

# Rapport : Séries Temporelles

EMSBD6 - Bruno KUBECZKA

7 juin 2023

## Abstract

Cette étude a pour objet la mise en application des techniques de modélisation des séries temporelles. Elle propose de se pencher sur l'évolution du nombre de postes vacants en France de 1989 à 2023 en identifiant les modèles les plus adaptés à la prédiction d'une part et en étudiant le comportement des modèles face aux années COVID d'autre part.

## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Présentation des données . . . . .	4
1.2	Objet de l'étude . . . . .	5
<b>2</b>	<b>Analyse exploratoire</b>	<b>7</b>
2.1	Chonogramme de la série étudiée . . . . .	7
2.2	Saisonnalité : Monthplot . . . . .	7
2.3	Saisonnalité : Lag-plot . . . . .	8
2.4	Modélisation additive . . . . .	9
2.4.1	Tendance générale par moyenne mobile . . . . .	9
2.4.2	Saisonnalité . . . . .	10
2.4.3	Bruit résiduel . . . . .	12
2.4.4	Décomposition automatique . . . . .	12
2.5	Modélisation multiplicative . . . . .	13
2.5.1	Tendance générale par moyenne mobile . . . . .	13
2.5.2	Saisonnalité . . . . .	14
2.5.3	Bruit résiduel . . . . .	16
2.5.4	Décomposition automatique . . . . .	16
2.6	Conclusion . . . . .	17

<b>3</b>	<b>Lissage Exponentiel</b>	<b>18</b>
3.1	Modélisation sur la série d'entraînement (Jan 2009 - Déc 2018)	18
3.1.1	Modèles candidats	18
3.1.2	Modèle MAM	19
3.1.3	Modèle MMM	22
3.1.4	Modèle MNM	25
3.1.5	Modèle suggéré par R	27
3.2	Prédiction de la série test 2019 - Choix du modèle	30
3.3	Entraînement sur la série d'étude entière (Jan 2009 - Dec 2019)	30
3.4	Conclusion	32
<b>4</b>	<b>Modèles ARMA</b>	<b>33</b>
4.1	Stationnarité	33
4.1.1	Chronogramme et corrélogramme	33
4.1.2	Transformation de la série	35
4.1.2.1	Normalité et stabilisation de la variance	35
4.1.2.2	Elimination de la saisonnalité	36
4.1.2.3	Elimination de la tendance	38
4.1.3	Analyse des résidus (série sans tendance ni saisonnalité)	40
4.2	Modélisation des résidus sur la série d'entraînement (Jan 2009 - Déc 2018)	41
4.2.1	ACF et PACF	41
4.2.2	Modèle ARMA(p,q) initial	42
4.2.3	ARMA(2,3)	43
4.2.4	ARMA(2,2)	45
4.2.5	ARMA(2,1)	48
4.2.6	ARMA(2,0) = AR(2)	50
4.2.7	auto-sélection du modèle : armaselect	53
4.2.8	auto-sélection du modèle : auto.arima	53
4.3	Prédiction de la série test 2019 - Choix du modèle ARMA	54
4.4	Entraînement sur la série d'étude entière (Jan 2009 - Dec 2019)	56
4.4.1	ARMA(2,0)=AR(2)	57
4.4.2	auto-sélection du modèle : armaselect	59
4.4.3	auto-sélection du modèle : auto.arima	60
4.5	Conclusion	60

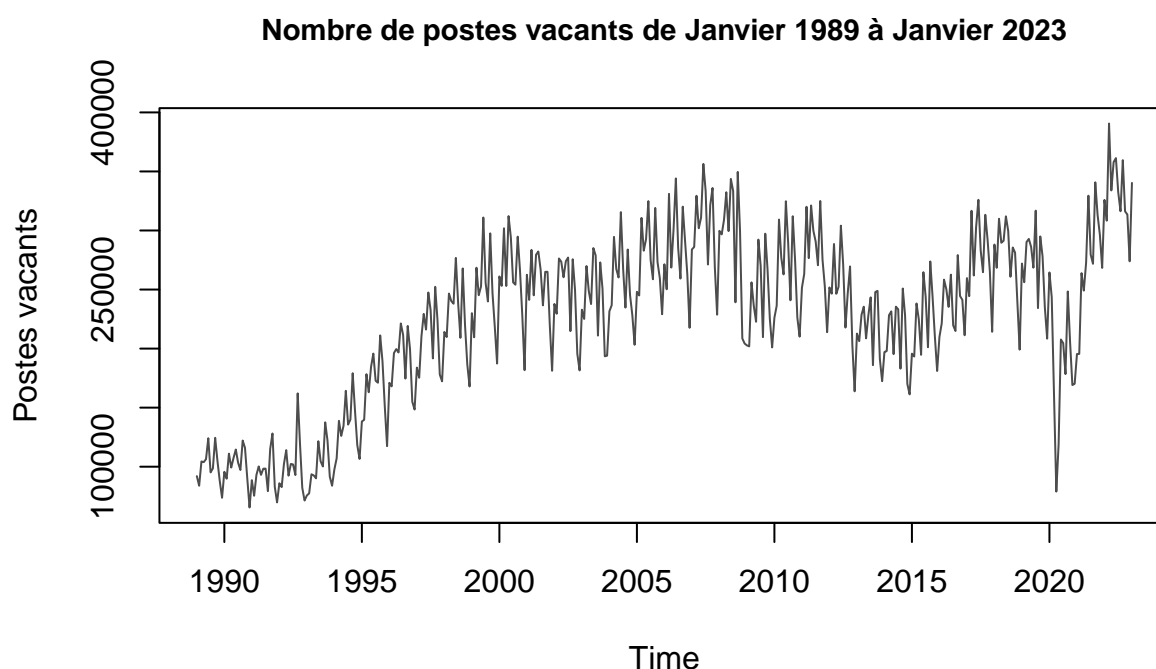
<b>5</b>	<b>Modèles SARIMA</b>	<b>61</b>
5.1	Modélisation sur la série d'entraînement (Jan 2009 - Déc 2018) . . . . .	61
5.1.1	Modèle SARIMA initial . . . . .	61
5.1.2	Auto-sélection du modèle SARIMA . . . . .	63
5.2	Prédiction de la série test 2019 . . . . .	64
5.3	Entraînement sur la série d'étude entière (Jan 2009 - Dec 2019) . . . . .	64
5.3.1	SARIMA(2,1,0)(2,1,0)[12] . . . . .	65
5.3.2	auto-sélection du modèle : auto.arima . . . . .	67
5.3.3	SARIMA(3,1,2)(0,1,2)[12] . . . . .	70
5.3.4	SARIMA(3,1,2)(0,1,1)[12] . . . . .	72
5.3.5	Choix du modèle . . . . .	75
<b>6</b>	<b>Résumé des modèles candidats et comparaison</b>	<b>76</b>
6.1	Comparaison des modèles sur la série temporelle de test (2019) . . . . .	76
6.2	Années COVID - prédiction et comparaison des modèles . . . . .	78
6.2.1	Prédiction de l'année 2020 . . . . .	78
6.2.2	et au-delà... Prédiction à horizon 36 mois . . . . .	80
6.3	Conclusion . . . . .	82
<b>7</b>	<b>Références</b>	<b>83</b>

# 1 Introduction

## 1.1 Présentation des données

Source des données:

- FRED Federal Reserve Economic Data - St Louis Fed : Total new job vacancies, Flow for France (Not Seasonally Adjusted)(Unité : nombre, Fréquence : mensuelle)  
<https://fred.stlouisfed.org/series/LMJVTTNVFRM647N>
- OCDE: <https://stats.oecd.org/> - Registered Unemployed and Job Vacancies for OECD countries



Les données qui nous intéressent sont issues du site de l'OCDE (Organisation de coopération et de développement économiques) ; elles représentent l'évolution du **nombre de postes vacants en France de Janvier 1989 à Janvier 2023**.

Les «postes vacants » (job vacancies), sont des postes libres, nouvellement créés ou inoccupés, ou encore occupés et sur le point de se libérer, pour lesquels des démarches actives sont entreprises pour trouver le candidat convenable. Le recrutement souhaité peut correspondre à un contrat à durée indéterminée (CDI), un contrat à durée déterminée (CDD), ou à un emploi saisonnier, même de courte durée.

Les données couvrent les entreprises de 10 salariés ou plus du champ privé, situés en France métropolitaine et dans les départements-régions d'Outremer (Drom) hors Mayotte. Sont exclus du champ l'agriculture, l'intérim, les particuliers employeurs et les emplois publics.

Le nombre de postes vacants qui fait partie des données dont l'objectif est d'estimer le niveau et la répartition de la demande de main-d'oeuvre non satisfaite, est à opposer au taux de chômage qui représente les offres de main-d'oeuvre non satisfaite.

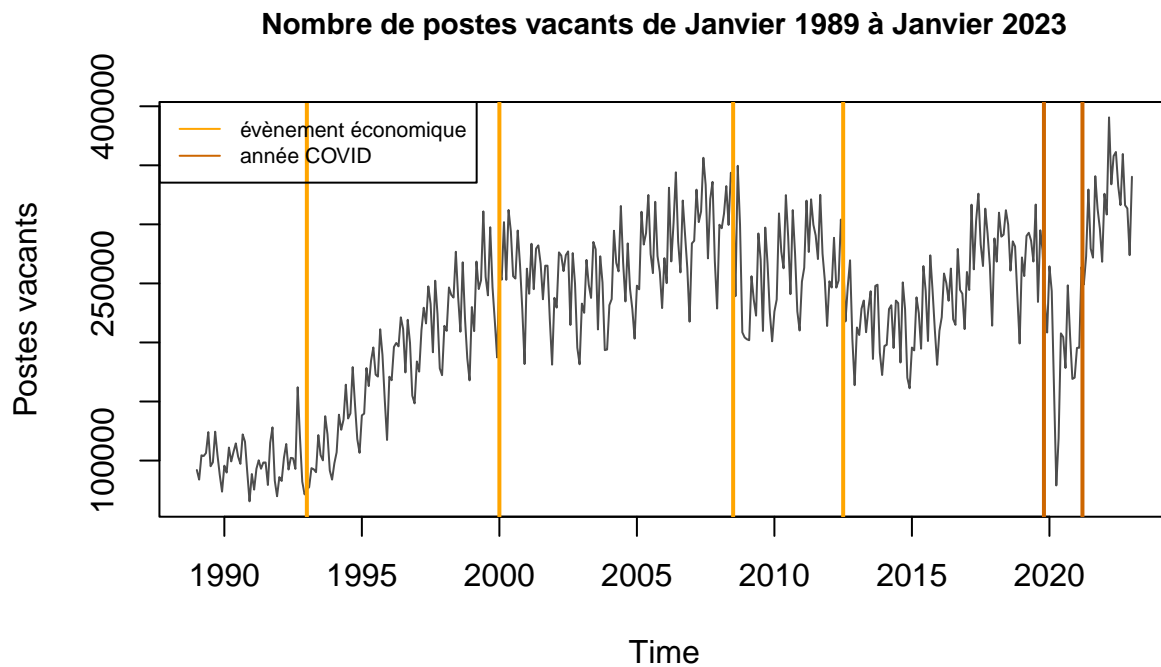
Les données issues de l'OCDE via la FRED (Federal Reserve Economic Data) sont fournies sous la forme d'un **fichier CSV** ; elles compilent les **données mensuelles** depuis **Janvier 1989**.

Dans le cadre de la mise en application des techniques de modélisation des séries temporelles, ces données présentent l'avantage

- de mettre à disposition un **grand nombre de données**,
- de présenter une **évolution avec tendance et saisonnalité**,
- de présenter des **données “accidentées”** des différentes crises économiques.

## 1.2 Objet de l'étude

Regardons le chronogramme de la série temporelle (de Janvier 1989 à Janvier 2023) pour comprendre l'orientation de l'étude.



Le chronogramme du nombre de postes vacants est jalonné des différents événements économiques impactant le marché du travail :

- Dès **1993** : politique volontariste dédiée à inciter le marché du travail
  - Diminution du coût du travail en 1993, renforcée en 1995-1996.
  - Dispositifs incitant à la réduction du temps de travail de 1996 jusque 2000 (développement du temps partiel, réduction collective des horaires de travail)
  - Dispositifs type “Emploi-jeune” en 1997
- **2000** : Le marché de l'emploi ayant été dopé, les incitations imaginées depuis 1993 sont arrêtées progressivement et le marché de l'emploi trouve son rythme de croisière.

- **2008** : crise économique mondiale consécutive à la crise des “subprimes”. La France et la zone euro en général entrent en récession en Janvier 2009. Le marché du travail s’effondre.
- **2013** : fin de la récession en zone Euro ; néanmoins les choix stratégiques européens pour répondre à la crise sclérosent l’économie, amènent des taux de croissance faibles et créent un chômage de masse.
- **2020** : l’année COVID stoppe l’économie mondiale

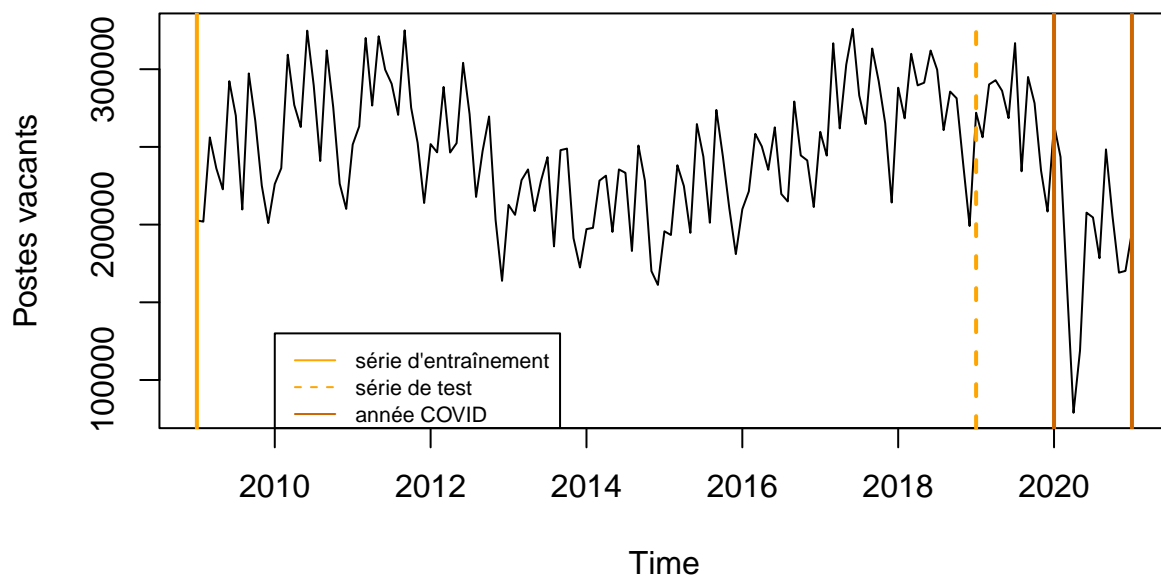
Parmi ces “cassures” économiques, les plus marquées sur le chronogramme sont la crise mondiale de 2008 et surtout l’arrêt brutal de l’économie en Mars 2020.

Si on regarde la situation en mars 2020, on peut légitimement s’interroger si un modèle de la série temporelle valable jusqu’en décembre 2019 peut toujours être considéré comme pertinent en 2020, alors que les valeurs observées sont visiblement accidentées.

Dans le cadre de ce projet, on va s’atteler à répondre à cette question et on propose de procéder de la façon suivante :

- Dans 1er temps, on extraira les données comprises entre les 2 crises majeures, à savoir la **série temporelle de Janvier 2009 à Décembre 2019** ; ce sera la **série d’étude**.
- La **série d’étude** sera alors scindée de la façon suivante
  - une **série d’entraînement** de **Janvier 2009 à Décembre 2018**, qui servira à identifier des modèles candidats à prédire une **série temporelle de test**.
  - l’**année 2019** servira de **série temporelle de test** pour valider les modèles candidats
- On entraînera ensuite les modèles les plus aptes à la prédiction sur l’ensemble de la **série d’étude**.
- Enfin, on prédira les valeurs pour l’année 2020 et au-delà et on statuera sur la pertinence des modèles et de leurs prédictions.

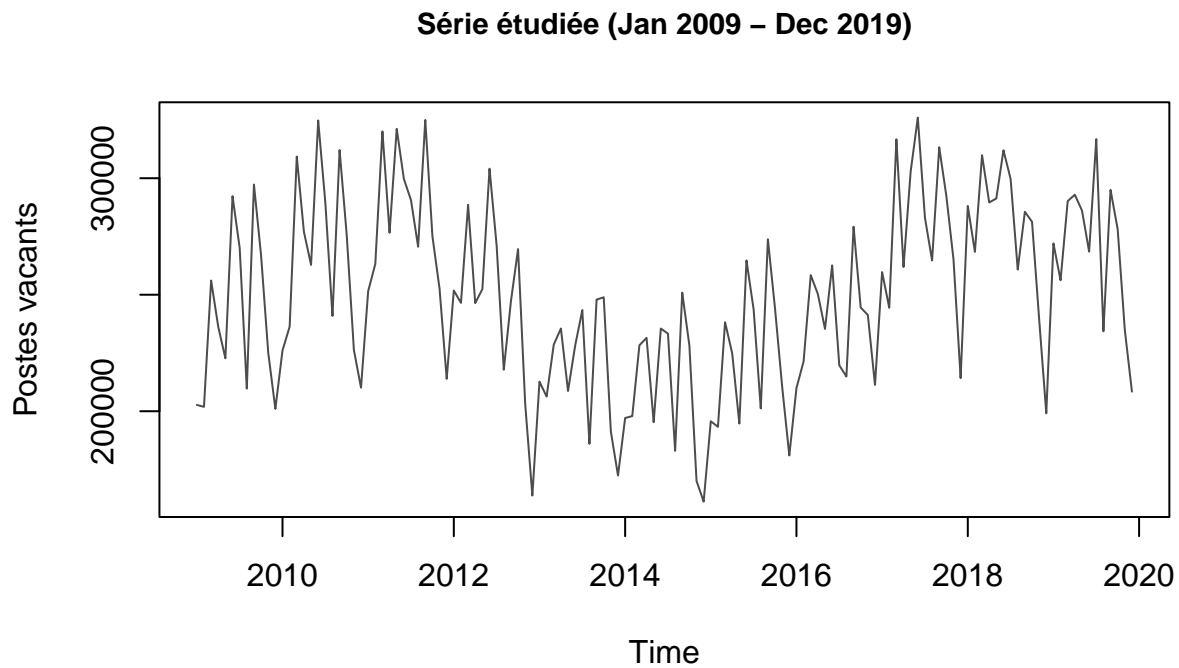
## Série d'étude + année 2020



## 2 Analyse exploratoire

L'analyse exploratoire propose une 1ère approche des caractéristiques de la série d'étude de Janvier 2009 à Décembre 2019.

### 2.1 Chonogramme de la série étudiée



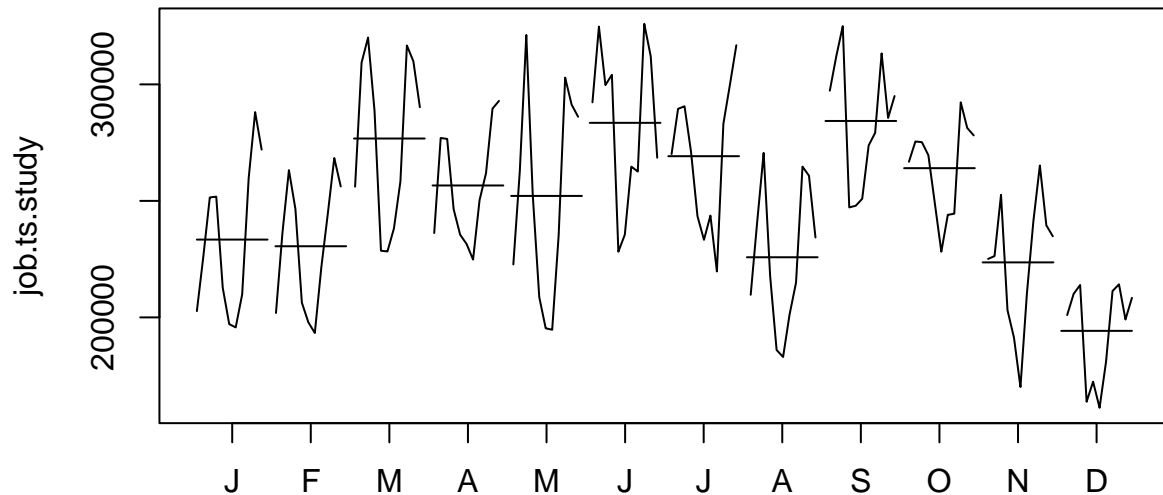
En 1ère analyse à l'oeil, la série présente

- une tendance globalement stable sur le long terme, mais localement en hausse ou en baisse,
- une saisonnalité marquée,
- une variabilité (raisonnable) des données.

### 2.2 Saisonnalité : Monthplot

Le *monthplot* donne le chronogramme associé à chaque mois de l'année.

```
monthplot(job.ts.study)
```



Ainsi le *monthplot* nous éclaire sur les caractéristiques de la saisonnalité :

- Pour chaque mois, le chronogramme présente une tendance à l'allure similaire, mais à l'amplitude différente
- En observant les moyennes mensuelles, on note
  - 3 pics annuels du nombre de postes vacants : Mars, Juin et Septembre,
  - une baisse importante du nombre de postes vacants de septembre à décembre,
  - un rebond en janvier par rapport au mois de décembre de l'année précédente,
  - une forte baisse du nombre de postes vacants en Août, avant la reprise de septembre.

**Autre point notable** qui pourra avoir de l'importance lors de la modélisation ARMA/SARIMA : Mars, Juin et Septembre sont des mois multiples de 3 ; on peut s'attendre à des corrélations plus ou moins marquées entre la série et la série décalée de 3 voire 6 mois.

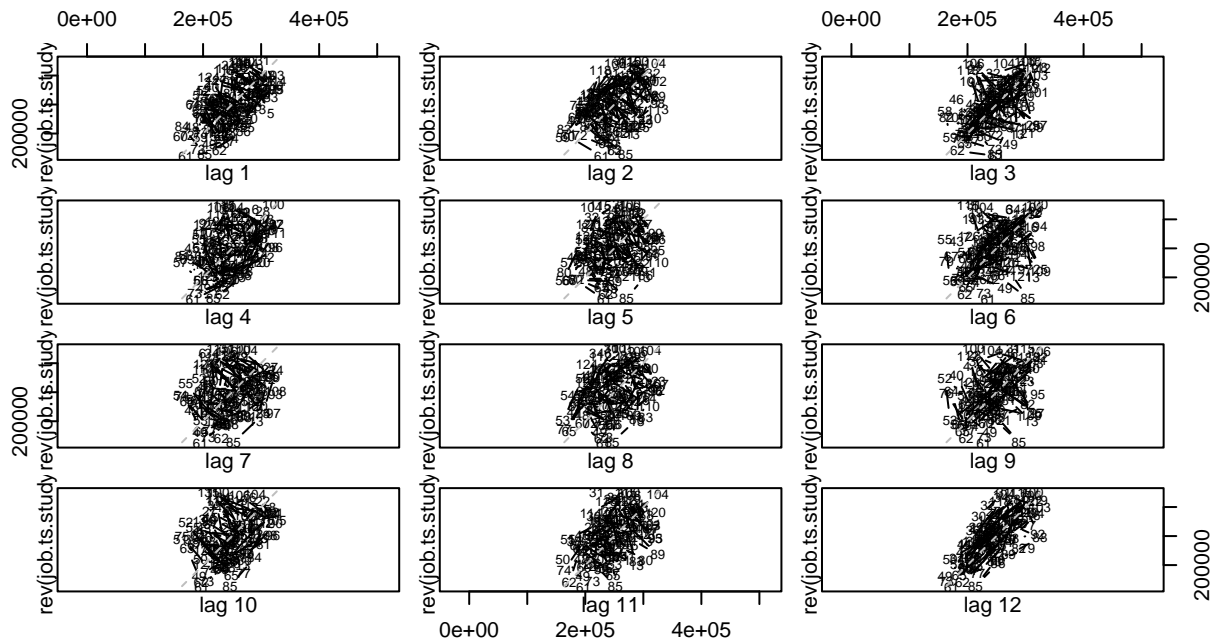
## 2.3 Saisonnalité : Lag-plot

Le lag-plot présente un diagramme mettant en avant les potentielles corrélations entre la série et la série retardée d'autant d'indices que la saisonnalité supposée (ici 12).

L'idée est de confirmer l'ordre de la saisonnalité ainsi que d'éventuelles corrélations à des ordres plus petits.



## Monthplot de la série d'étude (Jan 2009 – Dec 2019)



En l'occurrence, la corrélation la plus forte (i.e. la forme la plus allongée) est obtenue au **retard 12**.

On note aussi une tendance à la linéarité entre la série et la série retardée aux **décalages 3 et 6**, ce qui confirme la tendance du month-plot (une corrélation forte tous les 3 mois)

## 2.4 Modélisation additive

L'idée est de scinder la série en une **tendance générale**, une **saisonnalité**, une **composante résiduelle** combinées de façon additive, à savoir

$$X_t = M_t + S_t + Z_t$$

où

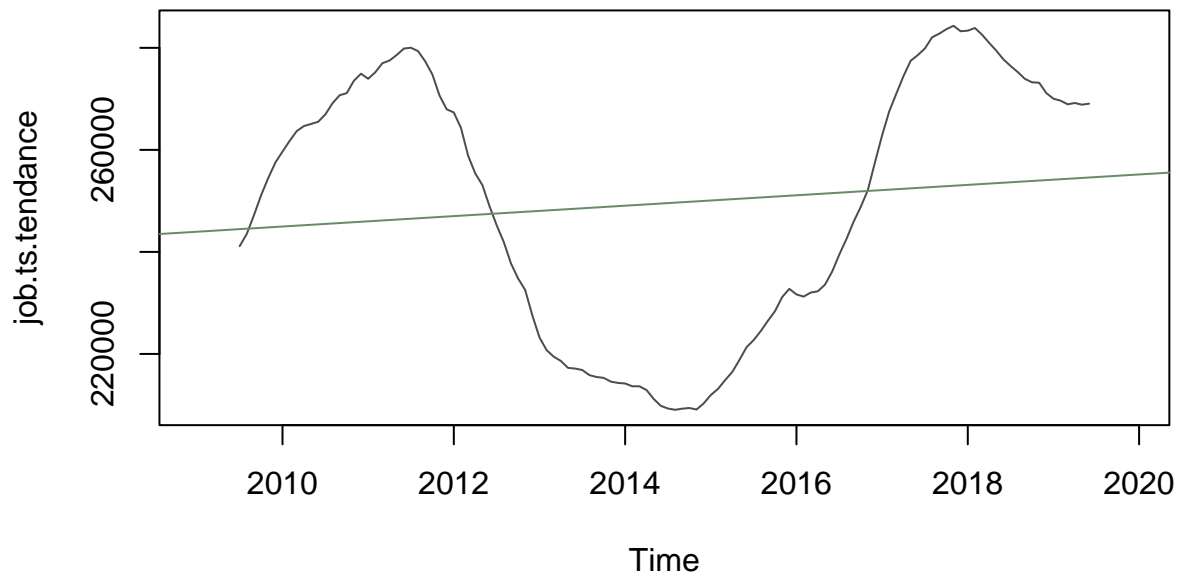
- $M_t$  est la **tendance générale** (évolution à long terme de la série)
- $S_t$  est la **saisonnalité** (schéma périodique)
- $Z_t$  sont les **résidus** que l'on souhaite idéalement de **moyenne nulle** et de **faible variabilité constante**.

