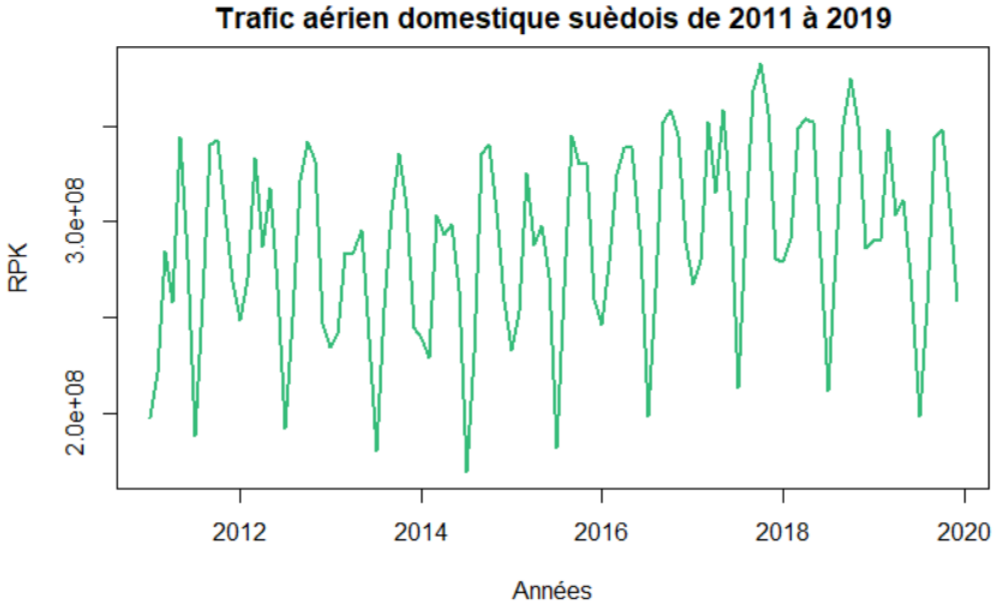
Projet séries temporelles

- Trafic aérien domestique suèdois -

Victor KLÖTZER

11/01/2021

Dans ce projet on se propose d'étudier la série suivante:



qui représente le trafic aérien domestique en Suède mensuel de janvier 2011 à décembre 2019 et exprimé en Revenue Passengers Kilometers (RPK). On a ainsi $12 \times 9 = 108$ données pour cette série temporelle.

L'objectif de ce projet est de modéliser cette série afin de founir des prévisions mensuelles de RPK pour le trafic domestique suédois entre janvier 2020 et décembre 2025. Pour ce faire, on va utiliser ici trois différentes techniques pour modéliser la série. On essayera d'abord de décomposer la série en une composante de saisonnalité et une composante de tendance que l'on ajustera par un modèle linéaire. Ensuite on utilisera le lissage exponentiel. Et enfin on modélisera cette série à l'aide de modèles ARIMA.

Afin de pouvoir mesurer la qualité de prévisions des modèles, on aimerait se réserver un échantillon de validation (entre 25% et 33% des données les plus récentes). Cependant, en observant un peu la série, on remarque assez clairement qu'à partir de 2018 le RPK ne croît plus comme avant, et semble même commencer à diminuer. Ainsi, si l'on se réservait même seulement les dernières 24 observations de la série comme échantillon de validation, on perdrait le changement de tendance qui se produit à partir d'environ janvier 2018. Prendre un plus petit échantillon (trop petit) ne semble pas non plus envisageable car cela ne permettrait pas d'apporter d'information fiable sur la qualité prédictif des modèles. On décide donc de ne pas utliser d'échantillon de validation dans ce projet et de se servir de toutes les observations pour la construction des modèles.

Une autre chose que l'on peut d'ors et déjà noter est que les valeurs des données sont très grandes (de l'ordre de 10^8). Afin de réduire la variance des données, on pourrait donc soit les diviser par 10^8 ou bien les passés au logarithme. Le passage au logarithme reviendrait finalement à vouloir ajuster un modèle mutliplicatif. Choissir ici un modèle multiplicatif semble cohérent, car si le nombre de voyageurs diminue de manière significative, on devrait également observer des variations moins grandes sur une saison (ou inversement si le nombre de voyageur augmente). On décide donc de passer les données au logarithme et d'appliquer un modèle sur ces données "loguées". Ainsi, en notant $(X_t)_{t\geq 1}$ la série temporelle des RPK, on souhaite créer le modèle suivant :

$$X_t = T_t \times S_t \times \epsilon_t$$

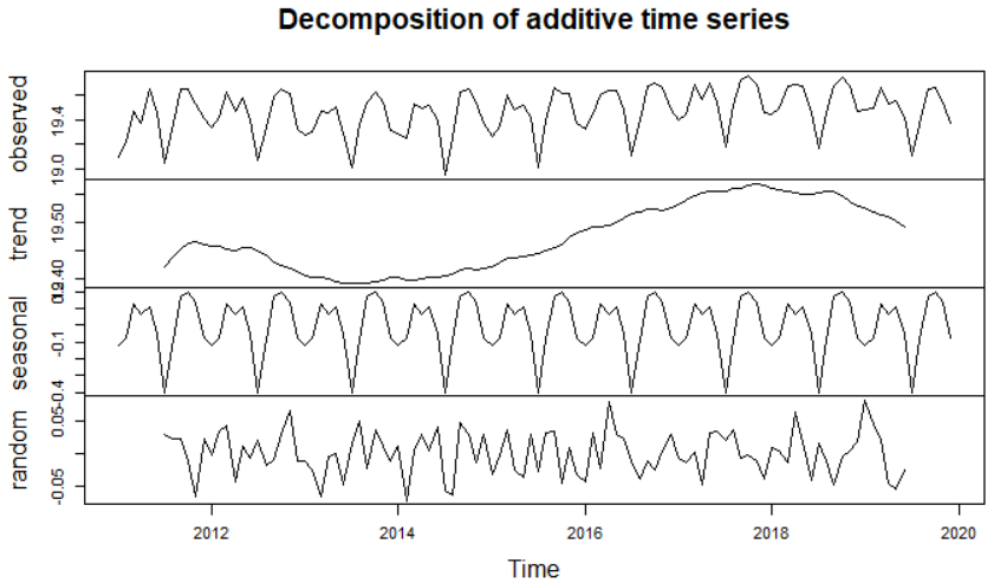
où T_t est la tendance, S_t la saisonnalité et ϵ_t l'erreur.
En passant les données au logarithme on obtient donc le modèle :

$$\tilde{X}_t := \log(X_t) = \log(T_t) + \log(S_t) + \log(\epsilon_t) =: \tilde{T}_t + \tilde{S}_t + \tilde{\epsilon}_t$$

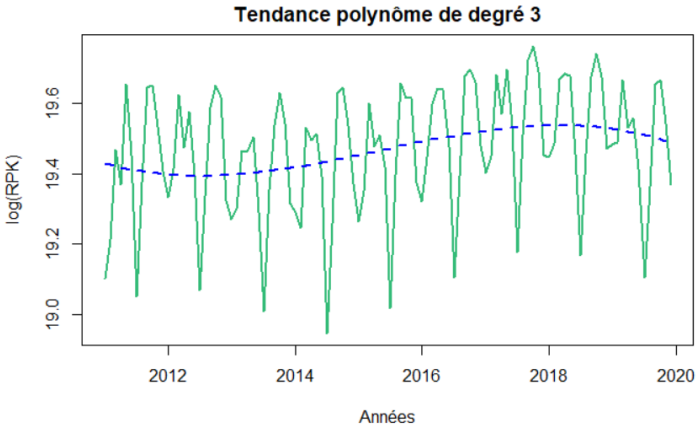
sur lequelle on va travailler par la suite.

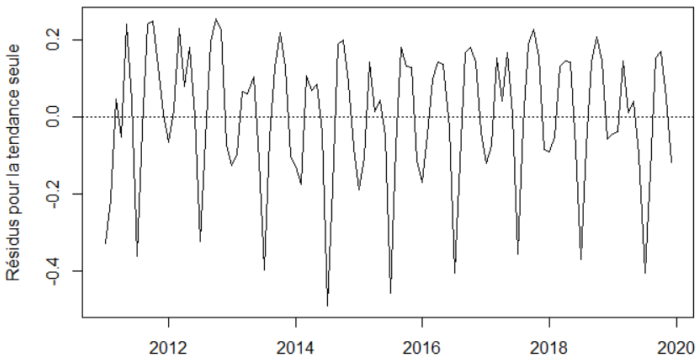
1. Premier modèle : tendance ajustée par un modèle linéaire

Dans ce premier modèle on veut essayer d'identifier la tendance et la saisonnalité de la série. On utilise d'abord la fonction `decompose` de `R` pour vérifier que l'on a bien une composante saisonnière et surtout pour obtenir une idée de la tendance de la série (calculée à partir de moyennes mobiles dans `decompose`) :



