# Projet de séries temporelles

Jean-Bapiste Dreveton
Alexandre Filiot

On s'intéresse à la modélisation et la prévision de l'indice de production industrielle observé en France. Il ne vous est pas demandé de connaître le processus de construction de l'indicateur. Vous ne travaillez que sur les données observées. A partir du répertoire des séries chronologiques de l'INSEE, `www.insee.fr/fr/statistiques?debut=0&categorie=10` , vous devez choisir une série agrégée brute, mensuelle, correspondant à n'importe quel secteur d'activité (à votre convenance) contenant au moins 25 années et contenant le mois de janvier 2018.

Date limite : 6 Juin 2018

# Partie 1 : Les données

## 1. Que représente la série choisie ? (secteur, périmètre, traitements éventuels, transformation logarithmique, ...)

Dans ce projet, nous nous intéresserons au secteur du transport, et plus précisement à la fréquentation de passagers lors des vols internationaux au départ de Paris. Les données sont mensuelles et s'étendent de Janvier 1994 à Janvier 2018. Ces données sont issues de l'Insee [1]. La série originale est affichée en page 3, graphique 1.

## 2&3. Transformer si besoin la série pour la rendre stationnaire (désaisonnalisation, différentiation, suppression de la tendance déterministe, ...). Justifier soigneusement vos choix. Représenter graphiquement la série choisie avant et après transformation.

Tout d'abord, il est souvent d'usage pour ce type de série d'opérer au préalable une transformation logarithmique. Ceci permet de réduire l'hétéroscédasticité des résidus.

Les touristes prenant leurs vacances à peu près au même moment de l'année, il en résulte que notre série présente une saisonnalité à l'ordre 12. Nous appliquons alors la transformation suivante :

$$\mathbf{Y_t} = (1 - B^{12})\mathbf{logX_t}$$

où $(\mathbf{X_t})_{12 \leq t \leq T}$ est la série initiale, et $B$ l'opérateur backward.

Suite à l'arrivée de compagnies low-cost sur le secteur du transport aéronautique, et la construction de nouveaux aéroports, notamment dans les pays émergents, il y a eu une démocratisation d'accès au transport aérien. Cela se traduit sur notre série par une tendance haussière. La série a donc été différenciée à l'ordre 1 pour supprimer cette tendance déterministe. La série finale est alors :

$$\mathbf{Z_t} = (1 - B)\mathbf{Y_t} = (1 - B) \circ (1 - B^{12})\mathbf{logX_t}$$

L'objectif de cette partie étant de stationnariser la série initiale, justifions nos transformations. Ces justifications sont également disponibles en commentaires du code. Étant donné que notre série présente une tendance croissante sur l'ensemble de la période observée, nous nous plaçons dans le cas avec constante non nulle et avec tendance du test de Dickey-Fuller augmenté (ADF). Nous utilisons en premier lieu ce test pour rejeter ou non la stationnarité de nos différentes séries. L'estimation du modèle ADF, pour une série quelconque $(\mathbf{X_t})$ est donnée par :

$$\mathbf{\Delta X_t} = c + at + \phi\mathbf{X_t} + \sum_{k=1}^{h} \alpha_k \mathbf{\Delta X_{t-k}} + \mathbf{u_t}$$

avec $h$ le lag maximum utilisé dans l'estimation. Systématiquement nous avons relevé la valeur de la statistique de test et l'avons comparée aux valeurs critiques associées aux tests à 1%, 5% et 10%.

### Tests sur la série originale

Commençons par notre série originale. Lorsque $h = 0$, nous trouvons une statistique de test de $-6,3258$ soit une $p$-valeur faible. En estimant le modèle simple :

$$\mathbf{\Delta logX_t} = c + at + \phi\mathbf{logX_t} + \mathbf{u_t}$$

nous rejetons donc aux seuils usuels l'hypothèse de racine unitaire. Quel est alors l'intérêt d'appliquer une désaisonnalisation et/ou une différenciation ? Nous allons y venir. Quand bien même l'hypothèse de non-stationnarité n'est pas rejetée à 0,1%, il faut néanmoins tester la validité du modèle en terme de persistance des résidus. Pour cela, nous utilisons un test de Ljung-Box avec un lag maximal égal à 24. Pour notre série $\mathbf{logX_t}$, il s'avère que le test de blancheur des résidus est rejeté à 5% à tous les ordres : les résidus du modèle sont donc corrélés de l'ordre 1 à l'ordre 24 , le modèle n'est donc pas valide. L'idée est donc d'augmenter le paramètre $h$ dans l'estimation du modèle ADF afin de relever le lag minimal permettant de ne pas rejeter la corrélation des résidus de

1. `https://insee.fr/fr/statistiques/serie/010001981`

l'ordre 1 à l'ordre 24 au seuil de 5%. Cette valeur est de $h_{min} = 23$ pour la série $\mathbf{logX_t}$, ce qui est beaucoup. Nous aimerions donc pouvoir diminuer cette valeur de $h_{min}$ à l'aide d'une première transformation, la désaisonnalisation.

**Tests sur la série désaisonnalisée**

La $p$-valeur du test ADF pour un lag $h = 0$ appliqué à la série désaisonnalisée $\mathbf{Y_t}$ est très faible, on rejette donc l'hypothèse de racine unité à 5% (*df*-statistique : -8,6601). Le test de Ljung-Box ne rejette pas à 5%, de peu, l'hypothèse de corrélation des résidus à l'ordre 1. En effet, la $p$-valeur est égale à 0,0542. Enfin, le lag minimal à partir duquel l'hypothèse de corrélation des résidus est rejetée à 5% jusqu'à l'ordre 24 est de 13, ce qui est nettement inférieur à 23 pour la série $\mathbf{logX_t}$.

**Tests sur la série désaisonnalisée et différenciée**

La $p$-valeur du test ADF pour un lag $h = 0$ appliqué à la série différenciée et désaisonnalisée $\mathbf{Z_t}$ est encore plus faible, on rejette donc logiquement l'hypothèse de racine unité à 5% (*df*-statistique : -24,697). Le test de Ljung-Box ne rejette pas à 5% l'hypothèse de corrélation des résidus à l'ordre 1. En effet, la $p$-valeur est égale à 0,3952. Enfin, le lag minimal à partir duquel l'hypothèse de corrélation des résidus est rejetée à 5% jusqu'à l'ordre 24 est lui aussi égal à 13.

**Comment choisir ?**

On remarque que la valeur de $h_{min}$ est la même pour la série désaisonnalisée $\mathbf{Y_t}$ que pour celle désaisonnalisée et différenciée $\mathbf{Z_t}$. Pour cette valeur de $h$, la $p$-valeur du test ADF sur $\mathbf{Y_t}$ est de 0,021, tandis que celle du test sur la série $\mathbf{Z_t}$ est inférieure à 0,01. Ainsi, les tests ADF rejettent (heureusement) l'hypothèse de racine unité sur les deux transformations. Il en est de même pour le test de Phillips-Perron (*df*-statistique respectives de -327,68 et -129,24 pour les séries $\mathbf{Z_t}$ et $\mathbf{Y_t}$) où les $p$-valeurs sont plus petites que 0,01. Enfin, le test KPSS ne rejette pas l'hypothèse de stationnarité à 5% puisque les $p$-valeurs sont toutes deux égales à 0,1. Systématiquement, nous remarquons que le rejet des hypothèses de non-stationnarité (ou le non-rejet de celles de stationnarité) est plus fort, en terme de $p$-valeurs, pour la série $\mathbf{Z_t}$. De plus, nous l'avons dit, l'hypothèse de corrélation des résidus à l'ordre 1 est "plus" rejetée à 5% pour cette série que pour celle désaisonnalisée $\mathbf{Y_t}$. Nous aurions pu nous contenter d'utiliser la série désaisonnalisée uniquement pour la suite, mais nous estimons que notre modèle est plus robuste lorsqu'il prend en compte la série $\mathbf{Z_t}$ en terme de stationnarité et de blancheur des résidus. Le calibrage du modèle ARMA aurait sans doute était sensiblement le même.

Enfin, nous n'aurions cependant pas pu utiliser la série $\mathbf{logX_t}$ initiale dans la mesure où le test ADF ne rejette pas l'hypothèse de racine unité à 5% ($p$-valeur égale à 0,553) lorsque $h = h_{min} = 23$.

# Partie 2 : Modèles ARMA

## 4. Choisir, en le justifiant, un modèle ARMA($p$,$q$) (avec éventuellement une composante saisonnière) pour votre série corrigée $\mathbf{Z_t}$. Estimer les paramètres du modèle et vérifier sa validité.

