur du test est 2.03e-14. Dans ce cas, nous rejetons, au seuil de 5%, l'hypothèse nulle selon laquelle les données sont normalement distribuées.

B Code R

```r
# Projet : séries temporelles Thomas Aujoux Elea Bordais
# Le projet peut être exécuter directement sur Rstudio sans modifications excepté le
↪  téléchargement de certains packages.




#--------------Imports des packages--------------------
rm(list=ls())

#install.packages("fUnitRoots") #tests de racine unitaire plus modulables
# Installer les packages non installés pour la suite

library(zoo) #format de serie temporelle pratique et facile d'utilisation (mais plus
↪  volumineux)
library(tseries) #diverses fonctions sur les series temporelles
library(fUnitRoots) # Pour les racines des polynômes
library("polynom")
library("ggplot2") # Pour l'affichage des courbes
library(forecast) # Pour la prédiction, dernière partie
library(car)
library(ellipse)


#---------------Mise en place l'espace de travail--------------
path <- "C:/Users/thoma/Documents/GitHub/serie_temporelle"
setwd(path) #definit l'espace de travail (working directory ou "wd")



#---------------Import des données--------------
datafile <- "donnees.csv" #definit le fichier de donnees
data <- read.csv(datafile,sep=";") #importe un fichier .csv dans un objet de classe
↪  data.frame #fichier csv



#---------------Traitement des données--------------
data_total = data[4:220,1:2]
data_debut = data[4:218,1:2] # On enlève la dernière colonne avec les codes et les 4
↪  premières lignes inutiles
data_fin = data[219:220,1:2] # On enlève les deux dernières données pour la partie III

colnames(data_total) <- c("dates","indice")
colnames(data_debut) <- c("dates","indice")
colnames(data_fin) <- c("dates","indice")
```

```r
dates_total <- as.yearmon(seq(from=2005, to=2023, by=1/12))
dates_debut <- as.yearmon(seq(from=2005, to=2022+10/12, by=1/12))
dates_fin <- as.yearmon(seq(from=2022+11/12, to=2023, by=1/12))
indice <- zoo(as.numeric(data_debut$indice), order.by=dates_debut) #convertit les
↪  premiers éléments de data en serie temporelle de type "zoo"
last_points <- zoo(as.numeric(data_fin$indice), order.by=dates_fin) #convertit les
↪  derniers éléments de data en serie temporelle de type "zoo"

T <- length(indice)


#---------------PARTIE 1--------------
#---------------Question 1--------------
# On trace la série temporelle pour faire les premières observations qualitatives.
plot.ts(indice, xlab="Années", ylab="Indice brut")
# commentaire : la série n'est pas stationnaire
# La série en niveau semble être très persistante et semble avoir une tendance linéaire
↪  croissante. A confirmer par la suite avec des analyses quantitatives.

dindice <- diff(indice,1)
plot.ts(dindice, xlab="Années", ylab="lag Indice brut")
# La série en différence première semble être stationnaire.
# La série est probablement I(1) (à vérifier, c'est seulement une première analyse en
↪  regardant uniquement la courbe).


#---------------Question 2--------------
# Etape 1 : Analyse qualitative de la non stationnarité de la série différenciée.
acf(dindice)  #trace les fonctions d'autocorrélation totale.
dev.print(device = png, file = "./Images_pour_rapport/acf_dindice.png", width = 600)
pacf(dindice)  #trace les fonctions d'autocorrélation partielle.
dev.print(device = png, file = "./Images_pour_rapport/pacf_dindice.png", width = 600)
# L'autocorrélation d'ordre 1 (totale ou partielle, c'est la même chose) est d'environ
↪  -0.45, soit petite et loin d'être égales à 1.
# La série semble donc stationnaire. Vérification du travail précédent.

# Etape 2 : Choix pour le test ADF
# Avant de procéder aux tests de racine unitaire, il convient de vérifier s'il y a une
↪  constante et/ou une tendance linéaire non nulle.
# La représentation graphique de spread a montré que la tendance est probablement
↪  positive linéaire.
summary(lm(indice ~ dates_debut))
# Le coefficient associé à la tendance linéaire (dates) est bien positif, et peut-être
↪  significatif
# (on ne peut pas vraiment le confirmer car le test n'est pas valide en présence de
↪  résidus possiblement autocorrélés).
# Il faudra donc se mettre dans le cas des tests de racine unitaire avec constante et
↪  éventuellement tendance non nulles.
```

```r
# Etape 3 : Choix pour le nombre de lags
# Vérifions que les résidus du modèle de régression sont bien non autocorrélés, sans
↪  quoi le test ADF ne serait pas valide.