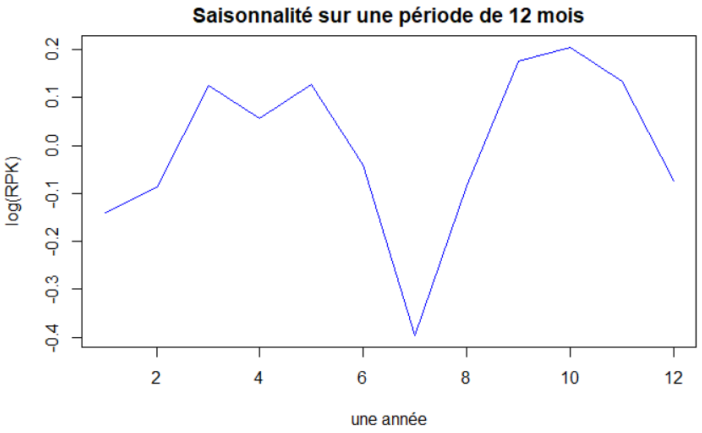
Cette série comporte donc bien une saisonnalité avec une période de 12 mois ainsi qu'une tendance qui s'apparente à un polynôme d'ordre 3. On pourrait vouloir utiliser une tendance linéraire pour commencer la modélisation, mais cela semble peut approprié car cette tendance linéaire serait croissante alors que l'on observe une décroissance à partir 2018. On décide donc d'ajuster pour la tendance, un modèle linéaire sur la série temporelle de type : $\tilde{T}_t = \beta_0 + \beta_1 t + \beta_2 t^2$. Pour les données loguées, on obtient ainsi la tendance suivante :

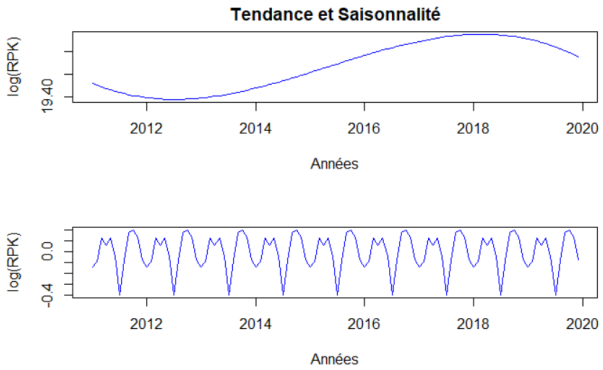




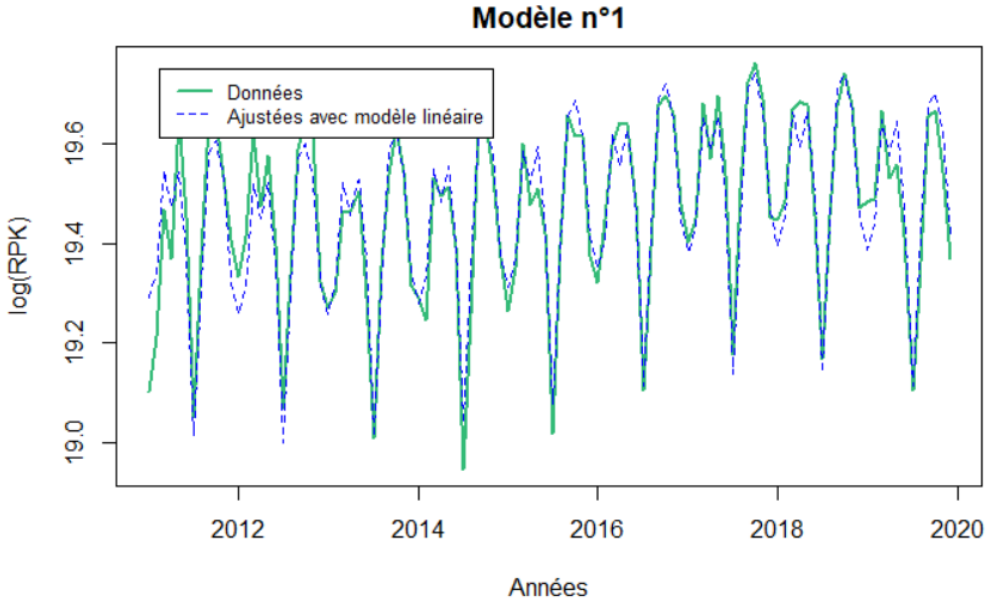
Les résidus de ce modèle linéaire ajusté sur une série qui a une saisonnalité, présentent eux aussi cette saisonnalité. On peut néanmoins noter que ces résidus sont bien centrés et reprennent la saisonnalité de manière assez régulière. On regardera les résidus du modèle final une fois que la saisonnalité aura été elle aussi expliquée.

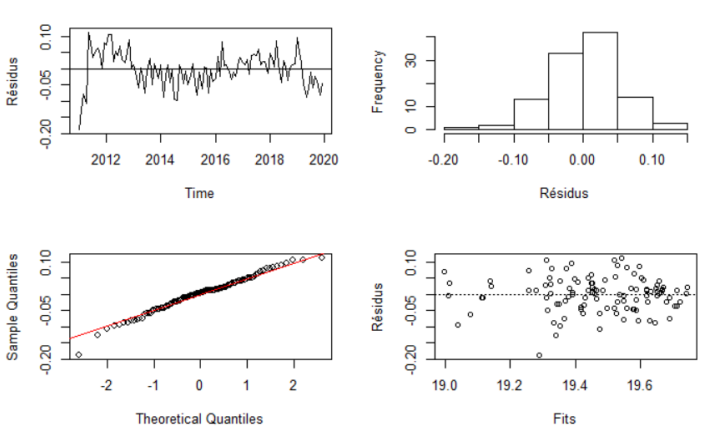
Afin de déterminer justement la composant saisonnière $\tilde{S}_t$ de la série, on retire la tendance des données puis on moyenne la valeur de la série sur chacun des 12 mois des 9 années que l'on observe. On retire ensuite la valeur moyenne des 12 points ainsi obtenus afin que la composante saisonnière soit centrée autour de 0. On a ainsi :

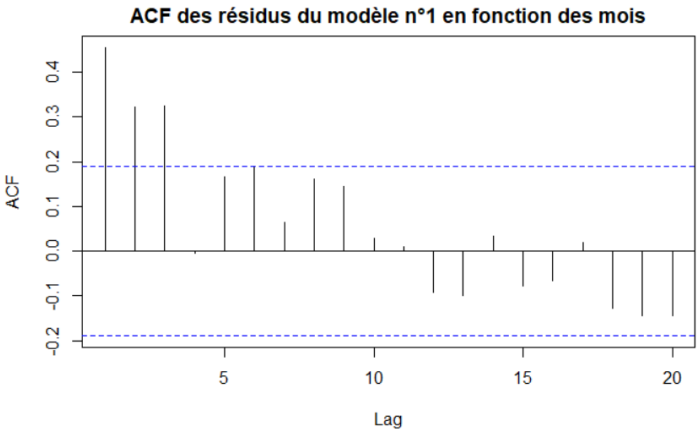




On peut désormais recontruire le modèle à partir de la tendance et de la saisonnalité calculées et analyser les résidus du modèle finalement obtenu.



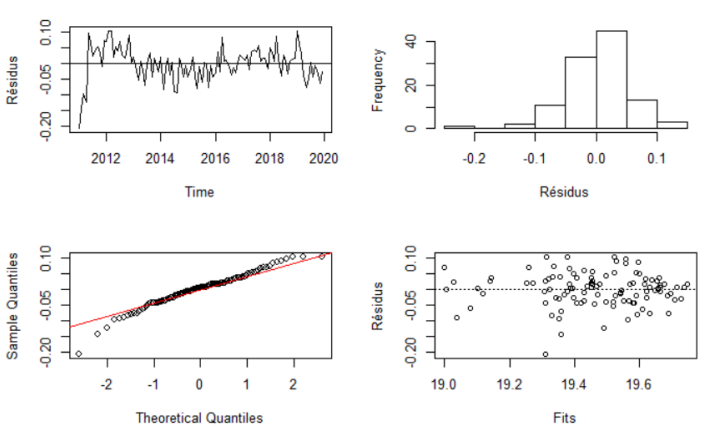


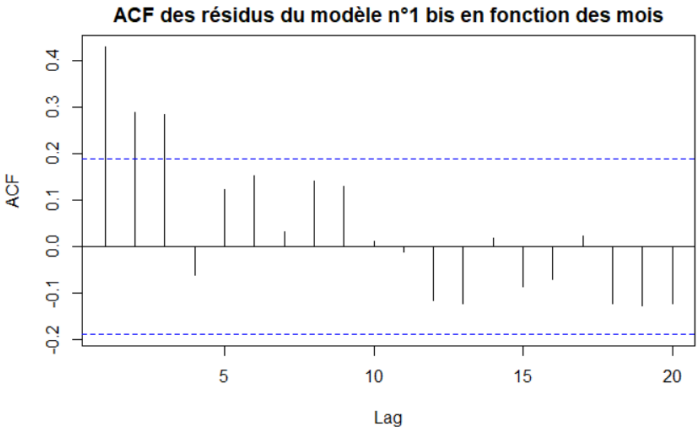


Les résidus ne présentent pas de forme particulière si ce n'est au début de la série. L'histogramme et le qqplot montrent que les résidus sont bien normalement distribués. Enfin, l'affichage des résidus en fonction des valeurs ajustées ne présente pas de structure particulière et montre ainsi que les résidus ont environ même variance (hypothèse d'homoscédasticité vérifiée). En ce qui concerne la fonction d'autocorrélation, on observe une corrélation pour des lags de 1 à 3 mois. Ces petites corrélations semblent acceptables car on étudie une série temporelle d'un phénomène dont deux observations proches dans le temps sont certainement corrélées.
Cette première modélisation semble donc convenable.

