INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Prévision du trafic aérien interne suédois

Module « Séries Temporelles »,4GM

Tangi TASSIN

Table des matières

Introduction & Observation qualitative

1.1 Du projet

Ce projet pédagogique issu du module 4GM "**Séries Temporelles**" a pour principal objectif de prédire le trafic aérien suédois interne (décrit en RPK) sur les années 2020-2021. L'échantillon de données inclus les années 2011 à 2019 (inclus). La prise en compte de la saisonnalité et l'étude de divers modèles de prédiction, étudiés cette année, va permettre une toute nouvelle approche d'apprentissage automatique complémentaire à notre formation sur les régressions linéaires généralisées. Pour ce projet 3 modélisations seront étudiées :

— Régression Linéaire & Moyenne mobile ;
— Lissage de Holt-Winters ;
— (S)ARIMA.

Après réalisation, ces modèles seront comparés entre-eux afin de choisir (via certains critères) le *"meilleur"* modèle.

1.2 De l'impact de la crise COVID19 sur le trafic

Il est certain que la crise sanitaire a eu un fort impact sur le trafic aérien, international ou domestique. Que ce soit par arrêté, pour confinement, par peur d'être contaminé etc..

Il est difficile d'avoir du recul sur l'impact qu'a eu cette crise sur l'année 2020 ; encore moins sur les années à venir. Les prédictions que nous ferons ici se basant sur les années précédentes, elles ne refléteront certainement pas le trafic réel de 2020, 2021 et s'en suit. Tout cela dépendant de l'avancée de la pandémie, des décisions des États, en particulier de la Suède.

Ce projet et les prédictions proposées ont donc l'aspect "formation" comme intérêt principal. Cependant compte tenu des (faibles) données que nous avons sur la diminution du RPK en 2020, sera appliquée une variation moyenne sur les années 2020, 2021 et les suivantes.

Si la tendance n'est plus valable, ce n'est plus le cas pour la saisonnalité également. Mais pour l'intérêt pédagogique du projet nous les considérerons tout de même. Cette modification ne sera faite qu'en synthèse et après choix du modèle et des prédictions sans prise en compte de la crise sanitaire.

2020 se trouve dans les valeurs à prédire pour ce projet, mais étant la seule source d'information sur l'impact COVID que nous avons à disposition nous serons obligés de les utiliser.

On peut voir sur la figure 1.1 l'évolution du RPK pour les trafics domestiques en 2020 (mondial). Source : A.

De manière empirique nous choisirons de prendre la moyenne à partir du mois d'avril 2020 et jusqu'à octobre 2020 (par manque de données). La moyenne obtenue est une diminution de **-61%**. Plutôt que de considérer cette valeur comme "variation par rapport à l'année précédente" nous la considérerons comme "variation par rapport aux valeurs prédites". Pour être plus précis il faudrait donc redresser légèrement le coefficient, 2020 ayant une valeur moyenne plus faible qu'en 2019 sans considérer la crise sanitaire (cette affirmation sera vérifiée dans les parties suivantes).

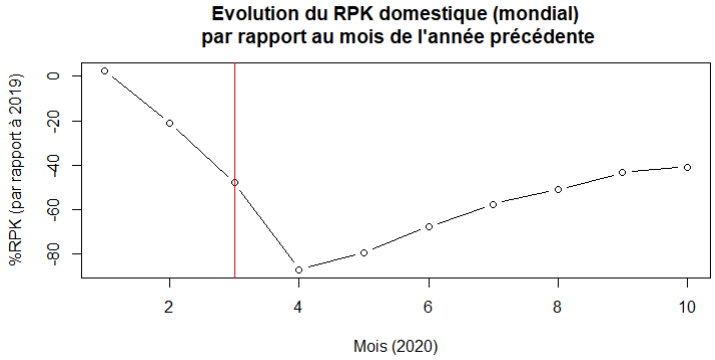

FIGURE 1.1 – Évolution du RPK domestique à l'internationale en 2020

1.3 Allure de la série temporelle

Nous allons pour commencer nous intéresser à l'aspect graphique de cette série temporelle.

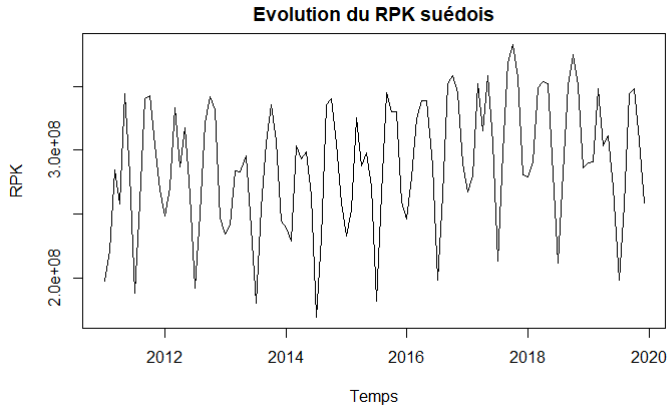

FIGURE 1.2 – Représentation graphique de l'évolution du RPK

Ce que l'on peut observer sur le graphique 1.2, c'est une série possédant une tendance, avec une saisonnalité (détaillée ci-après) d'un an. N'ayant pas travaillé sur les composantes cycliques il n'en sera pas fait mention ici : même si la représentation graphique suggère une absence de cette composante.

On peut remarquer que les valeurs du RPK sont grandes (avec un ordre de grandeur de 10^8). On peut imaginer que les valeurs varient beaucoup, auquel cas il serait intéressant de travailler sur les $\log RPK$.

Le graphique 1.3 montre qu'il n'en est rien. Les deux représentations sont équivalentes, au sens des variations. Pour la compréhension du lecteur et des résultats nous utiliserons donc les valeurs normalisées $RPK' := RPK * 10^{-}8$, afin d'avoir des valeurs proches de 1. Il aurait été aussi intéressant de travailler sur les valeurs $\frac{RPK}{\hat{\sigma}}$ avec $\hat{\sigma}$ l'écart-type estimé. Cette approche aurait eu le mérite d'être plus proche des habitudes statistiques.

Étant donnée cette représentation nous utiliserons le modèle additif par la suite :

$$RPK_t^{'} = T_t + S_t + \epsilon_t$$

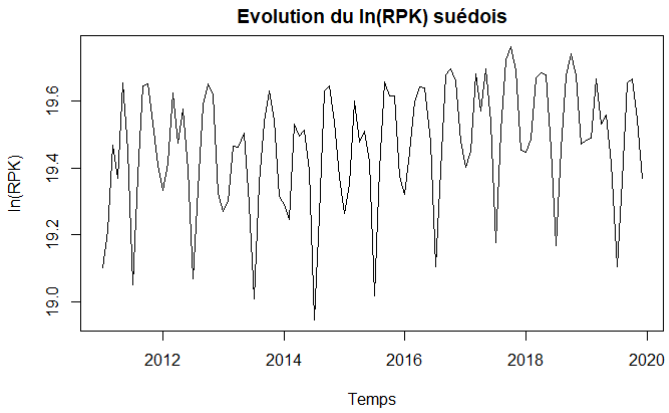

FIGURE 1.3 – Représentation graphique de l'évolution du $ln(RPK)$

1.4 Allure de la saisonnalité

Pour ce qui est de la saisonnalité, une étude des moyennes des valeurs sur la période 2011-2019 nous permet d'avoir une bonne idée de la forme de celle-ci.

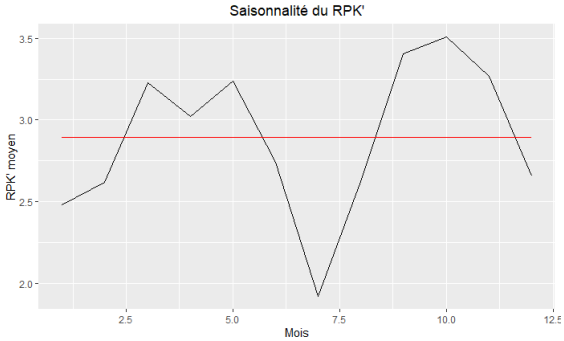

FIGURE 1.4 – Étude **rapide** de la saisonnalité

la figure 1.4 nous donne un aperçu de la saisonnalité. La moyenne de l'ensemble est représentée par la ligne rouge. Bien sûr pour avoir une meilleure précision il faudrait retirer la tendance, que nous n'avons pas encore étudiée. Nous nous en occuperons plus tard.

La forme de la tendance ne paraît pas évidente de prime abord. Nous utiliserons donc divers modèles quantitatifs pour la déterminer.

1.5 Échantillon d'apprentissage

Une part importante de ce projet est bien sûr la prévision des valeurs du RPK (et donc du RPK') sur des dates post-2019. Pour valider notre modèle nous répartirons donc nos données sur un échantillon d'apprentissage et un de validation.

Une variation de la saisonnalité est visible (figure 1.2) vers la fin de la série (une décroissance à partir de 2018). Il va donc être important de s'approcher le plus possible de ces années pour prendre en compte dans notre modèle cette décroissance. L'idéal aurait été d'avoir plus de données après 2018 pour prendre en compte cette période de décroissance dans le modèle, mais afin d'avoir un échantillon de validation d'une taille suffisante nous ne pourrons pas les prendre en compte.

Notre échantillon d'apprentissage sera constitué des années 2011 à 2017. L'échantillon de validation sera lui sur deux périodes : 2018 et 2019.

Moyennes mobiles & Régression linéaire

2.1 Étude de la série totale

Pour cette partie nous allons estimer la tendance par la méthode des moyennes mobiles. Plus exactement, nous allons nous intéresser à l'allure de celle-ci afin de récupérer quelques informations qui nous serons utiles par la suite. Les moyennes mobiles ici ne sont qu'une première étape à la réalisation du modèle ; mais elle nous permet de récupérer des informations (temporelles notamment) très importantes. On utilisera la moyenne mobile usuelle pour cette saisonnalité (12 mois), centrée et normalisée.

$$M * X_t = \frac{1}{12}(\frac{X_{t-6}}{2} + X_{t-5} + ... + X_{t-1} + X_t + X_{t+1} + ... + X_{t+5} + \frac{X_{t+6}}{2})$$

Avec M l'opérateur moyenne mobile.

En appliquant cet opérateur à la totalité de la série, de manière automatique grâce à la méthode *decompose()* de R notamment, on obtient la tendance représentée sur la figure 2.1. La forme de celle-ci peut faire penser à une sinusoïde.

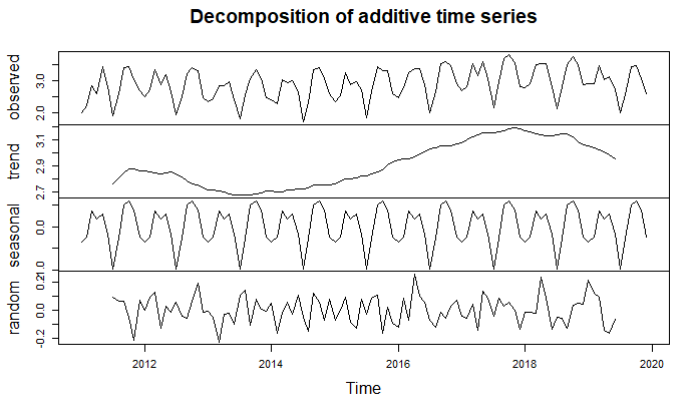

FIGURE 2.1 – Décomposition de la série totale

La période de cette sinusoïde peut-être (approximativement) déterminée en récupérant la valeur maximale (M), la valeur minimale (m) ; puis en réalisant le simple calcul $T = |2 * (t_M - t_m)|$. La pulsation associée est donc $\omega = \frac{2\pi}{T}$. On trouve ici une période de 8,5 années. Il est intéressant de noter ici que $t_m < t_M < 2018$, date limite de l'échantillon d'apprentissage.