### 2.4.1 Tendence générale par moyenne mobile

On procède à une estimation de la tendance par moyennes mobiles d'ordre 12 (saisonnalité supposée), qu'on complète par une régression linéaire simple de 1er ordre pour visualiser la tendance de façon plus explicite.

```
job.ts.tendance <- as.ts(ma(job.ts.study, order=12, centre = TRUE))
```

### Série d'étude – tendance générale par moyenne mobile



La tendance générale est localement fluctuante, à la hausse ou à la baisse.

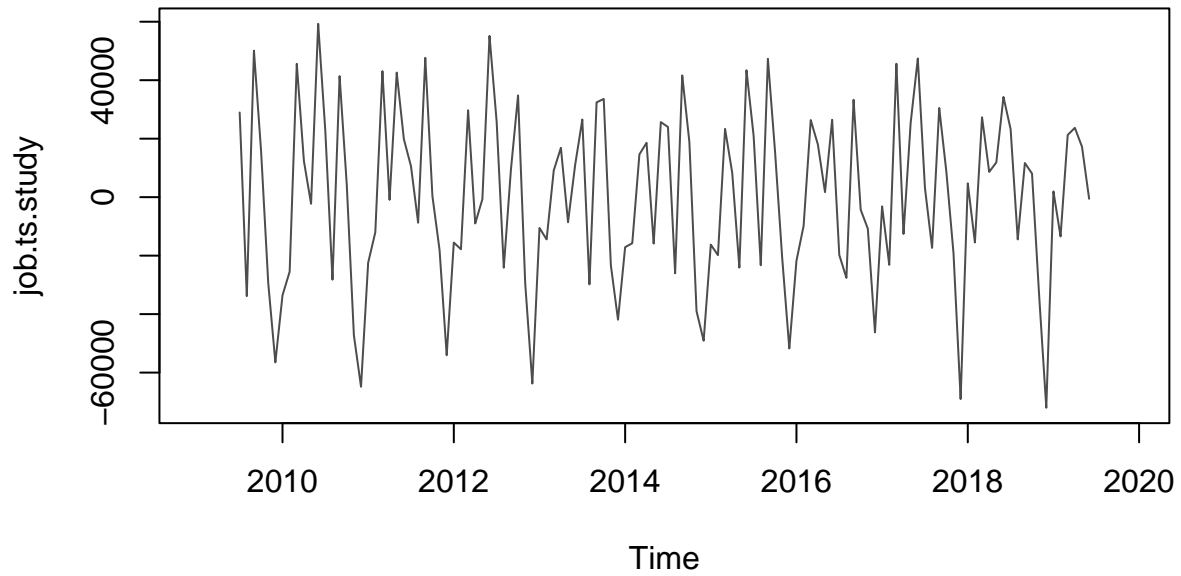
Globalement, elle est légèrement à la hausse sur la période Jan 2009 - Dec 2019.

#### 2.4.2 Saisonnalité

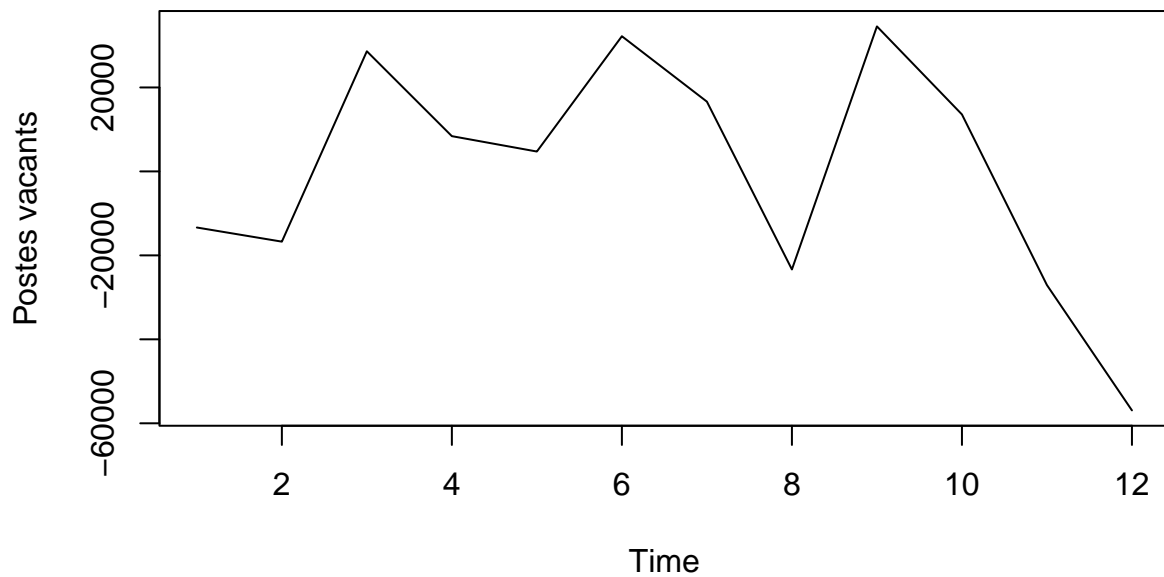
La saisonnalité est obtenue en **diminuant** la série de la tendance générale calculée précédemment.

```
job.ts.add.saisonnalite <- job.ts.study-job.ts.tendance
```

### Série d'étude – Saisonnalité (modèle additif)



### Série d'étude – Saisonnalité moyenne (modèle additif)



On visualise ici la saisonnalité sur l'ensemble de la série d'une part et la saisonnalité moyenne sur les 11 années de la série d'étude.

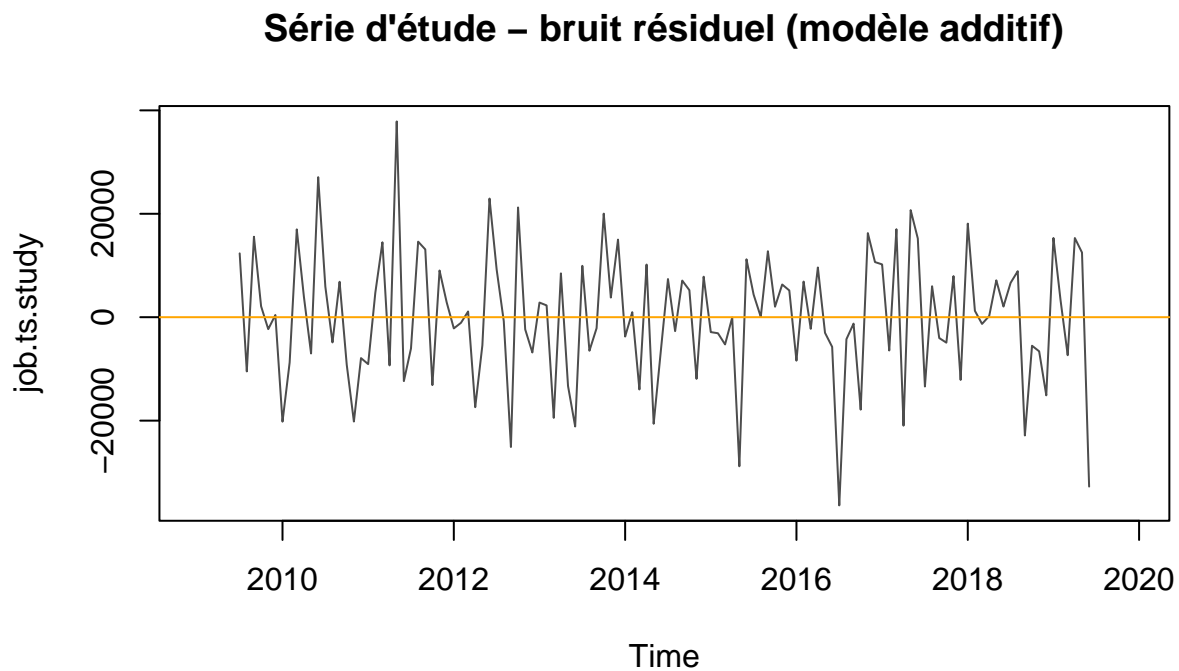
Le schéma imaginé dans le chapitre *monthplot* se confirme soit

- une saisonnalité à 3 pics,
- de fortes baisses en août et décembre,
- une reprise en janvier,

### 2.4.3 Bruit résiduel

Le bruit résiduel est la série d'étude diminuée de la tendance générale et de la saisonnalité moyenne.

```
job.ts.add.random <- job.ts.study - job.ts.tendance - job.ts.add.saisonnalite.mean
```



```
mean(job.ts.add.random, na.rm=TRUE)
```

```
## [1] -3.637683e-13
```

Le bruit résiduel est bien **centré en 0** et présente une **variabilité plutôt faible**, qui pourrait laisser penser à un bruit blanc.

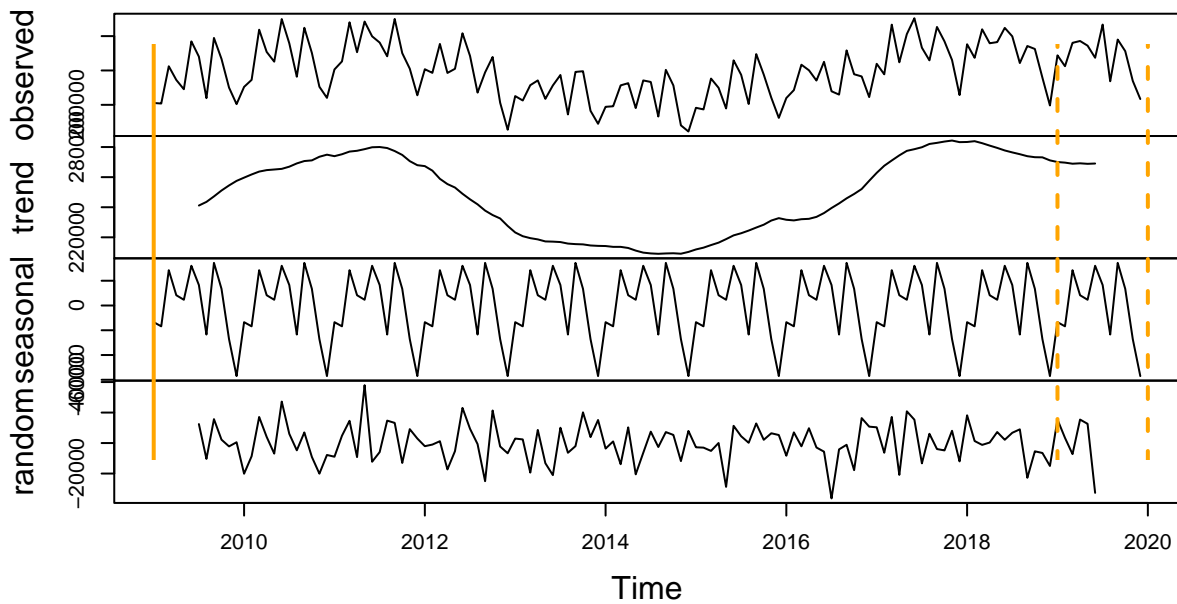
Les différentes modélisations à venir statueront sur la nature du bruit résiduel.

### 2.4.4 Décomposition automatique

On décompose la série d'étude grâce à la fonction *decompose*.

```
decomposition_additive=decompose(job.ts.study, type="additive")
```

## Decomposition of additive time series



La décomposition automatique reprend les graphes introduits dans les chapitres précédents (tendance, saisonnalité, erreur).

La représentation la plus intéressante reste celle de l'erreur.

Au vu du **bruit résiduel centré en 0 et de variance stable**, une **modélisation additive** semble être une **bonne approche de la série**.

## 2.5 Modélisation multiplicative

De la même façon que précédemment, on va décomposer la série en une **tendance générale**, une **saisonnalité** et une part d'**erreur**.

Néanmoins, cette fois-ci nous allons considérer une approche multiplicative de la série, à savoir

$$X_t = M_t * S_t * Z_t$$

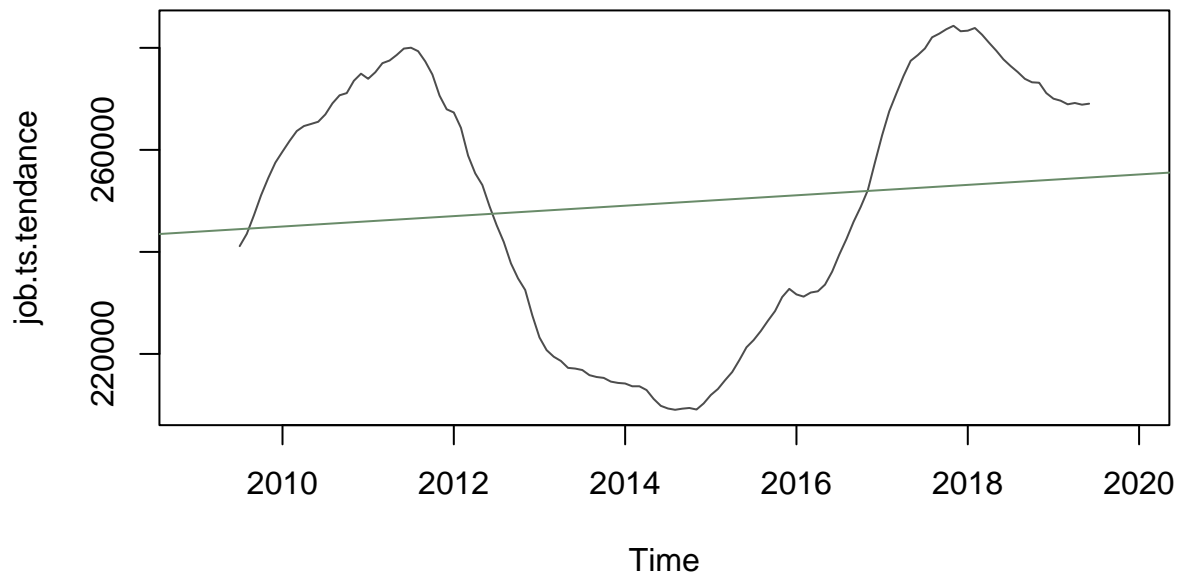
où

- $M_t$  est la **tendance générale** (évolution à long terme de la série)
- $S_t$  est la **saisonnalité** (schéma périodique)
- $Z_t$  sont les **résidus** que l'on souhaite idéalement **centré en 1** et de **faible variabilité constante**.

### 2.5.1 Tendence générale par moyenne mobile

On reprend l'estimation de la tendance par moyennes mobiles d'ordre 12 (saisonnalité supposée) calculée dans le modèle additif.

### Série d'étude – tendance générale par moyenne mobile

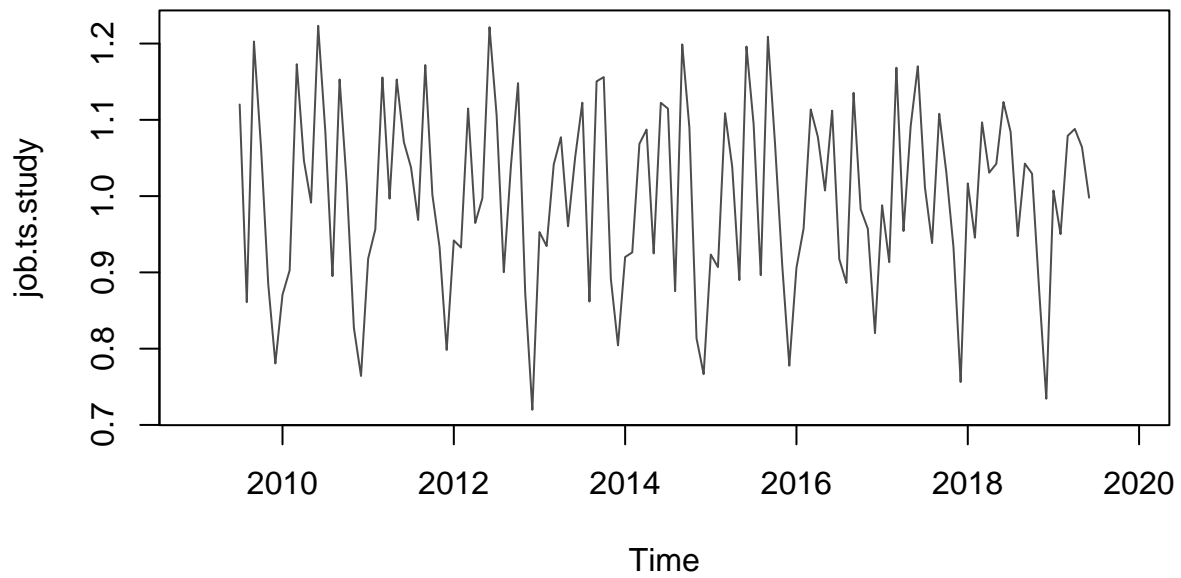


#### 2.5.2 Saisonnalité

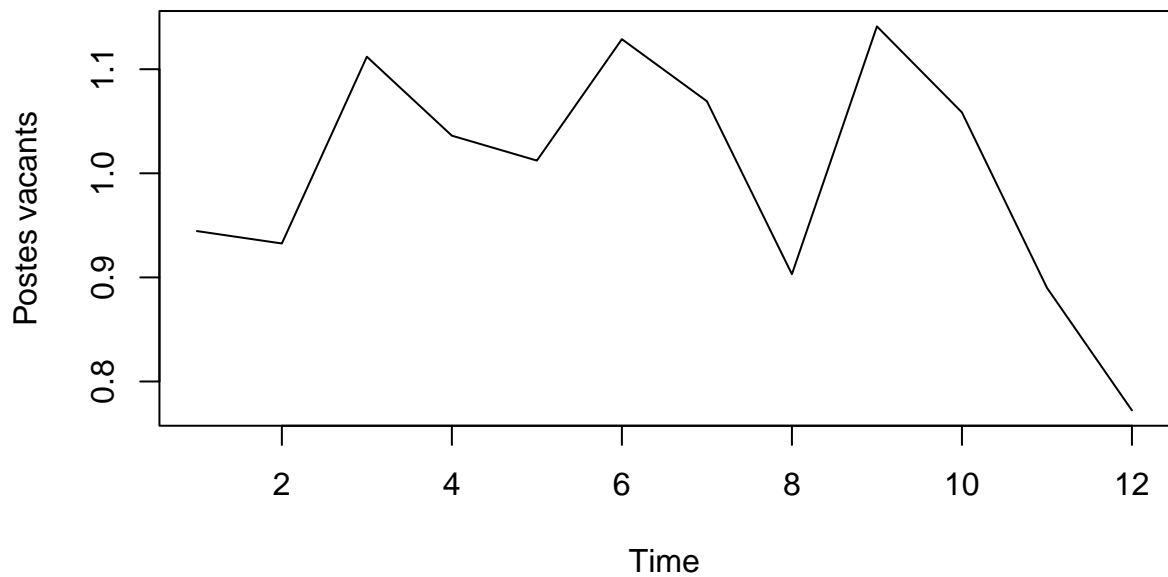
La saisonnalité est obtenue en **divisant** la série par la tendance générale calculée précédemment.

```
job.ts.mult.saisonnalite <- job.ts.study/job.ts.tendance
```

### Série d'étude – Saisonnalité (modèle multiplicatif)



### Série d'étude – Saisonnalité moyenne (modèle multiplicatif)



On visualise ici la saisonnalité sur l'ensemble de la série d'une part et la saisonnalité moyenne sur les 11 années de la série d'étude.

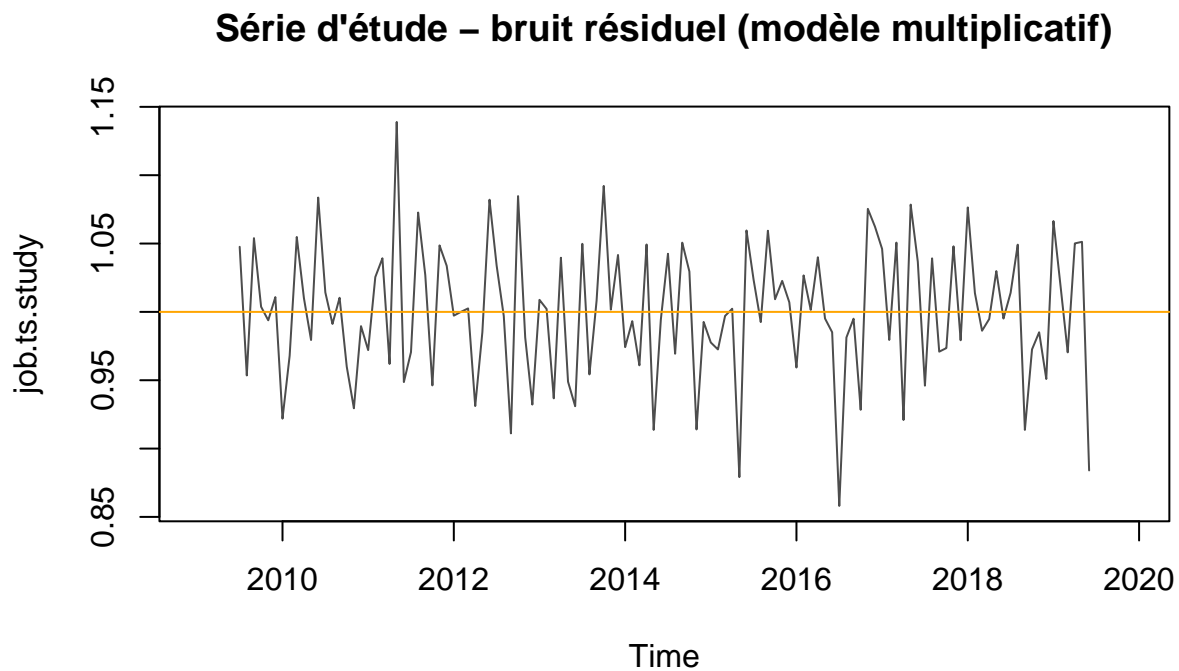
Le schéma imaginé dans le chapitre *monthplot* est bien

- une saisonnalité à 3 pics,
- de fortes baisses en août et décembre,
- une reprise en janvier,

### 2.5.3 Bruit résiduel

Le bruit résiduel est la série d'étude divisée par la tendance générale puis la saisonnalité moyenne.

```
job.ts.mult.random <- (job.ts.study / job.ts.tendance) / job.ts.mult.saisonnalite.mean
```



```
mean(job.ts.mult.random, na.rm=TRUE)
```

```
## [1] 1
```

Le bruit résiduel est bien **centré en 1** et présente une **variabilité plutôt faible et stable**, qui pourrait laisser penser à un bruit blanc.

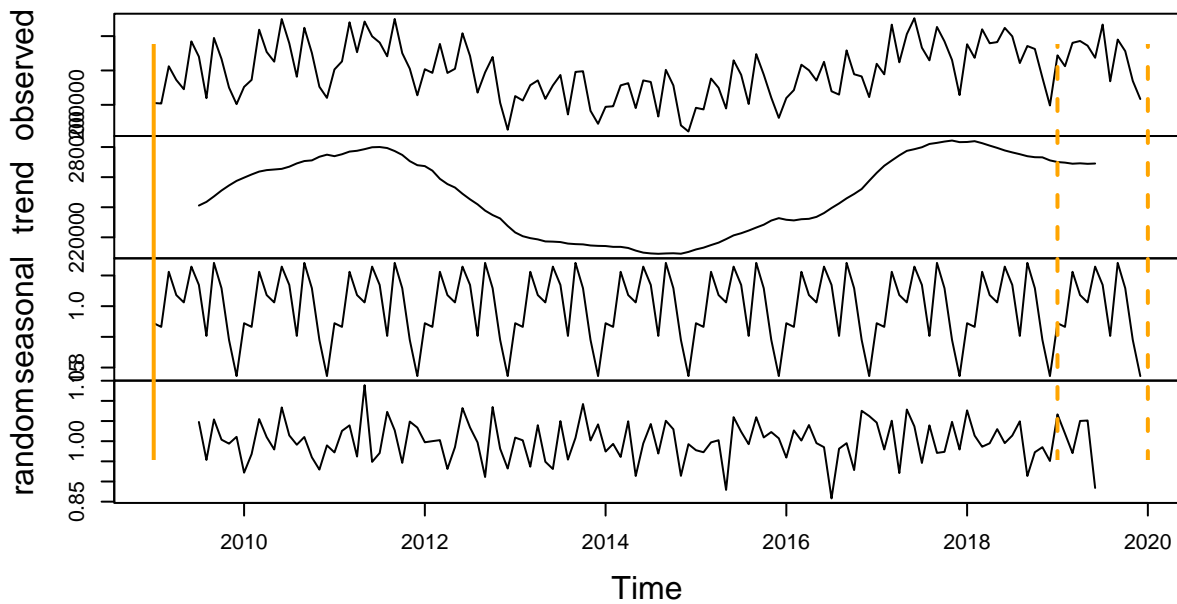
### 2.5.4 Décomposition automatique

On décompose la série d'étude grâce à la fonction *decompose*.

```
decomposition_multiplicative=decompose(job.ts.study, type="multiplicative")
```



## Decomposition of multiplicative time series



La décomposition automatique reprend les graphes introduits dans les chapitres précédents (Tendance, saisonnalité, erreur).

Au vu du **bruit résiduel centré en 1 et de variance stable**, une **modélisation multiplicative** semble aussi être une **bonne approche de la série**.

## 2.6 Conclusion

L'analyse exploratoire fait apparaître une série à **tendance légèrement croissante** et une **saisonnalité à 12 mois** avec de potentielles corrélations à 3 mois.

Les 2 modélisations descriptives amènent toutes les 2 à des **résidus centrés et de faible variance constante à l'oeil** ; il est difficile de statuer sur une meilleure approche entre la méthode additive et multiplicative. La recherche du meilleur modèle de lissage exponentiel devrait apporter un éclairage sur ce point.