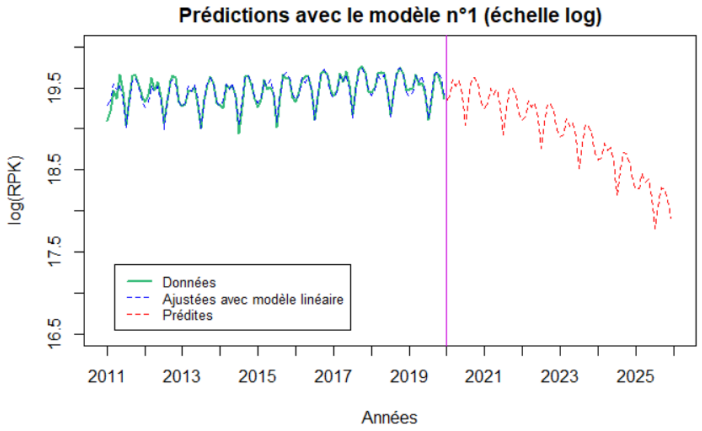
La fonction `tslm()` de R permet d'ajuster directement un modèle linéaire à une série temporelle comportant une saisonnalité (en utilisant la variable nommée `season`). On utilise cette fonction afin de remodéliser ce premier modèle et en particulier

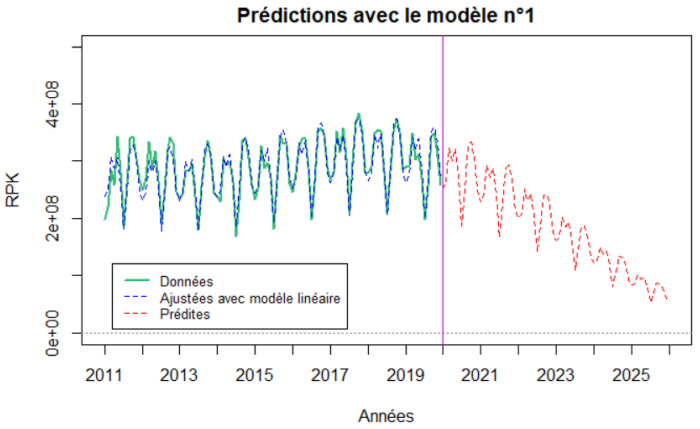
de pouvoir calculer des intervalles de confiances pour les prédicitions. Ce modèle 1 bis est quasi identitique au modèle 1, l'analyse de résidus notamment rendra le même résulat :

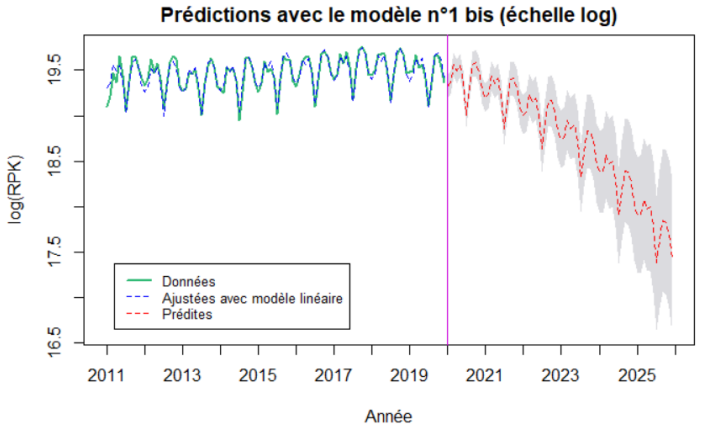


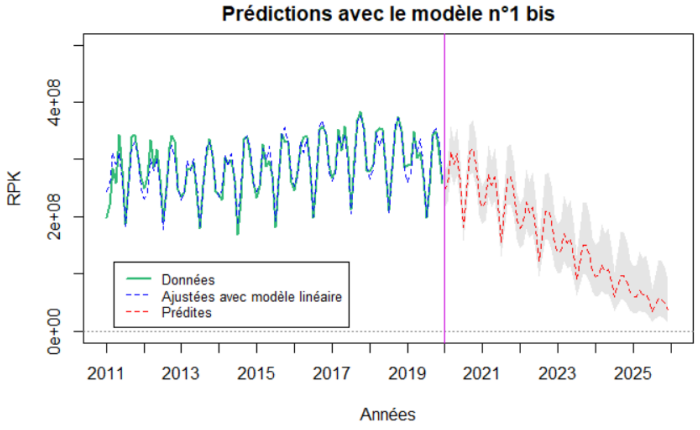


À partir de ces modèles 1 et 1 bis, on calcule finalement des prédictions pour les années 2020 à 2050 :





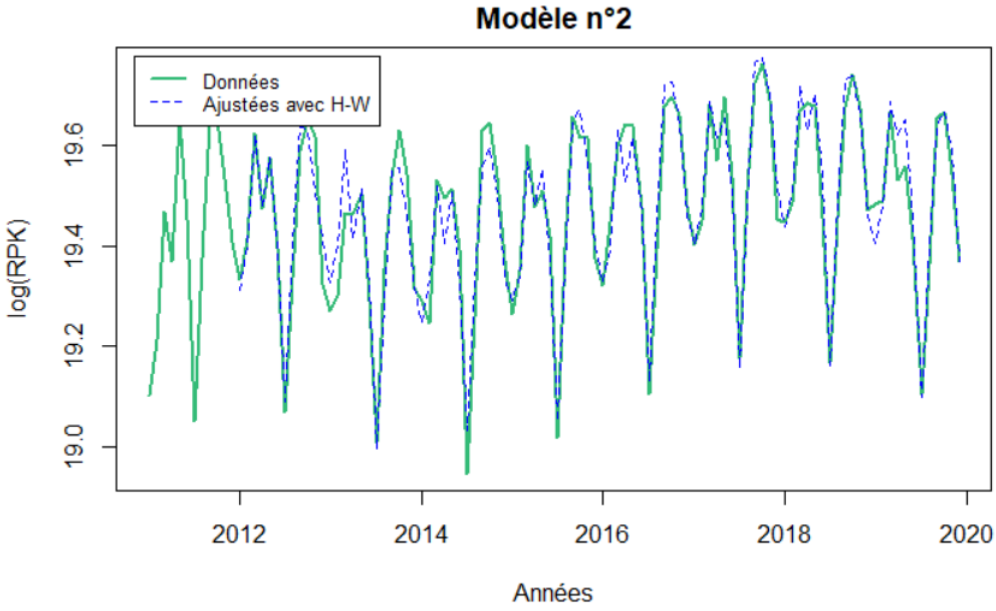


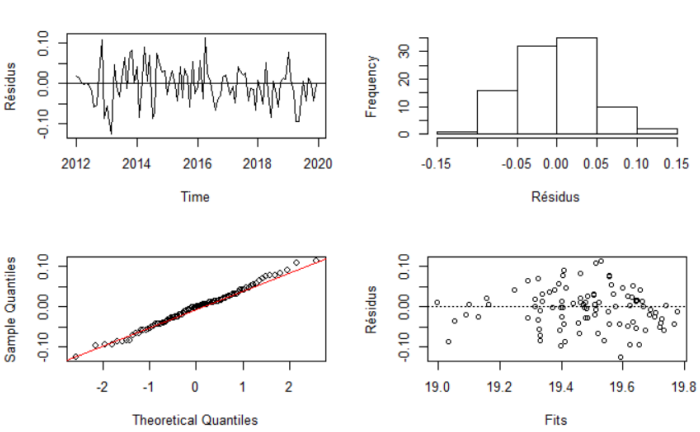


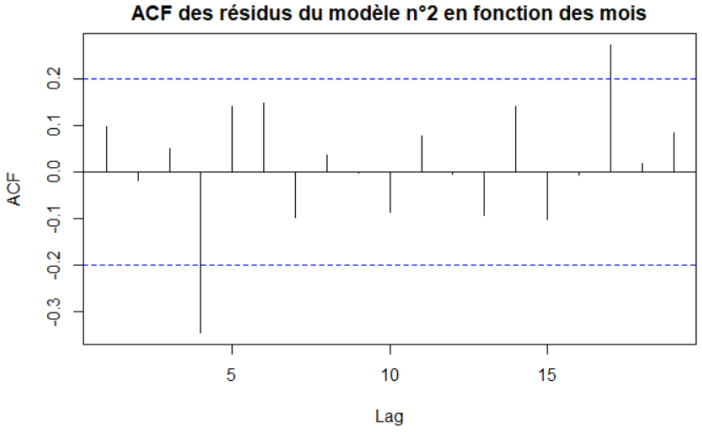
Une dernière chose que l'on peut noter pour ce modèle est qu'il comporte beaucoup de paramètres : 3 pour la tendance et 12 pour la saisonnalité, i.e. 15 au total. Ce n'est pas forcément un avantage car plus il y a de paramètres moins ceux-ci risque d'être précisément estimés. Pour avoir un critère de comparaison pour ce modèle, on peut déterminer sa vraissemblance que l'on pénalise par rapport au nombre de paramètres. Cela peut par exemple être fait avec le critère AIC qui est calculé comme suit : $\text{AIC} = -2 \times \log\text{Vraissemblance}(model_{1bis}) - 2 \times 15 = -359.78$. On pourra s'en servir pour comparer ce modèle aux autres modèles créés par la suite.