En réalisant une régression linéaire de la forme $\beta_0 + \beta_1 \sin[2\pi T * (t - t_0)] + \epsilon_t$, avec $t_0 := \frac{t_m + t_M}{2}$, on obtient une tendance estimée de la forme suivante :

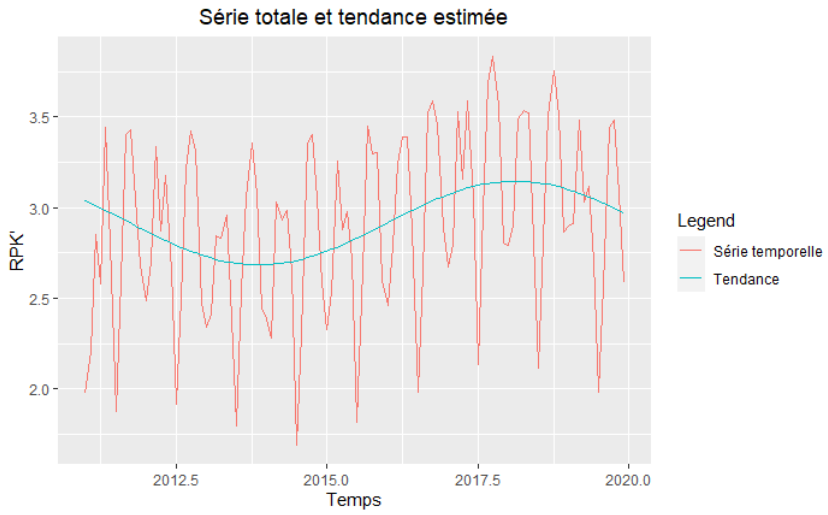

FIGURE 2.2 – Tendance estimée sur la série totale

La tendance estimée "collant" bien à la tendance que l'on peut observer qualitativement, on va pouvoir utiliser cette régression sur l'échantillon d'apprentissage.

2.2 Apprentissage

Intéressons-nous maintenant à l'échantillon d'apprentissage pour réaliser notre modèle. Suite à nos observations sur la série totale, nous connaissons les temps t_m et t_M qui sont inférieurs à 2018. Nous allons donc pouvoir utiliser ces temps pour notre échantillon d'apprentissage. La pulsation et t_0 restent donc les mêmes que précédemment.

Après une première étude sur l'échantillon d'apprentissage complet, il s'avère que les résidus studentisés pour les 4 premiers mois sont très élevés. Les différentes validations pour ce modèles sont disponibles en annexe (B). Ces premiers mois ne seront donc pas pris en compte pour notre modèle.

Notre jeu d'apprentissage est donc l'intervalle [mai 2011, décembre 2017].

On va de nouveau réaliser un modèle de régression linéaire afin de pouvoir faire des prévisions sur la tendance. On conservera l'hypothèse d'une tendance sinusoïdale, avec les caractéristiques (T, t_0) égales à celles trouvées via la décomposition totale : cela nous permet de conserver de l'information qui disparaît si on s'intéresse à la décomposition restreinte.

La régression linéaire se fera sur le modèle suivante :

$$RPK'_t = \beta_0 + \beta_1 sin(\omega(t - t_0)) + S_t + \epsilon_t$$

Les coefficients β_i et S_j seront estimés par MCO.

Les coefficients estimés (au nombre de $p = 13$) et les valeurs associées sont disponibles en annexe (B.2). On pourra retenir les quelques valeurs suivantes pour avoir en tête un ordre de grandeur.

	Estim.	P-valeur
$\hat{\beta_0}$	2,51	$< 2.10^-16$
$\hat{\beta_1}$	0,138	$< 2.10^-16$
$\hat{S_i}$	0,439	« 3 %

Avec l'estimation des S_i dans la tableau étant une valeur moyenne.

Le coefficient R^2 de Pearson associé à cette régression est élevé : *0,9594*.

En supposant que les hypothèses du modèle linéaire gaussien sont respectées, au risque $\alpha = 5\%$ l'ensemble des coefficients sont significatifs d'après le test de Student. A priori notre modèle semble correct ; ce que nous allons vérifier à l'étape suivante.

2.3 Validation

2.3.1 Des prédictions

Tendance et saisonnalité étant estimées, notre modèle est complet.

Nous allons pouvoir faire des prévisions sur les années 2018 et 2019, puis discuter de la validité du modèle grâce à notre échantillon de validation.

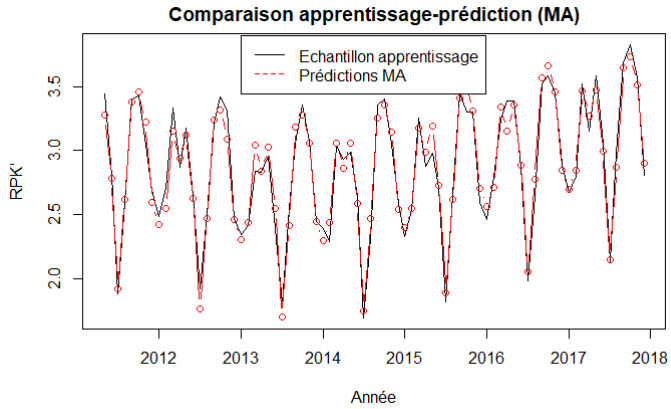

FIGURE 2.3 – Comparaison série d'apprentissage & prédictions

Ce que l'on peut remarquer avec la figure 2.3, c'est une très bonne prédiction du modèle linéaire sur la série d'apprentissage.

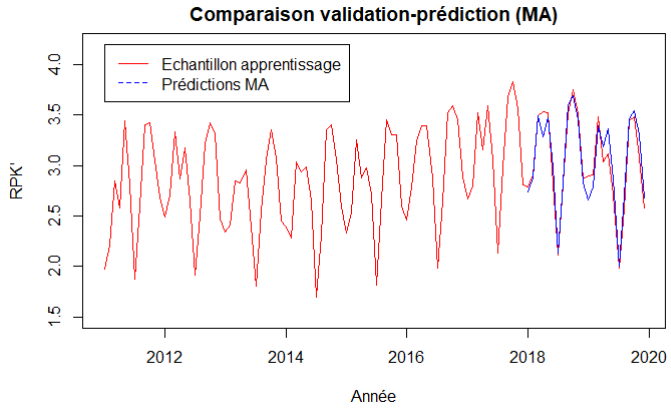

FIGURE 2.4 – Comparaison série de validation & prédictions

On remarquera également que les prédictions exprimées sur l'échantillon de validation sont elles aussi très bonnes. La validation des prédictions ne se fera ici que partiellement ; une étude plus poussée se faisant en Synthèse (5) en fin de ce rapport. Tout cela afin de comparer les modèles entre eux.

2.3.2 Du modèle linéaire

Afin de valider les hypothèses utilisées pour la création du modèle, il convient de considérer les divers représentations usuelles.

Le graphique 2.5 indique une indépendance entre les résidus $\hat{\epsilon_i}$ et les valeurs des $\hat{RPK'_i}$. On remarque également que les résidus sont centrés.

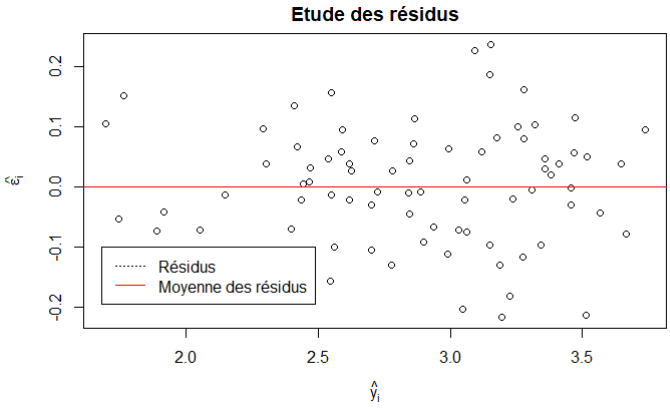

FIGURE 2.5 – Étude des résidus

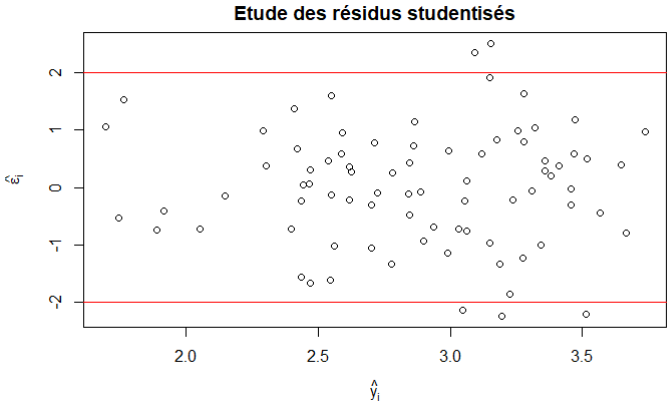

FIGURE 2.6 – Étude des résidus studentisés

La figure des résidus studentisés (2.6) montre que la grande majorité de ces résidus se trouvent dans l'intervalle [-2 ; 2] correspondant (approximativement) à l'intervalle de confiance de risque $\alpha = 5\%$. 5 points seulement (sur 80) sont très légèrement en dehors de l'intervalle mais de très peu. On peut donc considérer qu'il n'y a aucun point aberrant (contrairement aux 4 premiers points que l'on pouvait trouver sur le modèle avec les 4 premières valeurs). De plus, l'absence d'une structure spécifique laisse suggérer une homoscédasticité.

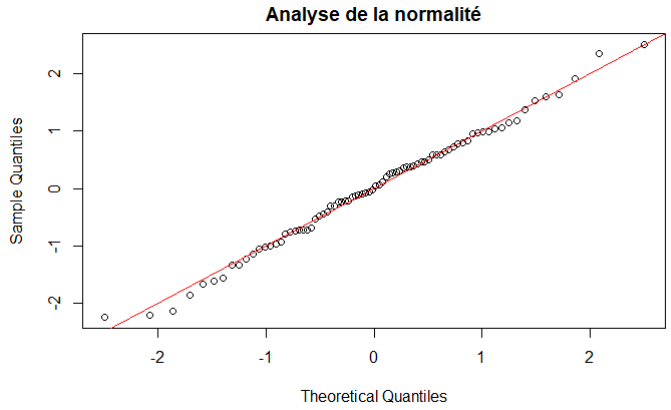

FIGURE 2.7 – Diagramme QQ des résidus studentisés (loi $N(0,1)$)

Enfin, le diagramme Quantile-Quantile 2.7 indique un suivi de la loi Normale Centrée Réduite.

On va pouvoir considérer que les hypothèses de régression gaussienne sont respectées. Les conclusions faites en 2.2 sur les coefficients sont donc applicables. Le modèle peut-done être validé.

Lissage de Holt-Winters

Le lissage de Holt-Winters pour les séries avec saisonnalité est une méthode de prédiction assez pratique pouvant donner de bons résultats. Pour rappel, trois coefficients $\alpha, \beta, \gamma \in [0,1]$ permettent de moduler le lissage. Ces coefficients peuvent être choisis manuellement ou déterminés via un critère, par exemple l'erreur de prédiction à 1 pas :

$$\Sigma_{t=1}^{T-1}(X_{t+1} - X_t\hat{(1)})^2$$

C'est ce critère que nous utiliserons par la suite. Comme précédemment nous utiliserons un modèle additif.

3.1 Apprentissage

Si l'on utilise le critère présenté précédemment, nous obtenons les valeurs indiquées dans le tableau 3.1 pour les coefficients.