### 3 Lissage Exponentiel

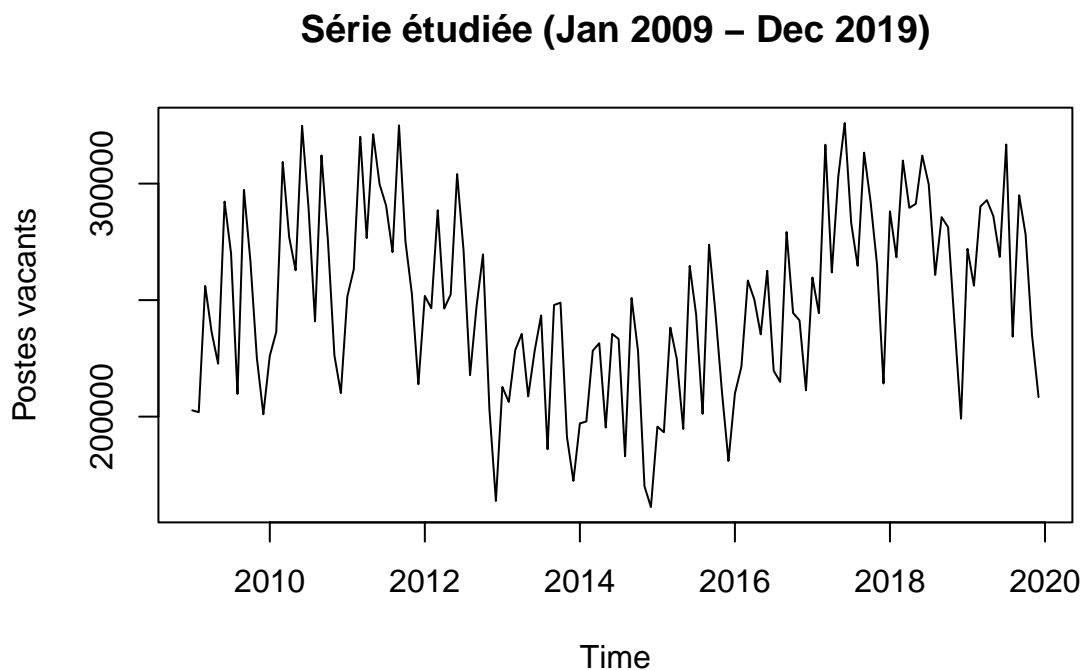
Ce chapitre a pour objet la modélisation par lissage exponentiel de la série d'étude.

Il s'agira

- d'identifier sur la série d'entraînement des modèles de lissage exponentiel candidats à la prédiction
- de repérer automatiquement le modèle le plus performant selon l'algorithme de la fonction *ets*
- de choisir le meilleur modèle selon les critères AIC, AICc et BIC

#### 3.1 Modélisation sur la série d'entraînement (Jan 2009 - Déc 2018)

##### 3.1.1 Modèles candidats



Afin de déterminer "à l'oeil" le meilleur modèle, on note les points suivants :

- la saisonnalité est bien présente ; néanmoins son amplitude est fluctuante au fil du temps (plus faible dans les années 2013-2015, plus forte sinon. On la prendra de préférence **multiplicative**).
- on prendra le parti de choisir une **erreur multiplicative** en concordance
- Le cas de la tendance est plus complexe ; sa fluctuation locale à la hausse et à la baisse ne permet pas de déterminer "à l'oeil" sa nature. On prend le parti de modéliser les cas de figure suivants
  - une tendance localement linéaire qu'on choisira additive et multiplicative (dans les 2 cas, on laissera l'ajustement déterminé si un amorti est requis)

- un modèle sans tendance, dans l’hypothèse où la meilleure modélisation repose sur une “simple” mise à jour du niveau.

Les modèles “candidats” seront donc les suivants:

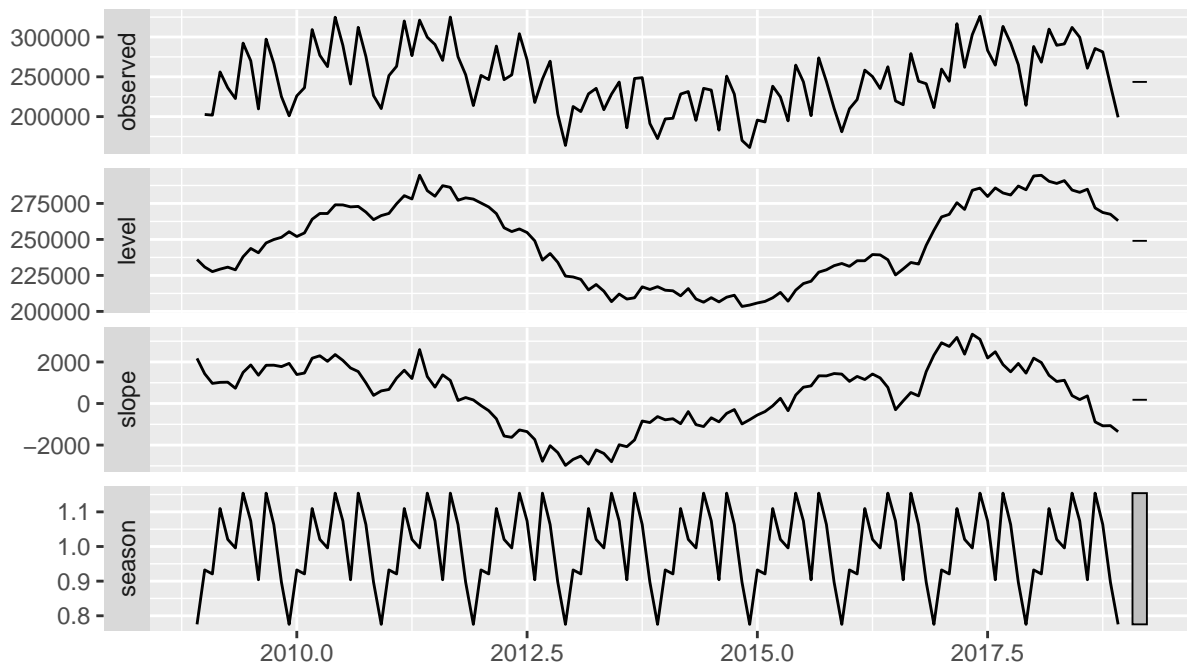
- **MAM** : Erreur Multiplicative, Tendance Additive (possiblement avec amorti), Saisonnalité Multiplicative
- **MMM** : Erreur Multiplicative, Tendance Multiplicative (possiblement avec amorti), Saisonnalité Multiplicative
- **MNM** : Erreur Multiplicative, sans tendance (N), Saisonnalité Multiplicative

### 3.1.2 Modèle MAM

```
modele.le.MAM.train <- ets(job.ts.train, model="MAM")
```

```
## ETS(M,Ad,M)
##
## Call:
## ets(y = job.ts.train, model = "MAM")
##
## Smoothing parameters:
##   alpha = 0.354
##   beta  = 0.0329
##   gamma = 1e-04
##   phi   = 0.9745
##
## Initial states:
##   l = 236075.3754
##   b = 2172.1049
##   s = 0.7753 0.8966 1.0632 1.154 0.904 1.0737
##       1.154 0.9959 1.0205 1.1096 0.9207 0.9324
##
## sigma: 0.0657
##
##      AIC      AICc      BIC
## 2916.624 2923.396 2966.799
##
## Training set error measures:
##               ME  RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -677.7014 15722 12427.8 -0.4198927 4.950247 0.5263952 -0.1310573
```

### Components of ETS(M,Ad,M) method



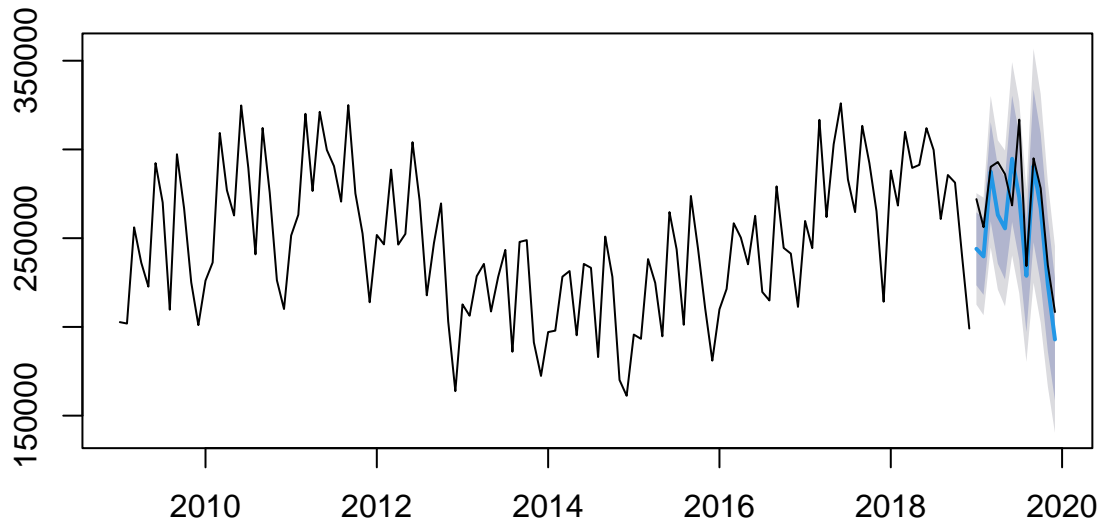
Les paramètres ajustés  $\alpha$ ,  $\beta$  et  $\gamma$  sont relativement faibles :

- La pente et la saisonnalité évoluent peu ( $\beta$  et  $\gamma$  sont presque nuls).
- Seul le niveau évolue sans néanmoins dépendre fortement du passé récent ( $\alpha < 0.4$ ).

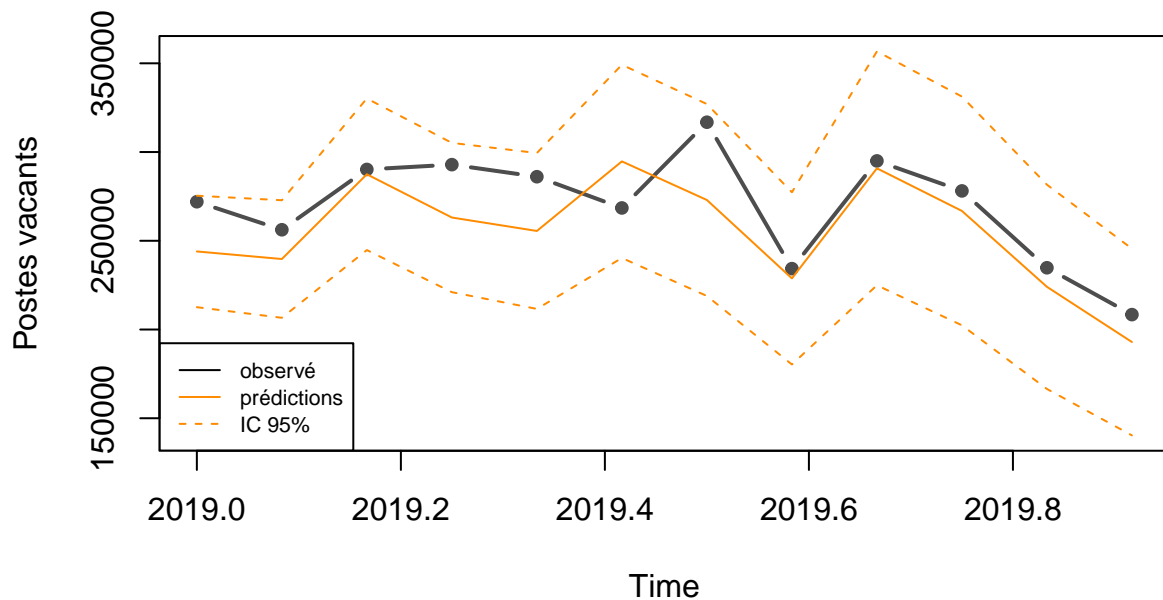
### prévisions à 12 mois

```
predictions.le.MAM.test <- forecast(modele.le.MAM.train, h=12)
```

Lissage exponentiel (M,Ad,M) – prédiction de l'année 2019



Lissage exponentiel (M,Ad,M) – prédiction de l'année 2019



L'ensemble des valeurs observées dans la série de test sont incluses dans l'intervalle de confiance à 95%.

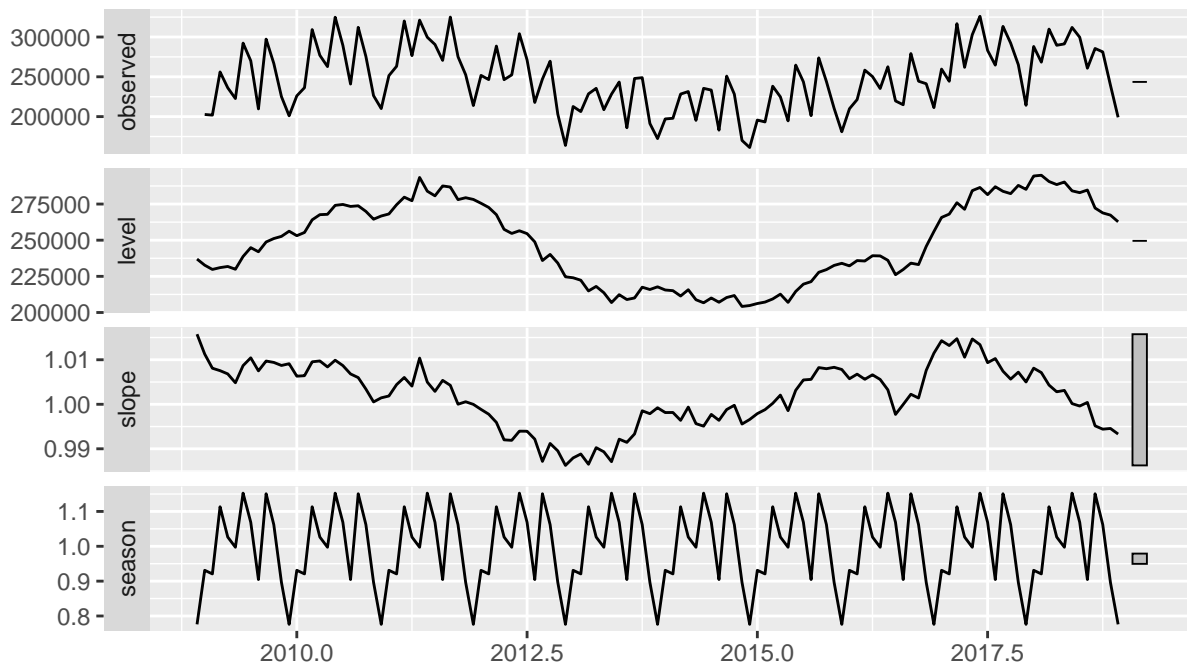
Les prédictions jusqu'en Juillet sont plutôt distantes des valeurs observées mais le modèle offre de bonnes performances dans ses prédictions d'août à décembre.

### 3.1.3 Modèle MMM

```
modele.le.MMM.train <- ets(job.ts.train, model="MMM")
```

```
## ETS(M,Md,M)
##
## Call:
## ets(y = job.ts.train, model = "MMM")
##
## Smoothing parameters:
##   alpha = 0.3428
##   beta  = 0.0408
##   gamma = 1e-04
##   phi   = 0.9665
##
## Initial states:
##   l = 236951.4556
##   b = 1.0157
##   s = 0.776 0.8963 1.062 1.1507 0.9046 1.0688
##         1.1525 0.9972 1.0266 1.1134 0.9207 0.9311
##
## sigma: 0.0654
##
##      AIC      AICc      BIC
## 2916.281 2923.054 2966.456
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -1196.966 15649.08 12363.55 -0.646491 4.934168 0.5236739
##              ACF1
## Training set -0.1363612
```

### Components of ETS(M,Md,M) method



On note que la fonction *ets* a préféré considérer un amorti pour l'ajustement des paramètres (**modèle M, Md, M**).

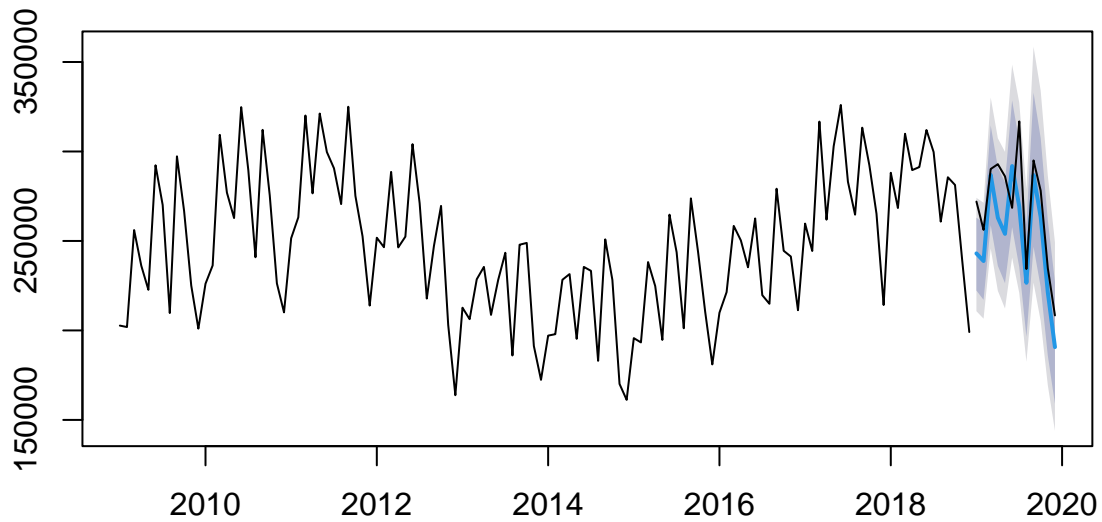
Les paramètres ajustés  $\alpha$ ,  $\beta$  et  $\gamma$  sont à nouveau relativement faibles:

- La pente et la saisonnalité évoluent peu ( $\beta$  et  $\gamma$  sont presque nuls).
- Seul le niveau évolue sans néanmoins dépendre fortement du passé récent ( $\alpha < 0.4$ ).

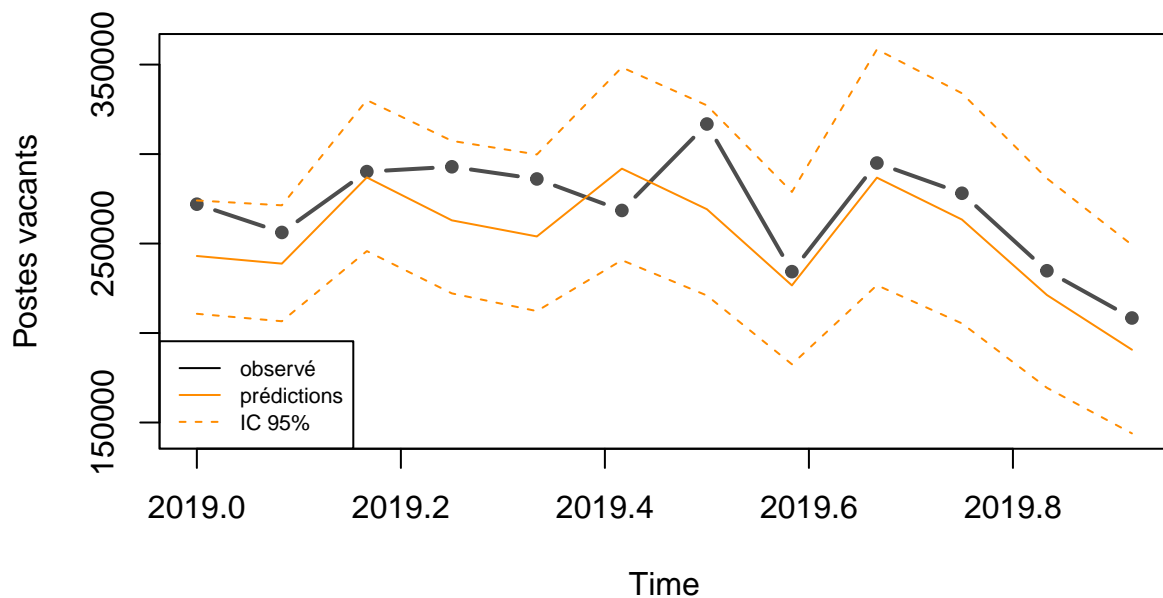
### prévisions à 12 mois

```
predictions.le.MMM.test <- forecast(modele.le.MMM.train, h=12)
```

**Lissage exponentiel (M,Md,M) – prédiction de l'année 2019**



**Lissage exponentiel (M,Md,M) – prédiction de l'année 2019**



L'ensemble des valeurs observées dans la série de test sont incluses dans l'intervalle de confiance à 95%.

Les prédictions sont plutôt distantes des valeurs observées, y compris d'août à décembre compte tenu de la meilleure performance du modèle (M,A,M) précédent.

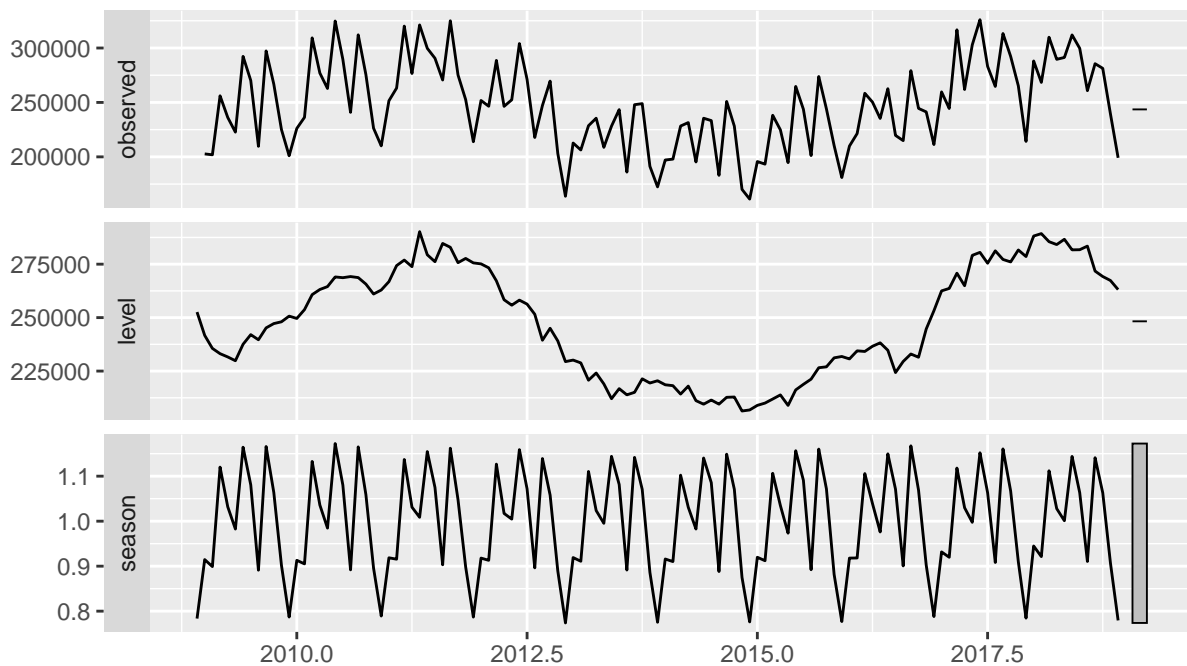


### 3.1.4 Modèle MNM

```
modele.le.MNM.train <- ets(job.ts.train, model="MNM")
```

```
## ETS(M,N,M)
##
## Call:
## ets(y = job.ts.train, model = "MNM")
##
## Smoothing parameters:
##   alpha = 0.3129
##   gamma = 0.1277
##
## Initial states:
##   l = 252600.558
##   s = 0.7836 0.9008 1.0602 1.155 0.8947 1.0723
##       1.1486 0.9856 1.0347 1.1249 0.9083 0.9311
##
## sigma: 0.0709
##
##      AIC      AICc      BIC
## 2932.123 2936.738 2973.935
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 42.34924 16935.15 13053.76 -0.2664633 5.246737 0.5529086
##           ACF1
## Training set -0.03931737
```

Components of ETS(M,N,M) method

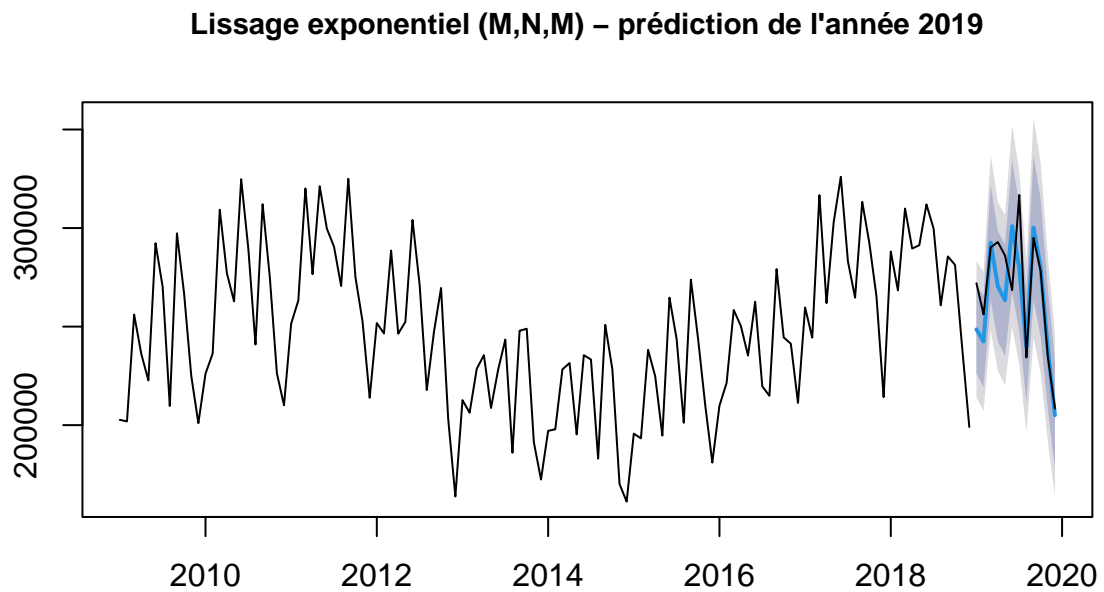


Les paramètres ajustés  $\alpha$  et  $\gamma$  sont relativement faibles:

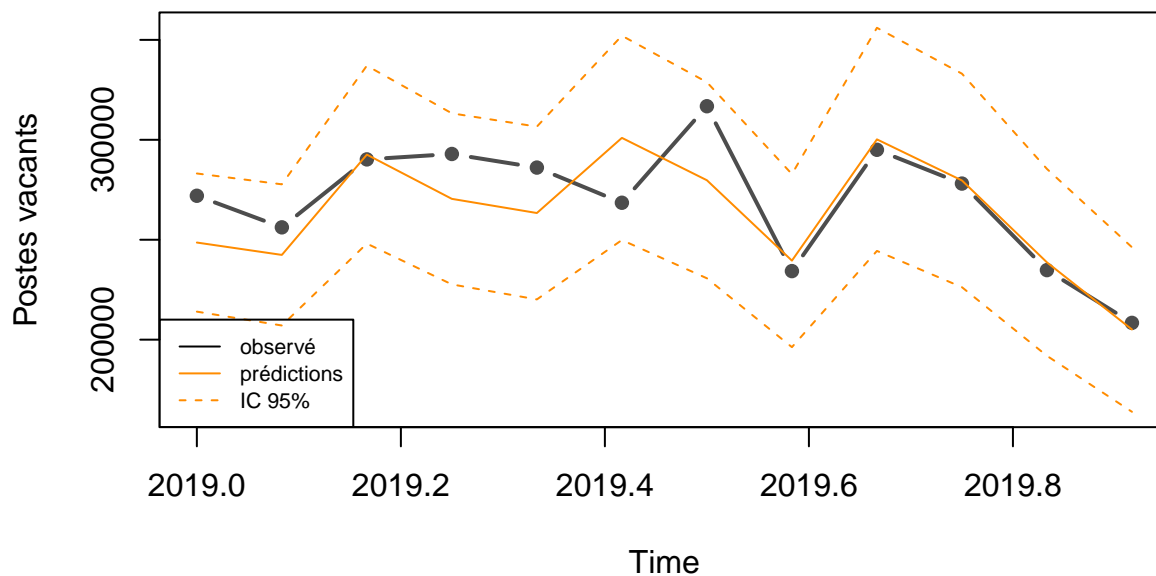
- La saisonnalité évolue peu ( $\gamma$  est presque nul).
- Le niveau évolue sans dépendre fortement du passé récent ( $\alpha < 0.4$ ).

prévisions à 12 mois

```
predictions.le.MNM.test <- forecast(modele.le.MNM.train, h=12)
```



### Lissage exponentiel (M,N,M) – prédiction de l'année 2019



A nouveau, l'ensemble des valeurs observées dans la série de test sont incluses dans l'intervalle de confiance à 95%.