Comme nous l'avons vu dans la partie précédente, notre série présente une période d'ordre 12 et une tendance haussière déterministe d'ordre 1. Ainsi,

$$\mathbf{logX_t} \sim SARIMA(p, d = 1, q)(P, D = 1, Q)_{s=12}$$

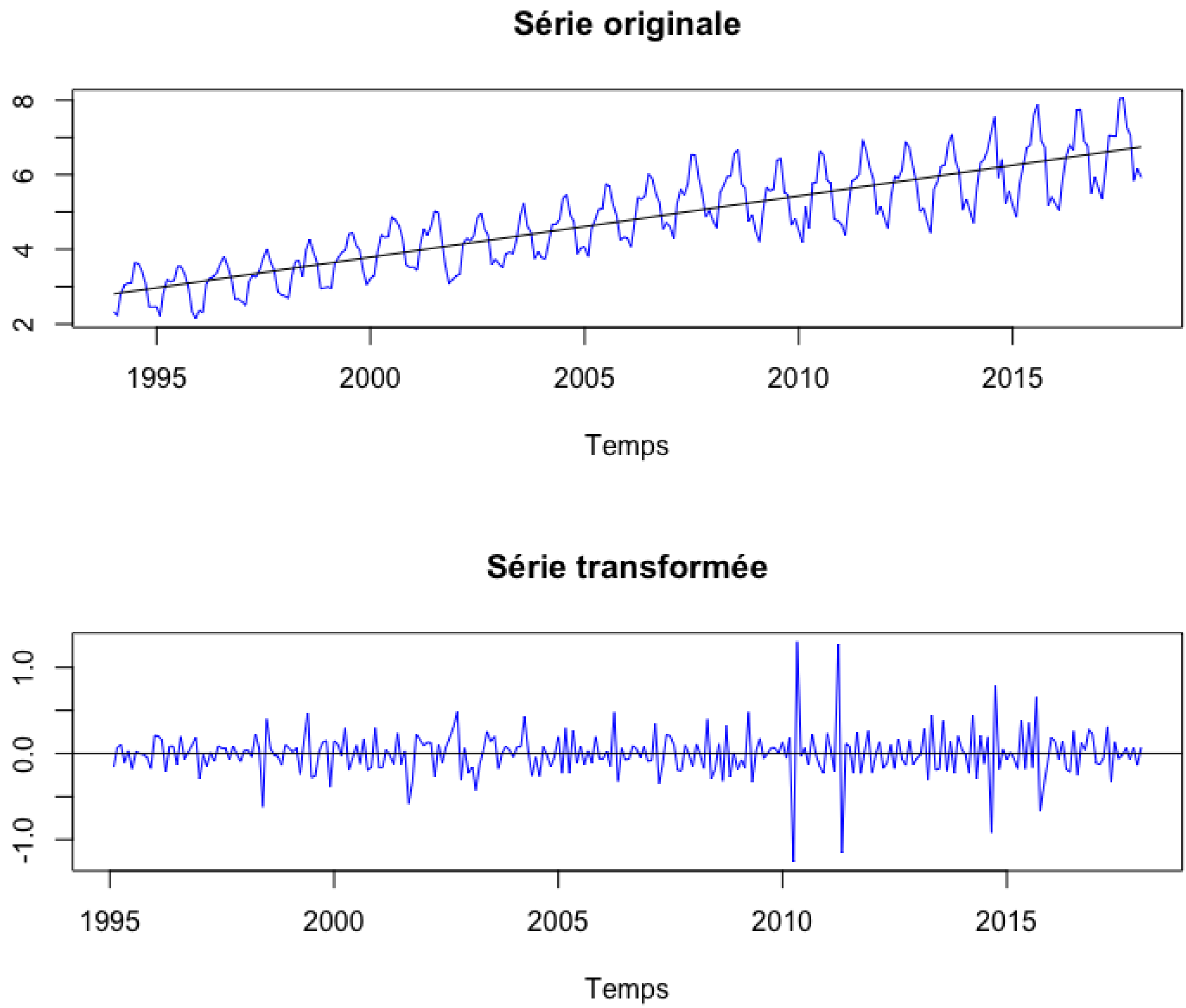
$$\mathbf{Z_t} \sim SARIMA(p, d = 0, q)(P, D = 0, Q)_{s=12}$$

Nous suspectons donc la série $\mathbf{logX_t}$ de satisfaire l'équation :

$$(1 - B)^d (1 - B^s)^D \phi(B)\Phi(B^s)\mathbf{logX_t} = \theta(B)\Theta(B^s)\boldsymbol{\epsilon_t}$$

où $p, q, P, Q$ sont respectivement les ordres des polynômes $\phi, \theta, \Phi, \Theta$ et sont à déterminer au même titre que la valeur des coefficients des différents polynômes. $\boldsymbol{\epsilon_t}$ est lui un bruit blanc.

Graphique 1 – Différence entre série brute et série corrigée




Nous déterminons les ordres maximums des coefficients $p$, $q$, $P$, $Q$ en regardant l'autocorrélograme et l'autocorrélogramme partiel de la série différenciée et désaisonnalisée $\mathbf{Z_t}$ (figure 2). Tout d'abord, les autocorrélations empiriques pour des lags compris entre 2 et 11 sont dans l'intervalle de confiance à 95%. Nous pouvons donc prendre $q_{max} = 3$. Pour la PACF, les autocorrélations partielles de lags 1,2,4,5,7,11,12,13,21 n'appartiennent pas aux bornes de significativité à 5%. Nous pouvons considérer que l'écart entre le lag 13 et le lag 21 permet de prendre $p_{max} = 14$.

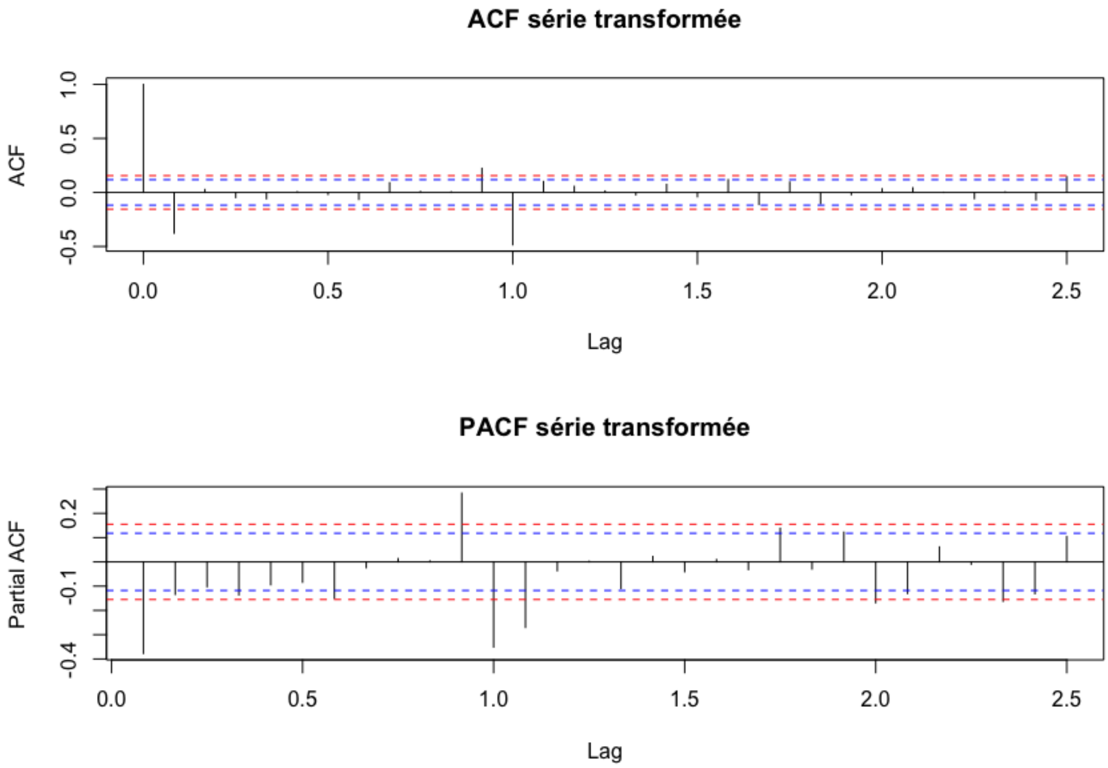
Pour ce qui est des ordres $P_{max}$ et $Q_{max}$, nous nous intéressons aux lags multiples de 12. À l'aide d'une instruction simple :

```
which(abs(acf$acf) > clim95)
```

avec `clim95` la valeur de la borne de significativité supérieure de l'ACF (ou PACF), nous pouvons avoir accès aux lags pour lesquels les autocorrélations ou autocorrélations partielles empiriques sont hors des bornes de significativité à 5%. En annexe, figure 4 sont représentés les ACF et PACF jusque lag 300 avec les bornes de significativité à 99%. En contrôlant la significativité à 5% nous devrions prendre en toute rigueur $P_{max} = 4$ et $Q_{max} = 12$ ; ces ordres sont trop élevés pour une éventuelle modélisation $SARIMA$. Nous contrôlons donc la significativité à 1%. Dans ce cas, nous pouvons prendre $Q_{max} = 5$, ce qui est déjà élevé. En conclusion, les ordres maximums vraisemblables pour la série $\mathbf{Z_t}$ sont :

$$\boxed{(p_{max}, q_{max}, P_{max}, Q_{max}) = (12, 3, 4, 5)}$$

Graphique 2 – ACF et PACF de la série stationnarisée jusque lag 30




Nous allons alors évaluer tous les modèles du type $SARIMA(p, 0, q)(P, 0, Q)_{s=12}$ avec $p \leq 5$ , $q \leq 6$, $P \leq 4$, $Q \leq 5$ et tester leurs validités et significativités à l'aide de tests statistiques.

La validité du modèle se teste à partir des tests de Ljung-Box comme précédemment. La significativité se vérifie en s'assurant que les coefficients associés aux ordres les plus élevés sont significativement non nuls, i.e que nous pouvons rejeter l'hypothèse de leur nullité à un certain seuil. Nous prendrons dans tous nos tests un seuil de significativitéà 5%.