Coefficient	Prédiction 1 pas	$\Sigma\epsilon_i^2$ (Validation)
α	0,1764	0,18
β	0.0801145	0,64
γ	0,3264452	0,29

TABLE 3.1 – Valeurs optimales des coefficients

3.2 Validation

Comme précédemment, la validation ici ne sera que partielle. Des caractéristiques seront détaillées davantage dans la Synthèse (5).

Si l'on s'intéresse qualitativement aux résultats : la figure 2.4 nous montre une très bonne prédiction sur le modèle d'entraînement, tant sur la tendance que sur la saisonnalité. On peut observer des résidus élevés en mi-2016, cette période ne respectant pas la saisonnalité usuelle.

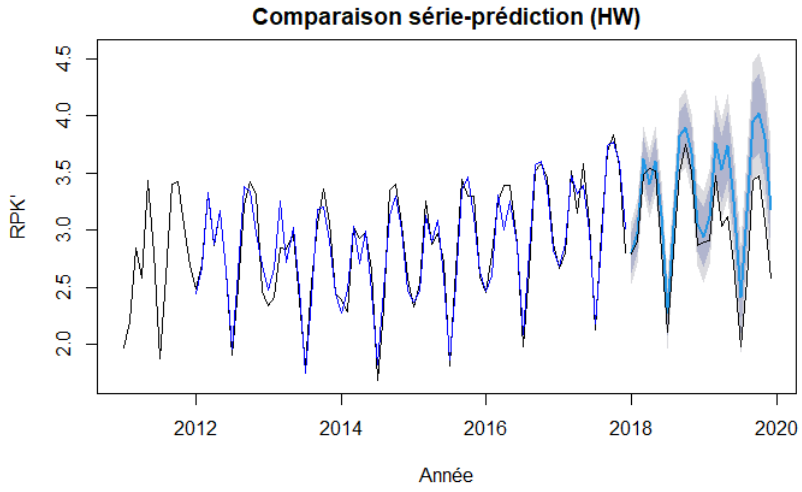

FIGURE 3.1 – Comparaison série de validation & prédictions

Pour ce qui est des prédictions sur l'échantillon de validation : ici la prédiction n'est vraiment pas bonne. On peut observer un 'décrochage' notamment sur l'année 2019 où 5 valeurs (sur 12) ne sont pas dans l'intervalle de confiance à 95%. A priori ce modèle semble moins efficace que le modèle basé sur la régression linéaire.

De plus, les valeurs des coefficients déterminés via l'erreur de prédiction à 1 pas ne sont pas les valeurs qui permettent d'obtenir les meilleurs prédictions sur l'échantillon de validation. En effectuant une recherche manuelle $((\alpha, \beta, \gamma) \in \{0.1; 0.2; ....; 1\}^3$, soit 1000 observations. Puis en réitérant une fois avec une précision au centième) ; on obtient des valeurs différentes (voir Annexe C.1). Ne pouvant se baser sur l'échantillon de validation nous allons donc rester avec les coefficients évalués

automatiquement.

Modèle (S)ARIMA

4.1 Un modèle ARIMA simple

La formation GM ne proposant pas d'étude des modèles SARIMA, il a été choisi de travailler prioritairement sur un modèle ARIMA afin de travailler sur des modèles bien connus.

4.1.1 Désaisonnalisation

La première problématique venant étant la présence de saisonnalité dans la série temporelle ; non compatible avec ce type de modèles. Afin de dessaisonnaliser la série un 'lag' de 12, i.e $\tilde{X}_t := X_t - X_{t-12}$ sera appliqué. A noter que l'on perd nécessairement 12 données sur la nouvelle série (les 12 premières, soit la première année).

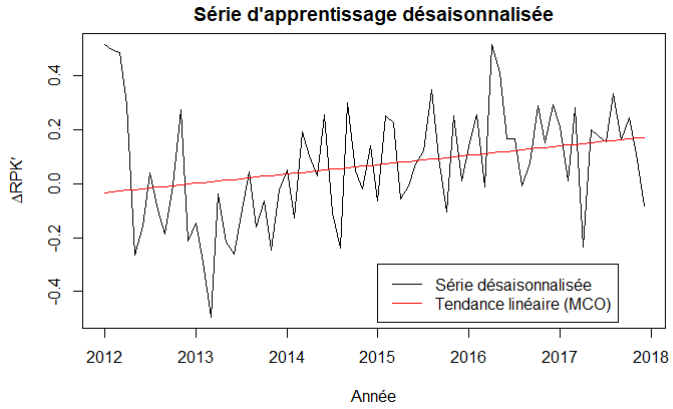

FIGURE 4.1 – Désaisonnalisation de la série d'apprentissage

Le graphique 4.1 nous montre que la série a bien été désaisonnalisée : pas besoin de différenciation supplémentaire.

4.1.2 Détendancialisation

Cependant on peut observer sur ledit graphique que la série possède toujours une tendance (quasi-linéaire). La série n'est donc toujours stationnaire. Cette information rejette directement l'hypothèse nulle du test du Dickey-Fuller qui sera détaillé plus loin (le résultat étant une série TS, c-à-d *Trend Stationnary*).

En réalisant des régressions linéaire du premier et du second ordre, on se rend compte que la tendance n'est que de degré 1. Afin de la détendancialiser nous allons devoir opérer une nouvelle différentiation d'ordre 1. Nous obtenons alors la formule suivante :

$$\bar{X}_t := \tilde{X}_t - \tilde{X_{t-1}} = X_t - X_{t-1} - (X_{t-12} - X_{t-13})$$

En appliquant cette modification sur la série des RPK', on obtient cette nouvelle série $R\bar{P}K'_t$. Il va maintenant être nécessaire de s'assurer de sa stationnarité.

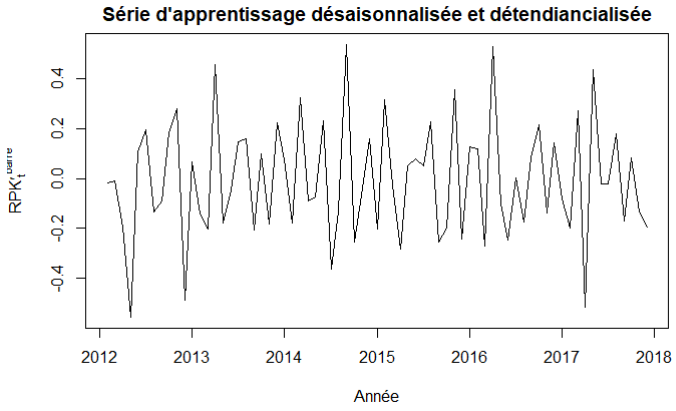

FIGURE 4.2 – Désaisonnalisation & détendancialisation de la série d'apprentissage

Pour ce faire le *test de Dickey-Fuller* sera utilisé. Pour rappel, le modèle de **Dickey-Fuller** peut s'écrire de la manière suivante :

$$\Delta R\bar{P}K'_t = c + b.t + (\phi_1 - 1)RP\bar{K}'_{t-1} + \epsilon_t$$

En suivant la procédure les étapes suivantes sont déroulées :
- Modèle général, b non significatif ;
- Modèle sans tendance ($b = 0$), c non significatif ;
- Modèle précédent ($b = 0$), $\phi_1 - 1$ significatif.

$Coeff\hat{icient}$	Valeur	$P(\|z\| >)$
$\phi_1 - 1$	-1,42	$< 2 * 10^{-16}$

TABLE 4.1 – Coefficients estimés au modèle de Dickey-Fuller

On peut donc au risque fixé $\alpha = 5\%$ valider la stationnarité de la série $R\bar{P}K'_t$.

4.1.3 Apprentissage

L'objectif ici est donc de créer un modèle ARIMA(p,d,q). plutôt que de travailler sur la série $R\bar{P}K'_t$ déterminée précédemment, nous allons travailler sur la série désaisonnalisée $\tilde{X}_t$. Il suffira pour la modélisation de fixer $\mathbf{d = 1}$ (on a vu précédemment qu'il n'y avait pas d'intérêt à prendre $d > 1$) ; d correspondant au *nombre de différenciations* à effectuer. On sait déjà que p et q sont tous deux non nuls. En effet les ACF & PACF de la série $R\bar{P}K'_t$ (4.3) ne sont pas nuls à partir d'une certaine valeur.

Nous utiliserons le critère **AIC** (*Akaike Information Criterion*) pour choisir notre modèle. Nous ferons une recherche automatique sur les modèles ayant les propriétés suivantes :
- $p, q < 5$;
- $d = 1$.

Sous le critère AIC le meilleur modèle obtenu est l'**ARIMA(2,1,1)**. Ce modèle peut être écrit de la manière suivante :

$$R\tilde{P}K'_t = (1 + \phi_1)\tilde{X_{t-1}} + \phi_2\tilde{X_{t-2}} + \epsilon_t + \psi_1\epsilon_{t-1}$$

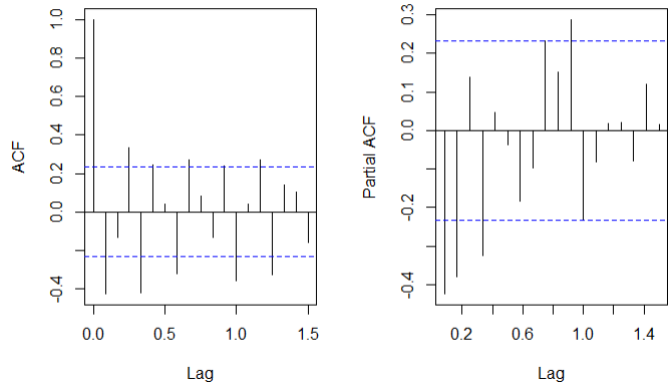

FIGURE 4.3 – ACF & PACF de la série $R\bar{P}K'_t$

Avec $R\tilde{P}K'_t$ la série désaisonnalisée et $(\epsilon_t)_{t\in\mathbb{N}}$ un bruit blanc gaussien $N(0,\sigma)$.

Sur la série d'apprentissage nous obtenons les coefficients indiqués sur le tableau 4.2.

$\hat{Coefficient}$	Valeur	$\hat{\sigma}$
ϕ_1	-1,1251	0,1542
ϕ_2	-0,5893	0,0933
ψ_1	0,7021	0,1969

TABLE 4.2 – Coefficients estimés du modèle ARIMA(2,1,1)

Plus de détails sur les données associées au modèle choisi ici peuvent se trouver en annexe (D.1).

4.1.4 Validation

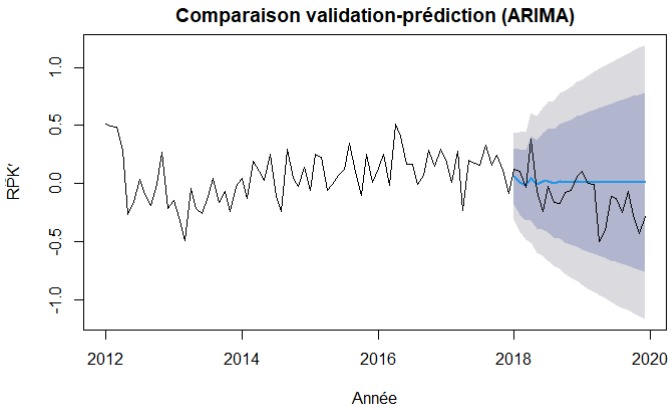

FIGURE 4.4 – Comparaison série de validation & prédictions

En commençant par la validation sur les prédictions, il est clair que ce modèle n'est pas adapté. Si les valeurs de validations se trouvent bien dans l'intervalle, elles sont loin de la prédiction qui ne suit pas son mouvement (graphique 4.4). Ce modèle peut de suite être rejeté.

Alors que pourtant, les hypothèses nécessaires à la réalisation du modèle et au calcul des coefficients sont vérifiées : série stationnaire par différentiation, normalité i.d des résidus.. (voir Annexe D.2). Cependant d'après le test de *Ljung-Box* **l'indépendance des résidus n'est pas respectée**.