Les prédictions jusqu'en Juillet sont plutôt distantes des valeurs observées.

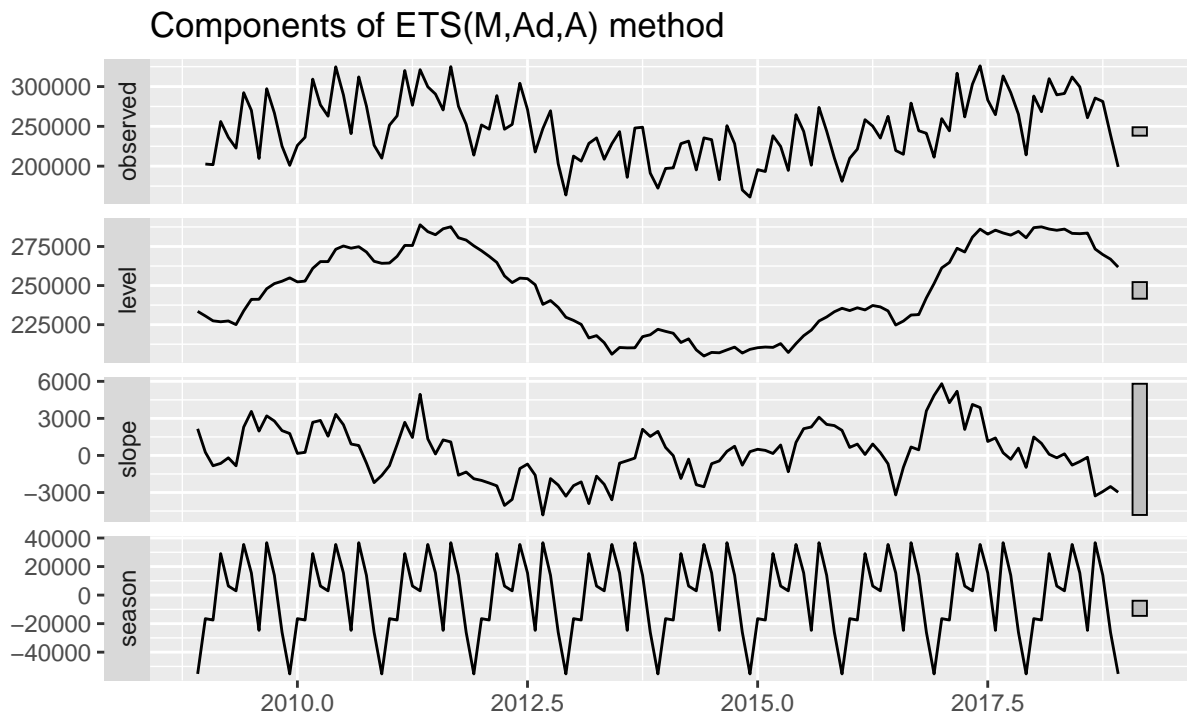
Néanmoins les prédictions d'août à décembre sont les meilleures proposées jusqu'à présent.

#### 3.1.5 Modèle suggéré par R

```
modele.le.ZZZ.train <- ets(job.ts.train, model="ZZZ")
```

```
## ETS(M,Ad,A)
##
## Call:
## ets(y = job.ts.train, model = "ZZZ")
##
## Smoothing parameters:
##   alpha = 0.2914
##   beta  = 0.0913
##   gamma = 1e-04
##   phi   = 0.8
##
## Initial states:
##   l = 233499.6707
##   b = 2154.1245
##   s = -55232.06 -25998.11 13817.32 36723.17 -24710 15516.18
##       35476 2978.602 6336.797 29068.69 -17460.46 -16516.12
##
```

```
## sigma: 0.0651
##
## AIC AICc BIC
## 2913.972 2920.744 2964.147
##
## Training set error measures:
## ME RMSE MAE MPE MAPE MASE
## Training set 98.73945 15075.35 11709.53 -0.2199421 4.725516 0.4959719
## ACF1
## Training set -0.1166306
```



Le modèle estimé comme le meilleur par la fonction *ets* (selon le critère par défaut AICc) est **(M, Ad, A)** c-à-d

- Une erreur multiplicative (M)
- Une tendance additive avec amorti (Ad)
- Une saisonnalité additive (A)

On note que contrairement à nos modèles “à l’oeil”, la fonction *ets* a estimé qu’une **saisonnalité additive** était plus pertinente selon le critère AICc.

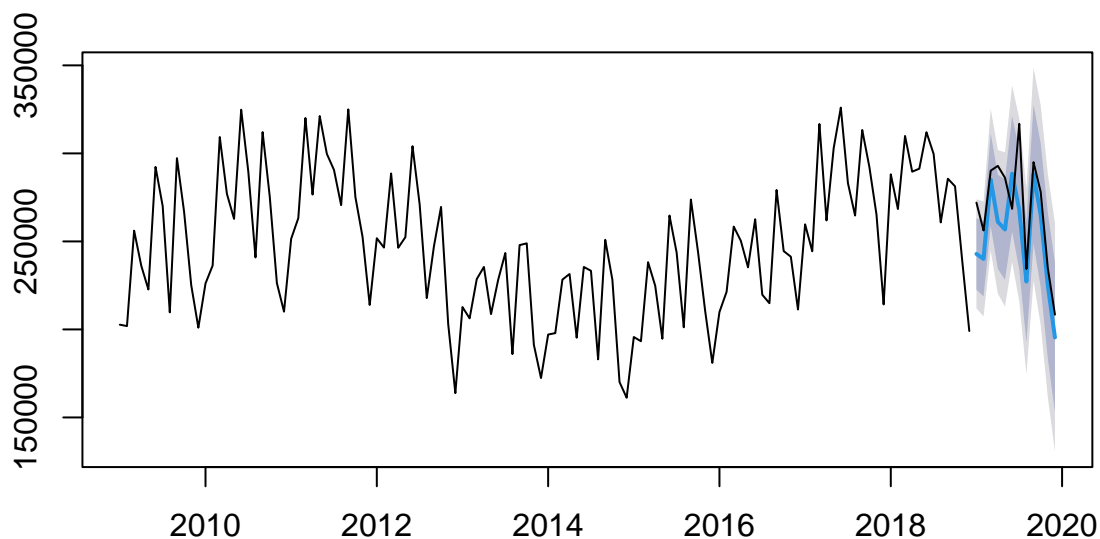
Les paramètres ajustés  $\alpha$ ,  $\beta$  et  $\gamma$  restent faibles.

- La pente et la saisonnalité évoluent peu ( $\beta$  et  $\gamma$  sont presque nuls).
- Seul le niveau évolue sans dépendre fortement du passé récent ( $\alpha < 0.3$ ).

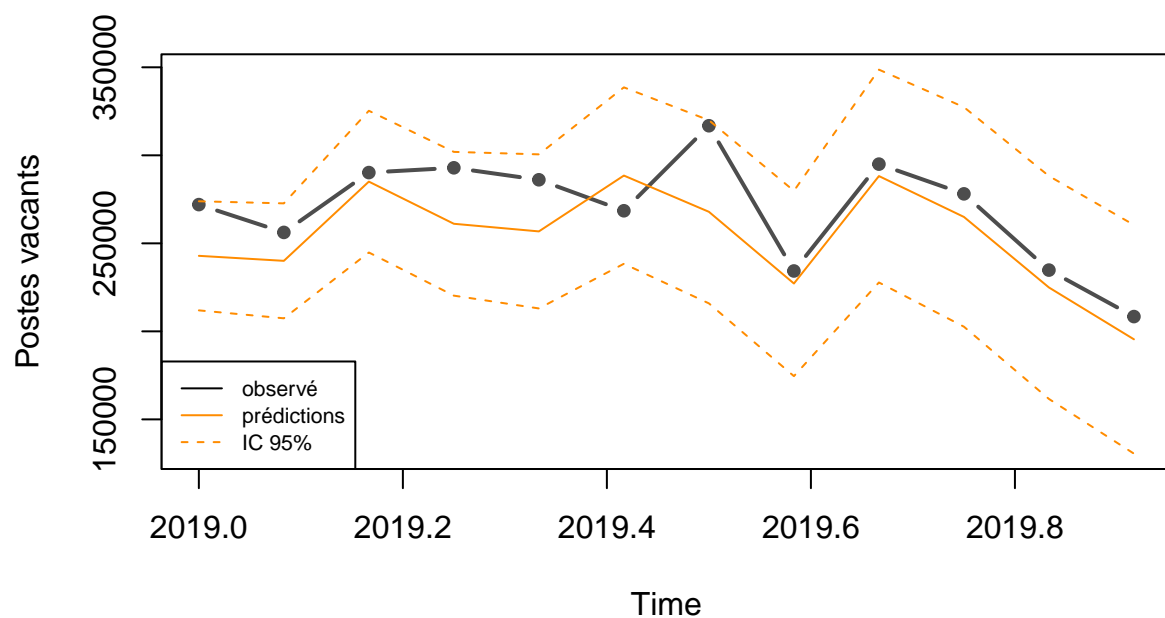
prévisions à 12 mois sur la série de test (2019)

```
predictions.le.ZZZ.test <- forecast(modele.le.ZZZ.train, h=12)
```

**Lissage exponentiel (M,Ad,A) – prédiction de l'année 2019**



**Lissage exponentiel (M,Ad,A) – prédiction de l'année 2019**



L'ensemble des valeurs observées dans la série de test sont incluses dans l'intervalle de confiance à 95%.

Le modèle identifié par R propose des performances similaires aux modèles étudiés précédemment.

### 3.2 Prédiction de la série test 2019 - Choix du modèle

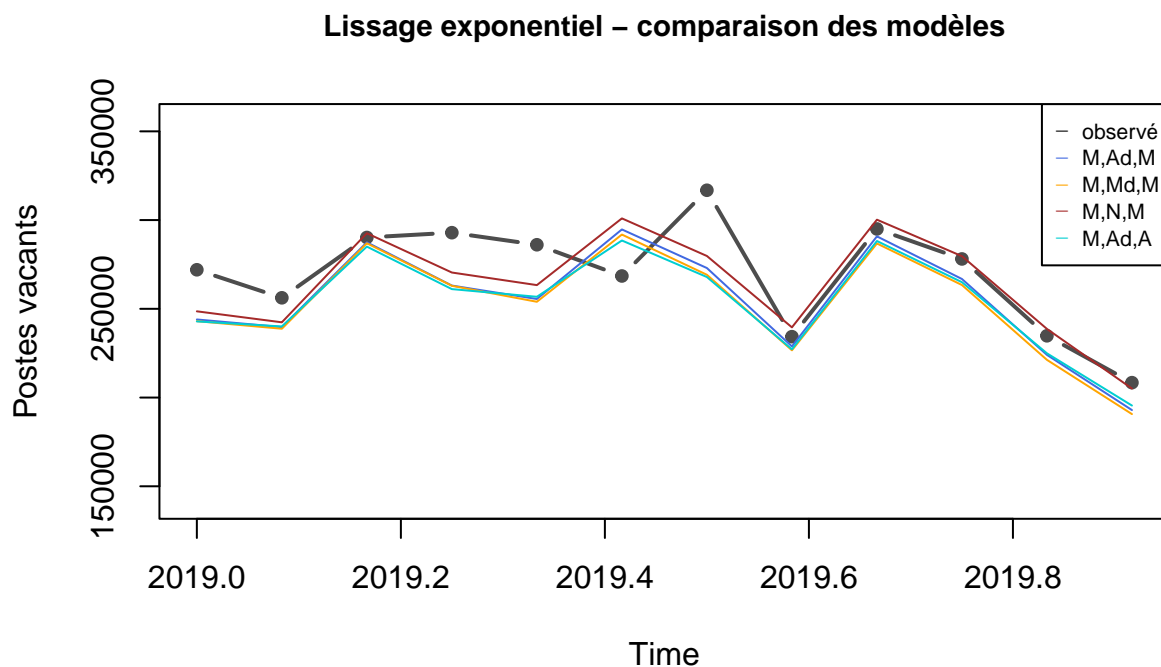


Table 1: Résumé des modèles

nom	aic	aicc	bic
M,Ad,M	2916.624	2923.396	2966.799
M,Md,M	2916.281	2923.054	2966.456
M,N,M	2932.123	2936.738	2973.935
M,Ad,A selon R	2913.972	2920.744	2964.147

D’après la comparaison graphique, l’ensemble des modèles performe de façon similaire.

On note cependant que le modèle (M, N, M) se comporte mieux après septembre, en proposant des prédictions très proches des observations.

Néanmoins, si l’on se base sur les critères de sélection de modèles, pour l’ensemble des critères AIC, AICc, BIC, le **modèle (M,Ad,A)** identifié par R avec une saisonnalité additive est le modèle disposant du meilleur pouvoir prédictif.

On prend le parti de faire confiance aux critères de sélection et on choisit le modèle (M,Ad,A) pour représenter la catégorie “lissage exponentiel”.

### 3.3 Entraînement sur la série d’étude entière (Jan 2009 - Dec 2019)

Afin de nous assurer de disposer du meilleur modèle de lissage exponentiel sur la série d’étude entière

- on va entraîner le modèle (M,Ad, A) sur la série d'étude entière (Janvier 2009 - Décembre 2019)
- on va utiliser la fonction *ets* pour préciser si ce modèle est bien le meilleur modèle sur l'entièreté de la série d'étude.

On entraîne le modèle (M,Ad,A) sur la série d'étude entière

```
modele.le.MAdA.study <- ets(job.ts.study, model="MAA", damped = TRUE)
```

```
## ETS(M,Ad,A)
##
## Call:
## ets(y = job.ts.study, model = "MAA", damped = TRUE)
##
## Smoothing parameters:
##   alpha = 0.2937
##   beta  = 0.0449
##   gamma = 1e-04
##   phi   = 0.8761
##
## Initial states:
##   l = 233073.9611
##   b = 2208.4909
##   s = -56237.08 -27175.21 13465.16 34443.55 -23457.77 16517.14
##       32098.11 4629.445 8295.622 28514.16 -16832.77 -14260.36
##
## sigma: 0.0661
##
##      AIC      AICc      BIC
## 3222.679 3228.732 3274.569
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 142.477 15606.07 12066.7 -0.2370243 4.809198 0.5323545 -0.1530498
```

On cherche maintenant de façon automatique le meilleur modèle sur la série d'étude entière:

```
modele.le.ZZZ.study <- ets(job.ts.study, model="ZZZ")
```

```
## ETS(M,Ad,A)
##
## Call:
## ets(y = job.ts.study, model = "ZZZ")
##
## Smoothing parameters:
##   alpha = 0.2937
##   beta  = 0.0449
##   gamma = 1e-04
##   phi   = 0.8761
##
## Initial states:
##   l = 233073.9611
##   b = 2208.4909
```

```
##      s = -56237.08 -27175.21 13465.16 34443.55 -23457.77 16517.14
##          32098.11 4629.445 8295.622 28514.16 -16832.77 -14260.36
##
##      sigma: 0.0661
##
##      AIC      AICc      BIC
## 3222.679 3228.732 3274.569
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 142.477 15606.07 12066.7 -0.2370243 4.809198 0.5323545 -0.1530498
```

La fonction *ets* confirme que le **modèle (M,Ad,A)** est bien le meilleur selon le critère par défaut AICc.

### 3.4 Conclusion

Le lissage exponentiel propose de bons modèles de prédiction pour notre série d'étude, le meilleur dans notre cas étant un modèle **(M,Ad,A)**.

Même si les prédictions restent relativement distantes des valeurs observées sur la série de test avant septembre, tous les modèles incluent les valeurs observées dans leur intervalle de confiance à 95%.

On notera aussi que les modèles analysés sont tous principalement basés sur la mise à jour du niveau ; ils proposent peu d'évolution de pente ou de saisonnalité. Ils indiquent aussi que la série est relativement peu dépendante de son passé récent.



## 4 Modèles ARMA

On va maintenant s'atteler à identifier un modèle ARMA pour la série d'étude

- en identifiant des candidats ARMA ajustés sur la série d'entraînement (Jan 2008 - Dec 2018)
- en identifiant le meilleur prédicteur sur la série test (2019)
- en entraînant le modèle sur la série entière en vue d'un travail sur l'année 2020

On prendra soin dans cette dernière étape de vérifier que le modèle en question est toujours bien identifié comme le meilleur modèle.

### 4.1 Stationnarité

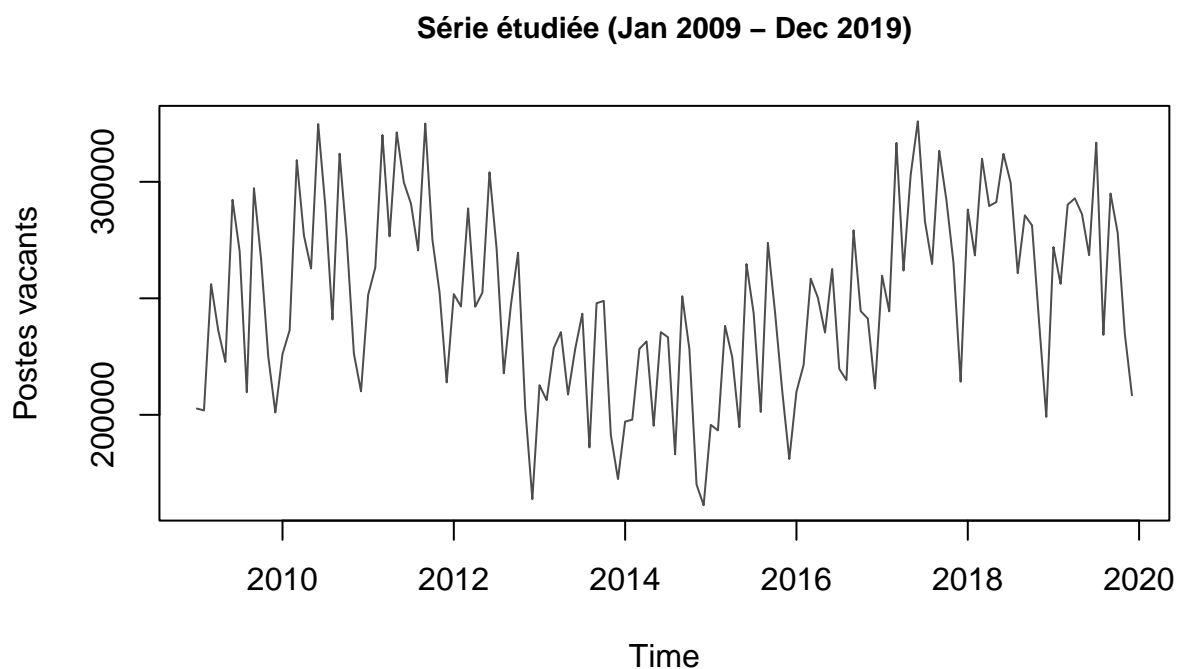
L'objet de ce chapitre est d'étudier la **stationnarité** de la série temporelle d'étude (Jan 2008 - Dec 2019).

Pour cela on va

- visualiser les caractéristiques du chronogramme et conclure ou non à la stationnarité,
- confirmer la conclusion avec la visualisation de la fonction de d'auto-corrélation empirique (corrélogramme)

Dans le cas (probable) où la série s'avérerait non-stationnaire, on procédera dans un second temps à une transformation de la série pour la faire tendre vers la stationnarité après modélisation.

#### 4.1.1 Chronogramme et corrélogramme



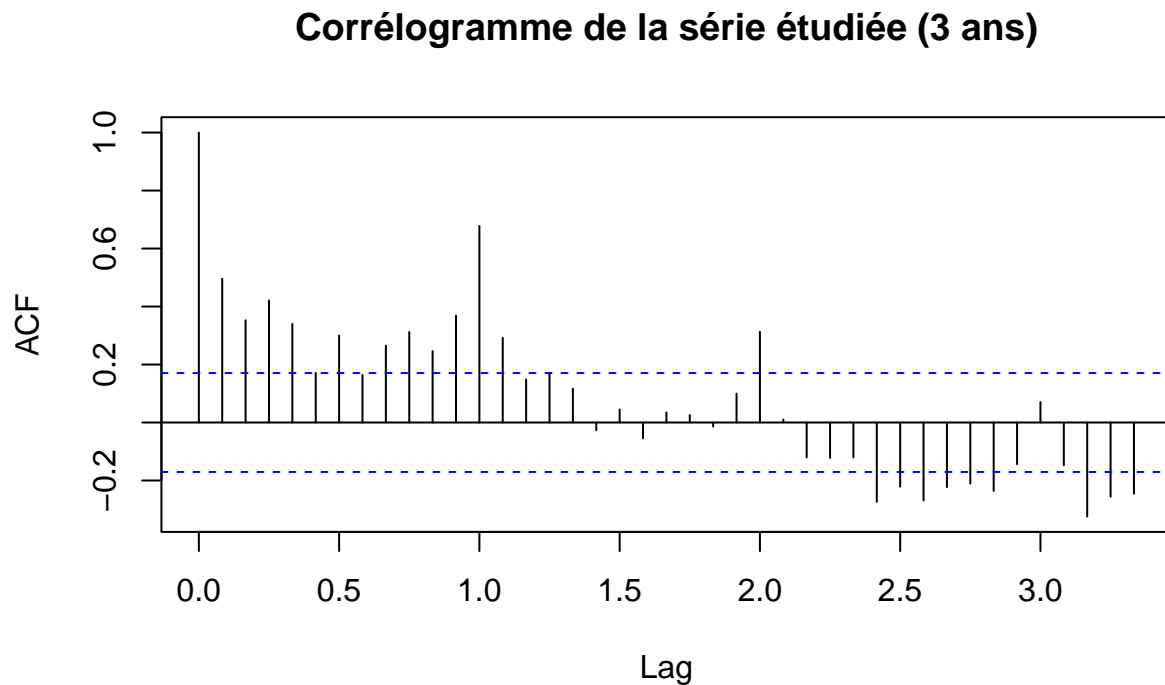
Un processus stationnaire doit présenter un chronogramme dont la moyenne et la variance sont constante dans le temps.

Or le chronogramme présente une série

- dont la moyenne n'est pas constante dans le temps
- dont la variance n'est pas constante dans le temps

On ne peut pas conclure à la stationnarité de la série au vu du chronogramme.

Pour confirmer la tendance, on visualise le corrélogramme.



Si le processus était stationnaire, il présenterait un corrélogramme

- dont la 1ère valeur est 1 (c'est le cas quelle que soit la série)
- dont les valeurs pour des décalages temporels supérieurs ou égaux à 1 convergent exponentiellement sous la barre de significativité à 95%.

Or on note ici

- une décroissance lente du lien linéaire de la série avec son passé
- des rebonds récurrents laissant imaginer une **saisonnalité de 12 mois**

Le corrélogramme confirme le **comportement non-stationnaire** de la série temporelle.

### 4.1.2 Transformation de la série

La série n'est pas stationnaire.

L'idée de la transformation est d'entamer une démarche qui permettra de réduire la série étudiée à des résidus stationnaires.

4 étapes pour aller dans ce sens:

- Vérifier la normalité de la série et stabiliser la variance  
La normalité de la série permet d'assurer que l'algorithme d'optimisation converge selon la méthode du maximum de vraisemblance (choisie par défaut par la méthode Arima).  
La stabilisation de la variance qui en découlera permettra de tendre vers la stationnarité de la série après modélisation.
- Entamer l'explication de l'information contenue dans la série en différenciant la tendance et la saisonnalité
- Sauf à ce que la série différenciée aboutisse à un processus stationnaire, s'atteler à modéliser le reste d'information par un modèle ARMA
- Valider que les résidus après modélisation ARMA sont bien un bruit blanc stationnaire.

#### 4.1.2.1 Normalité et stabilisation de la variance :

Dans l'optique de la modélisation de la série par un modèle ARMA, il est nécessaire d'examiner la normalité de la série par un test de d'Agostino.

## Le chargement a nécessité le package : fBasics

```
##
## Title:
## D'Agostino Normality Test
##
## Test Results:
## STATISTIC:
## Chi2 | Omnibus: 6.9611
## Z3 | Skewness: -0.1217
## Z4 | Kurtosis: -2.6356
## P VALUE:
## Omnibus Test: 0.03079
## Skewness Test: 0.9032
## Kurtosis Test: 0.008399
```

La p-valeur Omnibus du test de d'Agostino est faible (<5%), on rejette l'hypothèse  $H_0$  : Normalité de la série.

Pour tenter d'améliorer le résultat, on applique des transformations **logarithme** et **racine carrée** à la série d'étude pour identifier une éventuelle amélioration de la normalité.

Les résultats du test de d'Agostino (Omnibus) après transformation sont les suivants

```
##                p-values
## Serie brute 0.03079041
## log         0.09902901
## racine      0.06790228
```

Effectivement, les transformations “normalisantes” ont eu l’effet escompté ; les p-valeurs du test de d’Agostino pour les transformations **log** et **racine carrée** sont maintenant  $> 5\%$  ; on ne rejette pas la normalité de la série pour ces 2 transformations.

La transformation étant plus prononcée pour la transformation logarithmique, on appliquera pour la suite de l’étude une transformation **log** à la série, ce qui stabilisera la variance de la série d’étude.