En annexe, nous indiquons les modèles $AR$ (tableau 1), $MA$ (tableau 2), $ARMA$ (tableau 3) et $SARIMA$ (tableau 4) significatifs et valides à 5%. Au total, nous avions à l'origine retenu 266 modèles valides et significatifs (5 modèles $AR$ ; 4 modèles $MA$ ; 75 modèles $ARMA$ ; 182 modèles $SARIMA$). Ensuite, nous avons classé ces différents modèles retenus selon des indicateurs mesurant la qualité du modèle : les critères AIC et BIC. Nous avons privilégié de manière arbitraire l'AIC. Enfin, pour sélectionner le modèle final, nous avons calculé les erreurs de prédiction à horizon 4 (en tronquant la série $\mathbf{logX_t}$) et la somme des carrés résidus sur les données existantes (RMSE). Le tableau 5 recense les 10 modèles que nous avons finalement retenus : les modèles minimisateurs de l'AIC ou BIC, et 8 autres modèles dont les performances soit en terme de prédiction soit en terme de fidélité aux données étaient meilleures que celles des 2 premiers modèles. L'utilisation du critère d'erreur de prédiction n'est pas pertinente puisque nous ne cherchons pas à faire de la prédiction (car en patique nous ne connaissons pas les valeurs futures) mais bien de trouver un modèle valide, significatif et qui *fit* le mieux nos données observées. C'est la raison pour laquelle nous avons retenu comme modèle final un modèle de type :

$$\boxed{\mathbf{logX_t} \sim SARIMA(p = 0, d = 1, q = 3, P = 4, D = 1, Q = 2, s = 12)}$$

L'estimation des différents coefficients amène à la modélisation suivante :

$$\boxed{\begin{array}{c}(1 - B)(1 - B^{12})(1 + 0.0017B^{12} + 0.531B^{24} + 0.436B^{36} + 0.330B^{48})\ \mathbf{logX_t} \\ = \\ (1 + 0.441B + 0.048B^2 + 0.159B^3)(1 + 0.797B^{12} - 0.625B^{24})\ \boldsymbol{\epsilon_t}\end{array}}$$

## Partie 3 : Prévision

### 5&6. Écrire l'équation vérifiée par la région de confiance de niveau $\alpha$ sur les valeurs futures $(\mathbf{Z_{T+1}}, \mathbf{Z_{T+2}})$. Préciser les hypothèses utilisées pour obtenir cette région (réponses en jaune).

Comme notre série $(\mathbf{Z_t})_{t=1,...,T}$ est un $SARIMA(0,0,3,4,0,2,12)$ stationnaire, celle-ci admet une représentation ARMA canonique obtenue en supprimant les racines communes et en inversant les racines de module $< 1$. De fait, $\mathbf{Z_t}$ est solution de

$$\Phi(B)\mathbf{Z_t} = \Psi(B)\boldsymbol{\epsilon_t}$$

avec $B$ l'opérateur backward, $\Phi$ et $\Psi$ polynômes de degré respectifs $p \leq P \times s = 48$ et $q \leq q' + Q \times s = 27$ n'ayant pas de racine commune ni de racines à l'intérieur du disque unité, et $\boldsymbol{\epsilon_t}$ l'innovation linéaire de $\mathbf{Z_t}$ telle que $\boldsymbol{\epsilon_t} \sim \mathcal{N}(0, \sigma^2)$ et $(\boldsymbol{\epsilon_t})_{t=1,...,T}$ iid. Nous pouvons réécrire :

$$\mathbf{Z_t} = \boldsymbol{\epsilon_t} + \sum_{j=1}^{+\infty} c_j \boldsymbol{\epsilon_{t-j}} \ , \quad \sum_{j=1}^{+\infty} c_j^2 < \infty$$

De fait, comme nous ne disposons que des valeurs de la série entre $t = 1$ et $T$, nous avons plutôt :

$$\mathbf{Z_t} = \boldsymbol{\epsilon_t} + \sum_{j=1}^{T-1} c_j \boldsymbol{\epsilon_{t-j}} \ , \quad \sum_{j=1}^{T-1} c_j^2 < \infty$$

En projetant sur le passé fini de $Z_T$, on a donc pour tout $h \in \mathbb{N}$ :

$$\mathbf{Z_{T+h|Z_T,...,Z_1}} := \mathbf{\hat{Z}_{T+h|T}} = \sum_{j=h}^{T+h-1} c_j \boldsymbol{\epsilon_{T+h-j}} := EL\left(\mathbf{Z_{T+h}}|\mathbf{Z_T}, ..., \mathbf{Z_1}\right) = EL(\mathbf{Z_{T+h}}|\boldsymbol{\epsilon_T}, ..., \boldsymbol{\epsilon_1})$$

où $\mathbf{\hat{Z}_{T+h|T}}$ est la prévision linéaire optimale *tronquée* d'horizon $h > 0$ (au sens des moindres carrés). On en déduit au passage l'erreur de précision :

$$e(h) = \mathbf{Z_{T+h}} - \mathbf{\hat{Z}_{T+h|T}} = \sum_{j=0}^{h-1} c_j \boldsymbol{\epsilon_{T+h-j}} \tag{1}$$

et l'erreur quadratique moyenne :

$$MSE(h) = \mathbb{E}\left(\mathbf{Z_{T+h}} - \mathbf{\hat{Z}_{T+h|T}}\right)^2 = \sigma^2 \sum_{j=0}^{h-1} c_j^2 \tag{2}$$

Étant donné que les résidus sont gaussiens, les équations (1) et (2) permettent de conclure que, conditionnellement à $\mathbf{Z_u}, u \leq T$ :

$$\mathbf{Z_{T+h}} \sim \mathcal{N}\left(\mathbf{\hat{Z}_{T+h|T}} \ , \ \sigma^2 \sum_{j=0}^{h-1} c_j^2\right)$$

Pour finalement obtenir un intervalle de confiance de la valeur prédite de $\mathbf{Z_{T+h}}$ à 95%, $\forall h \in \mathbb{N}$ du type :

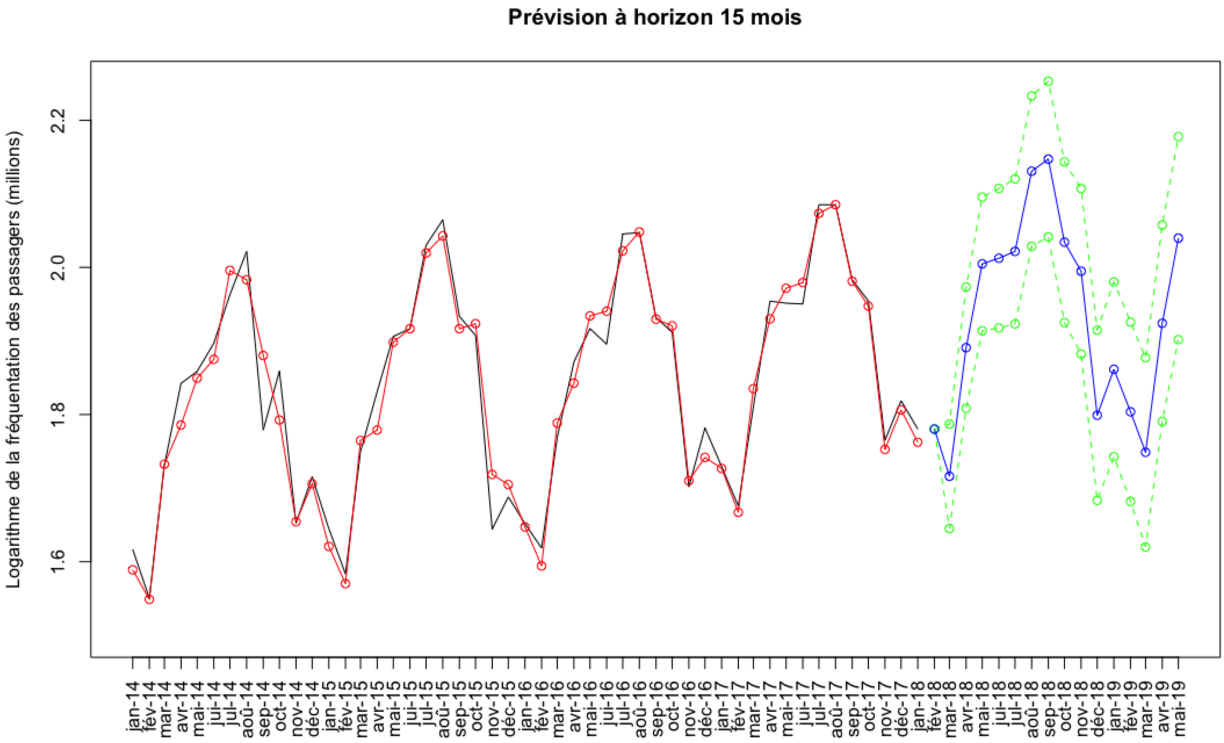
$$\boxed{\mathbf{Z_{T+h}}|\mathbf{Z_T}, ...\mathbf{Z_1} \in \left[\mathbf{\hat{Z}_{T+h|T}} \pm \sigma\sqrt{\textstyle\sum_{j=0}^{h-1} c_j^2} \times q_{1-\alpha/2}^{\mathcal{N}}\right] \ , \ \ q_{1-\alpha/2}^{\mathcal{N}} = 1,96}$$

On peut également remplacer $\sigma$ et les $c_i$ par leurs estimateurs respectifs $\hat{\sigma}$ et $\hat{c}_i$.

### 7. Déterminer graphiquement cette région pour $\alpha = 95\%$. Commenter.

Nous avons affiché ci-dessous (figure 3) les prévisions à horizon 15 mois de notre modèle $SARIMA$. Est tracé en vert l'intervalle de confiance à 95% des valeurs prédites. Nous remarquons que l'intervalle de confiance s'élargit au fur et à mesure que la prédiction est lointaine. Ceci normal étant donné la dépendance en $h$ au travers de l'expression $\sum_{j=0}^{\mathbf{h}-1} c_j^2$ dans l'intervalle de confiance théorique. Vous trouverez en annexe (figure 5) l'évolution de cet intervalle de confiance en fonction des 15 mois à prédire.

Graphique 3 – Prévisions




### 8. Question ouverte : soit $\mathbf{Y_t}$ une série stationnaire disponible de $t = 1$ à $T$. On suppose que $\mathbf{Y_{T+1}}$ est disponible plus rapidement que $\mathbf{Z_{T+1}}$. À quelles conditions cette information permet-elle d'améliorer la prévision de $\mathbf{Z_{T+1}}$ ?