# Comme la série est mensuelle, testons l'autocorrélation des résidus jusqu'à l'ordre
↪  24 (deux ans), sans oublier de corriger les degrés de libertés du nombre de
↪  régresseurs.
# tests d'autocorrélation des résidus
Qtests <- function(series, k, fitdf=0) {
  pvals <- apply(matrix(1:k), 1, FUN=function(l) {
    pval <- if (l<=fitdf) NA else Box.test(series, lag=l, type="Ljung-Box",
    ↪  fitdf=fitdf)$p.value
    return(c("lag"=l,"pval"=pval))
  })
  return(t(pvals))
}

adfTest_valid <- function(series, kmax, adftype){
  k <- 0
  noautocorr <- 0
  while (noautocorr==0){
    cat(paste0("ADF with ",k," lags: residuals OK? "))
    adf <- adfTest(series, lags=k, type=adftype)
    pvals <- Qtests(adf@test$lm$residuals, 24, fitdf =
    ↪  length(adf@test$lm$coefficients))[,2]
    if (sum(pvals<0.05,na.rm=T)==0) {
      noautocorr <- 1; cat("OK \n")
    } else cat("nope \n")
    k <- k+1
  }
  return(adf)
}

adf <- adfTest(indice, lag=0, type="ct")  #test ADF dans le cas avec constante et
↪  tendance.
Qtests(adf@test$lm$residuals, 24, fitdf = length(adf@test$lm$coefficients))
#L'absence d'autocorrélation des résidus est rejetée au moins une fois.
# Le test ADF avec aucun retard n'est donc pas valide.
# Ajoutons des retards de ΔXt jusqu'à ce que les résidus ne soient plus autocorrélés.
series <- indice
kmax <- 24
adftype="ct"
adf <- adfTest_valid(series,kmax,adftype=adftype)
# Il a fallu considérer 11 retards au test ADF pour supprimer l'autocorrélation des
↪  résidus.


# Etape 4 : Résultats du test ADF
adf #affichage des résultats du test valide maintenu
print(paste0("La pvaleur du test ADF est : ", round(adf@test$p.value,digits = 2)))
```

```r
# La racine unitaire n'est pas rejetée à un seuil de 95% pour la série en niveau, la
↪  série est donc au moins I(1).


# Etape 5 : On répéte le même raisonnement pour la série différentiée.
# Testons maintenant la racine unitaire pour la série différenciée dindice.
# La représentation graphique précédente semble montrer l'absence de constante et de
↪  tendance non nulle. Vérifions avec une régression :
summary(lm(dindice ~ dates_debut[-1])) #sans la premi`ere date car on a différencié la
↪  série
# Il y a bien ni constante ni tendance significative
# Effectuons donc le test ADF dans le cas sans constante ni tendance, en vérifiant
↪  l'absence autocorrélation des résidus.
adf <- adfTest_valid(dindice,24,"nc")
# Il est nécessaire d'inclure des retards dans le test ADF.
# Le test ADF avec aucun retard n'est donc pas valide.
# Ajoutons des retards de ΔXt jusqu'à ce que les résidus ne soient plus autocorrélés.
↪  Il faut 19 lag.
adf
print(paste0("La pvaleur du test ADF est : ", round(adf@test$p.value,digits = 2)))
# Le test rejette la racine unitaire (p-value<0.05), on dira donc que la série
↪  différenciée est "stationnaire". Indice est donc I(1).


#---------------Question 3--------------
p = ggplot(data=indice) + geom_line(aes(x=dates_debut,y=indice))
p
ggsave("Serie_brute.png",path="./Images_pour_rapport",width = 10, height = 5)

p_diff = ggplot(data=dindice) + geom_line(aes(x=dates_debut[-1],y=dindice))
p_diff
ggsave("Serie_differenciee.png",path="./Images_pour_rapport",width = 10, height = 5)



#---------------PARTIE 2--------------
#---------------Question 4--------------
# Etape 1 : Détermination des p et q miximaux
acf(dindice)  #trace les fonctions d'autocorrélation totale. q* = 3
pacf(dindice)  #trace les fonctions d'autocorrélation partielle. p* = 6
axis(side=1,at=seq(0,25))