2. Deuxième modèle : lissage exponentiel

Dans un second temps, on va utiliser le lissage exponentiel pour expliquer la série temporelle étudiée. La série comporte une tendance et une saisonnalité, c'est donc vers un lissage additif et saisonnier de Holt-Winters que l'on se tourne. La fonction `HoltWinters()` de R permet de créer un tel modèle et d'en estimer les paramètres α, β et γ optimaux en minimisant le carré de l'erreur de prédiction à lag 1 ("squared one-step prediction error" en anglais). On obtient les valeurs ajustées suivantes ainsi que les résidus suivant :

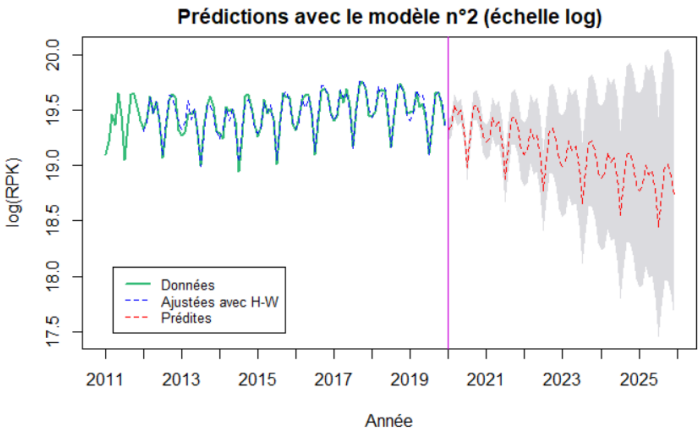


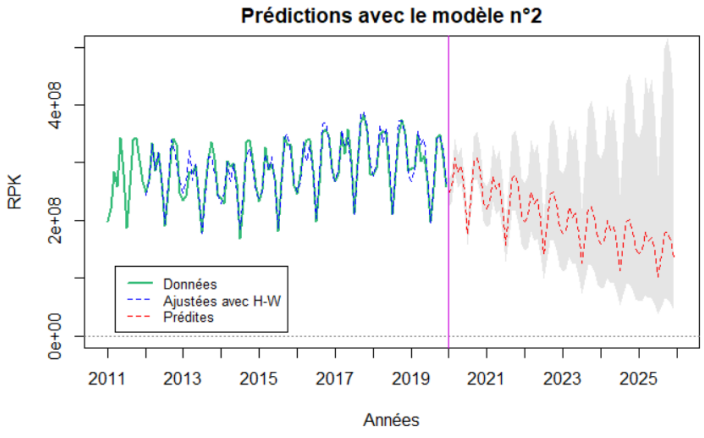




Les résidus sont décorréles, ne présentent pas de forme particulière et sont normalement distribués. Toutes les hypothèses sur les résidus sont donc vérifiées de sorte que l'on valide cette modélisation.

On peut alors prédire à partir de ce modèle 2, les RPK pour les années 2020 à 2050 :

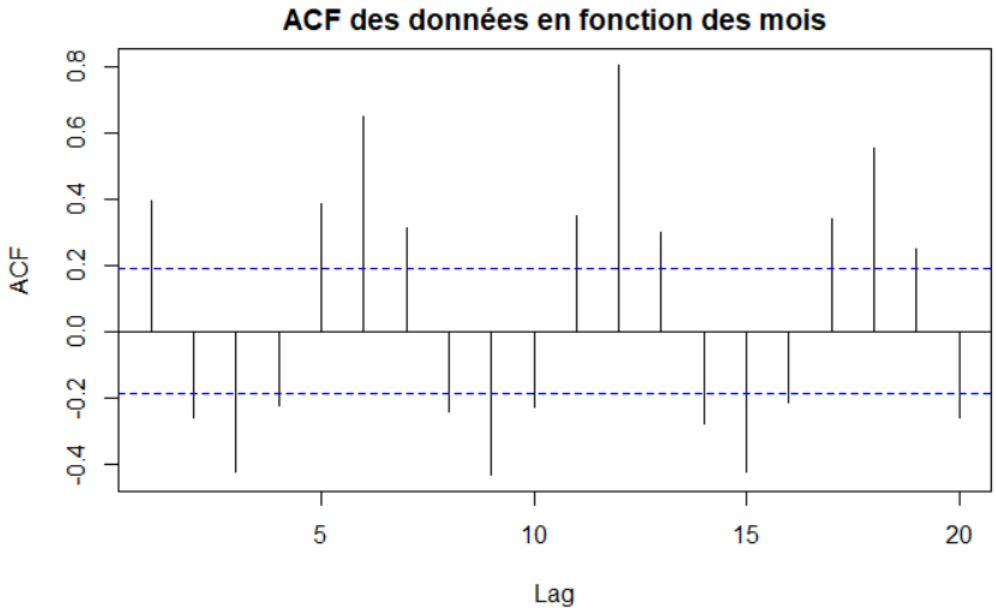




3. Troisième modèle : modélisation ARIMA

Le troisième et dernier type de modèle que l'on veut ajuster est un modèle ARIMA. Afin de pouvoir s'en servir, il faut d'abord retirer la saisonnalité de la série étudiée. Par la suite, il faudra également déterminer le nombre de différentiation qu'il faut faire de la série désaisonnalisée, afin de lui retirer sa tendance et d'obtenir ainsi une série stationnaire.

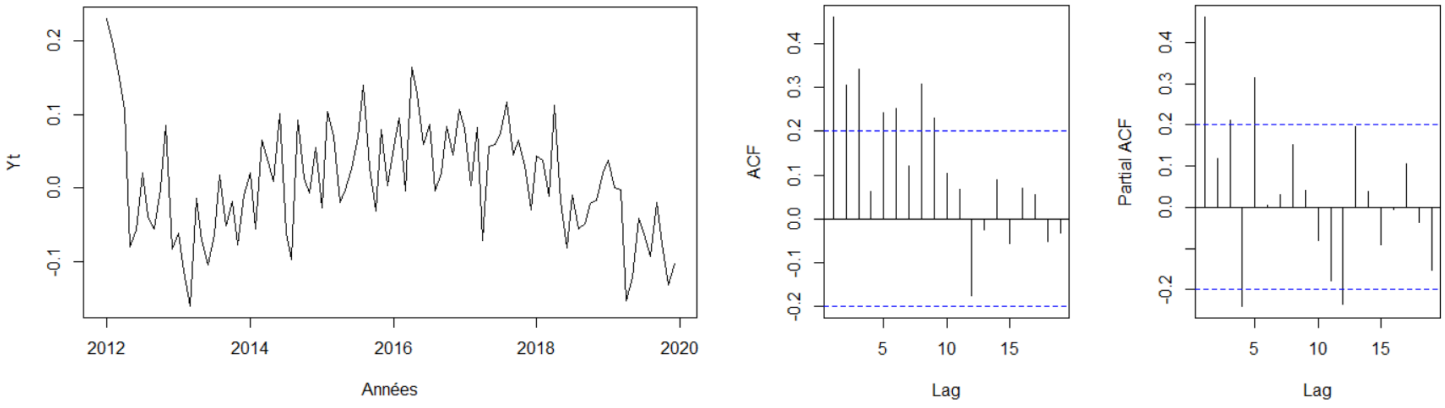
On regarde pour commencer la fonction d'autocorrélation de la série temporelle :



En observe clairement la saisonnalité. Pour retirer celle-ci on réalise dans un premier temps une différence de la série par rapport à elle-même décalée d'un lag de 12 mois :

$$Y_t := \tilde{X}_t - \tilde{X}_{t-12}, \quad \forall t \geq 13$$

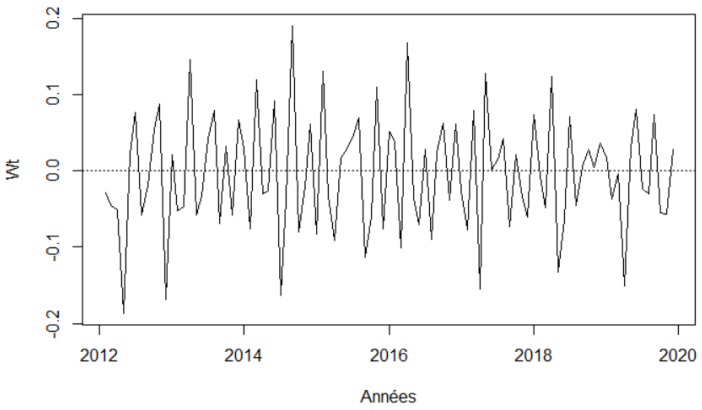
On obtient la série $(Y_t)_{t\geq 13}$ suivante avec sa fonction d'autocorrélation et d'autocorrélation partielle :