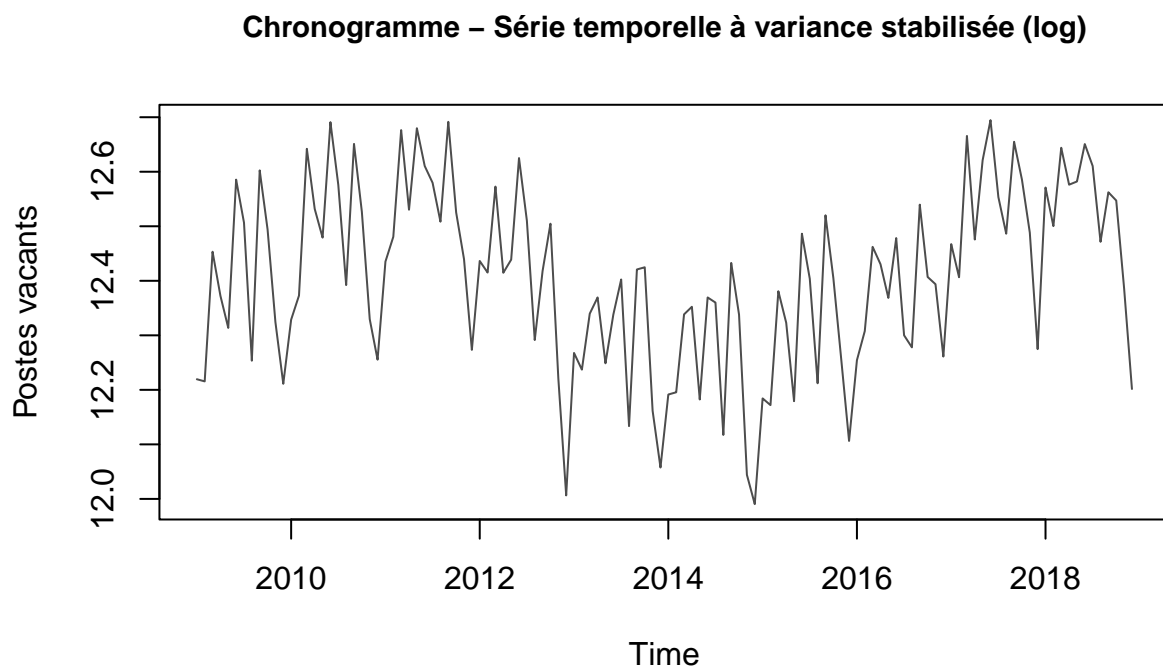
```
# stabilisation de la série d'étude (Jan 2009 - Dec 2019)
job.ts.study.log <- log(job.ts.study)

# stabilisation de la série d'entraînement (Jan 2009 - Dec 2018)
job.ts.train.log <- log(job.ts.train)

# stabilisation de la série de test (2019)
job.ts.test.log <- log(job.ts.test)
```

### Stabilisation de la variance

La transformation log étant appliquée, vérifions que la variabilité des données de la série a bien été atténuée.



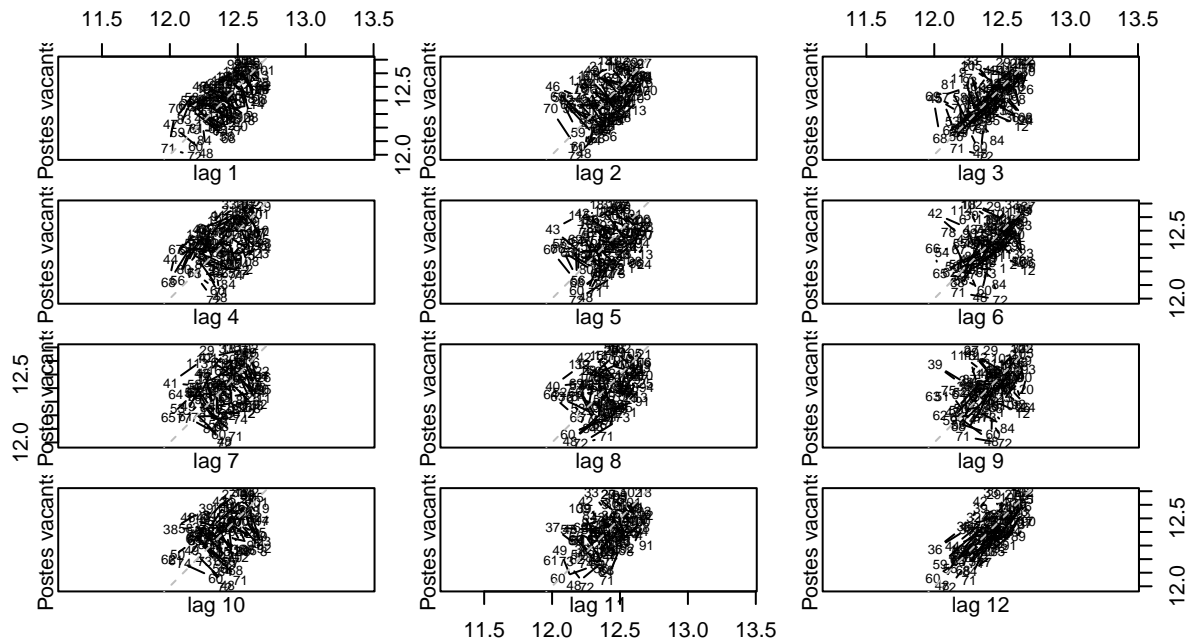
Les variations des données sont maintenant d’amplitude similaire ; l’opération de stabilisation de la variance est satisfaisante.

#### 4.1.2.2 Elimination de la saisonnalité :

Dans un 1er temps,

- on va confirmer la saisonnalité de la série stabilisée par un lag-plot
- on va différencier la série pour éliminer la saisonnalité estimée

## lag.plot de la série d'entraînement stabilisée par un log

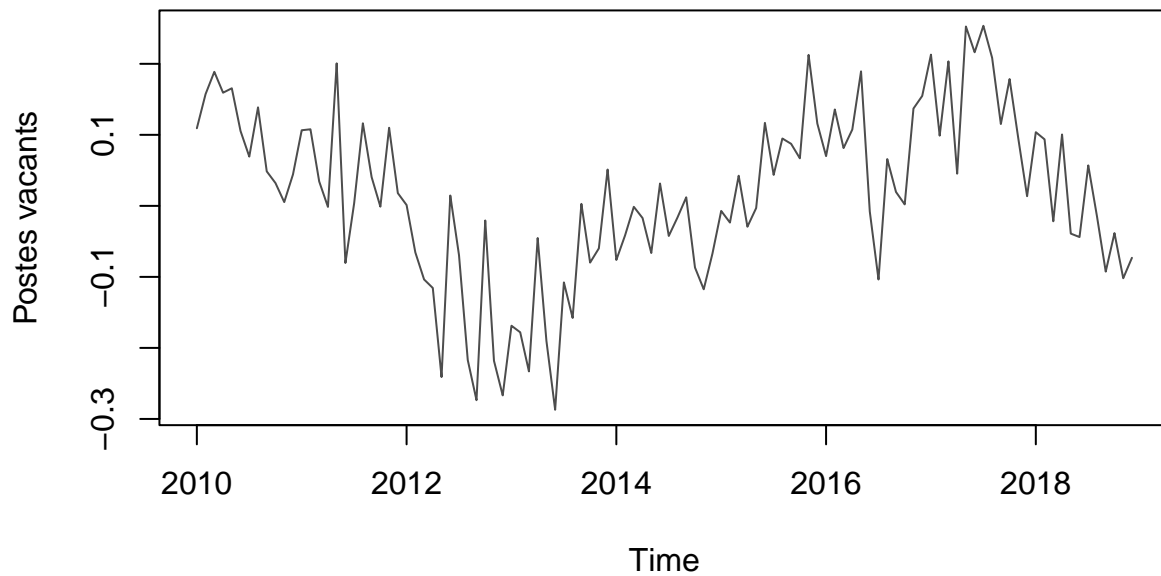


On note la plus forte corrélation de la série à un **décalage de 12 mois**. C'est cette valeur que l'on va considérer comme saisonnalité.

**Différenciation saisonnière** basée sur la **saisonnalité à 12 mois**, sur la série à variance stabilisée.

```
job.ts.train.log.sans_saison <- diff(job.ts.train.log, lag=12, difference=1)
```

### Série d'entraînement stabilisée, avec tendance, sans saisonnalité



Le chronogramme de la version sans saisonnalité présente une tendance qu'on va maintenant supprimer.

#### 4.1.2.3 Elimination de la tendance :

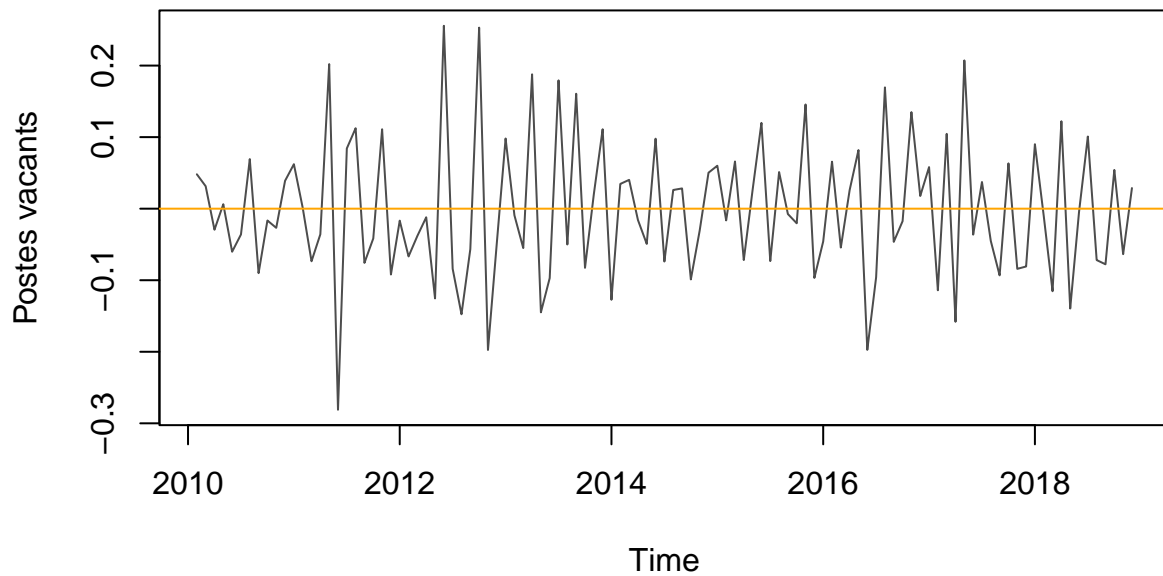
Le chronogramme de la série sans saisonnalité laisse penser à une tendance a priori linéaire, qu'on va tenter d'éliminer par une **différenciation d'ordre 1**.

L'élimination de la tendance à la série privée de saisonnalité donne les **résidus** de la série temporelle :

```
job.ts.train.log.residus <- diff(job.ts.train.log.sans_saison, lag=1, difference=1)
```

Chronogramme de la série sans tendance, sans saisonnalité (i.e. les résidus):

### Série d'entraînement sans saisonnalité, ni tendance (résidus)



```
mean(job.ts.train.log.residus)
```

```
## [1] -0.001704236
```

La question que l'on peut se poser sur les résidus de la série sans tendance et sans saisonnalité, est "sont-ils un bruit blanc."

S'il s'agit d'un bruit blanc, le chronogramme doit être centré sur 0 et de variance constante.

Dans notre cas,

- les données sont bien centrées en 0
- bien que ne présentant pas des extrêmes marqués, la variabilité des données ne peut pas être considérée comme constante ; la bande dans laquelle les valeurs évoluent n'est pas constante.

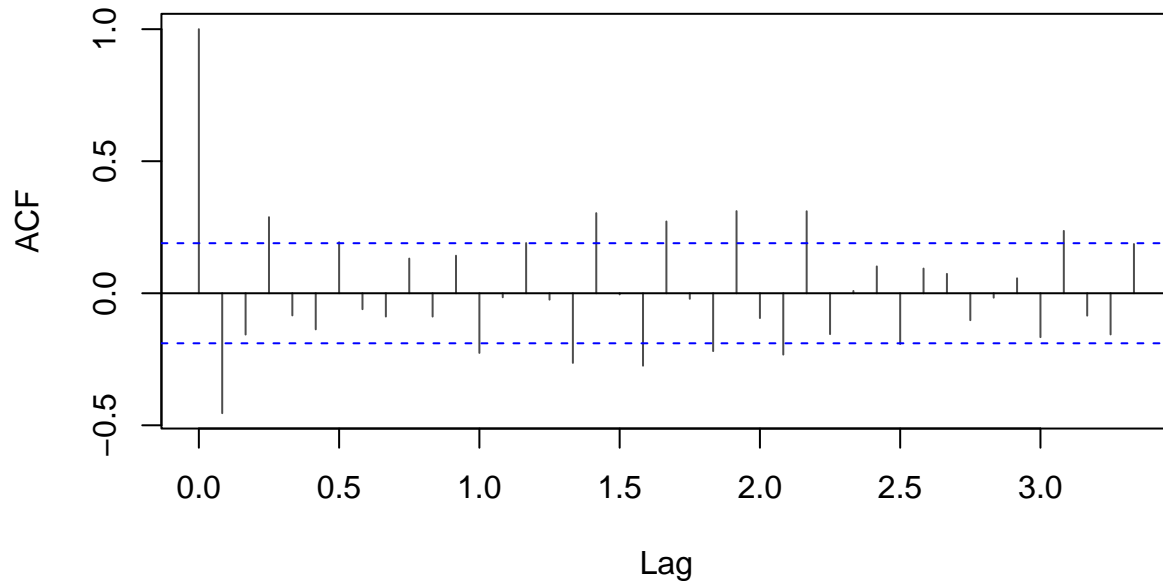
On soupçonne donc une variance non constante ; il reste vraisemblablement de l'information non expliquée dans ces résidus.

Pour aller plus loin, on se propose

- d'analyser de façon plus précise la blancheur des résidus de la série diminuée de la tendance et de la saisonnalité
- et dans le cas où ils ne seraient pas un bruit blanc, de modéliser ces résidus avec un modèle ARMA

#### 4.1.3 Analyse des résidus (série sans tendance ni saisonnalité)

### Corrélogramme des résidus



Dans le cas où il s'agirait d'un bruit blanc, le corrélogramme présenterait

- 1 première valeur égale à 1 (condition toujours vérifiée, quelle que soit la série)
- des valeurs qui convergent exponentiellement sous la bande de significativité de 95%.
- un maximum de 5% des corrélations en dehors de la bande de significativité

Ici, la convergence est réelle ; néanmoins on soupçonne que **plus de 5% des valeurs** sont en dehors de la bande des 95%.

Les résidus ne sont vraisemblablement **pas un bruit blanc**.

Le **test Portmanteau (test de blancheur)** devrait confirmer l'hypothèse.

Le test oppose

- $H_0$  : "toutes les auto-corrélations sont nulles"
- à  $H_1$  : "Il existe une auto-corrélation non nulle"

```
Box.test(job.ts.train.log.residus,  
        lag=length(job.ts.train.log.residus)/4,  
        type="Box-Pierce")  
  
##  
## Box-Pierce test  
##  
## data:  job.ts.train.log.residus  
## X-squared = 121.6, df = 26.75, p-value = 4.707e-14
```



En l'occurrence, la **p-valeur**  $< 5\%$  permet de **rejeter  $H_0$** .

Il existe au moins une corrélation non nulle : **les résidus NE sont PAS un bruit blanc**.

**Il reste de l'information à modéliser dans les résidus.**

On va tenter dans le chapitre suivant une **modélisation de ce bruit par un modèle ARMA**.

## 4.2 Modélisation des résidus sur la série d'entraînement (Jan 2009 - Déc 2018)

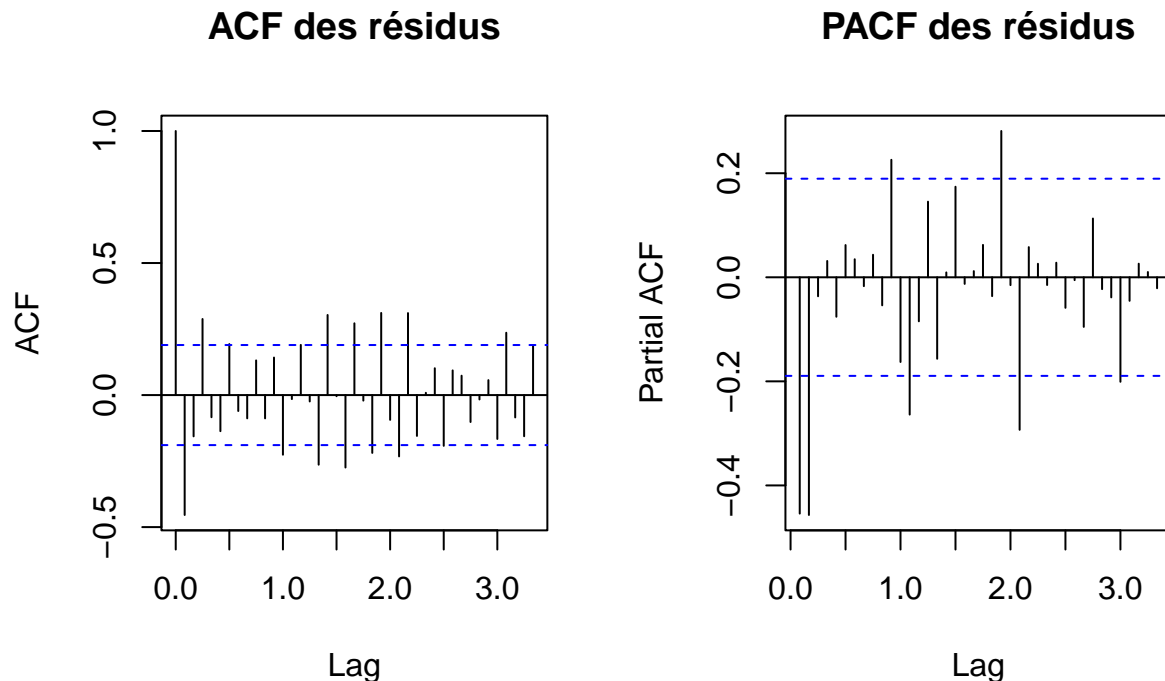
La série temporelle diminuée de la tendance et de la saisonnalité présente des résidus qui ne sont pas un bruit blanc.

Il reste de la variabilité inexpliquée dans ces résidus qu'il s'agit maintenant de modéliser par un modèle ARMA pour permettre la prédiction de nouvelles données.

### 4.2.1 ACF et PACF

Pour déterminer si les résidus peuvent être modélisés par un processus AR et/ou MA, on visualise

- la fonction d'auto-corrélation des résidus (ACF)
- la fonction d'auto-corrélation partielle (PACF)



Un **processus AR(p)** “**pur**” présenterait

- une ACF convergeant exponentiellement vers 0
- une PACF avec un point de rupture “décalé” au-delà duquel les valeurs seraient nulles

Un **processus MA(q)** “**pur**” présenterait

- une ACF avec un point de rupture “décalé” au-delà duquel les valeurs seraient nulles.
- une PACF convergeant exponentiellement vers 0

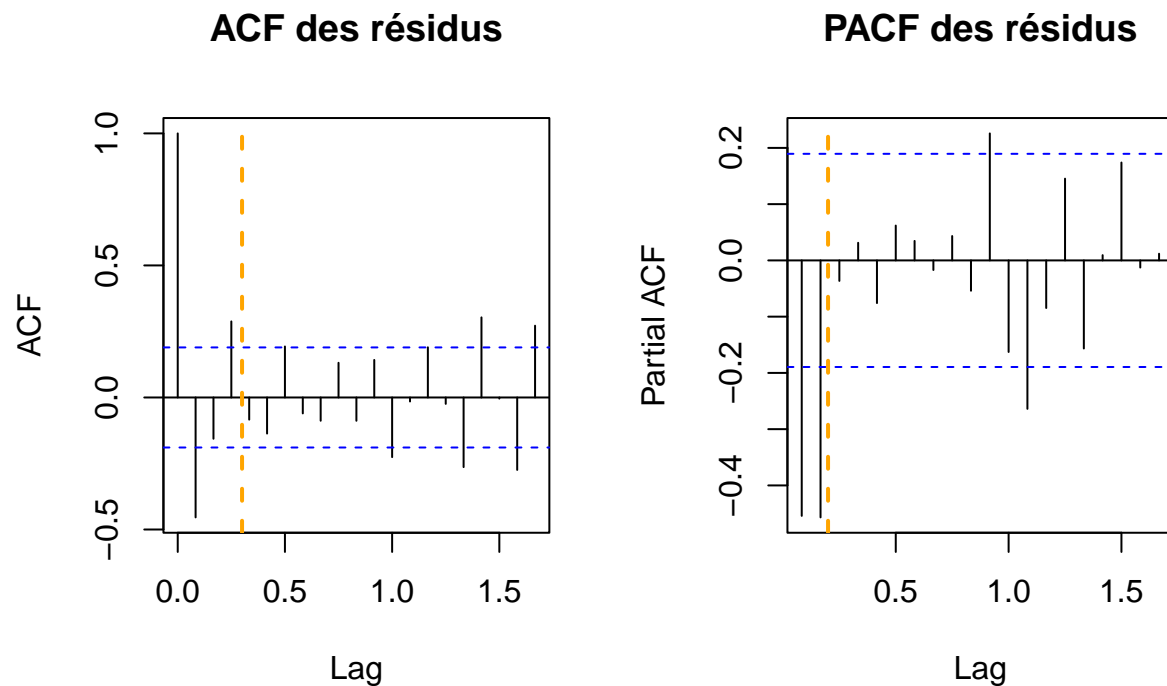
Ni l’ACF, ni la PACF ne présente ces caractéristiques exclusives :

- Les convergences exponentielles sont bien constatées
- Des rebonds significativement non nuls sont visibles de façon récurrente le long de l’ACF et de la PACF

Ainsi on imagine qu’un **processus mixte de type ARMA** serait le plus adéquat pour modéliser les résidus.

On se propose de dessiner un 1er modèle ARMA(p,q).

#### 4.2.2 Modèle ARMA(p,q) initial



ACF et PACF ne présentent pas les caractéristiques des processus AR et MA “purs”.

Néanmoins on peut en tirer des tendances qui peuvent dessiner un modèle ARMA mixte initial :

- L’ACF présente une décroissance exponentielle pour laquelle les valeurs entrent de façon persistante dans la bande de significativité de 95% **à partir d’un décalage h=4** ; ce sont les caractéristiques d’un **processus moyenne mobile MA(3)**.
- La PACF présente une décroissance exponentielle pour laquelle les valeurs entrent de façon persistante dans la bande de significativité de 95% **à partir d’un décalage h=3** ; ce sont les caractéristiques d’un **processus auto-régressif AR(2)**.

Ainsi, on peut imaginer un **modèle initial** basé

- sur une **composante auto-régressive AR d'ordre 2**
- sur une **composante moyenne mobile MA d'ordre 3**

**ARMA(2,3)** sera donc le **modèle initial** issu de l'intuition.

#### 4.2.3 ARMA(2,3)

```
# ajustement ARMA(3,2)
modele.arma.2_3 <- Arima(job.ts.train.log.residus, order=c(2,0,3))
```

```
## Series: job.ts.train.log.residus
## ARIMA(2,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2      ma3      mean
##      -0.6772 -0.6037  0.0098  0.1343 -0.1332 -0.0020
## s.e.    0.2513   0.2032  0.2833  0.2388   0.1602   0.0033
##
## sigma^2 = 0.006342: log likelihood = 121.64
## AIC=-229.29   AICc=-228.16   BIC=-210.58
```

```
# Significativité des coefficients
t_stat(modele.arma.2_3)
```

```
##          ar1      ar2      ma1      ma2      ma3 intercept
## t.stat -2.694403 -2.970747  0.034497  0.562373 -0.831405 -0.600952
## p.val   0.007051   0.002971  0.972481  0.573862   0.405745   0.547872
```

```
# corrélation des termes
cor.arma(modele.arma.2_3)
```

```
##          ar1      ar2      ma1      ma2      ma3
## ar1      1.00000000  0.39453704 -0.941730339  0.435007517  0.33203837
## ar2      0.394537043  1.00000000 -0.380226148 -0.574153810  0.83992969
## ma1     -0.941730339 -0.38022615  1.000000000 -0.401167729 -0.30062499
## ma2      0.435007517 -0.57415381 -0.401167729  1.000000000 -0.48061299
## ma3      0.332038373  0.83992969 -0.300624992 -0.480612990  1.00000000
## intercept -0.007645535 -0.01472487  0.009511262  0.008065677 -0.01444759
##          intercept
## ar1      -0.007645535
## ar2      -0.014724865
## ma1       0.009511262
## ma2       0.008065677
## ma3      -0.014447594
## intercept 1.000000000
```

```
# Test de blancheur des résidus
Box.test(modele.arma.2_3$residuals,
        lag=length(modele.arma.2_3$residuals)/4,
        type="Box-Pierce")
```

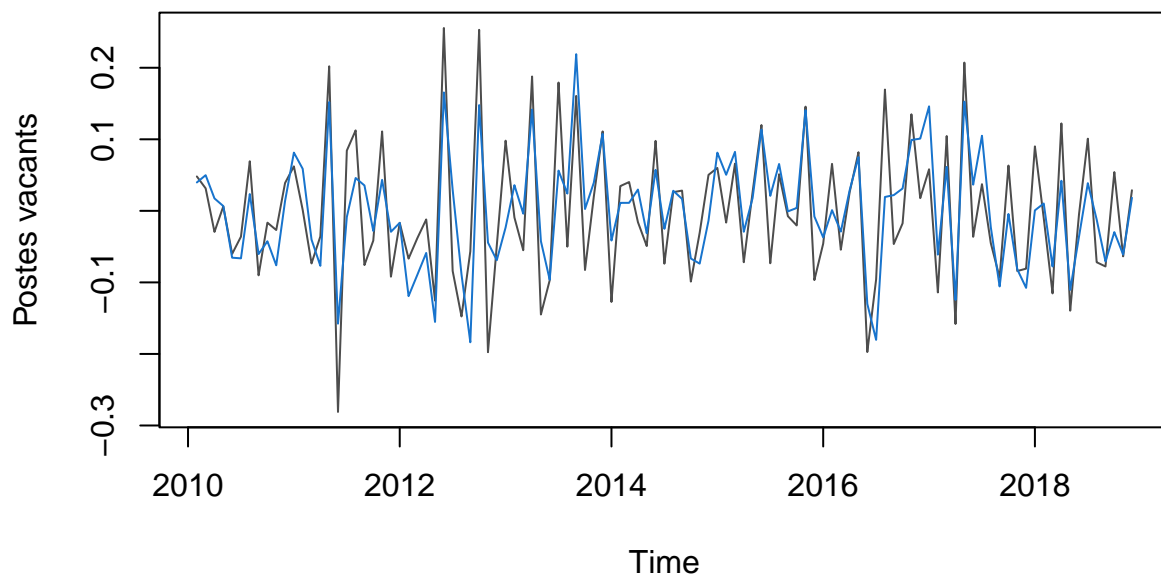
```
##
## Box-Pierce test
##
## data: modele.arma.2_3$residuals
## X-squared = 30.968, df = 26.75, p-value = 0.2614
```

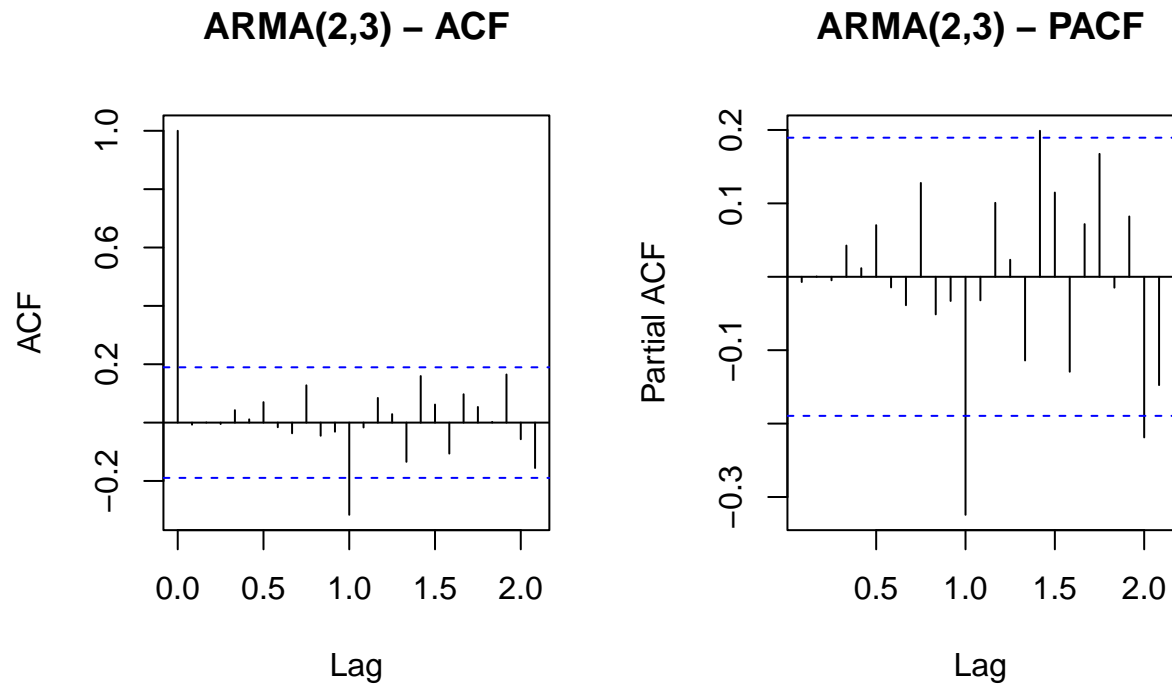
Après ajustement du **modèle ARMA(2,3)**,

- **MA1, MA2, MA3** sont considérés comme **non-significatifs** ;
- des **problèmes de corrélation** entre certains termes ( $>0.9$ ) sont détectés ;
- les **résidus** après modélisation sont un **bruit blanc** ;

la **p-valeur du test de blancheur** est  $> 5\%$ , et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus ARMA(2,3) soient un **bruit blanc**.