# Comme la série est stationnaire, elle est intégrée d'ordre d = 0.

# Etape 2 : validation des différents paramètres
#fonction de test des significativités individuelles des coefficients
signif <- function(estim){ #fonction de test des significations individuelles des
↪  coefficients
  coef <- estim$coef
  se <- sqrt(diag(estim$var.coef))
```

```
  t <- coef/se
  pval <- (1-pnorm(abs(t)))*2
  return(rbind(coef,se,pval))
}

# Test d'absence d'autocorrélation des résidus
Qtests <- function(series, k, fitdf=0) {
  pvals <- apply(matrix(1:k), 1, FUN=function(l) {
    pval <- if (l<=fitdf) NA else Box.test(series, lag=l, type="Ljung-Box",
    ↪  fitdf=fitdf)$p.value
    return(c("lag"=l,"pval"=pval))
  })
  return(t(pvals))
}

#fonction pour estimer un estimer un ARMA et en vérifier l'ajustement et la validité
modelchoice <- function(p,q,data=indice, k=24) {
  estim <- try(arima(data, c(p,0,q),optim.control=list(maxit=20000)))
  if (class(estim)=="try-error")
    return(c("p"=p,"q"=q,"arsignif"=NA,"masignif"=NA,"resnocorr"=NA, "ok"=NA))
  arsignif <- if (p==0) NA else signif(estim)[3,p]<=0.05
  masignif <- if (q==0) NA else signif(estim)[3,p+q]<=0.05
  resnocorr <-
  ↪  sum(Qtests(estim$residuals,24,length(estim$coef)-1)[,2]<=0.05,na.rm=T)==0
  checks <- c(arsignif,masignif,resnocorr)
  ok <- as.numeric(sum(checks,na.rm=T)==(3-sum(is.na(checks))))


  ↪  return(c("p"=p,"q"=q,"arsignif"=arsignif,"masignif"=masignif,"resnocorr"=resnocorr,"ok"=ok))
}

# Fonction pour estimer et vérifier tous les arima(p,q) avec p<=pmax et q<=max
armamodelchoice <- function(pmax,qmax){
  pqs <- expand.grid(0:pmax,0:qmax)
  t(apply(matrix(1:dim(pqs)[1]),1,function(row) {
    p <- pqs[row,1];
    q <- pqs[row,2]
    cat(paste0("Computing ARMA(",p,",",q,") nn"))
    modelchoice(p,q)
  }))
}

# On estime tous les ARMA possibles et on les stocke dans la variable armamodels
armamodels <- armamodelchoice(6,3)
# On garde les modèles bien ajustés et valides.
selec <- armamodels[armamodels[,"ok"]==1&!is.na(armamodels[,"ok"]),]
print("Les modèles valides et ajustés sont")
print(selec)
```

```r
#on a 3 modèles valides et ajustés : ARMA (4,1), ARMA(5,2) et ARMA(1, 3)
arima401 <- arima(dindice,c(4,0,1))
arima502 <- arima(dindice, c(5,0,2))
arima103 <- arima(dindice, c(1,0,3))


# Etape 3 : Calcul des AIC et BIC pour les modèles sélectionnés
models <- c("arima401","arima502","arima103"); names(models) <- models
apply(as.matrix(models),1, function(m) c("AIC"=AIC(get(m)), "BIC"=BIC(get(m))))
#ARMA(4,1) minimise l'AIC
#ARMA(1,3) minimise le BIC


# Etape 4 : On choisit le modèle parmis les 2 qui minimise le R2 ajusté

# Calcul de R2 ajusté
adj_r2 <- function(model){
  p <- model$arma[1]
  q <- model$arma[2]
  n <- model$nobs-max(p,q)
  ss_res <- sum(model$residuals^2)
  ss_tot <- sum(dindice[-c(1:max(p,q))]^2)
  adj_r2 <- 1-(ss_res/(n-p-q-1))/(ss_tot/(n-1))
  return(adj_r2)
}
adj_r2(arima401)
adj_r2(arima103)
# On sélectionne le ARIMA(4,0,1) qui a le plus grand R2 ajusté



#---------------Question 5--------------
# Il s'agit de montrer que le modèle ARMA(4,0,1) qu'on a pour la série différenciée est
↪  bien causal.
# Or un ARMA est causal ssi pas de racine dans le disque unité du polynôme phi
# Test la causalité du modèle
arma_causal = function(model){
  if(model$arma[1]==0){# gère le cas trivial d'un modèle MA
    return(TRUE)
  }
  else{
    phi = polynomial(c(1,-model$coef[1:(model$arma[1])]))
    racines = polyroot(phi) #les coefficients polynomiaux sont donnés dans l'ordre
    ↪  CROISSANT

    for(i in 1:length(racines)){
      if (abs(racines[i])<=1) {
        return(FALSE)
      }
```

```
    }
    return(TRUE) #si toutes les racines sont de module strictement supérieur à 1, alors
    ↪  ARMA causal
  }
}
arma_causal(arima401) # renvoie TRUE

png("./Images_pour_rapport/root401.png")
Arima(dindice, order = c(4, 0, 1), xreg = seq_along(dindice), include.mean=F) %>%
  autoplot()
# Cette fonction roduit un graphique des racines inverses AR et MA d'un modèle ARIMA.
# Les racines inverses situées en dehors du cercle unitaire sont représentées en rouge.
dev.off()

# Le modèle ARMA pour la série différenciée est causal, toutes les inverses des racines
↪  sont bien à l'intérieur du cercle unité.