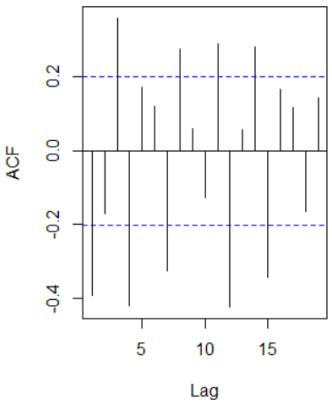


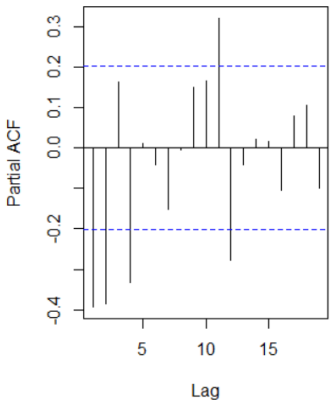
Comme souhaité, cette série (Y_t) semble effectivement avoir perdue sa saisonnalité. Mais on peut encore reconnaitre une tendance dans cette série et notamment la trop lente décroissance de sa fonction d'autocorrélation indique que (Y_t) n'est pas stationnaire. D'ailleurs, en effectuant un test augmenté de Dickey-Fuller pour tester l'hypothèse H_0 de non-stationnarité contre H_1 la série est stationnaire (à l'aide de la fonction `adf.test()` de R), on trouve effectivement une p-value de 0.6016 bien supérieure à 5. On différencie donc la série une fois avec un lag de 1 pour retirer la tendance :

$$W_t := Y_t - Y_{t-1}, \quad \forall t \geq 14$$

On obtient la série $(W_t)_{t\geq 14}$ suivante avec sa fonction d'autocorrélation et d'autocorrélation partielle :







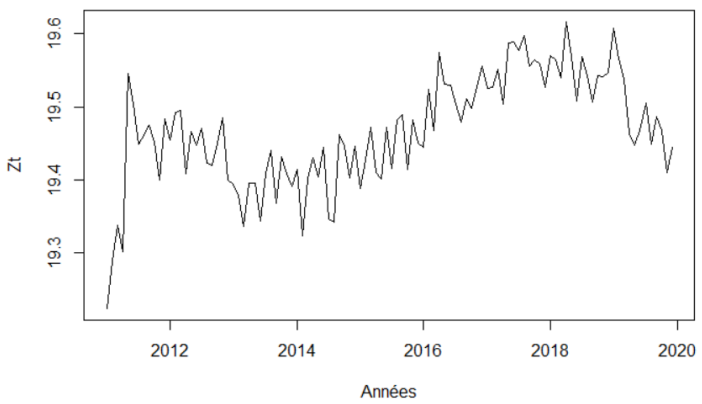
Cette fois-ci la fonction semble bien stationnaire. On le vérifie à l'aide d'un test augmenté de Dickey-Fuller qui retourne une p-value inférieure à 0.01 donc inférieure à 5. On rejete H_0 et accepte l'hypothèse de stationnarité de (W_t).

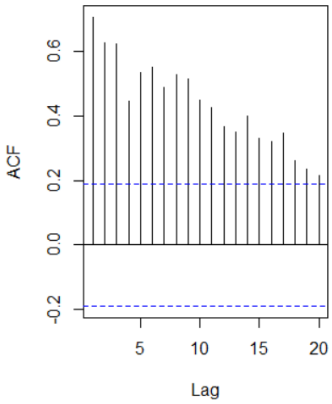
Au vu de la fonction d'autocorrélation dont on peut considérer qu'elle décroit exponentiellement (en alterné), et comme la fonction d'autocorrélation partielle présente quatre premiers pics principaux, on pourra donc par la suite adpater un modèle AR(4) à (W_t), ce qui revient à adapter un modèle ARIMA(4,1,0) à (Y_t).

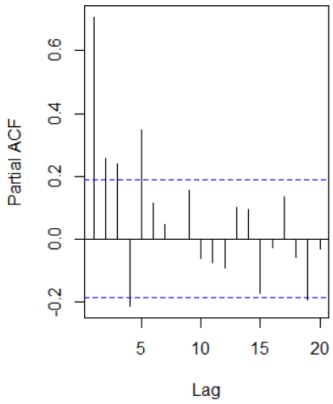
Afin de pouvoir comparer des modèles ARIMA entre eux, on essaie de stationnariser la série initiale $(\tilde{X}_t)$ différemment. On va retirer la saisonnalité d'une autre manière en utilisant tout simplement la décompositon de la série à partir de moyennes mobiles et de moyennes pour chaque mois. Pour récupérer cette composante saisonnière S_t, on se sert de la fonction `decompose()` de R et on obtient :

$$Z_t := \tilde{X}_t - S_t, \quad \forall t \geq 1$$

On a alors la série $(Z_t)_{t\geq 1}$ suivante avec sa fonction d'autocorrélation et d'autocorrélation partielle :



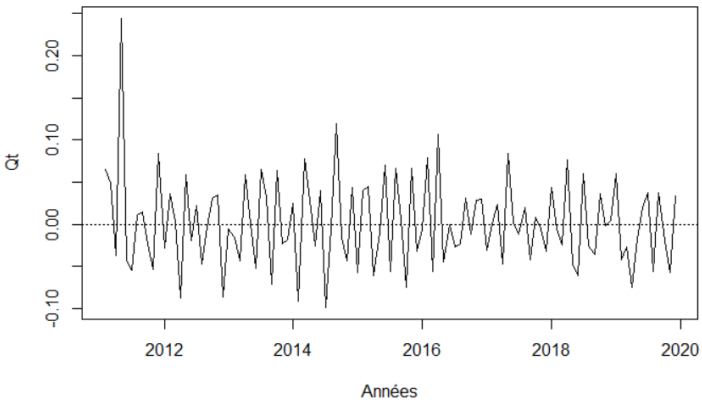


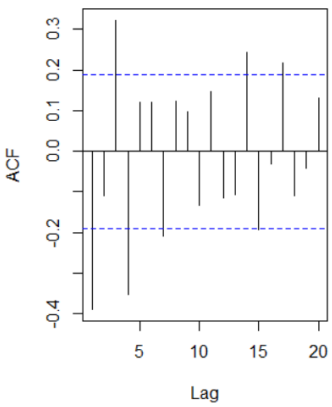


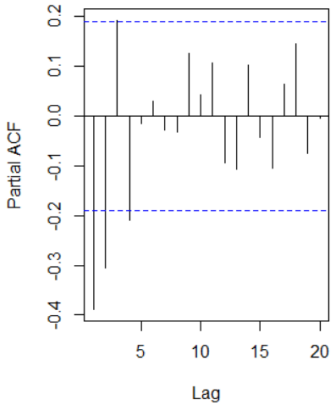
Cette série ne comporte plus de saisonnalité mais a encore une tendance. La fonction d'autocorrélation décroît trop lentement et le test augmenté de Dickey-Fuller avec une p-value de 0.7997 confirme que la série n'est pas staionnaire. Comme tout à l'heure, on différencie la série une fois avec un lag de 1 pour retirer la tendance :

$$Q_t := Z_t - Z_{t-1}, \quad \forall t \geq 2$$

On obtient finalement la série $(Q_t)_{t\geq 14}$ suivante avec sa fonction d'autocorrélation et d'autocorrélation partielle :







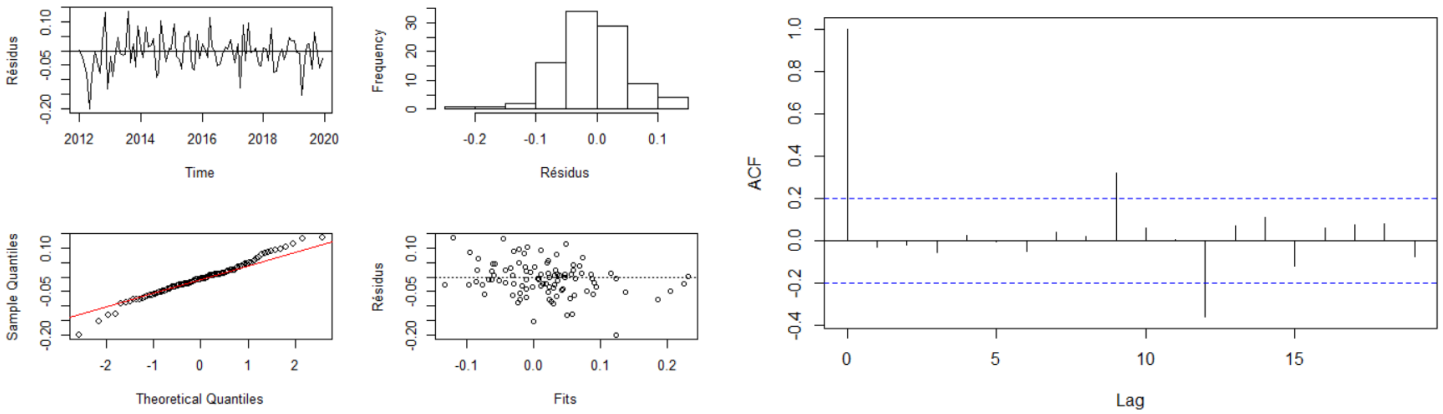
Cette série (Q_t) semble bien être stationnaire, ce que nous confirme d'ailleurs le test augmenté de Dickey-Fuller (p-value inférieure à 0.01).

Au vu des fonctions d'autocorrélation et d'autocorrélation partielle de (Q_t) et de leurs pics, on essayera donc aussi d'adapter un modèle ARMA(4,7) à (Q_t), ce qui reviendra à adapter un modèle ARIMA(4,1,7) à (Z_t).

3.a Troisième modèle version a : ARIMA(4,1,0) sur (Y_t)

Pour adapter un modèle ARIMA(4,1,0) à (Y_t) on utilise la fonction `arima()` de R qui permet de créer un tel modèle. Vient ensuite une étape de sélection de modèle, où l'on teste d'abord la significativité des paramètres estimés de l'ARIMA, puis on fixe à 0 les paramètres dont la p-value est supérieure à 5% tant que tous les coefficients restants ne sont pas significativement différents de 0.