Le modèle ARIMA n'est donc pas du tout intéressant ici. Il serait possible de s'arrêter là sur la partie ARMA, mais un modèle SARIMA peut également être intéressant à modéliser.

4.2 Un modèle SARIMA

4.2.1 Apprentissage

Nous l'avons vu précédemment, la série peut être considérée comme stationnaire en la désaisonnalisant et en la différenciant. Malheureusement les prévisions ne sont pas bonnes. Si la modélisation SARIMA n'a pas été vue en cours en spécialité GM (tout du moins jusque début S7), nous allons quand même tenter une modélisation. Celle-ci sera sans doute plus efficace que le modèle ARIMA, pouvant prendre directement en compte la saisonnalité.

En utilisant les mêmes critères que pour le modèle ARIMA, le modèle SARIMA(p,d,q)(P,D,Q) obtenu est le **SARIMA(4,1,0)(2,1,0)**. Les coefficients déterminés se trouvent en Annexe D.1

4.2.2 Validation

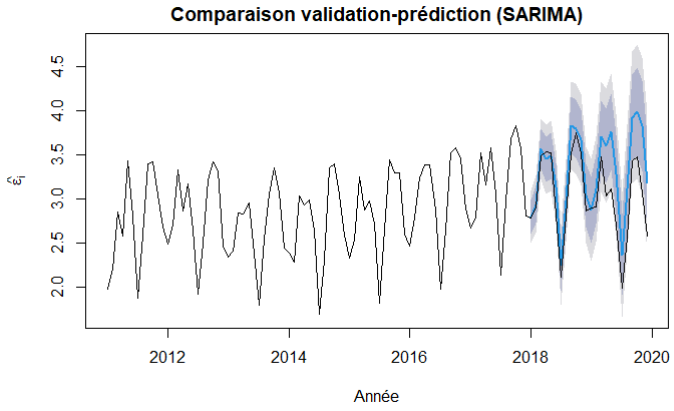

FIGURE 4.5 – Comparaison série de validation & prédictions

Il est clair ici que le modèle SARIMA(4,1,0)(2,1,0) suit bien mieux la courbe au niveau de l'échantillon de validation relativement à l'ARMA(2,1,1). On peut voir cependant que l'année 2019 est moins bien prédite.

Malheureusement, l'hypothèse d'un suivi i.i.d $N(0, \sigma)$ des résidus n'est pas respectée. Si la normalité (diagramme Q-Q et test de *Kolmogorov-Smirnov*, D.6) et l'homoscédasticité (diagramme des résidus studentisés, D.5) sont bien respectées, le test de *Ljung-Box* montre très clairement une forte corrélation entre les résidus (comme précédemment). L'hypothèse d'indépendance n'est donc pas respectée, ceci pouvant expliquer les mauvaises prédictions sur 2019 (mauvaise estimation des coefficients). La non-réalisation de cette hypothèse n'est pas un problème en soi pour les prédictions ; que nous pourrons donc comparer aux autres modèles.

Nous ne retiendrons que le modèle SARIMA (et non l'ARIMA) pour des raisons évidentes d'efficacité.

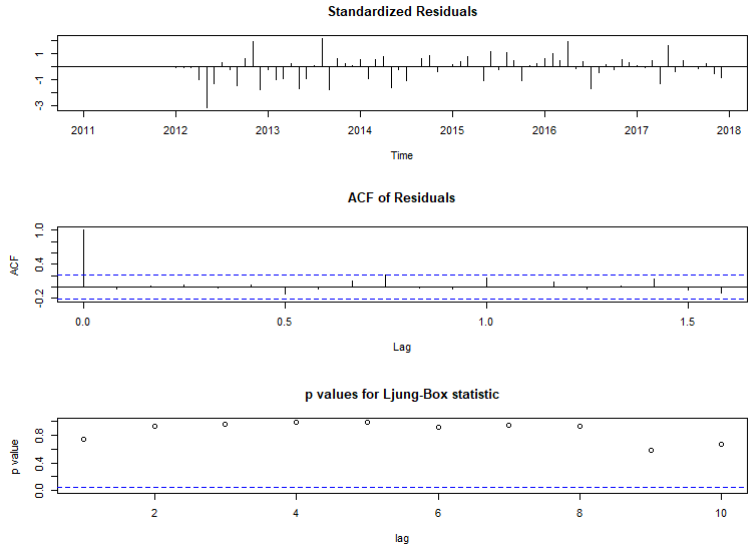

FIGURE 4.6 – Résidus et statistique de Ljung-Box (SARIMA)

Synthèse

5.1 Choix du modèle

Nos différents modèles (Régression Linéaire, souvent appelé ici moyenne mobile ; Holt-Winters ; SARIMA) étant réalisés ; nous allons pouvoir nous intéresser aux performances relatives. Dit autrement, "*quel est le meilleur modèle ?*". Nous nous intéresserons à la dispersion des résidus et à leur somme au carré ($\Sigma\epsilon_i^2$, erreur quadratique) sur l'échantillon de validation.

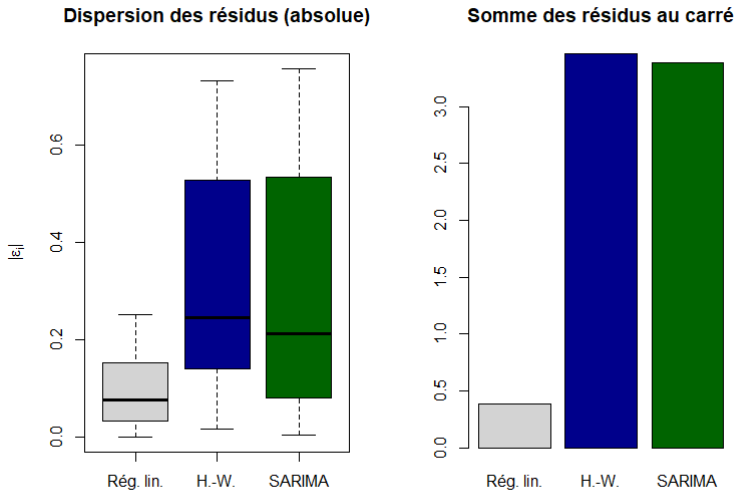

FIGURE 5.1 – Dispersion des résidus (échantillon de validation)

La figure 5.1 nous montre que le choix est vite fait : la régression linéaire est bien plus précise que les modèles Holt-Winters et SARIMA. La modélisation SARIMA partait déjà avec des coefficients mal calculés, dans le sens où l'hypothèse d'indépendance des erreurs n'est ici pas respectée. Contrairement au modèle de régression linéaire qui les vérifie toutes, tout du moins vu les conjectures que l'on a pu faire avec les diverses représentations graphiques.

La régression linéaire étant sélectionnée, nous allons pouvoir l'entraîner sur la série entière (aux 4 premiers mois près, ces derniers donnant des résidus très élevés car ne suivant pas la tendance sinusoïdale proposée).

Les conclusions sont les mêmes qu'en 3.2 : tous les coefficients calculés sont significatifs ; les hypothèses de normalité i.i.d des résidus est vérifiée. Peu de points sur 5.5 sont en dehors de l'intervalle de confiance à 95% et sont proches des bornes de ce dernier. Nous allons donc pouvoir conserver ce modèle.

```
Call:
tslm(formula = tsTotBis ~ sinus + season)

Residuals:
      Min        1Q    Median        3Q       Max
-0.223097 -0.075609  0.009536  0.065095  0.248424

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.54781    0.03926  64.894  < 2e-16 ***
sinus        0.23393    0.01555  15.044  < 2e-16 ***
season2      0.12180    0.05552   2.194 0.030810 *
season3      0.72851    0.05552  13.121  < 2e-16 ***
season4      0.53023    0.05552   9.550 2.23e-15 ***
season5      0.68798    0.05396  12.750  < 2e-16 ***
season6      0.19012    0.05396   3.523 0.000669 ***
season7     -0.62683    0.05396 -11.617  < 2e-16 ***
season8      0.07550    0.05396   1.399 0.165176
season9      0.85609    0.05396  15.866  < 2e-16 ***
season10     0.95918    0.05396  17.776  < 2e-16 ***
season11     0.71929    0.05396  13.330  < 2e-16 ***
season12     0.11355    0.05396   2.104 0.038106 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.111 on 91 degrees of freedom
Multiple R-squared:  0.9544,    Adjusted R-squared:  0.9483
F-statistic: 158.6 on 12 and 91 DF,  p-value: < 2.2e-16
```