# Donc, par définition d'un ARIMA, la série non transformée suit un modèle
↪  ARIMA(4,1,1).



#---------------PARTIE 3--------------
#---------------Question 6--------------
T # La longueur de la série
# On suppose pour la suite que les résidus de la série sont gaussiens.

# Voir rapport pour l'équation



#---------------Question 7--------------
# Voir rapport pour les hypothèses écrites formellement
# Nous allons tester l'hypothèse concernant les résidus.
arma <- arima (dindice, c(4,0,1), include.mean=F)

png("./Images_pour_rapport/ACF_LjunBoxTest.png")
tsdiag(arma)
# Nous n'observons rien de significatif pour les résidus concernant l'ACF, ils sont
↪  bien en dessous de la ligne tracée en pointillés bleu.
# Concernant le Ljung-Box Statistic les p-values pour différents laf sont au dessus de
↪  0.05.
# Cela signifie qu'il n'y a pas de significativité, pas de patternes.
dev.off()

# Pour le Q-Q Plot
png("./Images_pour_rapport/qqnorm.png")
qqnorm(arma$residuals)
qqline(arma$residuals)
dev.off()
```

```r
# Densité des résidus tracée par rapport à celle théorique en prennant la moyenne et
↪  l'écart type empirique.
png("./Images_pour_rapport/densite_res.png")
plot(density(arma$residuals ,lwd=0.5),xlim=c(-10,10), main="Densite des residus",
     xlab="Valeurs prises")
mu<-mean(arma$residuals)
sigma<-sd(arma$residuals)
x<-seq(-10,10)
y<-dnorm(x,mu,sigma)
lines(x,y,lwd=0.5,col="blue")
dev.off()

# Cette fonction applique le test de normalité proposé par Jarque et Bera (1980).
# Permet de voir un test différent de ceux vus précédemment.
jarque.bera.test(arma$residuals)
# Cela nous indique que la statistique du test est de 63.056 et que la valeur p du test
↪  est de 2.032e-14.
# Dans ce cas, nous rejetons l'hypothèse nulle selon laquelle les données sont
↪  normalement distribuées.


#---------------Question 8--------------
# Représentation graphique de la région pour alpha = 95%
# On fait les prédictions à horizons 1 et 2 de notre série temporelle
XT1 = predict (arma, n.ahead=2)$pred[1]
XT2 = predict (arma, n.ahead=2)$pred[2]

# On cherche d'abord a tracer le region de confiance univariee
# pour la serie originale a 95%.
png("./Images_pour_rapport/prevision.png")
arima <- arima(indice,c(4, 1, 1),include.mean=F)
forecast_indice = forecast(arima, h=2,level=95)
par(mfrow=c(1,1))
plot(forecast_indice,col=1,fcol=2,shaded=TRUE,xlab="Temps",ylab="Valeur série
↪  temporelle", xlim=c(2018, 2023), main="Prevision pour la série temporelle non
↪  différenciée")
dev.off()


#Ensuite, on represente la region de confiance bivariee a 95%.
arma_new = arima0(dindice, order=c(4,0,1))
# On associe les coefficients phi, psi et sigma2 pour la suite
phi <- as.numeric(arma_new$coef[1])
psi <- as.numeric(arma_new$coef[5])
sigma2 <- as.numeric(arma$sigma2)
sigma <- matrix (c(sigma2,(phi - psi)*sigma2,(phi - psi)*sigma2,(phi - psi)^2*sigma2 +
↪  sigma2),ncol =2)
Sigma = sigma2 * sigma
```

```r
inv_Sigma <- solve(Sigma)

png("./Images_pour_rapport/ellispe.png")
plot (XT1 ,XT2 , xlim =c(-250, 250) , ylim =c(-250,250) , xlab =" Prevision de X(T+1)
↪  sachant T ", ylab ="Prevision de X(T+2) sachant T", main =" Region de confiance
↪  bivariee à 95%")
lines(ellipse(Sigma, centre=c(XT1,XT2)), type="l", col="red", xlab="Xt+1",ylab="Xt+2",
↪  main="Ellipse de confiance pour (Xt+1,Xt+2)")
abline(h=XT1, v=XT2)
dev.off()
```