Se demander à quelles conditions la prévision de $\mathbf{Y_{T+1}}$ permet d'améliorer la prévision de $\mathbf{Z_{T+1}}$ est équivalent à savoir si $(\mathbf{Y_t})$ cause instantanément $(\mathbf{Z_t})$ au sens de Granger, c'est à dire :

$$\mathbb{E}\left(\mathbf{Z_{T+1}} \middle| \{\mathbf{Z_u}, \mathbf{Y_u}, u \leq T\} \cup \{\mathbf{Y_{T+1}}\}\right) \neq \mathbb{E}\left(\mathbf{Z_{T+1}} \middle| \{\mathbf{Z_u}, \mathbf{Y_u}, u \leq T\}\right)$$

Cette inégalité est vérifiée si et seulement si les innovations de $\mathbf{Z_{T+1}}$ et $\mathbf{Y_{T+1}}$ sont non corrélées (en effet, d'après les hypothèses, la série $(\mathbf{X_t}) := (\mathbf{Z_t}, \mathbf{Y_t})$ est stationnaire). Or, nous avons supposé que les résidus de $(\mathbf{Z_t})$ sont gaussiens. Étant donné que $(\boldsymbol{\epsilon_t})$ est un bruit blanc fort, les innovations sont donc indépendantes et identiquement distribuées de loi $\mathcal{N}\left(0, \sigma_\epsilon^2\right)$. Il faut donc distinguer les cas selon la loi des innovations de la série $(\mathbf{Y_t})$.

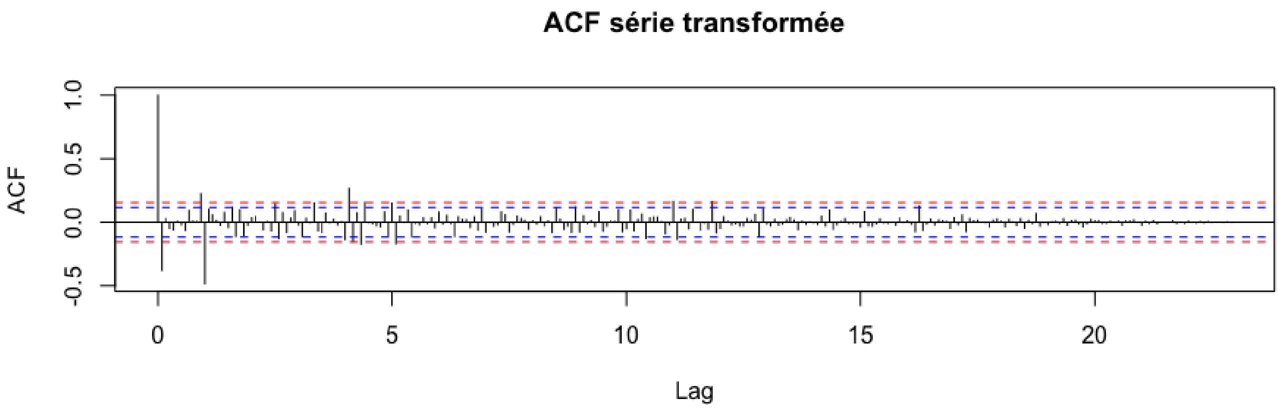
Dans le cas où les innovations de la série $\mathbf{Y_t}$ suivraient une loi gaussienne $\mathcal{N}(0, \sigma_y^2)$, un simple test de Student permet de tester la corrélation entre les innovations de la série $\mathbf{Z_t}$ et celles de la série $\mathbf{Y_t}$.

Si nous n'avons pas d'information sur la loi des innovations de la série $\mathbf{Y_t}$, mais que nous savons que les résidus sont i.i.d. (s'il s'agit un bruit blanc fort, par exemple), alors nous pouvons effectuer des tests de corrélations ne s'appuyant pas sur la distribution des variables (les tests de corrélation de Spearman ou de Kendall, par exemple).

En revanche, si nous n'avons pas d'information sur la loi des innovations de la série $\mathbf{Y_t}$, et qu'en plus elles ne sont pas i.i.d., il est difficile de conclure sur l'amélioration de la prévision, car nous ne pouvons tester la corrélation entre les innovations de la série $\mathbf{Z_t}$, et celles de la série $\mathbf{Y_t}$.

# Annexes

Graphique 4 – ACF et PACF de la série stationnarisée jusque lag 300



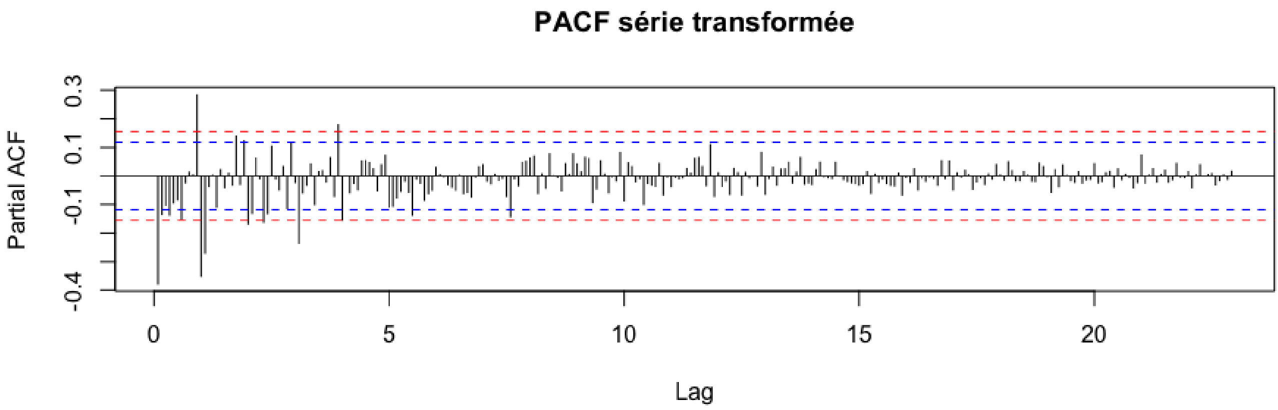




Tableau 1 – Modèles AR retenus

| p | q | AIC | BIC |
|---|---|---|---|
| 29 | 0 | -106.70 | 5.53 |
| 25 | 0 | -104.60 | -6.85 |
| 28 | 0 | -104.57 | 4.05 |
| 24 | 0 | -100.23 | -6.10 |
| 21 | 0 | -98.99 | -15.72 |

[1] $p_{max} = 30$.

Tableau 2 – Modèles MA retenus

| p | q | AIC | BIC |
|---|---|---|---|
| 0 | 13 | -145.07 | -90.76 |
| 0 | 20 | -139.97 | -60.32 |
| 0 | 27 | -134.39 | -29.40 |
| 0 | 26 | -134.37 | -33.00 |

[1] $q_{max} = 30$.

Tableau 3 – 20 premiers modèles ARMA retenus (sur 75)

| p | q | AIC | BIC |
|---|---|---|---|
| 1 | 13 | -146.66 | -88.73 |
| 2 | 15 | -146.48 | -77.69 |
| 6 | 19 | -146.41 | -48.66 |
| 5 | 16 | -145.62 | -62.35 |
| 8 | 20 | -145.37 | -36.76 |
| 0 | 13 | -145.07 | -90.76 |
| 18 | 14 | -145.03 | -21.93 |
| 9 | 20 | -144.87 | -32.64 |
| 3 | 15 | -144.66 | -72.26 |
| 12 | 20 | -144.44 | -21.35 |
| 7 | 12 | -141.18 | -65.15 |
| 3 | 14 | -141.01 | -72.23 |
| 19 | 18 | -140.90 | 0.30 |
| 5 | 18 | -140.78 | -50.27 |
| 9 | 20 | -139.97 | -60.32 |
| 16 | 15 | -139.91 | -20.44 |
| 2 | 12 | -139.43 | -81.50 |
| 4 | 17 | -139.21 | -55.94 |
| 17 | 20 | -138.38 | 2.82 |
| 4 | 16 | -137.42 | -57.77 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 3 | 10 | -85.78 | -31.48 |

[1] $p_{max} = 20$, $q_{max} = 20$.

Tableau 4 – 20 premiers modèles SAR(I)MA retenus (sur 182)

| p | d | q | P | D | Q | s | AIC | BIC |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | 4 | 0 | 2 | 12 | -159.39 | -119.56 |
| 4 | 0 | 5 | 4 | 0 | 1 | 12 | -159.28 | -101.35 |
| 4 | 0 | 5 | 4 | 0 | 2 | 12 | -158.85 | -97.30 |
| 4 | 0 | 5 | 0 | 0 | 5 | 12 | -157.57 | -99.65 |
| 0 | 0 | 3 | 2 | 0 | 4 | 12 | -157.40 | -117.58 |
| 3 | 0 | 3 | 4 | 0 | 2 | 12 | -157.07 | -106.38 |
| 5 | 0 | 4 | 2 | 0 | 4 | 12 | -156.61 | -95.06 |
| 0 | 0 | 3 | 3 | 0 | 5 | 12 | -156.33 | -109.26 |
| 5 | 0 | 2 | 2 | 0. | 4 | 12 | -156.17 | -101.86 |
| 5 | 0 | 4 | 3 | 0 | 3 | 12 | -155.88 | -94.33 |
| 5 | 0 | 5 | 0 | 0 | 5 | 12 | -155.81 | -94.27 |
| 3 | 0 | 2 | 0 | 0 | 5 | 12 | -155.68 | -112.23 |
| 3 | 0 | 5 | 2 | 0 | 4 | 12 | -155.50 | -97.58 |
| 4 | 0 | 3 | 3 | 0 | 4 | 12 | -155.04 | -97.12 |
| 0 | 0 | 1 | 4 | 0 | 2 | 12 | -155.01 | -122.43 |
| 0 | 0 | 1 | 5 | 0 | 3 | 12 | -154.62 | -114.80 |
| 0 | 0 | 3 | 0 | 0 | 5 | 12 | -154.53 | -118.32 |
| 0 | 0 | 3 | 3 | 0 | 3 | 12 | -154.42 | -114.60 |
| 4 | 0 | 2 | 4 | 0 | 2 | 12 | -154.42 | -103.73 |
| 4 | 0. | 1 | 4 | 0 | 3 | 12 | -154.35 | -103.67 |
| 0 | 0 | 3 | 5 | 0 | 0 | 12 | -154.28 | -118.08 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 5 | 0 | 4 | 1 | 0 | 0 | 12 | -111.55 | -68.11 |