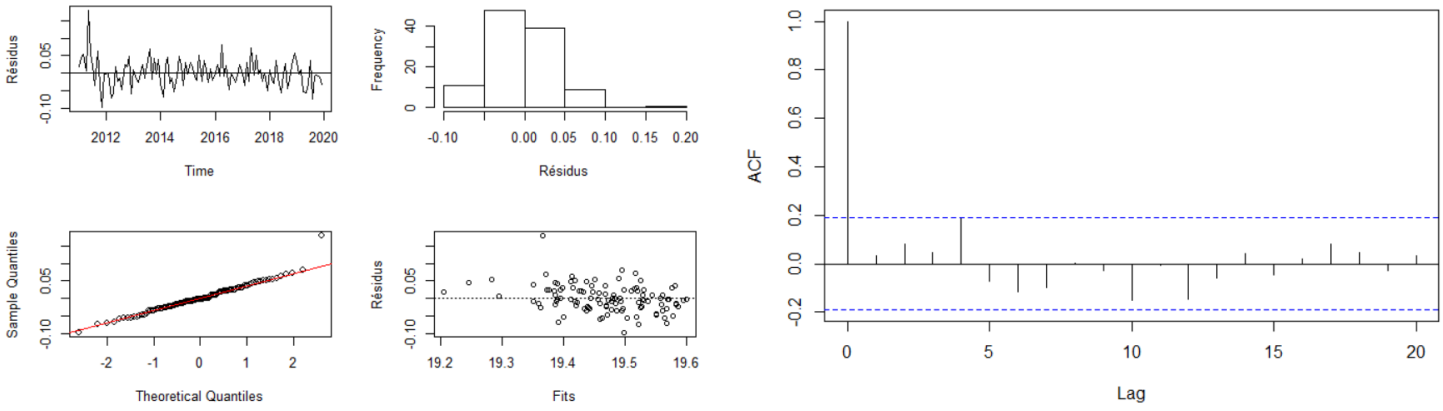
Après sélection, on obtient ici un ARIMA(4,1,0) où le coefficient AR3 est fixé à 0. On a ainsi quatre paramètres estimés dans ce modèle dont voici les résidus :



Les résidus ne présentent pas de forme particulière, sont normalement distribués et respectent l'hypothèse d'homoscédasticité. De plus, l'acf de résidus montre que ceux-ci sont bien décorrélés les uns des autres. Toutes les hypothèses sur les résidus sont donc vérifiées de sorte que l'on valide cette modélisation 3a.

3.b Troisième modèle version b : ARIMA(4,1,7) sur (Z_t)

On essaie ensuite d'adapter un modèle ARIMA(4,1,7) à (Z_t) en utlisant le même procédé que décrit ci-dessus. Après sélection, on obtient un ARIMA(4,1,7) où les coefficients MA1 et MA3 sont fixés à 0, de sorte que dix paramètres sont estimés dans ce modèle. Voici les résidus que l'on obtient :

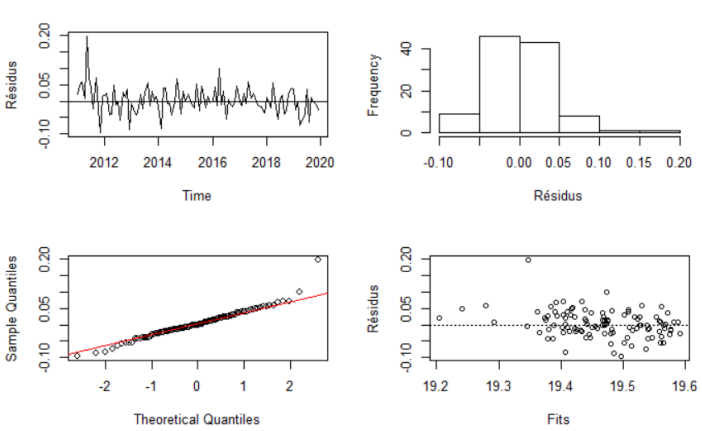


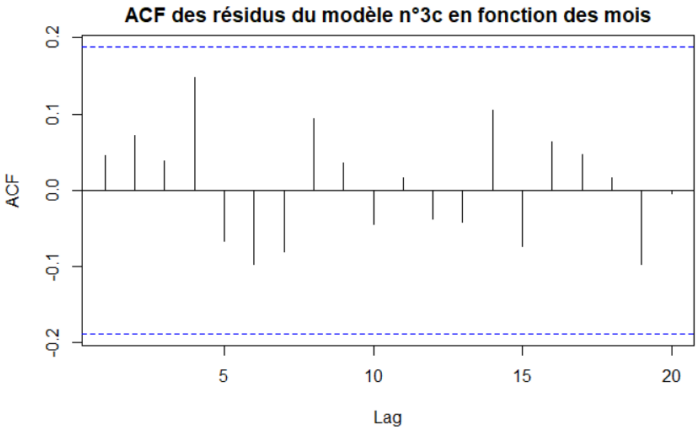
Comme ci-dessus, les résidus ne présentent pas de forme particulière, sont normalement distribués et respectent l'hypothèse d'homoscédasticité. De plus, l'acf de résidus montre que ceux-ci sont bien décorrélés les uns des autres. Toutes les hypothèses sur les résidus sont donc vérifiées de sorte que l'on valide aussi cette modélisation 3b.

3.c Troisième modèle version c : réduction du nombre de paramètres du modèle 3b

En cours de Séries Temporelles, le prof a évoqué le fait que souvent on peut se ramener à un modèle ARIMA avec peu de paramètres. Dans le modèle 3b ci-dessus on a dix paramètres ce qui semble relativement beaucoup sachant que plus on a de paramètres, moins précisément chacun d'entre-eux est estimés. On propose donc de reprendre le modèle précédant en retirant pour continuer le coefficient avec la p-value la plus élevée : en l'occurence, c'est le coefficient MA7 avec une p-value de 0.025513. On le retire (fixé à 0), puis on recommence la même procédure que précédemment jusqu'à ce que tous les coefficients restants soient significativement différents de 0.

Après sélection, on se retrouve cette fois-ci avec un ARIMA(2,1,6) sur (Z_t) dont les coefficients MA1, MA2 et MA3 sont fixés à 0, i.e dans ce modèle six paramètres sont estimés ce qui peut sembler plus raisonnable. Voici les résidus que l'on obtient pour ce modèle 3c :



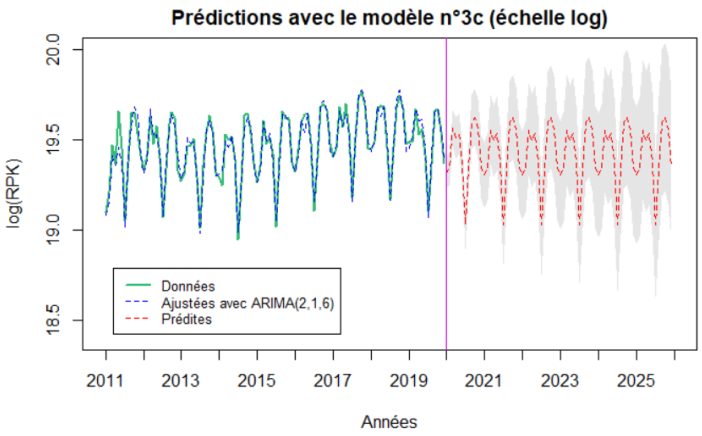


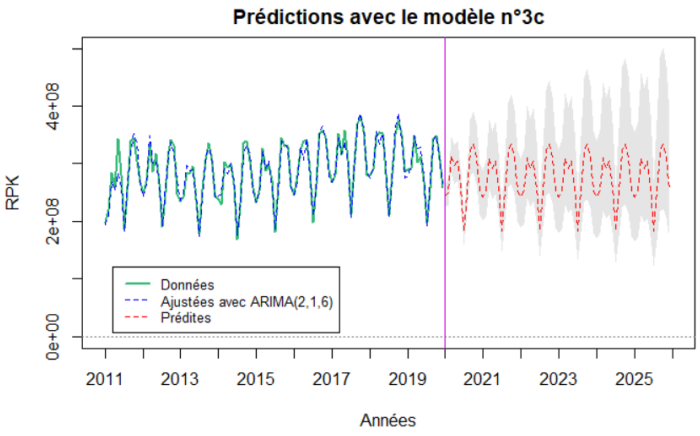
Encore une fois, les résidus ne présentent pas de forme particulière, sont normalement distribués et respectent l'hypothèse d'homoscédasticité. De plus, l'acf de résidus montre que ceux-ci sont bien décorrélés les uns des autres. Toutes les hypothèses sur les résidus sont donc vérifiées de sorte que l'on valide ici aussi cette modélisation 3c.

Choix du modèle ARIMA et prédictions

Afin de comparer ces trois modèles 3 et de choisir celui qui est le plus vraissemblable, on utilise le critère d'information d'Akaike (AIC) que l'on veut minimiser.
Le modèle 3a a un AIC de -256.06, le modèle 3b a un AIC de -362.21 et le modèle 3c un AIC de -364.88. C'est donc le modèle 3c qui semble le plus vraissemblable en prenant aussi en compte le nombre de paramètres du modèles.