FIGURE 5.2 – Résultats de la régression linéaire sur la série tronquée

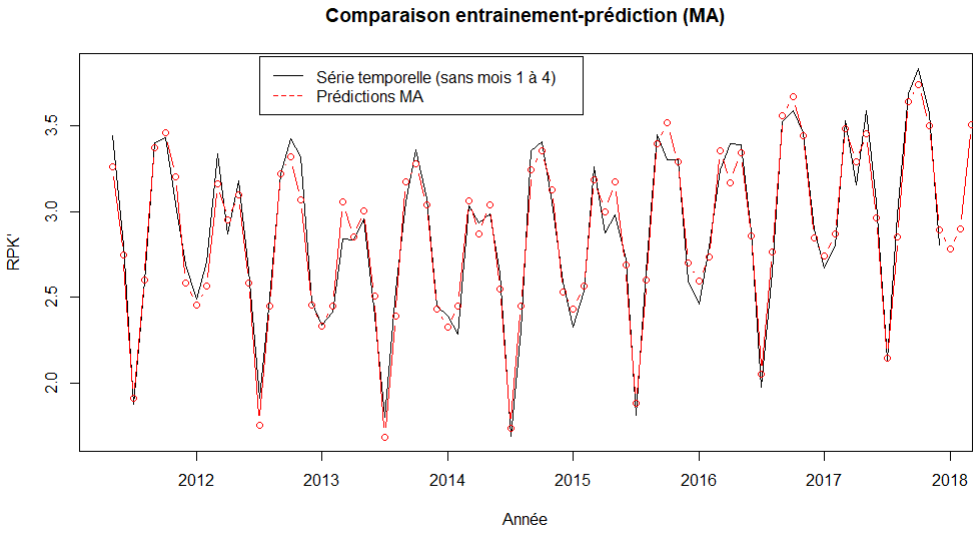

FIGURE 5.3 – Comparaison série d'entraînement & prédictions

FIGURE 5.4 – Étude des résidus

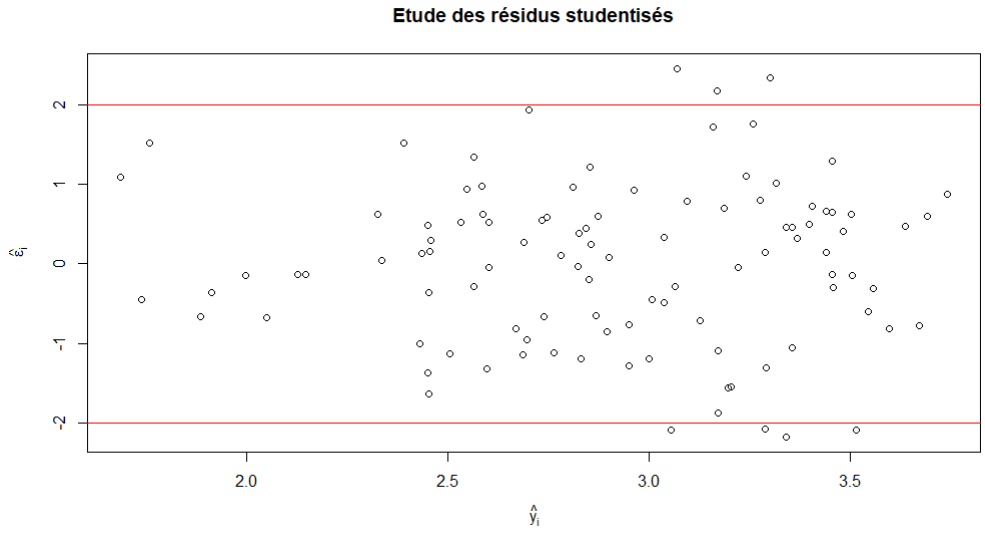

FIGURE 5.5 – Étude des résidus studentisés

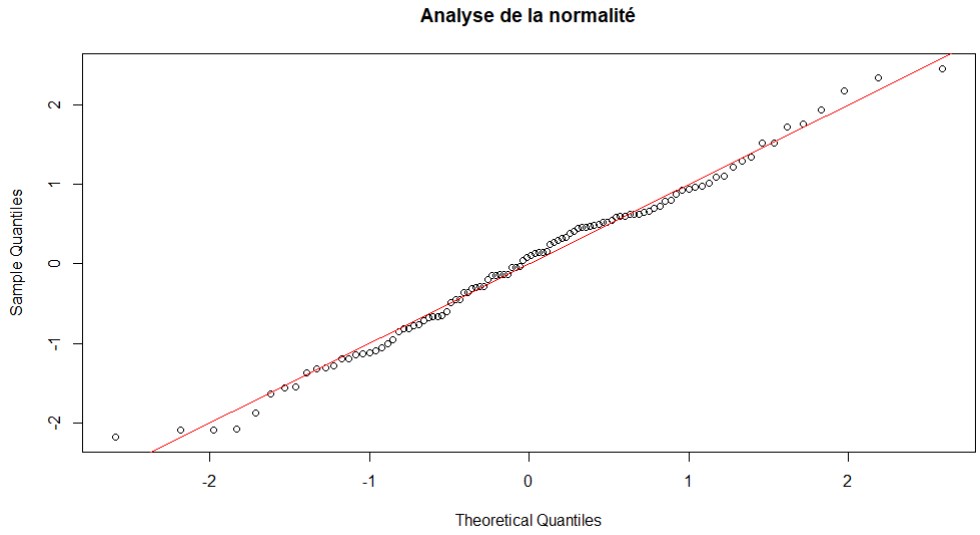

FIGURE 5.6 – Diagramme QQ des résidus studentisés (loi $N(0,1)$)

Le modèle de régression linéaire possède bien sûr des limites, notamment via la forme de la tendance que nous avons supposée (sinusoïdale) : il est probable que la tendance ne suive plus ce modèle au bout de quelques années. Le RPK a tendance à augmenter dans les différents pays du monde, que ce soit en vol intérieur ou international. De nombreux pays cependant tendent à réduire le trafic aérien en proposant des alternatives plus '*propres*' (au sens du développement durable), comme le train par exemple. Ces décisions et leur impact sont difficilement prévisibles.

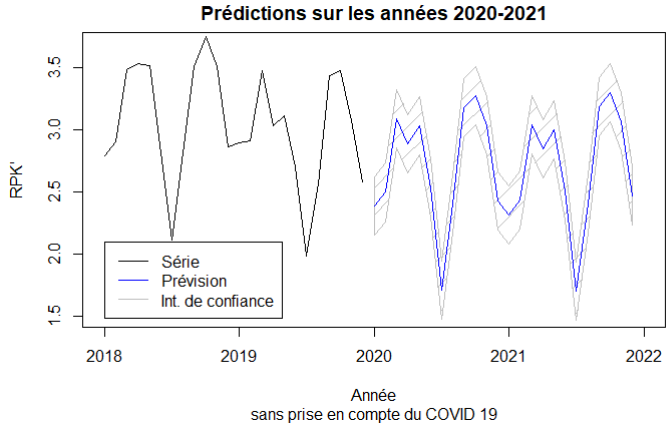

FIGURE 5.7 – Prévisions sur la période 20-21

5.2 Prise en compte de la crise sanitaire COVID-19

L'avantage de cette tendance supposée sinusoïdale, c'est que les années 2020-2021 vont être modélisées avec une tendance en forte décroissance. Cette modélisation prend sensiblement en compte la forte diminution du trafic aérien lors de la crise sanitaire COVID-19. Il est probable cependant que l'impact de la crise sur le RPK ne soit pas suffisamment pris en compte. Comme indiqué en introduction (1.2), la baisse du RPK vol intérieur a été énorme. En faisant l'hypothèse (très

forte) d'une conservation de la saisonnalité (intéressante seulement à titre pédagogique) ; nous allons pouvoir faire une prédiction sur les années 2020 et 2021 en appliquant un coefficient à celles-ci indiqué en 1.2.

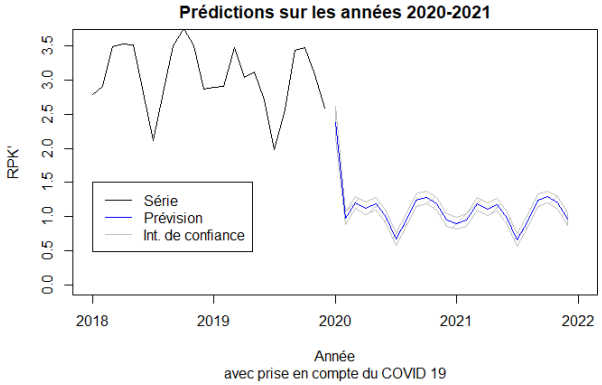

FIGURE 5.8 – Prévisions sur la période 20-21

A été conservée la valeur prédite pour janvier 2020 comme il n'y avait pas beaucoup d'impact de la crise en Europe à ce moment-là. Bien sûr l'impact de la crise sur le RPK de la Suède est très dur à concevoir et mesurer. Cette modélisation n'est qu'un modèle légèrement meilleur que le précédent.

En réajustant le modèle pour ré-obtenir les valeurs du RPK (et non pas le RPK'), notre modèle final est le suivant :

$$\hat{RPK}_t = [\hat{\beta_0} + \hat{\beta_1} \sin(\omega T * (t - t_0)) + \hat{S_t}] * C_t * 10^8$$

Avec C_t Coefficient de prise en compte du COVID19. Valant $1 \forall t \leq$ janvier 2020 et $(1 - 61\%)$ sinon. Les différents coefficients de la formule ci-dessous sont indiqués en Annexe B.2.

L'ensemble de toutes ces prédictions sont disponibles en Annexe E.

Sources

— The Impact of COVID-19 on Flight Networks, *Cornell University*
— Air Passenger Analysis, janvier à octobre 2020, *IATA*
— Summary of industry measures taken by Swedish airports and airline operators due to Covid-19, *Swedish Transport Agency*

MA & Régression linéaire

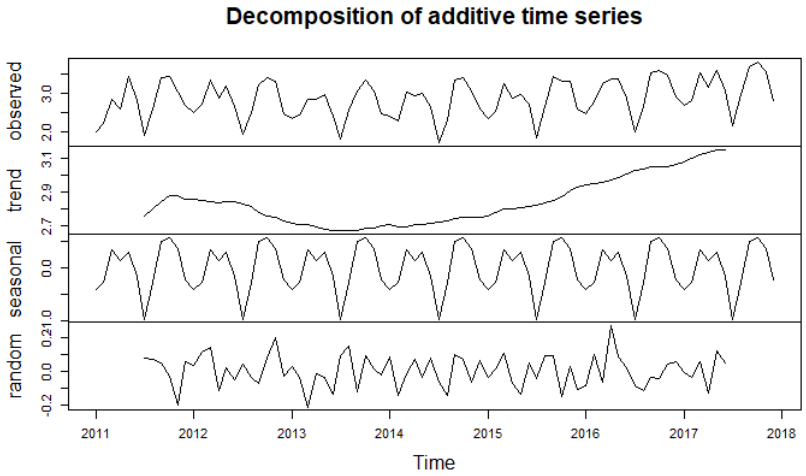

FIGURE B.1 – Décomposition de la série d'apprentissage

```
Call:
tslm(formula = tsApprBis ~ sinus + season)

Residuals:
      Min        1Q    Median        3Q       Max
-0.216580 -0.071733  0.001076  0.063935  0.236561

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.51194    0.04429  56.720  < 2e-16 ***
sinus        0.23032    0.01776  12.971  < 2e-16 ***
season2      0.13873    0.06222   2.230   0.0291 *
season3      0.75354    0.06222  12.111  < 2e-16 ***
season4      0.55288    0.06222   8.886 6.13e-13 ***
season5      0.74539    0.05998  12.426  < 2e-16 ***
season6      0.26207    0.05999   4.369 4.44e-05 ***
season7     -0.58721    0.05999  -9.788 1.52e-14 ***
season8      0.12833    0.06000   2.139   0.0361 *
season9      0.90561    0.06001  15.092  < 2e-16 ***
season10     0.99464    0.06001  16.574  < 2e-16 ***
season11     0.77525    0.06002  12.916  < 2e-16 ***
season12     0.15549    0.06003   2.590   0.0118 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1078 on 67 degrees of freedom
Multiple R-squared:  0.9594,    Adjusted R-squared:  0.9521
F-statistic: 131.8 on 12 and 67 DF,  p-value: < 2.2e-16
```

FIGURE B.2 – Résultats de la régression linéaire sur l'échantillon d'apprentissage tronqué

Lissage de Holt-Winters

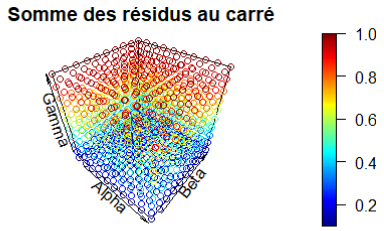

FIGURE C.1 – Représentation graphique de la recherche manuelle des coefficients α, β, γ minimisant les erreurs en validation

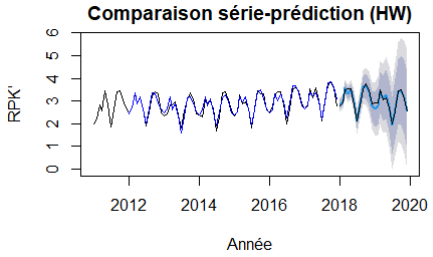

FIGURE C.2 – Résultat des prédictions sur la validation avec les coefficients optimaux

(S)ARIMA

```
Series: tsApprDes
ARIMA(2,1,1)

Coefficients:
          ar1      ar2     ma1
      -1.1251  -0.5893  0.7021
s.e.   0.1542   0.0933  0.1969

sigma^2 estimated as 0.03608:  log likelihood=18.29
AIC=-28.59   AICc=-27.98   BIC=-19.54

Training set error measures:
                      ME      RMSE       MAE      MPE     MAPE      MASE         ACF1
Training set -0.01097023 0.1846063 0.1429927 33.50792 234.1286 0.5772779 -0.03376433
```

FIGURE D.1 – Données de sortie pour la modélisation ARIMA(2,1,1)

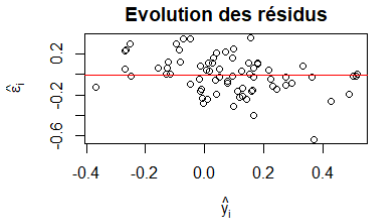

FIGURE D.2 – Représentation des résidus (modèle ARIMA)