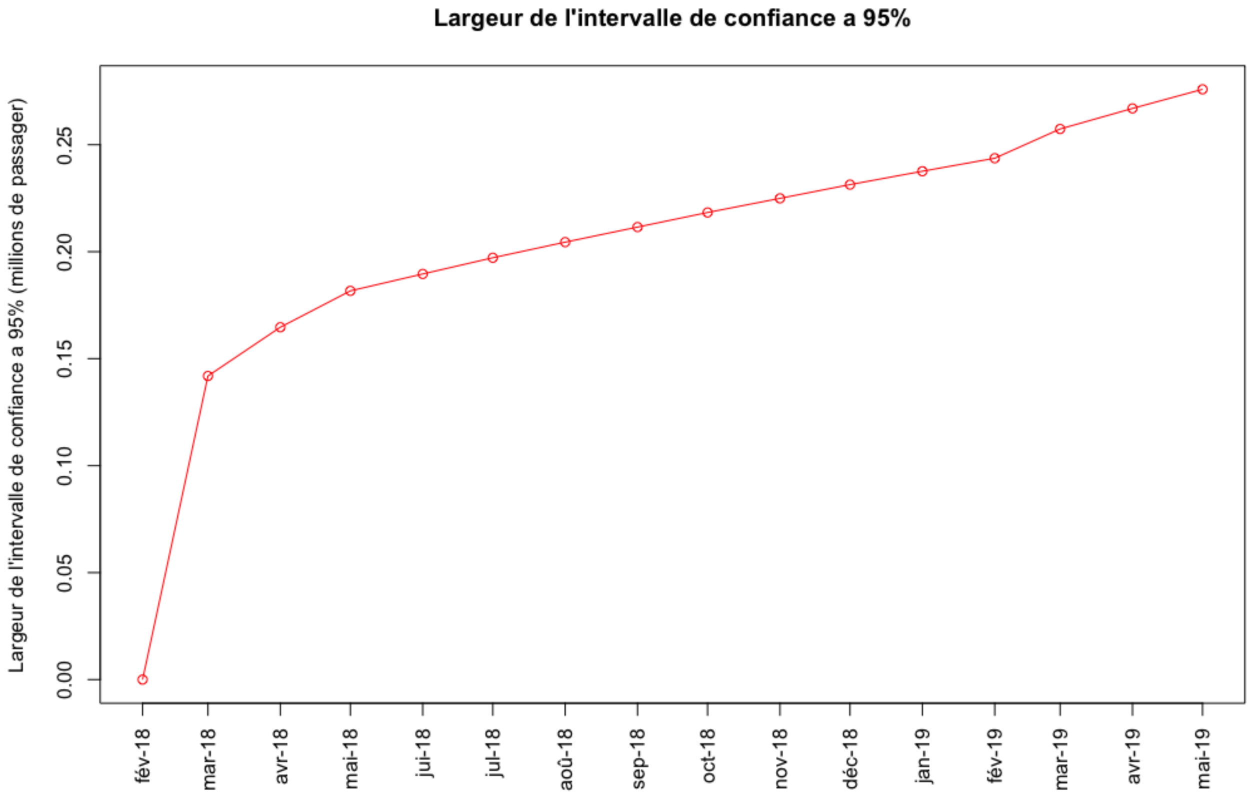
[1] $p_{max} = 5$, $q_{max} = 5$, $P_{max} = 5$, $Q_{max} = 5$.

Tableau 5 – Erreurs de prédiction à horizon 4 dates et RMSE des 10 modèles retenus

| Modèles | Critère minimisé | $\mathcal{E}_{prediction}$ | RMSE |
|---|---|---|---|
| $SARIMA(0,0,3,4,0,2,12)$ | AIC | 0.02326554 | 0.6010692 |
| $SARIMA(2,0,1,0,0,1,12)$ | BIC | 0.01947327 | 0.6296436 |
| $SARIMA(0,0,1,4,0,2,12)$ | - | 0.02467597 | 0.6135300 |
| $SARIMA(0,0,3,2,0,4,12)$ | - | 0.02277452 | 0.6032207 |
| $SARIMA(0,0,1,2,0,4,12)$ | - | 0.01623702 | 0.6582885 |
| $SARIMA(0,0,3,0,0,5,12)$ | - | 0.02662866 | 0.6212872 |
| $SARIMA(0,0,3,5,0,0,12)$ | - | 0.02478279 | 0.6079626 |
| $SARIMA(2,0,1,0,0,5,12)$ | - | 0.02810212 | 0.6227271 |
| $SARIMA(0,0,1,0,0,5,12)$ | - | 0.01918966 | 0.6952696 |
| $SARIMA(0,0,3,4,0,0,12)$ | - | 0.02534345 | 0.6243116 |

[1] $p_{max} = 5$, $q_{max} = 5$, $P_{max} = 5$, $Q_{max} = 5$.
[2] Erreurs minimales en jaune.

Graphique 5 – Largeur de l'intervalle de confiance à 95% dans le cadre de la prévision



# Table des figures



# Liste des tableaux

# Code

## Partie 1

```r
###############################################
########## Importation des packages ###########
###############################################
library(zoo)
library(tseries)
library(ggplot2)
library(fUnitRoots)
library(scales)
library(lm)
library(tseries)
library(forecast)
library(xtable)

###############################################
#### Importation et traitement des donnees ####
###############################################


setwd(dir="/Users/bfiliot/Desktop/ENSAE/S2/stprojet")
table = read.csv("valeurs_mensuelles.csv", head=T, sep=";")

names(table)[2] = "Frequentation de passagers vols internationaux - Paris"

table = table[-2,]  # suppression des deux premieres lignes
table = table[-1,]  # elles ne correspondent pas a des valeurs

table$year = substring(table[,1],1,4)  # date au format annee / mois
table$month = substring(table[,1],6,7)

data = table[order(table[4],table[5]),]   # ordonnencement du plus vieux au plus recent

xm = as.zoo(ts(data[[2]],
          start     = c(1994,1),
          end       = c(2018,1),
          frequency = 12)) # transformation en objet zoo

dates = seq(as.POSIXct("1994-01-01"), by = "month", length.out = 289)

###############################################
####### Affichage graphique de la serie #######
###############################################

# Utilisation de ggplot (uniquement ici)

p = ggplot(data = xm, aes(x = dates, y = xm)) +
  geom_line(colour = 'blue') +
  scale_x_datetime(labels = date_format("%Y"), breaks = date_breaks("years")) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = "Temps") +
  labs(y = "Fréquentation des passagers (millions)") +
  theme(text = element_text(size=20))
  #labs(title = "Serie temporelle d'origine") +

p # graphe de la serie temporelle

p2 = ggplot(data = xm[1:36], aes(x = seq(as.POSIXct("1994-01-01"), by = "month", length.out = 36), y = xm[1:36])) +
  geom_line(colour = 'blue') +
  scale_x_datetime(labels = date_format("%Y-%m"), breaks = date_breaks("months")) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = "Temps") +
  labs(y = "Fréquentation des passagers (millions)")

p2 # graphe de la serie temporelle les 3 premieres annees