On utilise donc ce modèle 3c, i.e. un ARIMA(2,1,6) sur la série (Z_t), pour expliquer la série temporelle étudiée et faire des prédictions pour les années 2020 à 2025. (Noter qu'ici il faut d'abord reconstruire la série originale $(\tilde{X}_t)$ à partir de valeurs ajustées et des prévisions que l'on obtient pour $Z_t = \tilde{X}_t - S_t, \quad \forall t \geq 1$)





Conclusion

Dans ce projet, on a modélisé de trois manières différentes un modèle multiplicatif sur la série temporelle des RPK du trafic domestique de Suède. Ceci a permis dans les trois cas de donner des prévisions sur les 6 prochaines années (de 2020 à 2025). Néanmoins, comme on veut faire des prévisons sur 6 ans à partir de données sur 9 années, il parait donc cohérent que les prévisions au delà de 2 ou 3 ans aient des intervalles de confiances assez larges.

ANNEXE : code de ce projet

```
##############################################
##       PROJET SÉRIES TEMPORELLES          ##
##  - Trafic aérien domestique suèdois -    ##
##             Victor KLÖTZER               ##
##               11/01/2021                 ##
##############################################

# Importation des librairies
library("tseries")
library("forecast")
library("caschrono")

# Emplacement de travail
setwd("~/Scolaire/INSA Rennes/4GM/Séries chronologiques/Projet")

# Importation et affichage des données
trafic = read.csv2("Projet_donnees.csv")
trafic_log = log(trafic)

ts_trafic = ts(trafic$RPK, start = c(2011,1), end = c(2011,108), freq = 12)
plot(ts_trafic, xlab="Années", ylab="RPK", main="Trafic aérien domestique suèdois de 2011 à 2019",
     col="#33bb77", lwd=2)

ts_trafic_log = ts(trafic_log$RPK, start = c(2011,1), end = c(2011,108), freq = 12)
plot(ts_trafic_log, xlab="Années", ylab="log(RPK)",
     main="Trafic aérien domestique suèdois de 2011 à 2019", col="#33bb77", lwd=2)


##############################
##      Premier modèle      ##
##############################

plot(decompose(ts_trafic_log, type="additive"))

# Pour la tendance
model_1_tendance = tslm(ts_trafic_log ~ trend + I(trend^2) + I(trend^3))
summary(model_1_tendance)

plot(ts_trafic_log, xlab="Années", ylab="log(RPK)", col="#33bb77", lwd=2,
     main="Tendance polynôme de degré 3")
lines(model_1_tendance$fitted.values, lty=2, lwd=2, col="blue")
```

```r
res = model_1_tendance$residuals

par(mfrow=c(1,1))
plot(res, xlab="Années", ylab="Résidus pour la tendance seule")
abline(h=0, lty=3)

Tendance = model_1_tendance$fitted


# Pour la saisonnalité
model_1_saisonnalite = mean((ts_trafic_log - Tendance)[trafic$Mois == 1])
for (i in 2:12){
  model_1_saisonnalite = c(model_1_saisonnalite, mean((ts_trafic_log - Tendance)[trafic$Mois == i]))
}

plot(model_1_saisonnalite - mean(model_1_saisonnalite), type="l", col="blue", xlab="une année",
     ylab="log(RPK)", main="Saisonnalité sur une période de 12 mois")

par(mfrow=c(2,1))
S = rep(model_1_saisonnalite,9) - mean(model_1_saisonnalite)
plot(model_1_tendance$fitted, col="blue", xlab="Années", ylab="log(RPK)")
title("Tendance et Saisonnalité")
plot(time(ts_trafic), S, type="l", col="blue", xy.labels=F, xlab="Années", ylab="log(RPK)")


# On reconstruit la série en additionnant tendance et saisonnalité
T_plus_S = Tendance + S
res_1 = ts_trafic_log - T_plus_S
par(mfrow=c(1,1))
plot(ts_trafic_log, xlab="Années", ylab="log(RPK)", main="Modèle n°1", col="#33bb77", lwd=2)
lines(T_plus_S, col="blue", lty=2, lwd=1.5)
legend('topleft', inset=.05, c("Données","Ajustées avec modèle linéaire"), lty=c(1,2), lwd=c(2,1),
       col=c("#33bb77","blue"), cex=.8)

par(mfrow=c(2,2))
plot(res_1, ylab="Résidus")
abline(h=0)
hist(res_1, main="", xlab="Résidus")
qqnorm(res_1, main="")
qqline(res_1, col="red")
plot(T_plus_S, res_1, xy.lines=FALSE, xy.labels=FALSE, xlab="Fits", ylab="Résidus")
abline(h=0, lty=3)

par(mfrow=c(1,1))
acf(res_1[1:length(res_1)], main="ACF des résidus du modèle n°1 en fonction des mois")


# Même modélisation mais en utilisant 'season' dans la fonction 'tslm()'
model_1_bis = tslm(ts_trafic_log ~ trend + I(trend^2) + I(trend^3) + season)
summary(model_1_bis)

plot(ts_trafic_log, xlab="Années", ylab="log(RPK)", col="#33bb77", lwd=2, main="Modèle n°1 bis")
lines(model_1_bis$fitted.values, lty=2, lwd=1.5, col="blue")

res_1_bis = model_1_bis$residuals

par(mfrow=c(2,2))
plot(res_1_bis, ylab="Résidus")
abline(h=0)
```

```r
hist(res_1_bis, main="", xlab="Résidus")
qqnorm(res_1_bis, main="")
qqline(res_1_bis, col="red")
plot(model_1_bis$fitted, res_1_bis, xy.lines=FALSE, xy.labels=FALSE, xlab="Fits", ylab="Résidus")
abline(h=0, lty=3)

par(mfrow=c(1,1))
acf(res_1_bis[1:length(res_1_bis)], main="ACF des résidus du modèle n°1 bis en fonction des mois")


## Prédictions pour le modèle 1
pred_1 = forecast(model_1_tendance, h=6*12, level=0.95)
S6 = rep(model_1_saisonnalite,6) - mean(model_1_saisonnalite)

# données loguées :
plot(ts_trafic_log, main = "Prédictions avec le modèle n°1 (échelle log)", xlab="Années",
     ylab="log(RPK)", xaxt="n", xlim=c(2011,2026), ylim=c(16.5,20), col="#33bb77", lwd=2)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(T_plus_S, lty=2, col="blue")
lines(pred_1$mean + S6, lty=2, col="red")
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec modèle linéaire","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)

# données réelles :
plot(ts_trafic, main = "Prédictions avec le modèle n°1", xlab="Années", ylab="RPK", xaxt="n",
     xlim=c(2011,2026), ylim = c(0,5e8), col="#33bb77", lwd=2)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(exp(T_plus_S), lty=2, col="blue")
lines(exp(pred_1$mean + S6), lty=2, col="red")
abline(h=0, col="#888888", lty=3, lwd=.5)
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec modèle linéaire","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)


## Prédictions pour le modèle 1 bis
pred_1_bis = forecast(model_1_bis, h=6*12, level=0.95)

# données loguées :
plot(pred_1_bis, xlab="Année", ylab="log(RPK)", xaxt="n",
     main = "Prédictions avec le modèle n°1 bis (échelle log)", fcol="red", flty=2, flwd=1)
lines(pred_1_bis$x, col="#33bb77", lwd=2)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(pred_1_bis$fitted, lty = 2, col = "blue")
abline(h=0, col="#888888", lty=3, lwd=.5)
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec modèle linéaire","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)

# données réelles :
plot(ts_trafic, main = "Prédictions avec le modèle n°1 bis", xlab="Années", ylab="RPK", xaxt="n",
     xlim=c(2011,2026), ylim = c(0,5e8), col="#33bb77", lwd=2)
polygon(c(index(pred_1_bis$lower), rev(index(pred_1_bis$lower))),
        c(exp(pred_1_bis$lower), rev(exp(pred_1_bis$upper))), col="grey90", border=F)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(exp(pred_1_bis$fitted), lty=2, col="blue")
lines(exp(pred_1_bis$mean), lty=2, col="red")
abline(h=0, col="#888888", lty=3, lwd=.5)
```

```r
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec modèle linéaire","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)


# AIC du modèle 1 bis
print(-2*logLik(model_1_bis)[1] - 2*15)



################################
##      Deuxième modèle     ##
################################

model_2 = HoltWinters(ts_trafic_log, seasonal="additive")
print(model_2)
plot(model_2$x, xlab="Années", ylab="log(RPK)", main="Modèle n°2", col="#33bb77", lwd=2)
lines(model_2$fitted[,"xhat"], col="blue", lty=2)
legend('topleft', inset=.02, c("Données","Ajustées avec H-W"), lty=c(1,2), lwd=c(2,1),