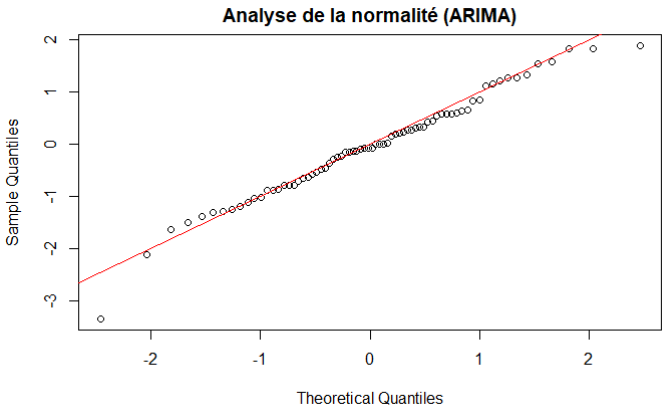

FIGURE D.3 – Diagramme Q-Q pour la loi N(0,1) (ARIMA)

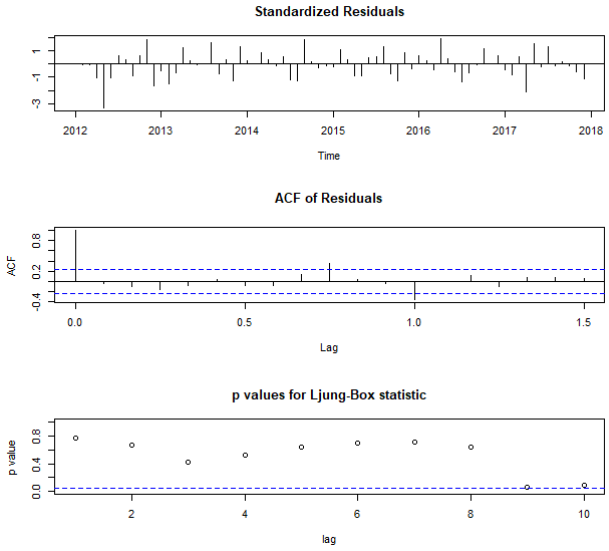

FIGURE D.4 – Résidus et statistique de Ljung-Box (ARIMA)

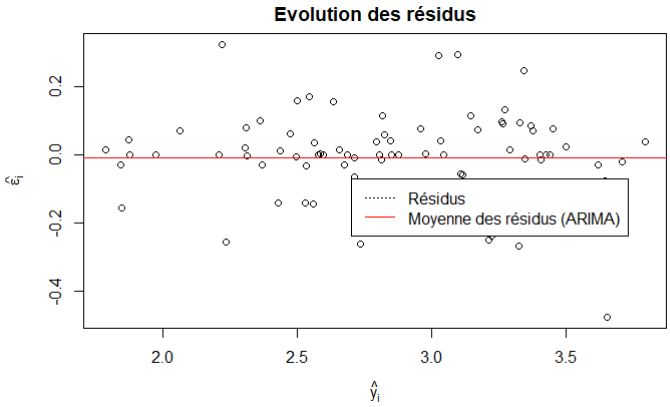

FIGURE D.5 – Représentation des résidus (modèle SARIMA)

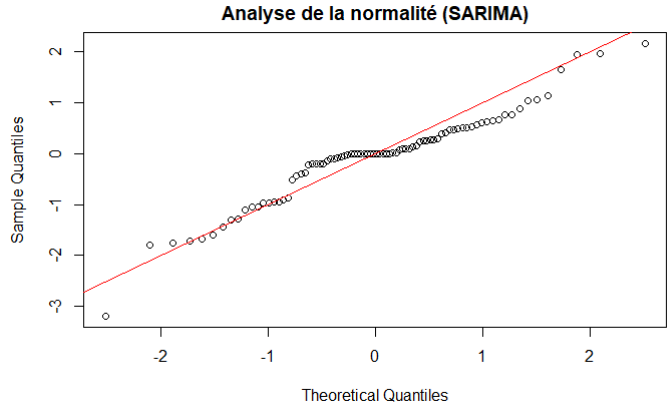

FIGURE D.6 – Diagramme Q-Q pour la loi N(0,1) (SARIMA)

$\hat{Coefficient}$	Valeur	$\hat{\sigma}$
ϕ_1	-0,5100	0,1168
ϕ_2	-0,3597	0,1370
ϕ_3	0,0168	0,1397
ϕ_4	-0,3240	0,1341
ϕ_1^S	-0,7601	0,1462
ϕ_2^S	-0,4544	0,1373

TABLE D.1 – Coefficients estimés du modèle SARIMA(4,1,0)(2,1,0)

Résultats des prévisions

Annee	season	RPK.s.C19.E8	RPK.inf.s.C19.E8	RPK.sup.s.C19.E8	RPK.a.C19.E8	RPK.inf.a.C19.E8	RPK.sup.a.C19.E8
2020	1	2.385	2.1501	2.6199	2.385	2.1501	2.6199
2020	2	2.4967	2.2617	2.7318	0.97373	0.88206	1.0654
2020	3	3.0941	2.8589	3.3293	1.2067	1.115	1.2984
2020	4	2.8871	2.6518	3.1224	1.126	1.0342	1.2177
2020	5	3.0369	2.8029	3.2709	1.1844	1.0931	1.2757
2020	6	2.5318	2.2977	2.766	0.98742	0.8961	1.0787
2020	7	1.7085	1.4742	1.9427	0.6663	0.57495	0.75765
2020	8	2.4052	2.1709	2.6395	0.93802	0.84664	1.0294
2020	9	3.181	2.9466	3.4154	1.2406	1.1492	1.332
2020	10	3.2801	3.0457	3.5146	1.2793	1.1878	1.3707
2020	11	3.0372	2.8027	3.2716	1.1845	1.093	1.2759
2020	12	2.4292	2.1947	2.6637	0.94739	0.85594	1.0388
2021	1	2.3143	2.0784	2.5503	0.90259	0.81057	0.99461
2021	2	2.4357	2.1997	2.6717	0.94992	0.85789	1.0419
2021	3	3.0428	2.8069	3.2788	1.1867	1.0947	1.2787
2021	4	2.8459	2.6099	3.0819	1.1099	1.0179	1.2019
2021	5	3.0058	2.7713	3.2404	1.1723	1.0808	1.2638
2021	6	2.5111	2.2766	2.7456	0.97932	0.88787	1.0708
2021	7	1.6981	1.4637	1.9325	0.66225	0.57083	0.75366
2021	8	2.4052	2.1709	2.6395	0.93802	0.84664	1.0294
2021	9	3.1914	2.9572	3.4256	1.2446	1.1533	1.336
2021	10	3.3009	3.0668	3.535	1.2874	1.1961	1.3786
2021	11	3.0682	2.8342	3.3022	1.1966	1.1054	1.2879
2021	12	2.4704	2.2366	2.7043	0.96347	0.87227	1.0547

Code R

F.1 Chargement

```
library(ggplot2)
library(ggfortify)
library(forecast)
library(car)
library(tseries)
library(latex2exp)
library(plot3D)
theme_update(plot.title = element_text(hjust = 0.5))


# tot pour "total"
dataTot = read.csv2("data.csv")
names(dataTot)[names(dataTot) == "Mois"] = "season"
dataTot$season = as.factor(dataTot$season)
nAnnees = 2019 - 2011 + 1


# S paration des    chantillons

dataTot$RPKprim = dataTot$RPK * 10^-8
tsTot = ts(dataTot$RPKprim, start=c(2011,1), frequency=12)

dataTot$time = as.numeric(time(tsTot))
nAnneesAppr = 7
dataTot$Echantillon = "Apprentissage"
dataTot$Echantillon[12*nAnneesAppr + 1:24] = "Validation"
tsAppr = ts(dataTot[dataTot$Echantillon == "Apprentissage","RPKprim"],
            start=c(2011,1), frequency=12)
tsValid = ts(dataTot[dataTot$Echantillon == "Validation","RPKprim"],
             start=c(2018,1), frequency=12)
```

F.2 Code restant

Régression linéaire

```
dataTotMA = dataTot
```

Série entière

Observation de la décomposition

```
plot(decompose(tsTot))
```

Régression linéaire

Avec la décomposition précédente on peut considérer que la tendance est d'une forme sinusoïdale . On va pouvoir déterminer sa période :

```
t = decompose(tsTot)$trend
t = na.remove(t)
tm = time(t)[t == min(t)]
tM = time(t)[t == max(t)]
T = 2*(tM - tm)
T
t0 = (tm + tM) / 2
t0
remove(t)
```

```
pi = 3.141593

# Renvoi les valeurs d'un sinus déphasé et d'une période définie.
# Entrées : ST, période, déphasage temporel
# Sortie : sinus associé sans les valeurs NA
sinTS = function(trend,T,t0){
  trend = na.remove(trend)
  return(sin(2*pi / T * (as.numeric(time(trend)) - t0)))
}
sinus = sinTS(tsTot,T,t0)


# Avec toutes les saisons :
MA.lm.total = tslm(tsTot~sinus+season)
# En réduisant les saisons :
#MA.lm.total = tslm(tsTot~sinus+dataTotMA$season)
summary(MA.lm.total)
```

```
plot(tsTot)
lines(as.numeric(time(tsTot)),MA.lm.total$fitted.values,col="blue")
```

```
resid = as.numeric(residuals(MA.lm.total))
fitted = as.numeric(fitted(MA.lm.total))
plot(resid~fitted,main="Evolution des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(2.7,-.07,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
dataTotMA$sinus = sinTS(tsTot,T,t0)
dataTotMA$sinTS = predict(MA.lm.total,dataTotMA)
ggplot(data=dataTotMA,aes(x=time(tsTot))) +
  geom_line(aes(y=RPKprim, color="Série temporelle")) +
  geom_line(aes(y = sinTS,color="Tendance")) +
  labs(y="RPK'",x="Temps",color="Legend",
       title="Série totale et tendance estimée")

ind = (1:ncol(dataTotMA))[colnames(dataTotMA) %in% c("sinTS","sinus")]
dataTotMA = dataTotMA[,-ind]
remove(ind, resid, fitted, sinus)
```

Série d'Apprentissage

Calcul des sinus, séparation des échantillons.

```
dataTotMA$sinus = sinTS(tsTot,T,t0)

dataApprMA = dataTotMA[dataTotMA$Echantillon == "Apprentissage",]
dataValidMA = dataTotMA[dataTotMA$Echantillon == "Validation",]
```

Observation de la décomposition

On conserve les valeurs de la périodicité et de la racine de la série totale, qui possède plus de données.

```
plot(decompose(tsAppr))
```

Régression linéaire

```
sinus = dataApprMA$sinus
MA.lm.appr = tslm(tsAppr~sinus+season)
summary(MA.lm.appr)
```

Validation

Des prédictions

```
p = predict(MA.lm.appr,dataValidMA)
plot(tsTot,col="red", main = "Comparaison série-prédiction (MA)",
     xlab="Année",ylab="RPK'",ylim=c(1.5,4.2))
```

```
lines(dataValidMA$time,p,col="blue")

legend(2011,4.2,legend=c("Echantillon apprentissage","Prédictions MA"),
       col=c("red","blue"),lty=1:2)
```

Du modèle linéaire

```
resid = as.numeric(residuals(MA.lm.appr))
fitted = as.numeric(fitted(MA.lm.appr))
plot(resid~fitted,main="Evolution des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(2.7,-.07,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
studres = as.numeric(rstudent(MA.lm.appr))
plot(studres,type='p',
     main="Etude des résidus studentisés",
     xlab="i",ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=2,b=0,col="red")
abline(a=-2,b=0,col="red")
```

Série d'Apprentissage 2 (sans les premiers mois)

```
tsApprBis = ts(dataApprMA$RPKprim[-(1:4)],start=c(2011,5),frequency = 12)
```

```
plot(tsApprBis)
```

Observation de la décomposition

```
plot(decompose(tsApprBis))
```

Régression linéaire

```
sinus = dataApprMA$sinus[-(1:4)]
MA.lm.appr.bis = tslm(tsApprBis~sinus+season)
summary(MA.lm.appr.bis)
```

```
mean(MA.lm.appr.bis$coefficients[-(1:2)])
```