```

```r

#################################################
####### Stationnarisation de la serie ########
#################################################

# Pour ce genre de serie temporelle il est utile
# de transformer en log pour reduire l'heteroscedasticite
# des residus et/ou problemes de non-linearite.

xmlog = log(xm)

# Comme nous l'avons vu, la serie presente une tendance haussiere
# ainsi qu'une saisonnalite annuelle. La tendance haussiere laisse
# presager d'une serie stationnaire.

# Tout d'abord verifions la nature de la tendance deterministe.

T     = length(xmlog)
time  = 1:T
fit1  = lm(xmlog ~ poly(time,1))
fit2  = lm(xmlog ~ poly(time,2))
pred1 = predict(fit1)
pred2 = predict(fit2)

plot(dates,
     xmlog,
     type = 'l',
     xlab = 'Temps',
     ylab = 'Frequentation des passagers (millions)')

lines(dates, pred1, type = "l", col = "red",   lty = 1, lwd = 2)
lines(dates, pred2, type = "l", col = "green", lty = 1, lwd = 2)

legend("bottomright",
       legend   = c("Série", "Approx. linéaire", "Approx. quadratique"),
       col      = c("black","red", "green"),
       text.col = c("black", "red", "green"),
       lty      = c(1,1,1),
       ncol     = 1,
       cex      = 0.5)

# Conclusion : tendance lineaire en a + bt. On se place donc dans le cas "ct".
# Notre serie log est-elle stationnaire ?

adf = adfTest(xmlog, lags=0, type="ct")
adf@test$p.value # < 0.01
# On rejette fortement la racine unite a 5%, la serie est donc stationnaire
# a 5%. Ce qui finalement n'est pas si contre-intuitif etant donne la tendance
# haussiere deterministe. Le modele est-il valide ?

# Regardons la validite du modele avec la persistance des residus.


Qtest <- function(x, kmax=24){
  t(apply(matrix(1:kmax), 1, FUN=function(k){
    pv = Box.test(x, lag=k, type="Ljung-Box")$p.value
    return(c("lag"=k,"pval"=pv, "Non autocorreles a 5%" = (pv>0.05)))
  }))
}

resid = adf@test$lm$residuals
plot(resid,type="l")
Qtest(adf@test$lm$residuals)
# Ici on voit que le test de blancheur des residus est rejete
# a 5% a tous les ordres. Les residus sont correles jusqu'a l'ordre 24.
# Le modele n'est donc pas valide si l'on considere un lag = 0 dans
# la formule de l'ADF.
# Mais nous pourrions augmenter ce lag de telle sorte a voir a partir duquel
# les residus ne sont pas correles (a 5%) et ceci pour tous les 24 ordres.
# Ceci est verifie pour le lag 23. C'est trop, il faut pouvoir diminuer
```

```r
# ce lag. Ce que nous decidons de faire est de garder un test adf avec un lag = 0
# et tester la blancheur des residus a l'ordre 1. Des qu'une transformation
# (elle existe par anticipation...) de la serie xmlog sera telle que
# le residu est non correle a l'ordre 1 a 5%, on choisira cette transformation
# et on regardera le lag minimum a partir duquel l'ADF/le modele est valide.
# On prend tout d'abord en compte la saisonnalite.

xdesaison = diff(xmlog,12)
adf = adfTest(xdesaison, lags=0, type="ct")
adf@test$p.value
# On rejette aussi la racine unite a 5%, la serie differenciee est stationnaire
# ce qui est logique.
# Ici on remarque que la p-valeur est superieure a 5% mais tres proche (0.05417)
resid = adf@test$lm$residuals
plot(resid,type="l")
Box.test(resid, lag=1, type="Ljung-Box")$p.value
# On decide donc de differencier egalement la serie pour avoir un rejet
# de la racine unitaire plus net.

# Appliquons donc une desaisonnalisation et une differenciation.

xdd = diff(desaison,1)
adf = adfTest(xdd, lags=0, type="c") #lag 11 pour 12 premiers ; lag 13 pour 24 premiers
adf@test$p.value
# On rejette la racine unite a 5%, la serie differenciee et desaisonnalisee
# est stationnaire ce qui est logique.
resid = adf@test$lm$residuals
plot(resid,type="l")
Box.test(resid, lag=1, type="Ljung-Box")$p.value
# Et cette fois l'absence d'autocorrelation n'est pas rejetee a l'ordre 1 a 5%
# car elle est egale a 0.39516. Nous gardons donc cette transformation.
# Verifions a partir de quel lag (dans la formule de l'ADF) le modele est valide,
# i.e les residus sont non-autocorreles jusqu'a l'ordre 24 a 5%.
# Au passage on peut verifier qu'a lag 0 le modele n'est pas valide meme
# si pour certains ordres les residus ne sont pas autocorreles a 5%.

# On s'intéresse au lag minimum tel que les residus ne sont pas autocorreles jusqu'a l'ordre 24.

lagmin <- function(serie){
  i = 0
  if (serie == xdd){
    type = "nc"} # la serie desaisonnalite et differenciee est centree
  if (serie == xdesaison){
    type = "c"} # la serie desaisonnalite est non centree
  if (serie == xmlog){
    type = "ct"} # la serie xmlog presente une tendance lineaire deterministe
  adf     = adfTest(serie, lags=i, type=type)
  bbxtest = Qtest(adf@test$lm$residuals)
  while (sum(bbxtest[,3]) < 24) {
    i       = i+1
    adf     = adfTest(serie, lags=i, type=type)
    bbxtest = Qtest(adf@test$lm$residuals)
  }
  return("lagminADF"=i)
}

lagmin(xdd)
# On trouve un lag min de 13 ce qui est bien mieux que le 23 pour
# la serie non transformee.
# On a donc un modele ADF valide lorsque celui-ci prend en compte
# 13 retards.
adfTest(xdd, lags=13, type="nc")
pp.test(xdd) # L'hypothese de racine unitaire est tres fortement rejetee.
kpss.test(xdd) # L'hypothese de stationnarite n'est pas rejetee a 5%.

# On se contentera donc de notre transformation dans la mesure ou
# elle garantit la non-autocorrelation des residus jusqu'a l'ordre 24
# pour un lagADF egal a 13, tout ceci pour des seuils de 5%.
# En soit nous aurions juste desaisonnaliser la serie...
# Car on remarque que le lag minimal de l'ADF pour lequel les residus
```

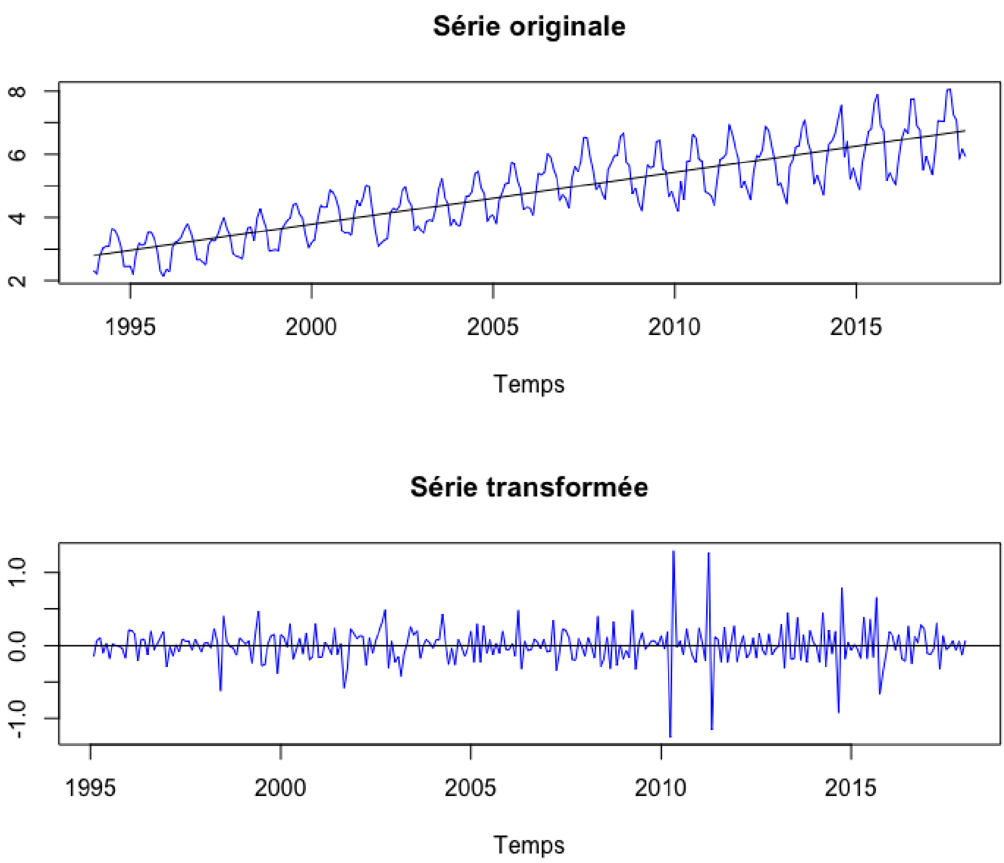
```
# sont non-autocorreles a tous les ordres est... ->
lagmin(xdesaison,50)
adfTest(xdesaison, lags=13, type="c")
pp.test(xdesaison) # L'hypothese de racine unitaire est tres fortement rejetee.
kpss.test(xdesaison)
#... -> 13, comme pour la serie desaisonnalisee. On retiendra donc qu'a
# lagADF = 0, les residus de la serie desaisonnalisee sont "presque"
# auto-correles a l'ordre 1 au niveau 5% (le terme "presque" est
# peu rigoureux) tandis que non pour la serie xdd desaisonnalisee et differenciee.

# Difference avant / apres

par(mfrow = c(1,2)) #ou (2,1)
plot(dates,
     xm,
     type = 'l',
     xlab = "Temps",
     ylab = 'Logarithme de la fréquentation des passagers (millions)',
     col  = 'blue',
     main = 'Série originale')

plot(dates[14:289],
     xdd,
     type = 'l',
     xlab = "Temps",
     ylab = 'Logarithme de la fréquentation des passagers (millions)',
     col  = 'blue',
     main = 'Série transformée')
abline(a=0, b=0)
```

Graphique 6 – Logarithme de la fréquentation des passagers (millions) - Comparaison avant après