**Série d'entraînement : modélisation ARMA(2,3)**





Les données estimées suivent bien la série temporelle.

ACF et PACF des résidus après **modélisation ARMA(2,3)** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Néanmoins on note un **rebond de corrélation à chaque saisonnalité**. Il reste vraisemblablement de l'information saisonnière non expliquée par le modèle.

Pour la suite, on prend le parti de **descendre l'ordre de la composante MA** (sachant que MA3 est considéré comme non significatif).

#### 4.2.4 ARMA(2,2)

```
# ajustement ARMA(2,2)
modele.arma.2_2 <- Arima(job.ts.train.log.residus, order=c(2,0,2))
```

```
## Series: job.ts.train.log.residus
## ARIMA(2,0,2) with non-zero mean
##
## Coefficients:
##      ar1      ar2      ma1      ma2      mean
##    -0.5770 -0.4365 -0.0988  0.0318 -0.0021
## s.e.   0.2306   0.1297   0.2467   0.1772   0.0035
##
```

```
## sigma^2 = 0.006308: log likelihood = 121.41
## AIC=-230.82 AICc=-229.98 BIC=-214.79
```

```
# Significativité des coefficients
```

```
t_stat(modele.arma.2_2)
```

```
##          ar1          ar2          ma1          ma2 intercept
## t.stat -2.502213 -3.366069 -0.400293 0.179368 -0.587858
## p.val   0.012342  0.000762  0.688940 0.857649  0.556628
```

```
# corrélation des termes
```

```
cor.arma(modele.arma.2_2)
```

```
##          ar1          ar2          ma1          ma2          intercept
## ar1      1.000000000  0.348794970 -0.919397585  0.570865249 -0.005079981
## ar2      0.348794970  1.000000000 -0.354051957 -0.385393901 -0.003734907
## ma1     -0.919397585 -0.354051957  1.000000000 -0.512015194  0.008519513
## ma2      0.570865249 -0.385393901 -0.512015194  1.000000000  0.000182513
## intercept -0.005079981 -0.003734907  0.008519513  0.000182513  1.000000000
```

```
# Test de blancheur des résidus
```

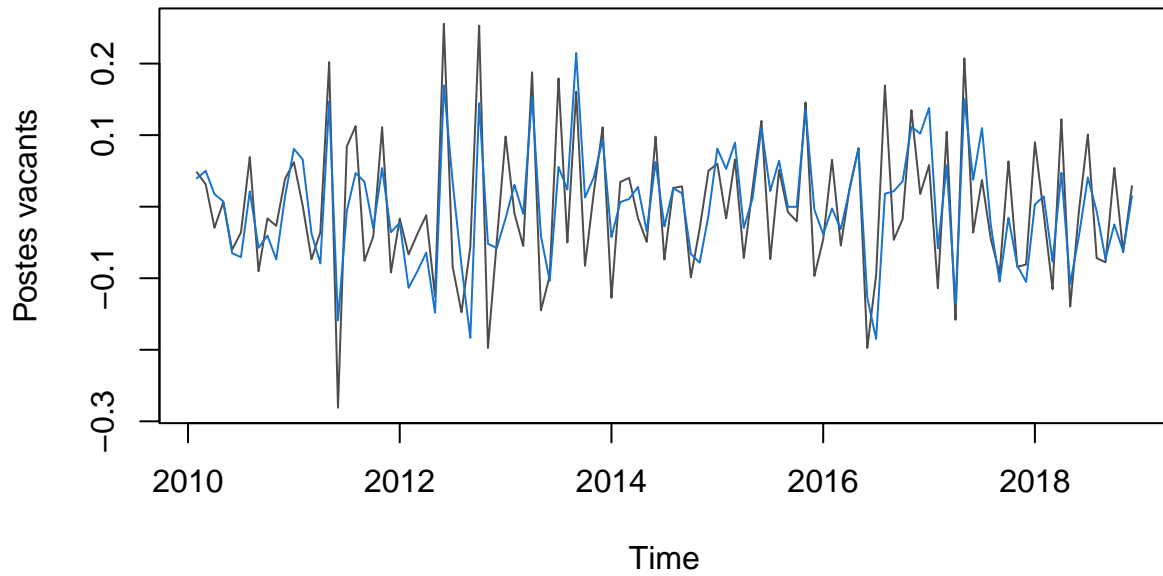
```
Box.test(modele.arma.2_2$residuals,
         lag=length(modele.arma.2_2$residuals)/4,
         type="Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: modele.arma.2_2$residuals
## X-squared = 31.677, df = 26.75, p-value = 0.2339
```

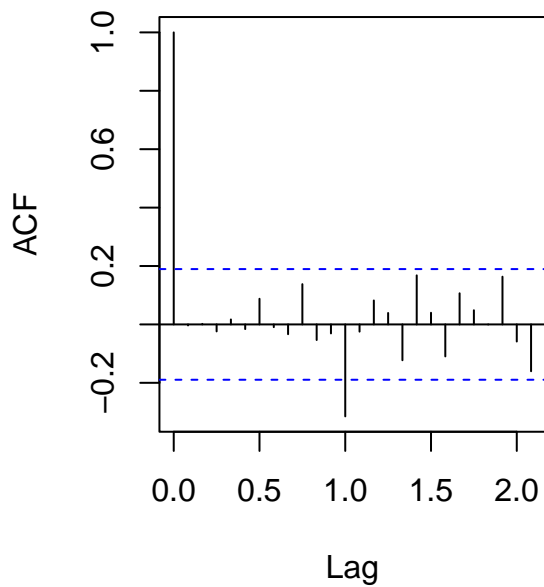
Après ajustement du **modèle ARMA(2,2)**,

- **MA1, MA2** sont considérés comme **non-significatifs**
- des **problèmes de corrélation** entre certains termes, notamment entre MA1 et AR1 ( $>0.9$ ) sont détectés
- les **résidus** après modélisation sont un **bruit blanc** ;  
la **p-valeur du test de blancheur** est  $> 5\%$ , et on **NE rejette PAS l'hypothèse** que les résidus après modélisation par un processus ARMA(2,2) soient un **bruit blanc**.

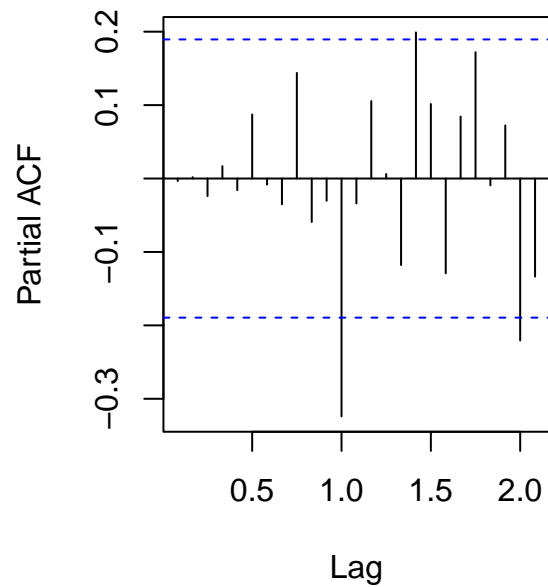
Série d'entraînement : modélisation ARMA(2,2)



ARMA(2,2) – ACF



ARMA(2,2) – PACF



Les données estimées suivent bien la série temporelle.

ACF et PACF des résidus après **modélisation ARMA(2,2)** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Néanmoins on note un **rebond de corrélation à chaque saisonnalité**. Il reste vraisemblablement de l'information saisonnière non expliquée par le modèle.

Pour la suite, on prend le parti de **descendre l'ordre de la composante MA** (sachant que MA2 est considéré comme non significatif).

#### 4.2.5 ARMA(2,1)

```
# ajustement ARMA(2,1)
modele.arma.2_1 <- Arima(job.ts.train.log.residus, order=c(2,0,1))
```

```
## Series: job.ts.train.log.residus
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1      mean
##      -0.5995  -0.4267  -0.0773  -0.0021
## s.e.    0.1835   0.1161   0.2019   0.0034
##
## sigma^2 = 0.006249: log likelihood = 121.4
## AIC=-232.79  AICc=-232.2  BIC=-219.43
```

```
# Significativité des coefficients
t_stat(modele.arma.2_1)
```

```
##          ar1      ar2      ma1 intercept
## t.stat -3.267425 -3.675295 -0.382952 -0.597222
## p.val   0.001085  0.000238  0.701755  0.550359
```

```
# corrélation des termes
cor.arma(modele.arma.2_1)
```

```
##          ar1      ar2      ma1 intercept
## ar1      1.000000000  0.736085037 -0.880151791 -0.004601346
## ar2      0.736085037  1.000000000 -0.667312035 -0.002495863
## ma1     -0.880151791 -0.667312035  1.000000000  0.008364039
## intercept -0.004601346 -0.002495863  0.008364039  1.000000000
```

```
# Test de blancheur des résidus
Box.test(modele.arma.2_1$residuals,
         lag=length(modele.arma.2_1$residuals)/4,
         type="Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: modele.arma.2_1$residuals
## X-squared = 31.553, df = 26.75, p-value = 0.2385
```

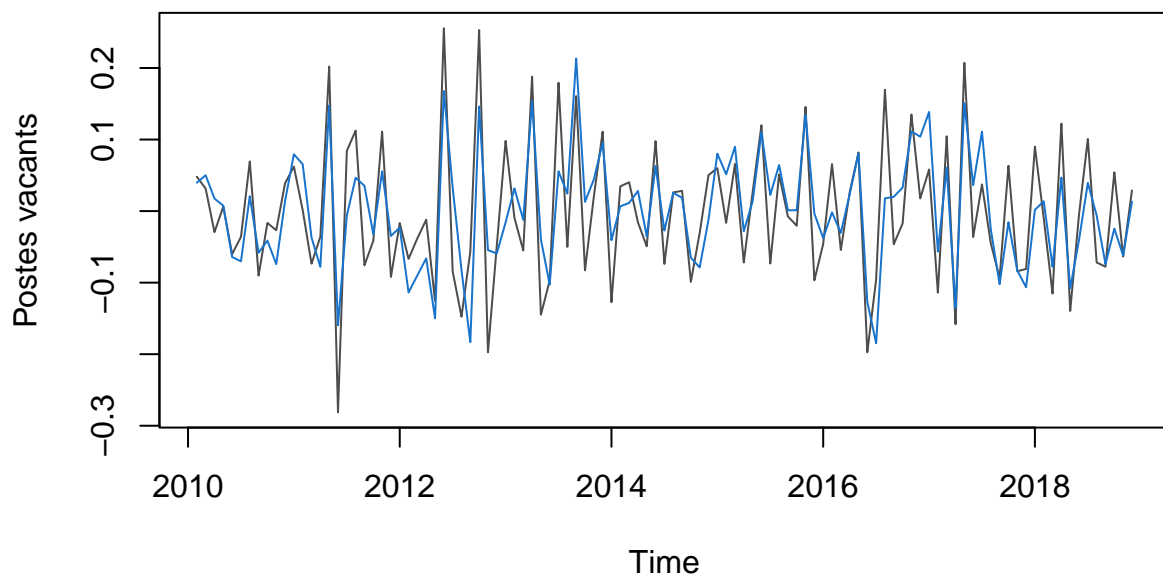


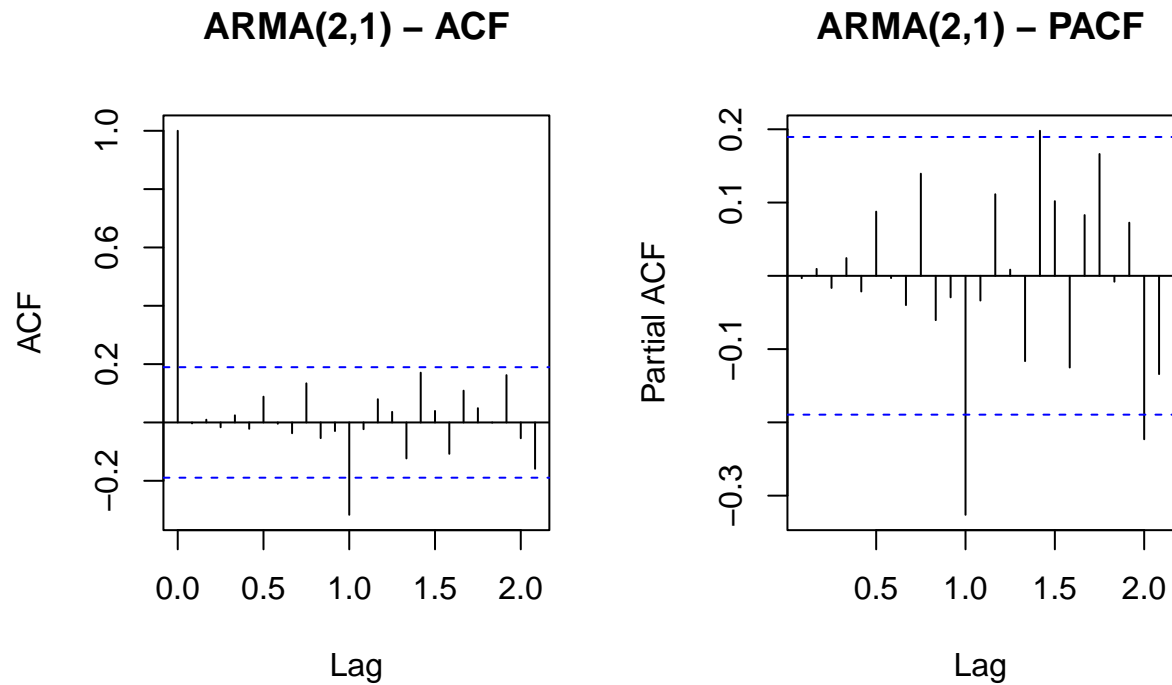
Après ajustement du **modèle ARMA(2,1)**,

- **MA1** est considéré comme **non-significatif**,
- Une corrélation certes  $< 0.9$  entre MA1 et AR1 reste élevée,
- les **résidus** après modélisation sont un **bruit blanc** ;

la **p-valeur du test de blancheur** est  $> 5\%$ , et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus ARMA(2,1) soient un **bruit blanc**.

### Série d'entraînement : modélisation ARMA(2,1)





Les données estimées suivent bien la série temporelle.

ACF et PACF des résidus après **modélisation ARMA(2,1)** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Néanmoins on note un **rebond de corrélation à chaque saisonnalité**. Il reste vraisemblablement de l'information saisonnière non expliquée par le modèle.

Pour la suite, on prend le parti de **descendre l'ordre de la composante MA** (sachant que MA1 est considéré comme non significatif).

#### 4.2.6 ARMA(2,0) = AR(2)

```
# ajustement AR(2)
modele.arma.2_0 <- Arima(job.ts.train.log.residus, order=c(2,0,0))
```

```
## Series: job.ts.train.log.residus
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##      ar1      ar2      mean
##    -0.6602 -0.4541 -0.0020
## s.e.   0.0857   0.0851   0.0036
##
```

```
## sigma^2 = 0.006197: log likelihood = 121.32
## AIC=-234.65 AICc=-234.25 BIC=-223.96
```

```
# Significativité des coefficients
t_stat(modele.arma.2_0)
```

```
##          ar1          ar2 intercept
## t.stat -7.701551 -5.336184 -0.572604
## p.val  0.000000  0.000000  0.566913
```

```
# corrélation des termes
cor.arma(modele.arma.2_0)
```

```
##          ar1          ar2 intercept
## ar1      1.000000000 0.453785740 0.005550796
## ar2      0.453785740 1.000000000 0.003896263
## intercept 0.005550796 0.003896263 1.000000000
```

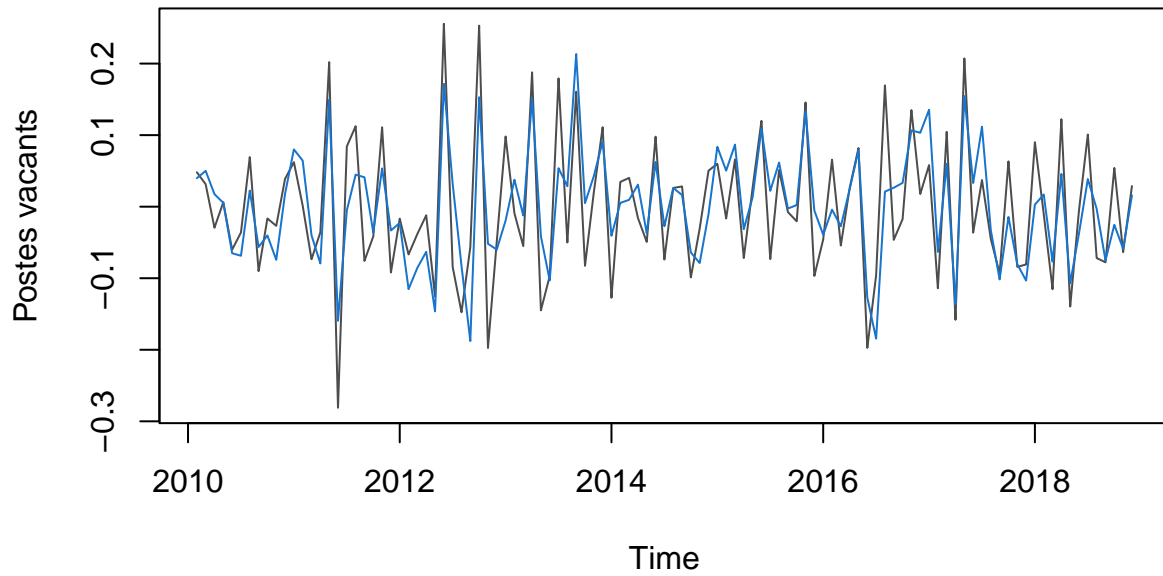
```
# Test de blancheur des résidus
Box.test(modele.arma.2_0$residuals,
         lag=length(modele.arma.2_0$residuals)/4,
         type="Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: modele.arma.2_0$residuals
## X-squared = 31.776, df = 26.75, p-value = 0.2302
```

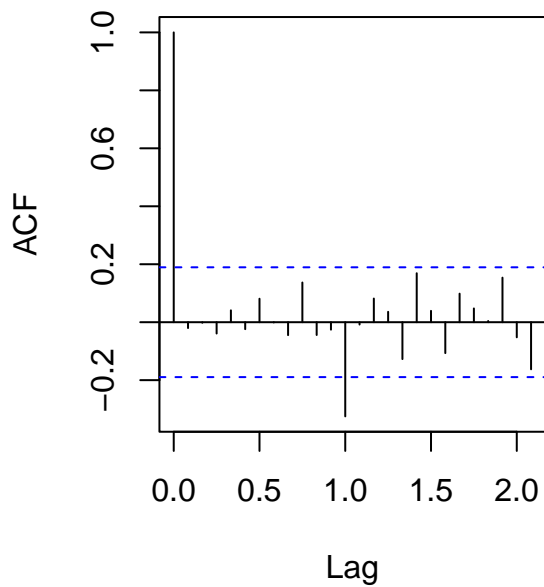
Après ajustement du **modèle AR(2)**,

- **tous les coefficients** sont considérés comme **significatifs**,
- **aucun problème de corrélation** n'est détecté,
- les **résidus** après modélisation sont un **bruit blanc** ;  
la **p-valeur** du **test de blancheur** est **> 5%**, et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus ARMA(2,0) soient un **bruit blanc**.

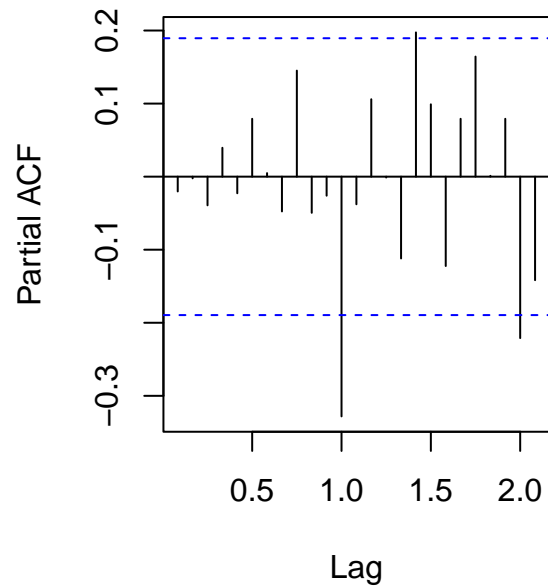
### Série d'entraînement : modélisation ARMA(2,0)



### ARMA(2,0) – ACF



### ARMA(2,0) – PACF



Les données estimées suivent bien la série temporelle.

ACF et PACF des résidus après **modélisation ARMA(2,0)** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Néanmoins on note un **rebond de corrélation à chaque saisonnalité**. Il reste vraisemblablement de l'information saisonnière non expliquée par le modèle.

De par ses **coefficients significatifs** et la **blancheur des résidus après modélisation**, le **modèle AR(2)** est un **modèle ARMA satisfaisant**.

On se propose maintenant de solliciter les fonctions de sélection automatique **armaselect** et **auto.arima** pour identifier les modèles les meilleurs selon leur algorithme respectif.

#### 4.2.7 auto-sélection du modèle : armaselect

```
armaselect(job.ts.train.log.residus)
```

```
##      p  q      sbc
## [1,] 2 13 -536.5251
## [2,] 2  0 -535.2944
## [3,] 2 12 -534.4276
## [4,] 7 12 -532.2216
## [5,] 2 14 -532.0679
## [6,] 3 13 -531.9105
## [7,] 2 15 -531.5449
## [8,] 7 13 -531.3929
## [9,] 6 12 -530.7387
## [10,] 6 13 -530.5757
```

La majorité des modèles identifiés par **armaselect** n'est **pas exploitable** ; sont proposés des modèles dont l'ordre de la composante moyenne mobile va de 12 à 15.

Néanmoins, *armaselect* a identifié parmi tous ces modèles complexes le **processus AR(2)** qu'il positionne en 2ème position de son classement des meilleurs modèles selon le critère sbc.

#### 4.2.8 auto-sélection du modèle : auto.arima

```
modele.arma.autoarima <- auto.arima(job.ts.train.log.residus)
```

```
## Series: job.ts.train.log.residus
## ARIMA(2,0,0)(2,0,0)[12] with zero mean
##
## Coefficients:
##      ar1      ar2      sar1      sar2
##      -0.7530 -0.4906 -0.4454 -0.2498
## s.e.   0.0874   0.0846   0.1014   0.1063
##
## sigma^2 = 0.005146: log likelihood = 130.13
## AIC=-250.25   AICc=-249.66   BIC=-236.89
```

```
# Significativité des coefficients
t_stat(modele.arma.autoarima)
```

```
##              ar1          ar2          sar1          sar2
## t.stat -8.620381 -5.798058 -4.392080 -2.350911
## p.val   0.000000  0.000000  0.000011  0.018728
```

```
# corrélation des termes
cor.arma(modele.arma.autoarima)
```

```
##              ar1          ar2          sar1          sar2
## ar1  1.0000000  0.51923174  0.2911222  0.16552793
## ar2  0.5192317  1.00000000  0.1514250  0.01538652
## sar1 0.2911222  0.15142502  1.0000000  0.44026620
## sar2 0.1655279  0.01538652  0.4402662  1.00000000
```

```
# Test de blancheur des résidus
Box.test(modele.arma.autoarima$residuals,
         lag=length(modele.arma.autoarima$residuals)/4,
         type="Box-Pierce")
```

```
##
## Box-Pierce test
##
## data:  modele.arma.autoarima$residuals
## X-squared = 27.785, df = 26.75, p-value = 0.4088
```

Auto-arma détecte un **processus ARMA(2,0)** avec une **composante saisonnière résiduelle** (pressentie dans l'analyse des fonctions d'auto-corrélation des résidus de la série diminuée de la tendance et de la saisonnalité).

Il confirme qu'une modélisation de type **ARMA saisonnière (SARMA)** serait pertinente dans le cas de notre série.

Plutôt que d'étudier le meilleur modèle SARMA, on se propose d'**étudier de façon équivalente les modèles intégrés SARIMA** dans lesquels la tendance sera conservée. C'est l'objet du **chapitre "Modèles SARIMA"**.