Validation

Des prédictions

```
plot(tsApprBis, main = "Comparaison apprentissage-prédiction (MA)",
     xlab="Année",ylab="RPK'")


lines(as.numeric(time(tsApprBis)),MA.lm.appr.bis$fitted.values,
      col="red",type="b")

legend(2013,3.9,legend=c("Echantillon apprentissage","Prédictions MA"),
       col=c("black","red"),lty=1:2)
```

```
p = predict(MA.lm.appr.bis,dataValidMA)
plot(tsTot,col="red", main = "Comparaison validation-prédiction (MA)",
     xlab="Année",ylab="RPK'",ylim=c(1.5,4.2))

lines(dataValidMA$time,p,col="blue")

legend(2011,4.2,legend=c("Echantillon apprentissage","Prédictions MA"),
       col=c("red","blue"),lty=1:2)
```

```
t = as.numeric(time(tsValid))
MAresValid = as.numeric(tsValid) - p
plot(t,MAresValid,xlab="Année",ylab="Résidus de validation",
     main="Observation des résidus (échantillon de validation)",
     sub = "Méthode MA")
abline(h=mean(MAresValid),col="red",lty="dotted")
abline(h=0,col="blue")
legend(2019.5, .2,legend=c("Résidus","Moyenne","y = 0"),fill=c("black","red","blue"))
remove(t)
```

```
summary(MAresValid)
summary(abs(MAresValid))
boxplot(abs(MAresValid),main="Répartition des résidus (valeur absolue) (MA)")
```

Du modèle linéaire

```
resid = as.numeric(residuals(MA.lm.appr.bis))
fitted = as.numeric(fitted(MA.lm.appr.bis))
plot(resid~fitted,main="Etude des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(1.68,-.1,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
studres = as.numeric(rstudent(MA.lm.appr.bis))
plot(studres~as.numeric(MA.lm.appr.bis$fitted.values),type='p',
     main="Etude des résidus studentisés",
     xlab=TeX('$\\hat{y _i}$'),ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=2,b=0,col="red")
abline(a=-2,b=0,col="red")
```

```r
qqnorm(studres,main="Analyse de la normalité")
abline(a=0,b=1,col="red")
```

```r
remove(dataApprMA, dataValidMA)
```

Série entière (sans les premiers mois)

```r
tsTotBis = ts(dataTotMA$RPKprim[-(1:4)],start=c(2011,5),frequency = 12)
```

Observation de la décomposition

```r
plot(decompose(tsTotBis))
```

Régression linéaire

```r
sinus = dataTotMA$sinus[-(1:4)]
MA.lm.total = tslm(tsTotBis~sinus+season)
summary(MA.lm.total)
```

```r
mean(MA.lm.total$coefficients[-(1:2)])
```

Validation

Des prédictions

```r
plot(tsApprBis, main = "Comparaison entrainement-prédiction (MA)",
     xlab="Année",ylab="RPK'")


lines(as.numeric(time(tsTotBis)),MA.lm.total$fitted.values,
      col="red",type="b")

legend(2012.5,3.9,legend=c("Série temporelle (sans mois 1 à 4)","Prédictions MA"),
       col=c("black","red"),lty=1:2)
```

Du modèle linéaire

```r
resid = as.numeric(residuals(MA.lm.total))
fitted = as.numeric(fitted(MA.lm.total))
plot(resid~fitted,main="Etude des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(1.68,-.1,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
studres = as.numeric(rstudent(MA.lm.total))
plot(studres~as.numeric(MA.lm.total$fitted.values),type='p',
     main="Etude des résidus studentisés",
     xlab=TeX('$\\hat{y _i}$'),ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=2,b=0,col="red")
abline(a=-2,b=0,col="red")
```

```
qqnorm(studres,main="Analyse de la normalité")
abline(a=0,b=1,col="red")
```

Fin

```
MAres = MAresValid
MAmodel = MA.lm.appr.bis
MAtrueModel = MA.lm.total
remove(dataApprMA,
       dataTotMA,
       dataValidMA,
       MA.lm.appr,
       MA.lm.appr.bis,
       MA.lm.total,
       tsApprBis,
       tsTotBis)

remove(p, sinus, fitted)

remove(studres, resid, MAresValid)
```

Régression linéaire

```
dataTotMA = dataTot
```

Série entière

Observation de la décomposition

```
plot(decompose(tsTot))
```

Régression linéaire

Avec la décomposition précédente on peut considérer que la tendance est d'une forme sinusoïdale . On va pouvoir déterminer sa période :

```
t = decompose(tsTot)$trend
t = na.remove(t)
tm = time(t)[t == min(t)]
tM = time(t)[t == max(t)]
T = 2*(tM - tm)
T
t0 = (tm + tM) / 2
t0
remove(t)
```

```
pi = 3.141593

# Renvoi les valeurs d'un sinus déphasé et d'une période définie.
# Entrées : ST, période, déphasage temporel
# Sortie : sinus associé sans les valeurs NA
sinTS = function(trend,T,t0){
  trend = na.remove(trend)
  return(sin(2*pi / T * (as.numeric(time(trend)) - t0)))
}
sinus = sinTS(tsTot,T,t0)


# Avec toutes les saisons :
MA.lm.total = tslm(tsTot~sinus+season)
# En réduisant les saisons :
#MA.lm.total = tslm(tsTot~sinus+dataTotMA$season)
summary(MA.lm.total)
```

```
plot(tsTot)
lines(as.numeric(time(tsTot)),MA.lm.total$fitted.values,col="blue")
```

```
resid = as.numeric(residuals(MA.lm.total))
fitted = as.numeric(fitted(MA.lm.total))
plot(resid~fitted,main="Evolution des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(2.7,-.07,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
dataTotMA$sinus = sinTS(tsTot,T,t0)
dataTotMA$sinTS = predict(MA.lm.total,dataTotMA)
ggplot(data=dataTotMA,aes(x=time(tsTot))) +
  geom_line(aes(y=RPKprim, color="Série temporelle")) +
  geom_line(aes(y = sinTS,color="Tendance")) +
  labs(y="RPK'",x="Temps",color="Legend",
       title="Série totale et tendance estimée")

ind = (1:ncol(dataTotMA))[colnames(dataTotMA) %in% c("sinTS","sinus")]
dataTotMA = dataTotMA[,-ind]
remove(ind, resid, fitted, sinus)
```

Série d'Apprentissage

Calcul des sinus, séparation des échantillons.

```
dataTotMA$sinus = sinTS(tsTot,T,t0)

dataApprMA = dataTotMA[dataTotMA$Echantillon == "Apprentissage",]
dataValidMA = dataTotMA[dataTotMA$Echantillon == "Validation",]
```

Observation de la décomposition

On conserve les valeurs de la périodicité et de la racine de la série totale, qui possède plus de données.

```
plot(decompose(tsAppr))
```

Régression linéaire

```
sinus = dataApprMA$sinus
MA.lm.appr = tslm(tsAppr~sinus+season)
summary(MA.lm.appr)
```

Validation

Des prédictions

```
p = predict(MA.lm.appr,dataValidMA)
plot(tsTot,col="red", main = "Comparaison série-prédiction (MA)",
     xlab="Année",ylab="RPK'",ylim=c(1.5,4.2))
```

```
lines(dataValidMA$time,p,col="blue")

legend(2011,4.2,legend=c("Echantillon apprentissage","Prédictions MA"),
       col=c("red","blue"),lty=1:2)
```

Du modèle linéaire

```
resid = as.numeric(residuals(MA.lm.appr))
fitted = as.numeric(fitted(MA.lm.appr))
plot(resid~fitted,main="Evolution des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(2.7,-.07,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
studres = as.numeric(rstudent(MA.lm.appr))
plot(studres,type='p',
     main="Etude des résidus studentisés",
     xlab="i",ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=2,b=0,col="red")
abline(a=-2,b=0,col="red")
```

Série d'Apprentissage 2 (sans les premiers mois)

```
tsApprBis = ts(dataApprMA$RPKprim[-(1:4)],start=c(2011,5),frequency = 12)
```

```
plot(tsApprBis)
```

Observation de la décomposition

```
plot(decompose(tsApprBis))
```

Régression linéaire

```
sinus = dataApprMA$sinus[-(1:4)]
MA.lm.appr.bis = tslm(tsApprBis~sinus+season)
summary(MA.lm.appr.bis)
```

```
mean(MA.lm.appr.bis$coefficients[-(1:2)])
```

Validation

Des prédictions

```
plot(tsApprBis, main = "Comparaison apprentissage-prédiction (MA)",
     xlab="Année",ylab="RPK'")


lines(as.numeric(time(tsApprBis)),MA.lm.appr.bis$fitted.values,
      col="red",type="b")

legend(2013,3.9,legend=c("Echantillon apprentissage","Prédictions MA"),
       col=c("black","red"),lty=1:2)
```

```
p = predict(MA.lm.appr.bis,dataValidMA)
plot(tsTot,col="red", main = "Comparaison validation-prédiction (MA)",
     xlab="Année",ylab="RPK'",ylim=c(1.5,4.2))

lines(dataValidMA$time,p,col="blue")

legend(2011,4.2,legend=c("Echantillon apprentissage","Prédictions MA"),
       col=c("red","blue"),lty=1:2)
```

```
t = as.numeric(time(tsValid))
MAresValid = as.numeric(tsValid) - p
plot(t,MAresValid,xlab="Année",ylab="Résidus de validation",
     main="Observation des résidus (échantillon de validation)",
     sub = "Méthode MA")
abline(h=mean(MAresValid),col="red",lty="dotted")
abline(h=0,col="blue")
legend(2019.5, .2,legend=c("Résidus","Moyenne","y = 0"),fill=c("black","red","blue"))
remove(t)
```

```
summary(MAresValid)
summary(abs(MAresValid))
boxplot(abs(MAresValid),main="Répartition des résidus (valeur absolue) (MA)")
```

Du modèle linéaire

```
resid = as.numeric(residuals(MA.lm.appr.bis))
fitted = as.numeric(fitted(MA.lm.appr.bis))
plot(resid~fitted,main="Etude des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'),type='p')
abline(a=mean(resid),b=0,col="red")
legend(1.68,-.1,legend=c("Résidus","Moyenne des résidus"),col=c("black","red"),
       lty=c(3,1))
```

```
studres = as.numeric(rstudent(MA.lm.appr.bis))
plot(studres~as.numeric(MA.lm.appr.bis$fitted.values),type='p',
     main="Etude des résidus studentisés",
     xlab=TeX('$\\hat{y _i}$'),ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=2,b=0,col="red")
abline(a=-2,b=0,col="red")
```

```r
qqnorm(studres,main="Analyse de la normalité")
abline(a=0,b=1,col="red")
```

Fin

```r
MAres = MAresValid
MAmodel = MA.lm.appr.bis
remove(dataApprMA,
       dataTotMA,
       dataValidMA,
       MA.lm.appr,
       MA.lm.appr.bis,
       MA.lm.total,
       tsApprBis)

remove(p, sinus, fitted)

remove(studres, resid, MAresValid)
```

Holt-Winters

Apprentissage

Calcul automatique des coefficients

```
hw = HoltWinters(tsAppr)
hw
```

Validation

```
pred = forecast(hw,24)
t = as.numeric(time(hw$fitted))
plot(pred, main="Comparaison série-prédiction (HW)",
      xlab="Année",ylab="RPK'")
lines(t,(hw$fitted)[,1],col="blue")
lines(tsValid)
legend(2011,4.5,legend=c("Série","Prédictions"),col=c("black","blue"),lty=1)
```

```
HWres =  as.numeric(tsValid) - as.numeric(pred$mean)
```

```
sqRes = as.numeric(tsValid) - as.numeric(pred$mean)
sqRes = sum(sqRes^2)
sqRes
```