       col=c("#33bb77","blue"), cex=0.8)

res_2 = ts_trafic_log - model_2$fitted[,"xhat"]

par(mfrow=c(2,2))
plot(res_2, ylab="Résidus")
abline(h=0)
hist(res_2, main="", xlab="Résidus")
qqnorm(res_2, main="")
qqline(res_2, col="red")
plot(model_2$fitted[,"xhat"], res_2, xy.lines=FALSE, xy.labels=FALSE, xlab="Fits", ylab="Résidus")
abline(h=0, lty=3)

par(mfrow=c(1,1))
acf(res_2[1:length(res_2)], main="ACF des résidus du modèle n°2 en fonction des mois")


## Prédicitons avec le modèle 2
pred_2 = forecast(model_2, h=6*12, level=0.95)

# données loguées :
plot(pred_2, xlab="Année", ylab="log(RPK)", xaxt="n",
     main = "Prédictions avec le modèle n°2 (échelle log)", fcol="red", flty=2, flwd=1)
lines(pred_2$x, col="#33bb77", lwd=2)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(pred_2$fitted, lty = 2, col = "blue")
abline(h=0, col="#888888", lty=3, lwd=.5)
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec H-W","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)

# données réelles :
plot(ts_trafic, main = "Prédictions avec le modèle n°2", xlab="Années", ylab="RPK", xaxt="n",
     xlim=c(2011,2026), ylim = c(0,5e8), col="#33bb77", lwd=2)
polygon(c(index(pred_1$lower), rev(index(pred_1$lower))), c(exp(pred_2$lower), rev(exp(pred_2$upper))),
        col="grey90", border=F)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(exp(pred_2$fitted), lty=2, col="blue")
lines(exp(pred_2$mean), lty=2, col="red")
abline(h=0, col="#888888", lty=3, lwd=.5)
```

```r
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec H-W","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)



################################
##      Troisième modèle      ##
################################

# ACF des données loguées Xt
acf(ts_trafic_log[1:length(ts_trafic_log)], main="ACF des données en fonction des mois")

## Yt = Xt - X(t-12)
Yt = diff(ts_trafic_log, lag=12)
par(mfrow=c(1,1))
plot(Yt, xlab="Années")
par(mfrow=c(1,2))
acf(Yt[1:length(Yt)], main="ACF de Yt")
pacf(Yt[1:length(Yt)], main="PACF de Yt")

# Test augmenté de Dickey-Fuller pour tester l'hypothèse H0 de non-stationnarité contre H1 stationnaire
# on trouve ici une p-value > 5%, on considère donc que la série n'est pas stationnaire
adf.test(Yt)


## Wt = Yt - Y(t-1)
Wt = diff(Yt, lag=1)
par(mfrow=c(1,1))
plot(Wt, type="l", xlab="Années")
abline(h=0, lty=3)
par(mfrow=c(1,2))
acf(Wt[1:length(Wt)], main="ACF de Wt")
pacf(Wt[1:length(Wt)], main="PACF de Wt")

# Test augmenté de Dickey-Fuller
# On trouve ici une p-value < 5%, on considère donc que la série est stationnaire
adf.test(Wt)


## Zt = Xt - St (où St est la saisonnalité)
Saisonnalite = decompose(ts_trafic_log, type="additive")$seasonal

Zt = ts_trafic_log - Saisonnalite
par(mfrow=c(1,1))
plot(Zt, xlab="Années")
par(mfrow=c(1,2))
acf(Zt[1:length(Zt)], main="ACF de Zt")
pacf(Zt[1:length(Zt)], main="PACF de Zt")

# Test augmenté de Dickey-Fuller
# On trouve ici une p-value > 5%, on considère donc que la série n'est pas stationnaire
adf.test(Zt)


## Qt = Zt - Z(t-1)
Qt = diff(Zt, lag=1)
par(mfrow=c(1,1))
plot(Qt, type="l", xlab="Années")
```

```r
abline(h=0, lty=3)
par(mfrow=c(1,2))
acf(Qt[1:length(Qt)], main="ACF de Qt")
pacf(Qt[1:length(Qt)], main="ACF de Qt")

# Test augmenté de Dickey-Fuller
# On trouve ici une p-value < 5%, on considère donc que la série est stationnaire
adf.test(Qt)


### Modèle 3a
ar_1 = arima(Yt, order=c(4,1,0))
t_stat(ar_1)

# on supprime le coefficient AR3 (i.e. on le fixe à 0 pour la suite)
ar_2 = arima(Yt, order=c(4,1,0), fixed = c(NA,NA,0,NA))
t_stat(ar_2)

model_3a = ar_2
res_3a = resid(model_3a)

par(mfrow=c(2,2))
plot(res_3a, ylab="Résidus")
abline(h=0)
hist(res_3a, main="", xlab="Résidus")
qqnorm(res_3a, main="")
qqline(res_3a, col="red")
plot(Yt - res_3a, res_3a, xy.lines=FALSE, xy.labels=FALSE, xlab="Fits", ylab="Résidus")
abline(h=0, lty=3)

par(mfrow=c(1,1))
acf(res_3a[1:length(res_3a)], main="ACF des résidus du modèle n°3a en fonction des mois")


### Modèle 3b
arima_1 = arima(Zt, order=c(4,1,7))
t_stat(arima_1)

# on supprime le coefficient MA3 (i.e. on le fixe à 0 pour la suite)
arima_2 = arima(Zt, order = c(4,1,7), fixed = c(NA,NA,NA,NA, NA,NA,0,NA,NA,NA,NA))
t_stat(arima_2)

# on supprime le coefficient MA1
arima_3 = arima(Zt, order = c(4,1,7), fixed = c(NA,NA,NA,NA, 0,NA,0,NA,NA,NA,NA))
t_stat(arima_3)

model_3b = arima_3
res_3b = resid(model_3b)

par(mfrow=c(2,2))
plot(res_3b, ylab="Résidus")
abline(h=0)
hist(res_3b, main="", xlab="Résidus")
qqnorm(res_3b, main="")
qqline(res_3b, col="red")
plot(Zt - res_3b, res_3b, xy.lines=FALSE, xy.labels=FALSE, xlab="Fits", ylab="Résidus")
abline(h=0, lty=3)

par(mfrow=c(1,1))
```

```r
acf(res_3b[1:length(res_3b)], main="ACF des résidus du modèle n°3b en fonction des mois")


### Modèle 3c
t_stat(arima_3)

# on supprime le coefficient MA7 (cela revient donc un MA(6))
arima_4 = arima(Zt, order = c(4,1,6), fixed = c(NA,NA,NA,NA, 0,NA,0,NA,NA,NA))
t_stat(arima_4)

# on supprime le coefficient AR3
arima_5 = arima(Zt, order = c(4,1,6), fixed = c(NA,NA,0,NA, 0,NA,0,NA,NA,NA))
t_stat(arima_5)

# on supprime le coefficient MA2
arima_6 = arima(Zt, order = c(4,1,6), fixed = c(NA,NA,0,NA, 0,0,0,NA,NA,NA))
t_stat(arima_6)

# on supprime le coefficient AR4 (cela revient donc un AR(2))
arima_7 = arima(Zt, order = c(2,1,6), fixed = c(NA,NA, 0,0,0,NA,NA,NA))
t_stat(arima_7)

model_3c = arima_7
res_3c = resid(model_3c)

par(mfrow=c(2,2))
plot(res_3c, ylab="Résidus")
abline(h=0)
hist(res_3c, main="", xlab="Résidus")
qqnorm(res_3c, main="")
qqline(res_3c, col="red")
plot(Zt - res_3c, res_3c, xy.lines=FALSE, xy.labels=FALSE, xlab="Fits", ylab="Résidus")
abline(h=0, lty=3)

par(mfrow=c(1,1))
acf(res_3c[1:length(res_3c)], main="ACF des résidus du modèle n°3c en fonction des mois")


## Comparaison des modèles
print(model_3a$aic)
print(model_3b$aic)
print(model_3c$aic)


## Prédictions pour le modèle 3c
pred_3c = forecast(model_3c, h=6*12, level=0.95)
Saisonnalite_6 = Saisonnalite[1:(6*12)]

# données loguées :
plot(ts_trafic_log, main = "Prédictions avec le modèle n°3c (échelle log)", xlab="Années",
     ylab="log(RPK)", xaxt="n", xlim=c(2011,2026), ylim=c(18.4,20), col="#33bb77", lwd=2)
polygon(c(index(pred_3c$lower), rev(index(pred_3c$lower))),
        c(pred_3c$lower + Saisonnalite_6, rev(pred_3c$upper + Saisonnalite_6)), col="grey90", border=F)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(pred_3c$fitted + Saisonnalite, lty=2, col="blue")
lines(pred_3c$mean + Saisonnalite_6, lty=2, col="red")
abline(h=0, col="#888888", lty=3, lwd=.5)
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec ARIMA(2,1,6)","Prédites"), lty=c(1,2,2),
```

```r
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)

# données réelles :
plot(ts_trafic, main = "Prédictions avec le modèle n°3c", xlab="Années", ylab="RPK", xaxt="n",
     xlim=c(2011,2026), ylim = c(0,5e8), col="#33bb77", lwd=2)
polygon(c(index(pred_3c$lower), rev(index(pred_3c$lower))),
        c(exp(pred_3c$lower + Saisonnalite_6), rev(exp(pred_3c$upper + Saisonnalite_6))),
        col="grey90", border=F)
axis(1, at = seq(2011, 2026, 1), labels = format(seq(2011, 2026, 1)))
lines(exp(pred_3c$fitted + Saisonnalite), lty=2, col="blue")
lines(exp(pred_3c$mean + Saisonnalite_6), lty=2, col="red")
abline(h=0, col="#888888", lty=3, lwd=.5)
abline(v=2020, col="#cc00dd", lwd=1.5)
legend('bottomleft', inset=.05, c("Données","Ajustées avec ARIMA(2,1,6)","Prédites"), lty=c(1,2,2),
       lwd=c(2,1,1), col=c("#33bb77","blue","red"), cex=0.8)
```