### 4.3 Prédiction de la série test 2019 - Choix du modèle ARMA

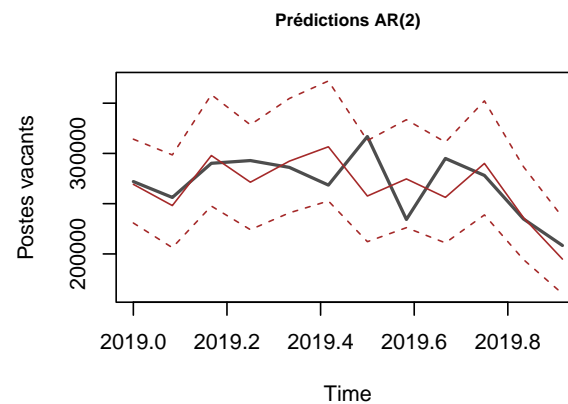
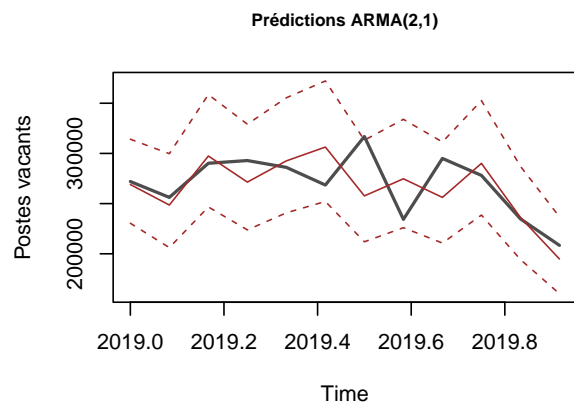
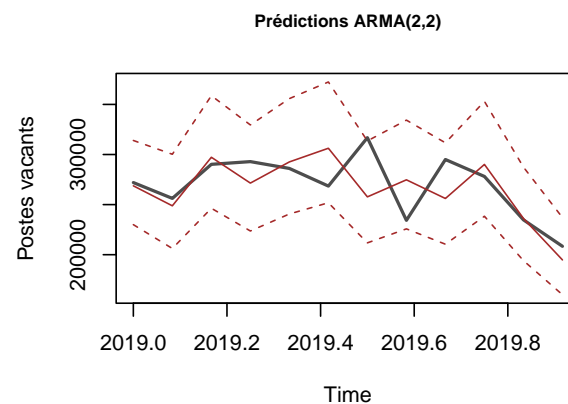
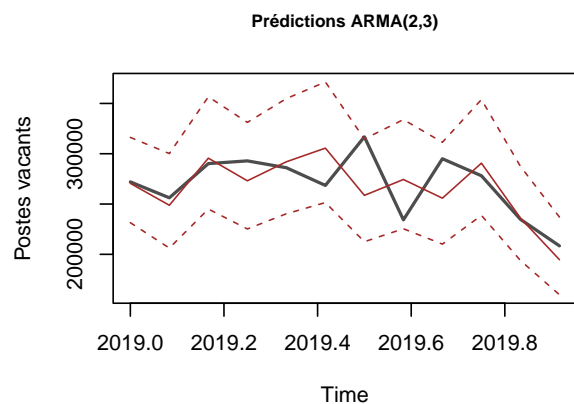
Afin de déterminer le modèle le plus apte à la prédiction parmi les modèles ARMA identifiés,

- on procède à la prédiction des valeurs de la série de test (année 2019) pour chacun des modèles ARMA,
- on compare visuellement les valeurs prédites et les intervalles de confiance à 95% associés
- on compare les critères de sélection de modèles AIC, AICc et BIC

On note que pour la visualisation des valeurs, on procédera à la "reconstruction" de la série :

aux valeurs prédites des résidus de la série, **on ajoute la tendance et la saisonnalité** préalablement retirées pour la modélisation ARMA.

```
# Prédications de la série test (année 2019, horizon=12)
predictions.test.arma.2_3 <- forecast(modele.arma.2_3, h=12)
predictions.test.arma.2_2 <- forecast(modele.arma.2_2, h=12)
predictions.test.arma.2_1 <- forecast(modele.arma.2_1, h=12)
predictions.test.arma.2_0 <- forecast(modele.arma.2_0, h=12)
```



### ARMA(p,q) – superposition des prédictions sur l'année 2019

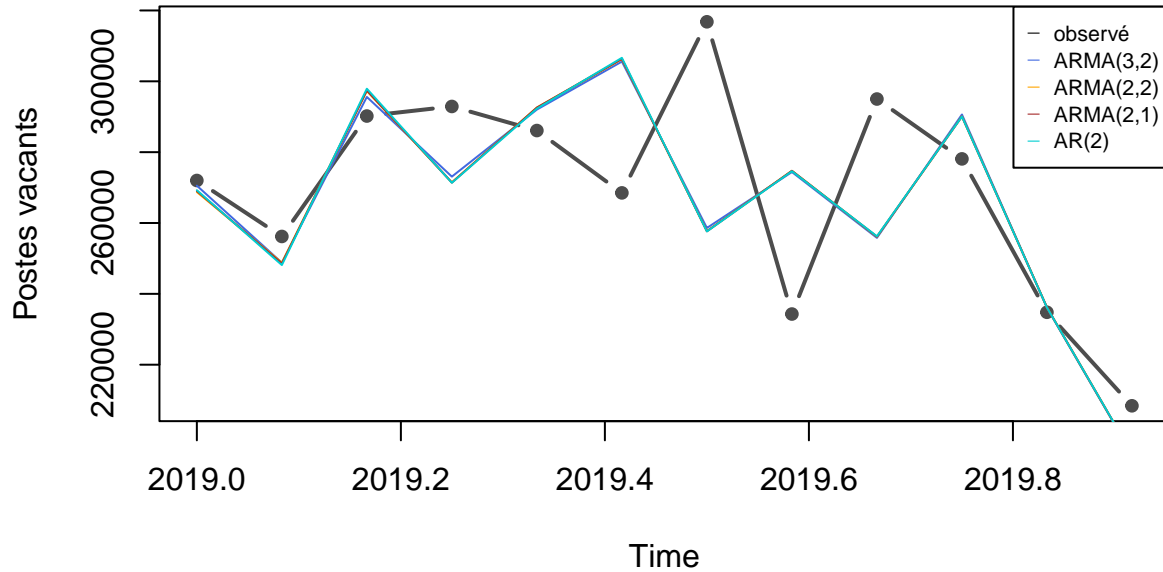


Table 2: Critères de sélection des modèles ARMA

	AIC	AICc	BIC
ARMA(2,3)	-229.29	-228.16	-210.58
ARMA(2,2)	-230.82	-229.98	-214.79
ARMA(2,1)	-232.79	-232.2	-219.43
AR(2)	-234.65	-234.25	-223.96

Tous les modèles présentant des résidus après modélisation ARMA assimilables à un bruit blanc, on les considère tous comme de bons candidats pour prédire l'année 2019 (série de test).

Graphiquement, on note

- que pour tous les modèles ARMA considérés, les valeurs observées en 2019 sont toutes dans les intervalles de confiance à 95%
- que les prédictions des modèles considérés sont tellement proches qu'elles se superposent

On départage les modèles sur leurs critères de sélection AIC, AICc et BIC ; pour l'ensemble des critères, c'est le **processus AR(2)** qui est le plus apte à la prédiction.

C'est ce modèle qu'on retiendra pour représenter la catégorie ARMA(p,q).

#### 4.4 Entraînement sur la série d'étude entière (Jan 2009 - Dec 2019)

Afin de nous assurer de disposer du meilleur modèle ARMA sur la série d'étude entière (Jan 2009 - Dec 2019).



- on va entraîner le modèle AR(2) sur la série d'étude entière (Janvier 2009 - Décembre 2019)
- on va utiliser les fonctions *armaselect* et *auto-arima* pour nous indiquer si ce modèle est toujours bien le meilleur modèle sur l'entièreté de la série d'étude.

#### 4.4.1 ARMA(2,0)=AR(2)

On entraîne le modèle AR(2) sur la série d'étude

```
# série d'étude stabilisée,
# diminuée de sa tendance et de sa saisonnalité
job.ts.study.log.residus <-
  diff(diff(job.ts.study.log, lag=12, differences = 1),
        lag=1, differences = 1)

modele.arma.2_0.study <- Arima(job.ts.study.log.residus, order=c(2,0,0))

## Series: job.ts.study.log.residus
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      mean
##      -0.7024  -0.4464  -0.0011
## s.e.   0.0819   0.0812   0.0033
##
## sigma^2 = 0.006037: log likelihood = 136.34
## AIC=-264.68   AICc=-264.33   BIC=-253.56

# Significativité des coefficients
t_stat(modele.arma.2_0.study)

##          ar1      ar2 intercept
## t.stat -8.578734 -5.494761 -0.333412
## p.val   0.000000  0.000000  0.738823

# corrélation des termes
cor.arma(modele.arma.2_0.study)

##          ar1      ar2 intercept
## ar1      1.000000000 0.48568074 0.008286564
## ar2      0.485680735 1.00000000 0.009534750
## intercept 0.008286564 0.00953475 1.000000000

# Test de blancheur des résidus
Box.test(modele.arma.2_0.study$residuals,
         lag=length(modele.arma.2_0.study$residuals)/4,
         type="Box-Pierce")

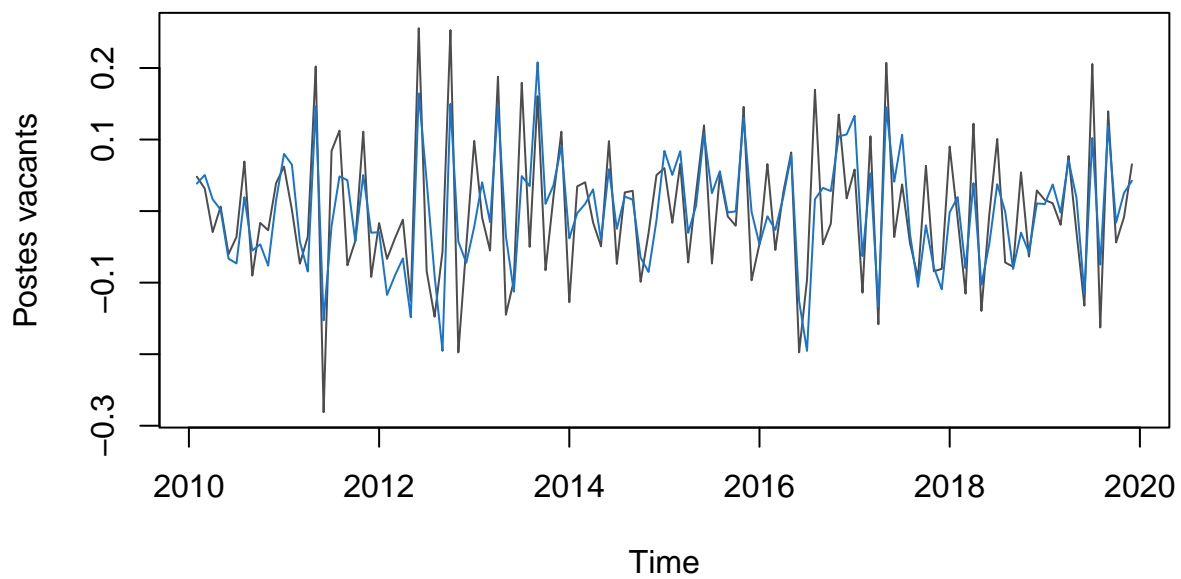
##
## Box-Pierce test
##
## data: modele.arma.2_0.study$residuals
## X-squared = 38.78, df = 29.75, p-value = 0.1242
```

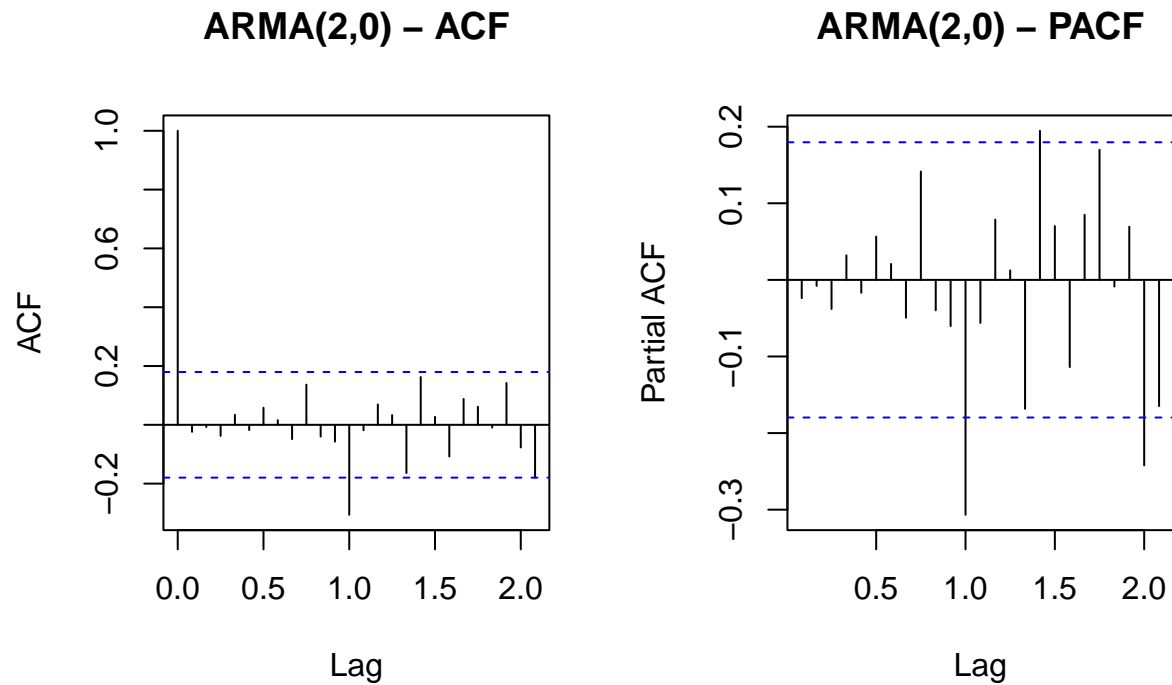
Après ajustement du **modèle AR(2)**,

- **tous les coefficients** sont considérés comme **significatifs**,
- **aucun problème de corrélation** n'est détecté,
- les **résidus** après modélisation sont un **bruit blanc** ;

la **p-valeur du test de blancheur** est **> 5%**, et **on NE rejette PAS l'hypothèse** que les résidus après modélisation par un processus ARMA(2,0) soient un **bruit blanc**.

### Série d'entraînement : modélisation ARMA(2,0)





Les données estimées suivent bien la série temporelle.

ACF et PACF des résidus après **modélisation AR(2)** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Le **rebond de corrélation à chaque saisonnalité** est présent.

#### 4.4.2 auto-sélection du modèle : armaselect

```
armaselect(job.ts.study.log.residus)
```

```
##      p  q      sbc
## [1,] 7 12 -602.5496
## [2,] 2 13 -602.4472
## [3,] 7 13 -601.3186
## [4,] 2 12 -601.1230
## [5,] 6 13 -599.3295
## [6,] 2  0 -599.1994
## [7,] 6 12 -598.8779
## [8,] 8 12 -597.9507
## [9,] 10 12 -597.8513
## [10,] 3 13 -597.6680
```

La majorité des modèles identifiés par **armaselect** n'est **pas exploitable** ; sont proposés des modèles dont l'ordre de la composante moyenne mobile vaut 12 ou 13.

Néanmoins **armaselect** confirme le choix du **modèle AR(2)** positionné en 6ème position du classement des meilleurs modèles selon le critère sbc.

#### 4.4.3 auto-sélection du modèle : **auto.arima**

On cherche à déterminer si la fonction *auto-arima* considère toujours AR(2) comme le meilleur modèle ; pour cette analyse, on prend le soin de préciser à la fonction de ne pas considérer la partie saisonnière résiduelle (*seasonal=FALSE* : on impose de chercher un processus ARMA, sans composante saisonnière).

```
modele.arma.autoarima.study <- auto.arima(job.ts.study.log.residus, seasonal=FALSE)
```

```
## Series: job.ts.study.log.residus
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##      -0.7017  -0.4456
## s.e.   0.0819   0.0812
##
## sigma^2 = 0.005992: log likelihood = 136.28
## AIC=-266.57  AICc=-266.36  BIC=-258.23
```

Auto-arima confirme qu'un **processus ARMA(2,0)** est bien le meilleur modèle selon le critère par défaut AICc.

C'est ce modèle que l'on conserve pour poursuivre l'étude de l'année 2020.

## 4.5 Conclusion

L'étude des modèles ARMA sur la série d'étude par une méthode entraînement/test a mise en évidence les points suivants :

- **ARMA(2,0)=AR(2)** est la meilleure modélisation ARMA(p,q) de la série d'étude dans l'optique d'une prédiction
- **Une saisonnalité résiduelle** dans la série diminuée de sa tendance et de sa saisonnalité n'est pas prise en compte par les modèles ARMA(p,q).

Une modélisation plus complexe comme SARMA/SARIMA serait plus adéquate.

## 5 Modèles SARIMA

On a vu pendant l'identification d'un modèle ARMA qu'une saisonnalité persistait dans les résidus du modèle. La sélection automatique du meilleur modèle selon la fonction `auto.arima` sur la série différenciée de sa tendance et de sa saisonnalité et stabilisée par un log, a fait apparaître une **composante saisonnière d'ordre 12**.

Pour répondre à la modélisation complexe de la saisonnalité, on va utiliser un **modèle SARIMA** en étudiant cette fois-ci la **série temporelle avec tendance et saisonnalité, stabilisée par un log**.

La démarche sera la suivante:

- on identifie “manuellement” et automatiquement des candidats SARIMA ajustés sur la série d'entraînement (Jan 2008 - Dec 2018)
- on identifie le meilleur prédicteur sur la série test (2019)
- on entraîne le modèle sur la série entière en vue d'un travail sur l'année 2020 ;  
on prendra soin dans cette dernière étape de vérifier que le modèle en question est toujours bien identifié comme le meilleur modèle.

### 5.1 Modélisation sur la série d'entraînement (Jan 2009 - Déc 2018)

#### 5.1.1 Modèle SARIMA initial

Pour rappel, lors de l'étude des modèles ARMA, `auto.arima` avait détecté dans la série d'entraînement stabilisée, diminuée de sa tendance et de sa saisonnalité, une **composante saisonnière d'ordre 12 de type ARMA(2,0)**.

Sachant que par ailleurs la tendance se modélisait par un modèle AR(2), on va considérer le modèle **SARIMA(2,1,0)(2,1,0)[12]** comme **modèle initial**.

```
modele.sarima.2_1_0.2_1_0 <- Arima(job.ts.train.log, order=c(2,1,0), seasonal = c(2,1,0))
```

```
## Series: job.ts.train.log
## ARIMA(2,1,0)(2,1,0)[12]
##
## Coefficients:
##          ar1          ar2          sar1          sar2
##      -0.7530  -0.4906  -0.4454  -0.2498
## s.e.    0.0874   0.0846   0.1014   0.1063
##
## sigma^2 = 0.005165: log likelihood = 130.13
## AIC=-250.26   AICc=-249.66   BIC=-236.89
```

```
# Significativité des coefficients
t_stat(modele.sarima.2_1_0.2_1_0)
```

```
##          ar1          ar2          sar1          sar2
## t.stat -8.620398 -5.798091 -4.391907 -2.350528
## p.val   0.000000  0.000000  0.000011  0.018747
```

```
# corrélation des termes
```

```
cor.arma(modele.sarima.2_1_0.2_1_0)
```

```
##          ar1          ar2          sar1          sar2
## ar1  1.0000000  0.51922156  0.2911099  0.16551922
## ar2  0.5192216  1.00000000  0.1514036  0.01536422
## sar1 0.2911099  0.15140361  1.0000000  0.44026835
## sar2 0.1655192  0.01536422  0.4402683  1.00000000
```

```
# Test de blancheur des résidus
```

```
Box.test(modele.sarima.2_1_0.2_1_0$residuals,
         lag=length(modele.sarima.2_1_0.2_1_0$residuals)/4,
         type="Box-Pierce")
```

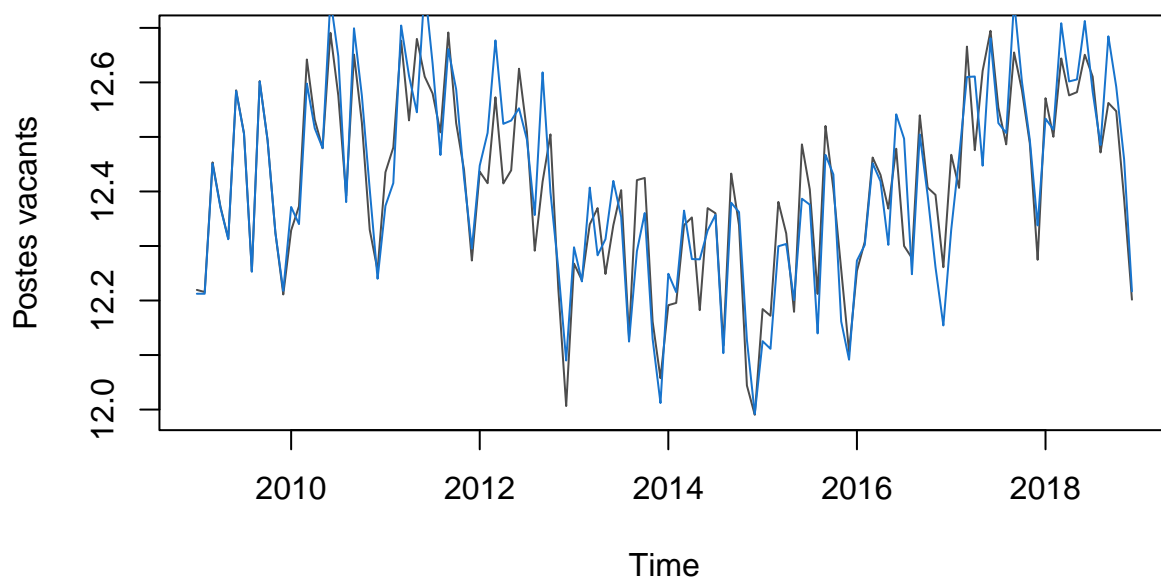
```
##
## Box-Pierce test
##
## data:  modele.sarima.2_1_0.2_1_0$residuals
## X-squared = 35.329, df = 30, p-value = 0.2309
```

Après ajustement du **modèle SARIMA(2,1,0)(2,1,0)[12]**,

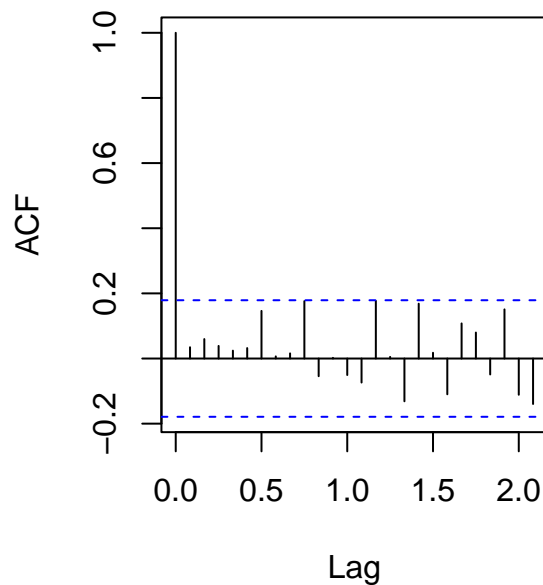
- tous les coefficients sont considérés comme **significatifs**,
- aucun problème de **corrélacion** n'est détecté,
- les **résidus** après modélisation sont un **bruit blanc** ;

la **p-valeur** du **test de blancheur** est **> 5%**, et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus SARIMA(2,1,0)(2,1,0)[12] soient un **bruit blanc**.

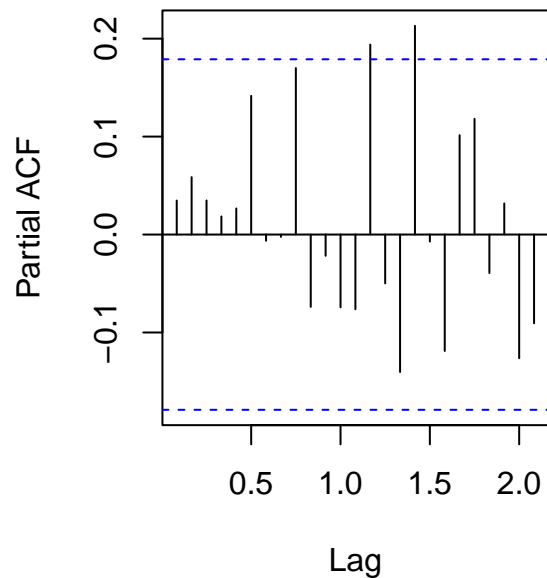
## modélisation SARIMA(2,1,0)(2,1,0)[12]



## SARIMA(2,1,0)(2,1,0)[12] – ACF



## SARIMA(2,1,0)(2,1,0)[12] – PACF



Les données estimées suivent bien la série temporelle.

ACF et PACF des résidus après **modélisation SARIMA(2,1,0)(2,1,0)[12]** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

On ne note **plus de rebond de corrélation aux saisonnalités**. On peut considérer que toute l'information portée par la série a été expliquée.

De par ses **coefficients significatifs** et la **blancheur des résidus après modélisation**, le **modèle SARIMA(2,1,0)(2,1,0)[12]** est un **bon modèle SARIMA** issue d'une recherche "manuelle".

Voyons si une sélection automatique du modèle SARIMA converge vers ce même modèle.

### 5.1.2 Auto-sélection du modèle SARIMA

```
modele.sarima.auto <- auto.arima(job.ts.train.log)
```

```
## Series: job.ts.train.log
## ARIMA(2,1,0)(2,1,0)[12]
##
## Coefficients:
##      ar1      ar2      sar1      sar2
##    -0.7530 -0.4906 -0.4454 -0.2498
## s.e.   0.0874   0.0846   0.1014   0.1063
```

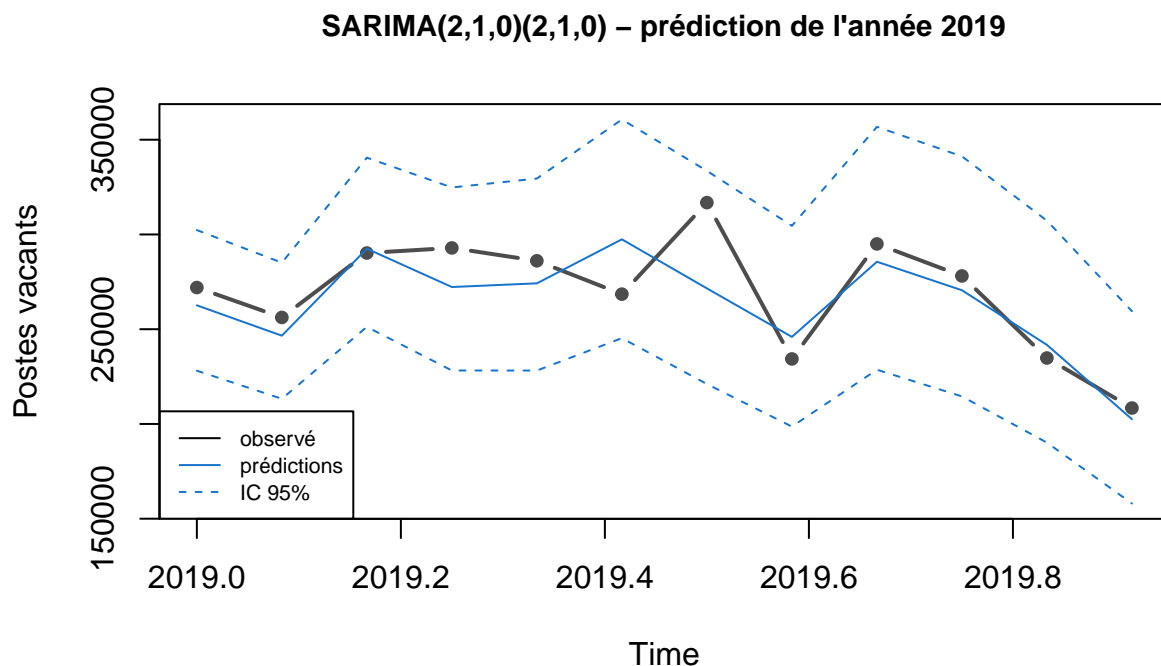
```
##
## sigma^2 = 0.005165: log likelihood = 130.13
## AIC=-250.26 AICc=-249.66 BIC=-236.89
```

La fonction *auto.arima* confirme le choix du modèle **SARIMA(2,1,0)(2,1,0)[12]**

C'est donc ce modèle qu'on va choisir pour estimer l'année test 2019.