## Partie 2

```r
##################################################
########## Calibrage d'un modèle ARMA ##########
##################################################

########### Fonctions prealables ###############

# Fonction de test de l'absence d'autocorrelation des residus
Qtest <- function(x, kmax=24){
  t(apply(matrix(1:kmax), 1, FUN=function(k){
    pv = Box.test(x, lag=k, type="Ljung-Box")$p.value
    return(c("lag"=k,"pval"=pv, "Non autocorreles a 5%" = (pv>0.05)))
  }))
}

# Fonction de test des significativites individuelles des coefficients
signif <- function(estim){
  coef = estim$coef
  se   = sqrt(diag(estim$var.coef))
  t    = coef/se
  pval = (1-pnorm(abs(t)))*2
  return(rbind(coef,se,pval))
}

# Fonction annexe
tf_binaire <- function(signif){
  if (is.na(signif)) return(1)
  else return(signif)
}

# Selection des valeurs pmax, qmax, Pmax, Qmax #
# Pour cela nous sommes alles voir comment sont calcules les
# limites de l'intervalle de confiance a 95% pour l'acf (et donc pacf)
# via la fonction
getS3method("plot", "acf")
# et nous avons retrouve la variable clim egale a la borne
par(mfrow=c(2,1))
clim95 = qnorm((1 + 0.95)/2)/sqrt(length(xdd))
acf    = acf(xdd,  lag.max = 300, main = "ACF série transformée")
pacf   = pacf(xdd, lag.max = 300, main = "PACF série transformée")
# Les 2 valeurs coincident.

# On veut determiner explicitement quels sont les lags pour lesquels les
# valeurs observees sont hors de la zone de significativite a 5%.

which(abs(acf$acf) > clim95)
which(abs(pacf$acf) > clim95)

# On remarque que les valeurs des autocorrelations pour des lag
# entre q=2 et q=11 sont dans l'intervalle de confiance a 95%  donc on peut
# supposer legitiment que qmax = 3.
# Pour la PACF, c'est un peu plus delicat. En effet les autocorrelations
# partielles de lag 1,2,4,5,7,11,12,13,21 ne sont pas dans l'intervalle de
# confiance a 95%. On peut considerer que l'ecart entre lag 13 et lag 21
# permet de prendre pmax = 14.
# Au niveau des ordres Pmax et Qmax, qu'en est-il ?
# Etant donne que les mutliples de 12 sont : 12,24,36,48,60,72,84,96,108,120,132,144
# on peut prendre Pmax = 4 et on pourrait aller jusque Qmax = 12...
```

```r
# Si l'on regarde avec un intervalle de confiance a 99%... ->

clim99 = qnorm((1 + 0.99)/2)/sqrt(length(xdd))
par(mfrow = c(2,1))
acf    = acf(xdd,  lag.max = 300, main = "ACF série transformée")
abline(clim99, 0, col = "red", lty = 2)
abline(-clim99,0, col = "red", lty = 2)
pacf   = pacf(xdd, lag.max = 300, main = "PACF série transformée")
abline(clim99, 0, col = "red", lty = 2)
abline(-clim99,0, col = "red", lty = 2)

which(abs(acf$acf) > clim99)
which(abs(pacf$acf) > clim99)

#... -> on peut prendre Qmax = 5.

# On a donc les ordres maximums vraisemblables
# (p*,d*,q*,P*,D*,Q*,s) = (14,0,3,4,0,5)

# La fonction suivante automatise le processus
# de significativite et validation puis calcul
# des AIC, BIC.

modelchoice <- function(p,q,P,Q,serie,k=24){

  print(paste0("SARIMA (p=",p,",d=",0,",q=",q,",P=",P,",D=",0,",Q=",Q,",s=",12,")"))

  estim = try(arima(serie, order=c(p,0,q), seasonal=list(order = c(P,0,Q), period=12),
                    method="ML",
                    optim.control=list(maxit=20000)))

  if (class(estim)=="try-error") return(c("p"=p,"d"=0,"q"=q,"P"=P,"D"=0,"Q"=Q,"s"=12,
                                          "ARsignif"       = NA,
                                          "MAsignif"       = NA,
                                          "ARSAISONsignif" = NA,
                                          "MASAISONsignif" = NA,
                                          "ModeleSignif"   = NA,
                                          "ModeleValide"   = NA,
                                          "Selectionne"    = NA,
                                          "AIC"            = NA,
                                          "BIC"            = NA))

  arsignif        = if (p==0) NA else signif(estim)[3,p]<=0.05
  masignif        = if (q==0) NA else signif(estim)[3,p+q]<=0.05
  ARsaisonsignif  = if (P==0) NA else signif(estim)[3,p+q+P]<=0.05
  MAsaisonsignif  = if (Q==0) NA else signif(estim)[3,p+q+P+Q]<=0.05

  ModeleSignif    = prod(apply(matrix(c(arsignif,masignif,ARsaisonsignif,MAsaisonsignif)), 1, tf_binaire))

  resnocorr       = sum(Qtest(estim$residuals,k)[,3])==k

  aic             = AIC(estim)
  bic             = BIC(estim)

  return(c("p"=p,"d"=0,"q"=q,"P"=P,"D"=0,"Q"=Q,"s"=12,
           "ARsignif"       = arsignif,
           "MAsignif"       = masignif,
           "ARSAISONsignif" = ARsaisonsignif,
           "MASAISONsignif" = MAsaisonsignif,
           "ModeleSignif"   = ModeleSignif,
           "ModeleValide"   = resnocorr,
           "Selectionne"    = ModeleSignif*resnocorr,
           "AIC"            = aic,
           "BIC"            = bic))
}
```

```r
# On verifie si des modeles AR sont selectionnes...

pmax = 30 #par extreme securite
resultAR = data.frame(t(mapply(function(p) modelchoice(p,0,0,0,xdd), c(0:pmax))))
resultAR = resultAR[with(resultAR, order(resultAR$Selectionne, decreasing = TRUE)),]
View(resultAR)
ARselec  = resultAR[resultAR$Selectionne == 1,]
ARselec  = ARselec[with(ARselec, order(ARselec$AIC, decreasing = FALSE)),]
View(ARselec)

# On selectionne AR(29), AR(25), AR(28), AR(24), AR(21)... dans l'ordre des
# AIC les plus faibles. Les ordres p sont beaucuop trop eleves pour
# ce type de modelisation. On ecarte donc une modelisation de la serie
# xdd par un AR. D'autant plus que les BIC sont tres grands.

# Des modeles MA ?

qmax = 30 #par extreme securite
resultMA = data.frame(t(mapply(function(q) modelchoice(0,q,0,0,xdd), c(0:qmax))))
resultMA = resultMA[with(resultMA, order(resultMA$Selectionne, decreasing = TRUE)),]
View(resultMA)
MAselec  = resultMA[resultMA$Selectionne == 1,]
MAselec  = MAselec[with(MAselec, order(MAselec$AIC, decreasing = FALSE)),]
View(MAselec)

# On selectionne MA(13), MA(20), MA(27), MA(26)... dans l'ordre des
# AIC les plus faibles. On retient donc le MA(13) puisqu'il minimise
# a la fois l'AIC et le BIC.

# Des modeles ARMA ?

pmax = 20
qmax = 20
resultARMA = data.frame(t(mapply(function(p,q) modelchoice(p,q,0,0,xdd),
                             c(apply(matrix(c(0:pmax)),1, function(k) rep(k,(qmax+1)))),
                             c(0:qmax))))

resultARMA = resultARMA[with(resultARMA, order(resultARMA$Selectionne, decreasing = TRUE)),]
View(resultARMA)
ARMAselec  = resultARMA[resultARMA$Selectionne == 1,]
ARMAselec  = ARMAselec[with(ARMAselec, order(ARMAselec$AIC, decreasing = FALSE)),]
View(ARMAselec)
length(which(is.na(ARMAselec[,16])==0))

# On garde 75 modeles ARMA au total.
# Les deux modeles ARMA optimaux sont ARMA(0,13) et ARMA(1,13). Ils minimisent
# chacun un critère.

# Des modeles SARIMA ?

pmax = 5
qmax = 5
Pmax = 5
Qmax = 5
resultSARIMA = data.frame(t(mapply(function(p,q,P,Q) modelchoice(p,q,P,Q,xdd),
                      c(apply(matrix(c(0:pmax)),1, function(k) rep(k,(qmax+1)*(Pmax+1)*(Qmax+1)))),
                      c(apply(matrix(c(0:qmax)),1, function(k) rep(k,(Pmax+1)*(Qmax+1)))),
                      c(apply(matrix(c(0:Pmax)),1, function(k) rep(k,(Qmax+1)))),
                      c(0:Qmax))))
resultSARIMA = resultSARIMA[with(resultSARIMA, order(resultSARIMA$Selectionne, decreasing = TRUE)),]
View(resultSARIMA)
SARIMAselec  = resultSARIMA[resultSARIMA$Selectionne == 1,]
SARIMAselec  = SARIMAselec[with(SARIMAselec, order(SARIMAselec$AIC, decreasing = FALSE)),]
View(SARIMAselec)
length(which(is.na(SARIMAselec[,16])==0))

# On selectionne 182 modeles. Les 5 premiers sont par ordre croissant d'AIC:
# SARIMA(0,0,3,4,0,2,12), SARIMA(4,0,5,4,0,1,12), SARIMA(4,0,5,4,0,2,12)
# SARIMA(4,0,5,0,0,5,12), SARIMA(0,0,3,2,0,4,12)
```

```r
# On concatene tous les modeles retenus.

resultats_retenus            = rbind(ARselec,MAselec,ARMAselec,SARIMAselec)
resultats_retenus            = resultats_retenus[with(resultats_retenus,
                                        order(resultats_retenus$AIC, decreasing = FALSE)),]
resultats_retenus["AIC+BIC"] = resultats_retenus$AIC + resultats_retenus$BIC
resultats_retenus["AIC+BIC"] = resultats_retenus$AIC + resultats_retenus$BIC
View(resultats_retenus)
length(which(is.na(resultats_retenus[,16])==0))

# Au total, ce sont 266 modeles qui sont retenus. Bien sur,
# il reste maintenant a selectionner le ou les meilleurs du point
# de la minimisation des criteres.

# Pour le latex :
print(xtable(ARselec,     type = "latex", tabular.environment = "longtable"),
      file = "ARselec.tex")
print(xtable(MAselec,     type = "latex", tabular.environment = "longtable"),
      file = "MAselec.tex")
print(xtable(ARMAselec,   type = "latex", tabular.environment = "longtable"),
      file = "ARMAselec.tex")
print(xtable(SARIMAselec, type = "latex", tabular.environment = "longtable"),
      file = "SARIMAselec.tex")

##################################################
########## Selection d'un modèle ARMA ##########
##################################################

# Conclusion : il nous reste a comparer les dix premiers modeles suivants :
# celui maximisant l'AIC : SARIMA(0,0,3,4,0,2,12)
# celui maximisation le BIC : SARIMA(2,0,1,0,0,1,12)
# 8 autres modèles ayant une somme de criteres AIC + BIC parmi les plus petites :
# SARIMA(0,0,1,4,0,2,12) ; SARIMA(0,0,3,2,0,4,12)
# SARIMA(0,0,1,2,0,4,12) ; SARIMA(0,0,3,0,0,5,12)
# SARIMA(0,0,3,5,0,0,12) ; SARIMA(2,0,1,0,0,5,12)
# SARIMA(0,0,1,0,0,5,12) ; SARIMA(0,0,3,4,0,0,12)
# Ces modeles sont bien sur valides et bien ajustes.
# Les modèles ne sont pas imbriques, on ne peut donc pas faire un LR test
# (rapport de vraisemblance).Il nous reste un critere de prevision pour choisir
# entre les 10 modeles. On choisit de comparer les erreurs de prediction
# pour selectionner un unique modele.

T           = length(xmlog)
trend       = 1:(T-4)
xmlog_tronq = xmlog[trend]
obs         = xmlog[(T-3):T]
lt          = lm(xmlog[trend] ~ poly(trend,2))

# Ici on rebascule sur xmlog, on doit donc le prendre en compte : d = 1, D = 1
# faisant reference a notre differenciation et desaisonnalisation.
sarima1  = arima(xmlog_tronq, order=c(0,1,3), seasonal=list(order = c(4,1,2), period=12), method="ML", optim.control=list(maxit=20000))
sarima2  = arima(xmlog_tronq, order=c(2,1,1), seasonal=list(order = c(0,1,1), period=12), method="ML", optim.control=list(maxit=20000))
sarima3  = arima(xmlog_tronq, order=c(0,1,1), seasonal=list(order = c(4,1,2), period=12), method="ML", optim.control=list(maxit=20000))
sarima4  = arima(xmlog_tronq, order=c(0,1,3), seasonal=list(order = c(2,1,4), period=12), method="ML", optim.control=list(maxit=20000))
sarima5  = arima(xmlog_tronq, order=c(0,1,0), seasonal=list(order = c(2,1,4), period=12), method="ML", optim.control=list(maxit=20000))
sarima6  = arima(xmlog_tronq, order=c(0,1,3), seasonal=list(order = c(0,1,5), period=12), method="ML", optim.control=list(maxit=20000))
sarima7  = arima(xmlog_tronq, order=c(0,1,3), seasonal=list(order = c(5,1,0), period=12), method="ML", optim.control=list(maxit=20000))
sarima8  = arima(xmlog_tronq, order=c(2,1,1), seasonal=list(order = c(0,1,5), period=12), method="ML", optim.control=list(maxit=20000))
sarima9  = arima(xmlog_tronq, order=c(0,1,0), seasonal=list(order = c(0,1,5), period=12), method="ML", optim.control=list(maxit=20000))
sarima10 = arima(xmlog_tronq, order=c(0,1,3), seasonal=list(order = c(4,1,0), period=12), method="ML", optim.control=list(maxit=20000))
```

```
modelComp = list(sarima1, sarima2, sarima3, sarima4, sarima5, sarima6, sarima7, sarima8, sarima9,
                 sarima10)

# On introduit la fonction rmserror qui calcule la somme des carres des
# residus sur l'ensemble de la periode tronquee.

rmserror <- function(model) sqrt(sum(model$residuals^2))

# Et on les calcule.

i = 0
predmodel = rep(list(list(),10))
erreur = c("SARIMA(0,0,3,4,0,2,12)minAIC"= 0,
           "SARIMA(2,0,1,0,0,1,12)minBIC"= 0,
           "SARIMA(0,0,1,4,0,2,12)"      = 0, "SARIMA(0,0,3,2,0,4,12)" = 0,
           "SARIMA(0,0,1,2,0,4,12)"      = 0, "SARIMA(0,0,3,0,0,5,12)" = 0,
           "SARIMA(0,0,3,5,0,0,12)"      = 0, "SARIMA(2,0,1,0,0,5,12)" = 0,
           "SARIMA(0,0,1,0,0,5,12)"      = 0, "SARIMA(0,0,3,4,0,0,12)" = 0)

RMSE = erreur

for (model in modelComp){
  i = i+1
  print(model)

  predmodel[[i]] = as.zoo(ts(predict(model,4)$pred,
                         start     = c(2017,10),
                         end       = c(2018,1),
                         frequency = 12))

  erreur[i]      = sqrt(sum((predmodel[[i]]-obs)^2)/4)
  RMSE[i]        = rmserror(model)
}

# Erreur correspond a la somme des carres des erreurs de prediction
# sur les 4 derniers mois de la serie xmlog pour chaque modele.
# RMSE regroupe les RMSE des differents modeles.

erreur = data.frame(erreur)
colnames(erreur) = c("Erreur prediction")

RMSE = data.frame(RMSE)
colnames(RMSE) = c("RMSE")

resultsPRED = cbind(erreur,RMSE) # resultsPRED regroupe l'ensemble des erreurs
View(resultsPRED)

print(xtable(resultsPRED, type = "latex", tabular.environment="longtable"), file = "resultsPRED.
    tex")

# On remarque que l'erreur de prediction est minimisee par
# le SARIMA(0,0,1,2,0,4,12). Le RMSE est minimise
# par le SARIMA(0,0,3,4,0,2,12), i.e le SARIMA minimisant
# l'AIC. On remarque que l'erreur de prediction a horizon 4 periodes
# est plus petite pour le SARIMA minimisant le BIC. Cependant
# le RMSE de ce meme modele est plus faible.
# On a don d'un cote un modele minimisant l'AIC et
# qui fit mieux la serie sur les donnees observees (en fait,
# parmi les modeles retenus, il fit LE mieux la serie xmlog) mais predit moins bien.
# De l'autre, un autre modele minimisant le BIC et
# predit mieux mais qui fit moins bien la serie sur les donnees observees.
# Enfin, il existe 2 autres modeles ne minimisant aucun des 2 criteres
# mais qui pourtant predisent mieux a horizon 4 periodes.
```

```
# On affiche les erreurs de prediction.

pred = plot(dates[(T-3):T],
            obs,
            type = 'o',
            xlab = "Temps",
            ylim = c(1.7,2),
            ylab = 'Logarithme de la fréquentation des passagers (millions)',
            col  = 'black',
            main = 'Prévision',
            xaxt = "n")

axis(side=1, at=dates[(T-3):T], labels=format(dates[(T-3):T], '%b-%y'))

lines(predmodel[[1]],                         type = "o", col = "red",      lty = 1, lwd = 2)
lines(dates[(T-3):T], predmodel[[2]],  type = "o", col = "blue",     lty = 1, lwd = 2)
lines(dates[(T-3):T], predmodel[[3]],  type = "o", col = "green",    lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[4]],  type = "o", col = "gray",     lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[5]],  type = "o", col = "orange",   lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[6]],  type = "o", col = "darkblue", lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[7]],  type = "o", col = "cyan",     lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[8]],  type = "o", col = "brown",    lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[9]],  type = "o", col = "yellow",   lty = 1, lwd = 1)
lines(dates[(T-3):T], predmodel[[10]], type = "o", col = "pink",     lty = 1, lwd = 1)

legend(y          = 2.0,
       x          = dates[[T-1]],
       bty        = "n",
       legend     = c("Valeurs observées","SARIMA(0,0,3,4,0,2,12) minimisateur AIC",
                      "SARIMA(2,0,1,0,0,1,12) minimisateur BIC",
                      "SARIMA(0,0,1,4,0,2,12)", "SARIMA(0,0,3,2,0,4,12)",
                      "SARIMA(0,0,1,2,0,4,12)", "SARIMA(0,0,3,0,0,5,12)",
                      "SARIMA(0,0,3,5,0,0,12)" , "SARIMA(2,0,1,0,0,5,12)",
                      "SARIMA(0,0,1,0,0,5,12)", "SARIMA(0,0,3,4,0,0,12)"),

       col        = c("black","red", "blue", "green", "gray", "orange", "darkblue", "cyan",
                      "brown", "yellow", "pink"),

       text.col   = c("black","red", "blue", "green", "gray", "orange", "darkblue", "cyan",
                      "brown", "yellow", "pink"),
       lty        = c(1,1,1,1,1,1,1,1,1,1,1),
       pch        = c(NA,1,1,1,1,1,1,1,1,1,1),
       y.intersp  = 0.5,
       ncol       = 1)
```