Calcul manuel des coefficients

```
alpha  = rep(1:10,each=100) / 10
beta = rep(1:10,each=10) / 10
beta = rep(beta,10)
gamma = rep(1:10,100) / 10
sqRes = rep(Inf,1000)
for (i in 1:1000)
{
  f = forecast(HoltWinters(tsAppr,alpha[i],beta[i],gamma[i],
               seasonal="additive"),h=24)
  res = as.numeric(tsValid) - as.numeric(f$mean)
  sqRes[i] = sum(res^2)
}
min(sqRes)
remove(i,f,res)
```

```
scatter3D(alpha,beta,gamma,sqRes,
          main = "Somme des résidus au carré", xlab="Alpha",
          ylab="Beta", zlab="Gamma")
```

```
ind = (1:1000)[sqRes == min(sqRes)]
alphaMin = alpha[ind]
alphaMin
betaMin = beta[ind]
betaMin
gammaMin = gamma[ind]
gammaMin
min(sqRes)
remove(alpha,beta,gamma)
```

```
alpha = alphaMin + (-4:5) / 100
beta = betaMin + (-4:5) / 100
gamma = gammaMin + (-4:5) / 100
alpha = rep(alpha,each=100)
beta = rep(beta,each=10)
beta = rep(beta,10)
gamma = rep(gamma,100)

sqRes = rep(Inf,1000)
for (i in 1:1000)
{
  f = forecast(HoltWinters(tsAppr,alpha[i],beta[i],gamma[i]),
               seasonal="additive",h=24)
  res = as.numeric(tsValid) - as.numeric(f$mean)
  sqRes[i] = sum(res^2)
}
min(sqRes)
remove(i,f,res)
```

```
scatter3D(alpha,beta,gamma,sqRes)
```

```
ind = (1:1000)[sqRes == min(sqRes)]
alphaMin = alpha[ind]
alphaMin
betaMin = beta[ind]
betaMin
gammaMin = gamma[ind]
gammaMin
min(sqRes)
remove(alpha,beta,gamma,ind)
```

```
hwBis = HoltWinters(tsAppr,alphaMin,betaMin,gammaMin)
hwBis
remove(alphaMin,betaMin,gammaMin)
```

```
sqRes = as.numeric(tsValid) - as.numeric(forecast(hwBis)$mean)
sqRes = sum(sqRes^2)
sqRes
```

```
plot(forecast(hwBis), main="Comparaison série-prédiction (HW)",
      xlab="Année",ylab="RPK'")
```

```
lines(tsValid)
lines(t,(hw$fitted)[,1],col="blue")
```

```
HWresValid =  as.numeric(tsValid) - as.numeric(pred$mean)

t = as.numeric(time(tsValid))
plot(t,HWresValid,xlab="Année",ylab="Résidus de validation",
     main="Observation des résidus (échantillon de validation)",
     sub = "Méthode Holt-Winters")
abline(h=mean(HWresValid),col="red",lty="dotted")
abline(h=0,col="blue")
legend(2018, -.5,legend=c("Résidus","Moyenne","y = 0"),fill=c("black","red","blue"))
```

```
HWModel = hw
remove(t,hw,hwBis,pred,fitted)
remove(sqRes)
```

SARIMA

Etude préliminaire

```
acf(tsAppr)
pacf(tsAppr)
```

Différenciation

Calculs

```
tsApprDes = diff(tsAppr,lag=12)
tsValidDes = diff(tsValid,lag=12)
```

```
plot(tsApprDes,main="Série d'apprentissage désaisonnalisée",
     xlab="Année",ylab=TeX("$\\Delta{RPK'}$"))
lm = tslm(tsApprDes~trend)
lines(as.numeric(time(tsApprDes)),lm$fitted.values,col="red")
legend(2015,-.3,legend=c("Série désaisonnalisée","Tendance linéaire (MCO)"),lty=1,col=c("black","red"))
remove(lm)
```

```
acf(tsApprDes)
pacf(tsApprDes)
```

Test de Dickey-Fuller (désaisonnalisation)

```
t = as.numeric(time(tsApprDes))
data = as.numeric(tsApprDes)
n = length(t)
yP = data[-1]
yM = data[-n]
yDelta = yP - yM
model = lm(yDelta ~ t[-1] + yM)
summary(model)
remove(model,t,n,data,yDelta,yP,yM)
```

b != 0 phi - 1 != 0 : Série TS

Test de Dickey-Fuller (désaisonnalisation + détendancialisation)

```
tsApprDesDet = diff(tsApprDes)
```

```
par(mfrow=c(1,2))
acf(tsApprDesDet)
pacf(tsApprDesDet)
```

```
plot(tsApprDesDet, main="Série d'apprentissage désaisonnalisée et détendiancialisée",
     xlab="Année",ylab=TeX("$RPK'_t ^{barre}$"))
```

```
t = as.numeric(time(tsApprDesDet))
data = as.numeric(tsApprDesDet)
n = length(t)
yP = data[-1]
yM = data[-n]
yDelta = yP - yM
model = lm(yDelta ~ t[-1] + yM)
summary(model)
remove(model,n,data)
```

b = 0

```
model = lm(yDelta ~ yM)
summary(model)
```

c = 0

```
model = lm(yDelta ~ yM -1)
summary(model)
```

pih - 1 != 0 : série stationnaire

ARIMA avec d > 0

Création

```
ARIMAauto = auto.arima(tsApprDes,seasonal=F,ic="aic",d=1)
```

```
summary(ARIMAauto)
```

```
plot(forecast(ARIMAauto),main="Comparaison validation-prédiction (ARIMA)",
     xlab="Année",ylab=TeX("$\\tilde{RPK_'}$"))
lines(diff(tsTot,12))
```

```
ARIMA1 = arima(tsApprDes,order=c(2,0,2))
summary(ARIMA1)
```

```
plot(forecast(ARIMA1))
```

Validation

```
resid = as.numeric(residuals(ARIMAauto))
fitted = as.numeric(fitted(ARIMAauto))
plot(resid~fitted,main="Evolution des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=mean(resid),b=0,col="red")
```

```
legend(2.7,-.07,legend=c("Résidus","Moyenne des résidus (ARIMA)"),col=c("black","red"),
       lty=c(3,1))
```

```
hist(resid,main="Histogramme des résidus (ARIMA)",
     xlab="Résidus",ylab="Fréquence")
```

```
sd = sqrt(ARIMAauto$sigma2)
qqnorm(resid / sd,main="Analyse de la normalité (ARIMA)")
abline(a=0,b=1,col="red")
```

```
tsdiag(ARIMAauto)
```

SARIMA

Création

```
SARIMAauto = auto.arima(tsAppr)
```

```
SARIMAauto
```

```
plot(forecast(SARIMAauto),main="Comparaison validation-prédiction (SARIMA)",
     xlab="Année",ylab=TeX('$\\hat{\\epsilon_i}$'))
lines(tsTot)
```

Validation

```
resid = as.numeric(residuals(SARIMAauto))
fitted = as.numeric(fitted(SARIMAauto))
plot(resid~fitted,main="Evolution des résidus",xlab=TeX('$\\hat{y _i}$'),
     ylab=TeX('$\\hat{\\epsilon_i}$'))
abline(a=mean(resid),b=0,col="red")
legend(2.7,-.07,legend=c("Résidus","Moyenne des résidus (ARIMA)"),col=c("black","red"),
       lty=c(3,1))
```

```
hist(resid,main="Histogramme des résidus (SARIMA)",
     xlab="Résidus",ylab="Fréquence")
```

```
sd = sqrt(SARIMAauto$sigma2)
qqnorm(resid / sd,main="Analyse de la normalité (SARIMA)")
abline(a=0,b=1,col="red")
```

```
ks.test(resid / sd,"pnorm")
```

```
tsdiag(SARIMAauto)
```

```
SARIMAModel = SARIMAauto
SARIMAres = as.numeric(tsValid - forecast(SARIMAauto)$mean)
```

```
remove(ARIMAauto,tsApprDes,tsApprDesDet,tsValidDes,resid,sd,fitted)
```

Synthèse

Tangi Tassin

08/01/2021

Comparaison

```
colors = c("lightgray","darkblue","darkgreen")
noms = c("Rég. lin.","H.-W.","SARIMA")
```

```
MAres.abs = abs(MAres)
HWres.abs = abs(HWres)
SARIMAres.abs = abs(SARIMAres)
```

```
MAres.abs = abs(MAres)
plot(HWres.abs,col=colors[2])
points(MAres.abs,col=colors[1])
points(SARIMAres.abs,col=colors[3])
```

```
t = as.numeric(time(tsValid))
plot(tsValid,ylim=c(2,4))
lines(t, as.numeric(tsValid - MAres),col=colors[1])
lines(t, as.numeric(tsValid - HWres),col=colors[2])
lines(t, as.numeric(tsValid - SARIMAres), col=colors[3])
```

```
par(mfrow=c(1,2))
boxplot(MAres.abs,HWres.abs,SARIMAres.abs,col=colors,
        names=noms, main="Dispersion des résidus (absolue)",
        ylab=TeX("$|\\epsilon_i|$"))

barplot(c(sum(MAres^2),sum(HWres^2),sum(SARIMAres^2)),
        col=colors,main="Somme des résidus au carré",names.arg=noms)
```

Prédiction

```
dataPred = data.frame(Annee=rep(2020:2021,each=12), season = rep(1:12,2),
                      time=c(2020 + 0:11 / 12, 2021 + 0:11 / 12))
tsPred = ts(rep(1,24),start=c(2021,1),frequency=12)
dataPred$season = as.factor(dataPred$season)
dataPred$sinus = sinTS(tsPred,T,t0)


predict = predict(MAtrueModel,dataPred,interval="prediction")
colnames(predict) <- c("RPK.s.C19.E8","RPK.inf.s.C19.E8", "RPK.sup.s.C19.E8")
dataPred = cbind(dataPred,signif(predict,5))
```

```r
plot(tsValid,xlim=c(2018,2022),ylim=c(1.5,3.7),
     main = "Prédictions sur les années 2020-2021",
    xlab="Année",ylab="RPK'",sub="sans prise en compte du COVID 19")

polygon(x = c(dataPred$time,rev(dataPred$time)),c(predict[,2],rev(predict[,3])),
        density=5,col="grey")
lines(dataPred$time,predict[,1],col="blue")

legend(2018,2.1,legend=c("Série","Prévision","Int. de confiance"),
       col=c("black","blue","grey"),lty=1)
```

Prédiction avec prise en compte du COVID

```r
plot(tsValid,xlim=c(2018,2022),ylim=c(0,3.6),
     main = "Prédictions sur les années 2020-2021",
     xlab="Année",ylab="RPK'",sub="avec prise en compte du COVID 19")

predict2 = predict
colnames(predict2) <- c("RPK.a.C19.E8","RPK.inf.a.C19.E8", "RPK.sup.a.C19.E8")
predict2[-1,] = (1-.61)*predict[-1,]
dataPred = cbind(dataPred, signif(predict2,5))
polygon(x = c(dataPred$time,rev(dataPred$time)),
        c(predict2[,2],rev(predict2[,3])),density=5,col="grey")
lines(dataPred$time,predict2[,1],col="blue")

legend(2018,1.5,legend=c("Série","Prévision","Int. de confiance"),
       col=c("black","blue","grey"),lty=1)
```

```r
write.csv(dataPred[-c(3,4)],"predictions.csv",row.names=F,quote=F)
```

INSA Rennes
20 Avenue des Buttes de Coësmes
CS 70839
35708 Rennes Cedex 7

Tél. +33 (0) 2 23 23 82 00
Fax +33 (0) 2 23 23 83 96

www.insa-rennes.fr

INSA

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR,
DE LA RECHERCHE
ET DE L'INNOVATION
Liberté
Égalité
Fraternité