### Partie 3

```
##################################################
################ Predictions ###################
##################################################

# Modele retenu

model = arima(xmlog, order=c(0,1,3), seasonal=list(order = c(4,1,2), period=12), method="ML",
    optim.control=list(maxit=20000))

# On realise une prevision sur 15 mois
lag        = 15
lastobs    = xmlog[T]
forecasts  = forecast(model, h=lag, level=95)
prediction = as.zoo(ts(c(lastobs,forecasts$mean),  frequency = 12, start=c(2018,1)))
up         = as.zoo(ts(c(lastobs,forecasts$upper), frequency = 12, start=c(2018,1)))
down       = as.zoo(ts(c(lastobs,forecasts$lower), frequency = 12, start=c(2018,1)))
datesPRED  = seq(as.POSIXct("2018-02-01"), by = "month", length.out = lag+1)

plot(c(dates[(T-48):T],datesPRED),
     c(xmlog[(T-48):T],as.zoo(ts(rep(NA,length(datesPRED)), start = c(2018,2),frequency = 12))),
     type = 'l',
     xlab = '',
     ylab = 'Logarithme de la fréquentation des passagers (millions)',
     col  = "black",
     ylim = c(1.5,2.25),
     xaxt = "n",
     main = paste0("Prévision à horizon ",lag," mois"))

axis(side=1, at=c(dates[(T-48):T],datesPRED), labels=format(c(dates[(T-48):T],datesPRED),
     '%b-%y'), las=2)

lines(dates[(T-48):T], fitted(model)[(T-48):T] , type = "o", col = "red",   lty = 1, lwd = 1)
lines(datesPRED, up                             , type = "o", col = "green", lty = 2, lwd = 1)
lines(datesPRED, down                           , type = "o", col = "green", lty = 2, lwd = 1)
lines(datesPRED, prediction                     , type = "o", col = "blue",  lty = 1, lwd = 1)

legend("toplef",
       bty      = "n",
       legend   = c("Série observée", "Modèle SARIMA", "Prédiction",
                    "Borne supérieures et inférieures à 95%"),
       col      = c("black","red", "blue", "green"),
       text.col = c("black","red", "blue", "green"),
       lty      = c(1,1,1,2),
       pch      = c(NA,1,1,1,1),
       ncol     = 1,
       y.intersp = 0.5)

# On verifie bien que l'intervalle de confiance grandit selon que la
# prevision se fait a horizon plus large.

plot(datesPRED,
     up-down,
     type = 'o',
     xlab = '',
     ylab = "Largeur de l'intervalle de confiance a 95% (millions de passager)",
     col  = "red",
     xaxt = "n",
     main = "Largeur de l'intervalle de confiance a 95%")

axis(side=1, at=datesPRED, labels=format(datesPRED, '%b-%y'), las=2)
```