## 5.2 Prédiction de la série test 2019

```
predictions.test.sarima_2_1_0.2_1_0 <- forecast(modele.sarima.2_1_0.2_1_0, h=12)
```



Graphiquement, on note

- que les valeurs observées en 2019 sont toutes dans l'intervalle de confiance à 95% du modèle
- que les valeurs prédites suivent bien la tendance réellement observée, notamment en tout début et en fin d'année.

On retient ce modèle pour la suite de l'étude.

## 5.3 Entraînement sur la série d'étude entière (Jan 2009 - Dec 2019)

Afin de nous assurer de disposer du meilleur modèle SARIMA sur la série d'étude entière

- on va entraîner le modèle **SARIMA(2,1,0)(2,1,0)[12]** sur la série d'étude entière (Janvier 2009 - Décembre 2019)



- on va utiliser la fonction *auto-arima* pour préciser si ce modèle est toujours bien le meilleur modèle sur l'entièreté de la série d'étude.
- Si ce n'est pas le cas, on affinera le modèle et on gardera le meilleur modèle au sens des critères de sélection AICc ou BIC

### 5.3.1 SARIMA(2,1,0)(2,1,0)[12]

On entraîne le modèle SARIMA(2,1,0)(2,1,0)[12] sur la série d'étude stabilisée (avec tendance et saisonnalité).

```
modele.sarima.2_1_0.2_1_0.study <- Arima(job.ts.study.log, order=c(2,1,0), seasonal=c(2,1,0))
```

```
## Series: job.ts.study.log
## ARIMA(2,1,0)(2,1,0)[12]
##
## Coefficients:
##          ar1      ar2      sar1      sar2
##      -0.8067  -0.5101  -0.4408  -0.2459
## s.e.   0.0815   0.0801   0.0963   0.0943
##
## sigma^2 = 0.005047: log likelihood = 146.06
## AIC=-282.12  AICc=-281.59  BIC=-268.23
```

```
# Significativité des coefficients
```

```
t_stat(modele.sarima.2_1_0.2_1_0.study)
```

```
##          ar1      ar2      sar1      sar2
## t.stat -9.90292 -6.366173 -4.576606 -2.607461
## p.val   0.00000  0.000000  0.000005  0.009122
```

```
# corrélation des termes
```

```
cor.arma(modele.sarima.2_1_0.2_1_0.study)
```

```
##          ar1      ar2      sar1      sar2
## ar1  1.0000000  0.55435956  0.2847843  0.15455914
## ar2  0.5543596  1.00000000  0.2184642  0.03247674
## sar1 0.2847843  0.21846416  1.0000000  0.41020469
## sar2 0.1545591  0.03247674  0.4102047  1.00000000
```

```
# Test de blancheur des résidus
```

```
Box.test(modele.sarima.2_1_0.2_1_0.study$residuals,
          lag=length(modele.arma.2_0.study$residuals)/4,
          type="Box-Pierce")
```

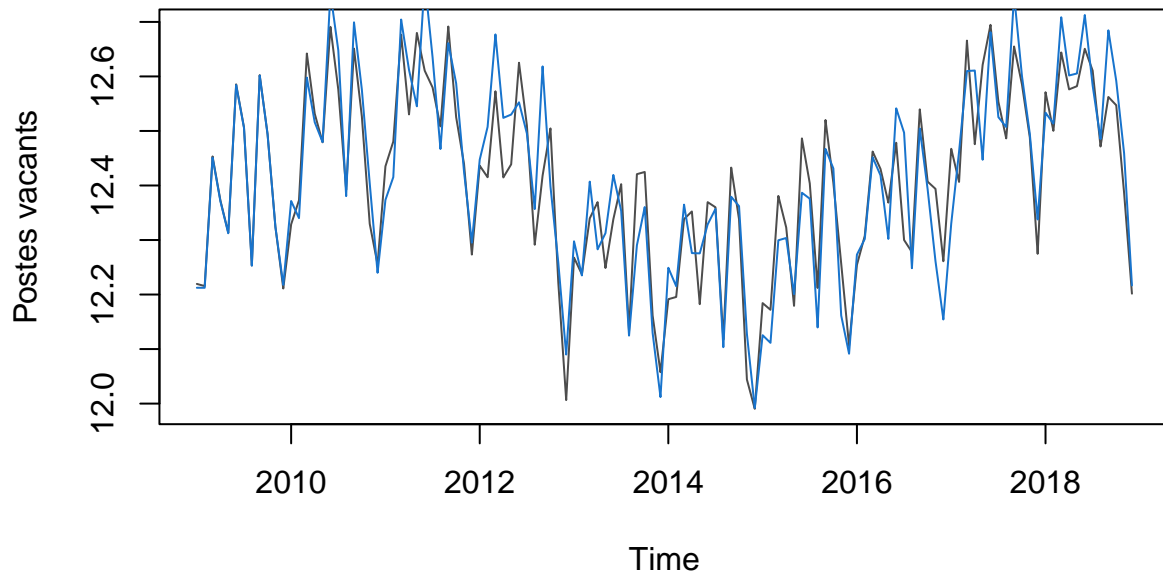
```
##
## Box-Pierce test
##
## data: modele.sarima.2_1_0.2_1_0.study$residuals
## X-squared = 37.134, df = 29.75, p-value = 0.1653
```

Après ajustement du modèle SARIMA(2,1,0)(2,1,0)[12],

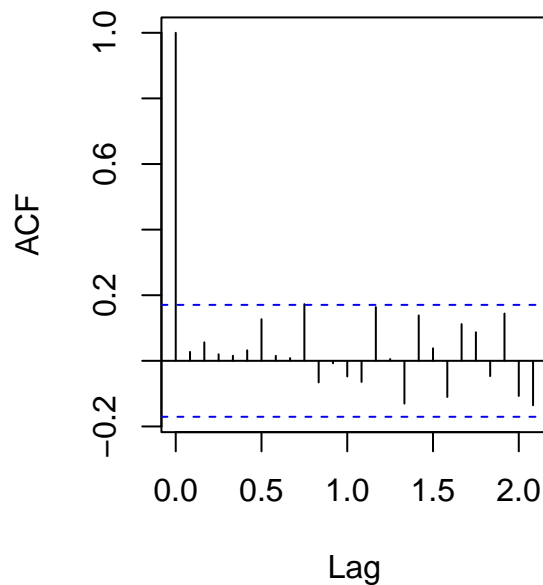
- tous les coefficients sont considérés comme **significatifs**,
- aucun problème de **corrélacion** n'est détecté,
- les **résidus** après modélisation sont un **bruit blanc** ;

la **p-valeur** du **test de blancheur** est **> 5%**, et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus  $SARIMA(2,1,0)(2,1,0)[12]$  soient un **bruit blanc**.

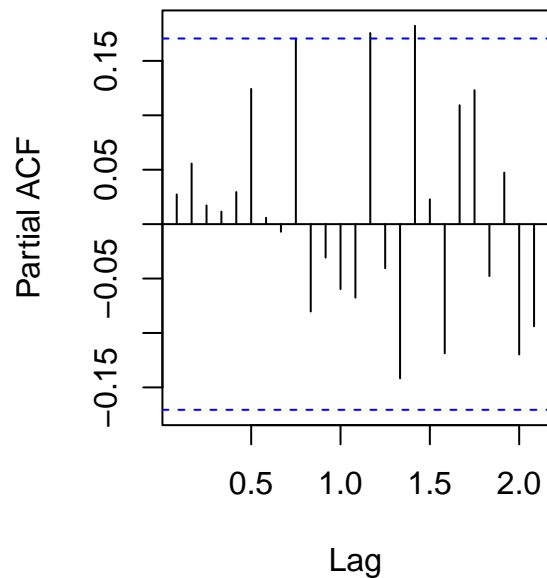
### modélisation $SARIMA(2,1,0)(2,1,0)[12]$



## SARIMA(2,1,0)(2,1,0)[12] – ACF



## SARIMA(2,1,0)(2,1,0)[12] – PACF



Par ailleurs, les données suivent plutôt bien la série temporelle.

ACF et PACF des résidus après modélisation SARIMA(2,1,0)(2,1,0)[12] présentent des caractéristiques satisfaisantes (décroissance exponentielle, des valeurs respectant la bande de significativité de 95%).

### 5.3.2 auto-sélection du modèle : `auto.arima`

On cherche à déterminer si la fonction *auto-arima* considère toujours **SARIMA(2,1,0)(2,1,0)[12]** comme le meilleur modèle sur la série d'étude entière.

```
modele.sarima.autoarima.study <- auto.arima(job.ts.study.log)
```

```
## Series: job.ts.study.log
## ARIMA(0,1,2)(0,1,1)[12]
##
## Coefficients:
##      ma1      ma2      sma1
##    -0.8583  0.3486 -0.8794
## s.e.   0.0850  0.0857  0.1795
##
## sigma^2 = 0.00442: log likelihood = 146.56
## AIC=-285.12  AICc=-284.76  BIC=-274
```

```
# Significativité des coefficients
t_stat(modele.sarima.autoarima.study)
```

```
##      ma1      ma2      sma1
## t.stat -10.09243  4.069938 -4.899491
## p.val   0.00000  0.000047  0.000001
```

```
# corrélation des termes
cor.arma(modele.sarima.autoarima.study)
```

```
##           ma1           ma2           sma1
## ma1  1.00000000 -0.61147768 -0.01434167
## ma2 -0.61147768  1.00000000 -0.09042106
## sma1 -0.01434167 -0.09042106  1.00000000
```

```
# Test de blancheur des résidus
Box.test(modele.sarima.autoarima.study$residuals,
         lag=length(modele.sarima.autoarima.study$residuals)/4,
         type="Box-Pierce")
```

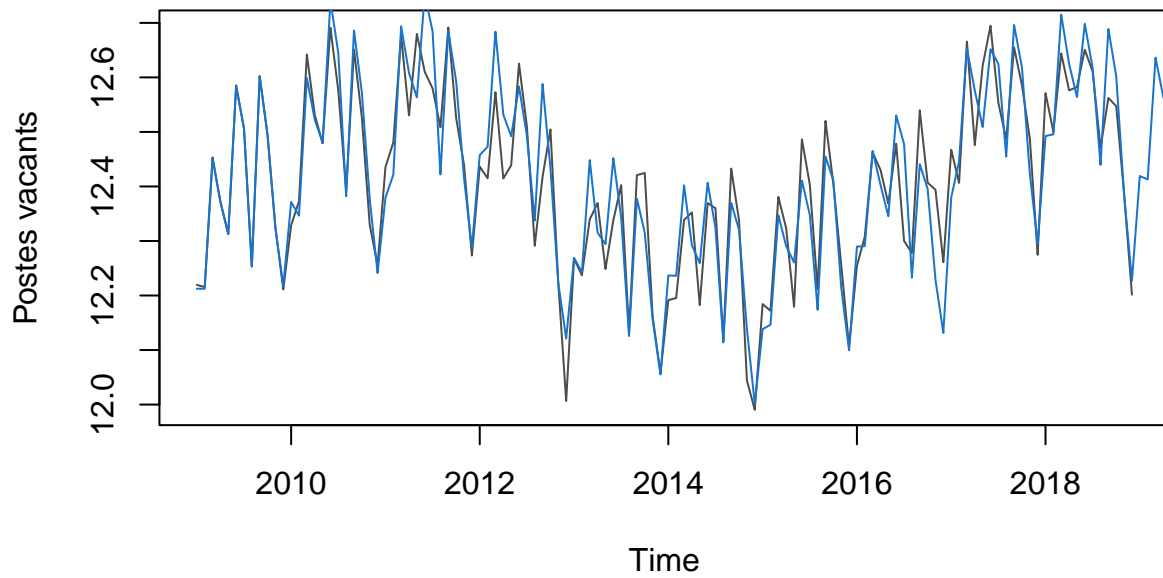
```
##
## Box-Pierce test
##
## data:  modele.sarima.autoarima.study$residuals
## X-squared = 102.08, df = 33, p-value = 5.538e-09
```

Auto-arima propose un **modèle alternatif SARIMA(0,1,2)(0,1,2)** qu'il considère le meilleur modèle selon le critère par défaut AICc.

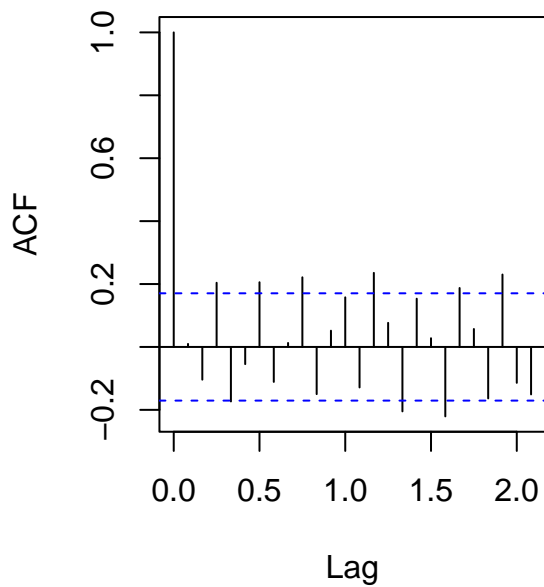
Après ajustement du **modèle SARIMA(0,1,2)(0,1,2)[12]** sur la série d'étude entière, incluant sa tendance et de sa saisonnalité, et stabilisée par un log :

- tous les coefficients sont considérés comme **significatifs**,
- aucun problème de corrélation n'est détecté,
- les résidus après modélisation NE sont PAS un **bruit blanc** ;  
la p-valeur du test de blancheur est  $< 5\%$ , et on rejette l'hypothèse que les résidus après modélisation par un processus SARIMA(0,1,2)(0,1,2) soient un **bruit blanc**.

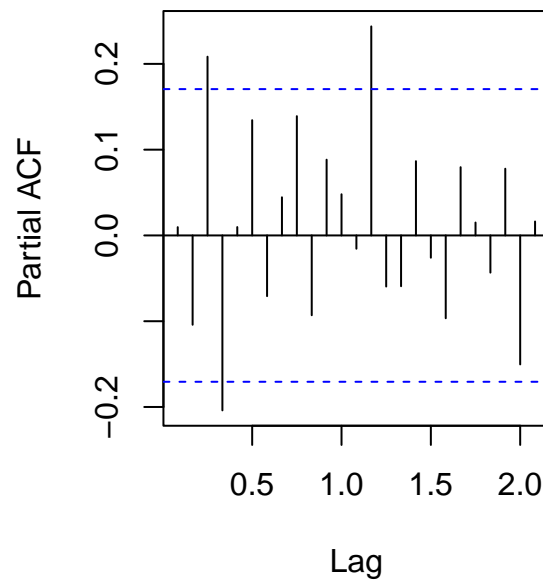
### modélisation SARIMA(2,1,0)(2,1,0)[12]



### SARIMA(0,1,2)(0,1,2)[12] – ACF



### SARIMA(0,1,2)(0,1,2)[12] – PACF



La fonction d'auto-corrélation des résidus après modélisation SARIMA(0,1,2)(0,1,2)[12] présente les caractéristiques suivantes

- les décroissances exponentielles sur ACF et PACF sont satisfaisantes

- néanmoins, les auto-corrélations ne sont visiblement pas nulles (ce qui est confirmé par le test de blancheur)
- la PACF présente des valeurs significativement **non nulles aux décalages 3, 4 et 14** ; laissant supposer qu'une **composante auto-régressive** améliorerait la compréhension de la série.

Pour **affiner le modèle proposé par auto.arima**, on introduit une **composante AR(3)** pour prendre en considération les accidents de la PACF.

### 5.3.3 SARIMA(3,1,2)(0,1,2)[12]

On entraîne le modèle SARIMA(3,1,2)(0,1,2)[12] sur la série d'étude stabilisée (avec tendance et saisonnalité)

```
modele.sarima.3_1_2.0_1_2.study <- Arima(job.ts.study.log, order=c(3,1,2), seasonal=c(0,1,2))
```

```
## Series: job.ts.study.log
## ARIMA(3,1,2)(0,1,2)[12]
##
## Coefficients:
##          ar1          ar2          ar3          ma1          ma2          sma1          sma2
##      -1.7848  -1.7096  -0.6253   1.2973   0.9997  -0.9624  -0.0372
## s.e.    0.0739   0.0857   0.0735   0.0527   0.0713   0.2887   0.1130
##
## sigma^2 = 0.003131: log likelihood = 161.4
## AIC=-306.79  AICc=-305.48  BIC=-284.56
```

*# Significativité des coefficients*

```
t_stat(modele.sarima.3_1_2.0_1_2.study)
```

```
##          ar1          ar2          ar3          ma1          ma2          sma1          sma2
## t.stat -24.15564 -19.94437 -8.506653  24.62616  14.01345 -3.334031 -0.329632
## p.val   0.00000   0.00000  0.000000  0.00000  0.00000  0.000856  0.741678
```

*# corrélation des termes*

```
cor.arma(modele.sarima.3_1_2.0_1_2.study)
```

```
##          ar1          ar2          ar3          ma1          ma2
## ar1   1.000000000  0.98438091  0.975697536 -0.122272011 -5.836595e-03
## ar2   0.984380909  1.00000000  0.985795722 -0.084043609 -3.757690e-03
## ar3   0.975697536  0.98579572  1.000000000 -0.107688534 -3.742136e-03
## ma1  -0.122272011 -0.08404361 -0.107688534  1.000000000  8.890891e-01
## ma2  -0.005836595 -0.00375769 -0.003742136  0.889089137  1.000000e+00
## sma1  0.051341740  0.06260985  0.060284375 -0.007081139 -8.081016e-05
## sma2 -0.132163256 -0.16113503 -0.154953510  0.017778936  1.640868e-04
##          sma1          sma2
## ar1   5.134174e-02 -0.1321632565
## ar2   6.260985e-02 -0.1611350258
## ar3   6.028437e-02 -0.1549535105
## ma1  -7.081139e-03  0.0177789357
## ma2  -8.081016e-05  0.0001640868
## sma1  1.000000e+00 -0.3052340598
## sma2 -3.052341e-01  1.0000000000
```

```
# Test de blancheur des résidus
Box.test(modele.sarima.3_1_2.0_1_2.study$residuals,
        lag=length(modele.sarima.3_1_2.0_1_2.study$residuals)/4,
        type="Box-Pierce")
```

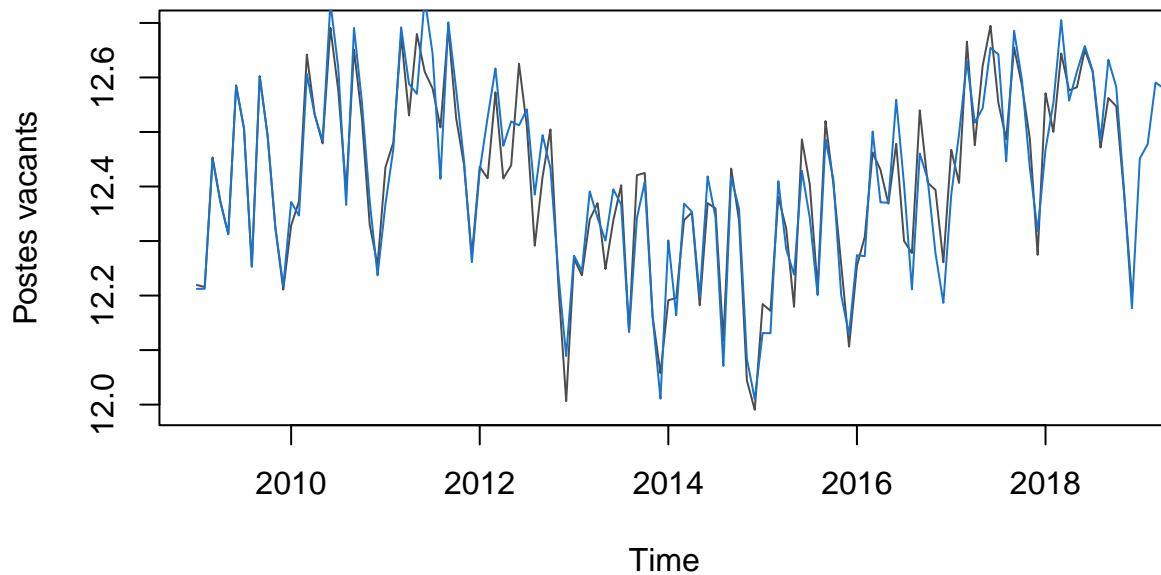
```
##
## Box-Pierce test
##
## data: modele.sarima.3_1_2.0_1_2.study$residuals
## X-squared = 38.615, df = 33, p-value = 0.2307
```

Après ajustement du **modèle SARIMA(3,1,2)(0,1,2)[12]** sur la série d'étude entière stabilisée par un log :

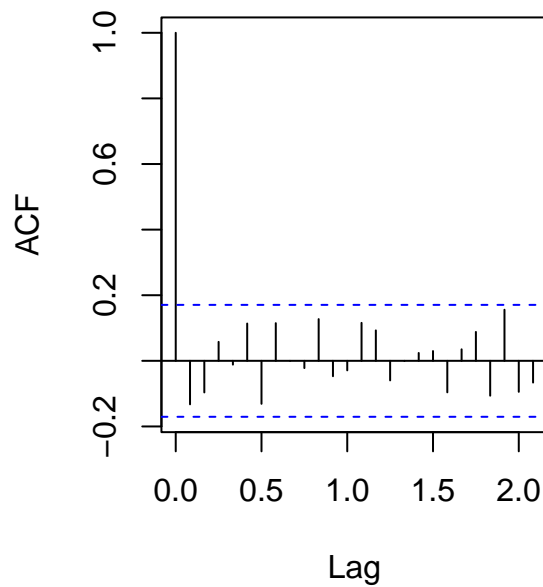
- parmi les 7 paramètres du modèle, le **paramètre SMA2** est considéré comme **non significatif**,
- des **problèmes de corrélation** sont détectés (des valeurs  $> 0.9$ ),
- les **résidus** après modélisation sont maintenant un **bruit blanc** ;

la **p-valeur du test de blancheur** est  $> 5\%$ , et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus SARIMA(3,1,2)(0,1,2)[12] soient un **bruit blanc**.

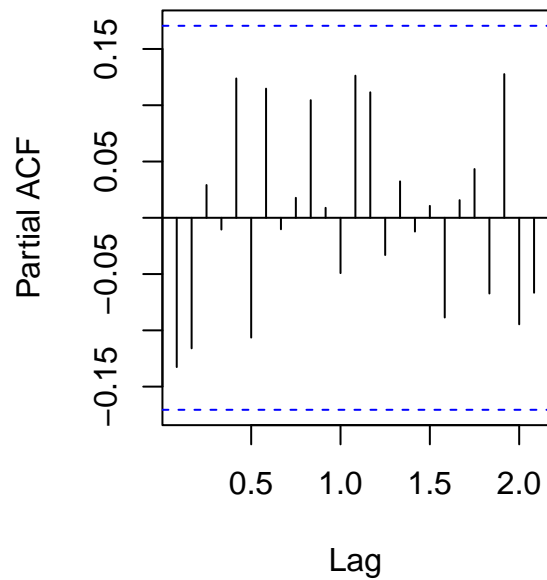
### modélisation SARIMA(2,1,0)(2,1,0)[12]



### SARIMA(3,1,2)(0,1,2)[12] – ACF



### SARIMA(3,1,2)(0,1,2)[12] – PACF



Avec l'ajout de la composante auto-régressive, ACF et PACF sont maintenant satisfaisantes.

Les valeurs ajustées suivent bien la série temporelle.

ACF et PACF des résidus après **SARIMA(3,1,2)(0,1,2)[12]** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Pour la suite, on prend cependant le parti de **supprimer le paramètre SMA2** considéré comme non significatif dans le modèle.

#### 5.3.4 SARIMA(3,1,2)(0,1,1)[12]

On entraîne le modèle SARIMA(3,1,2)(0,1,1)[12] sur la série d'étude stabilisée (avec tendance et saisonnalité)

```
modele.sarima.3_1_2_0_1_1.study <- Arima(job.ts.study.log, order=c(3,1,2), seasonal=c(0,1,1))
```

```
## Series: job.ts.study.log
## ARIMA(3,1,2)(0,1,1)[12]
##
## Coefficients:
##      ar1      ar2      ar3      ma1      ma2      sma1
##     -1.7881  -1.7143  -0.6291  1.2977  0.9999  -0.9967
## s.e.    0.0730   0.0843   0.0725  0.0520  0.0707   0.5155
##
## sigma^2 = 0.003095: log likelihood = 161.34
## AIC=-308.68   AICc=-307.67   BIC=-289.23
```



```
# Significativité des coefficients
```

```
t_stat(modele.sarima.3_1_2.0_1_1.study)
```

```
##           ar1           ar2           ar3           ma1           ma2           sma1
## t.stat -24.47864 -20.32825 -8.681854 24.93614 14.1504 -1.933473
## p.val   0.00000  0.00000  0.000000  0.00000  0.0000  0.053178
```

```
# correlation des termes
```

```
cor.arma(modele.sarima.3_1_2.0_1_1.study)
```

```
##           ar1           ar2           ar3           ma1           ma2
## ar1  1.000000000  0.985135378  0.976126107 -0.120510552 -2.270388e-03
## ar2  0.985135378  1.000000000  0.985884454 -0.083713751 -1.470960e-03
## ar3  0.976126107  0.985884454  1.000000000 -0.107437414 -1.497188e-03
## ma1 -0.120510552 -0.083713751 -0.107437414  1.000000000  8.852873e-01
## ma2 -0.002270388 -0.001470960 -0.001497188  0.885287318  1.000000e+00
## sma1 -0.001111334 -0.001385961  0.000225248 -0.003446331 -6.904668e-05
##           sma1
## ar1 -1.111334e-03
## ar2 -1.385961e-03
## ar3  2.252480e-04
## ma1 -3.446331e-03
## ma2 -6.904668e-05
## sma1 1.000000e+00
```

```
# Test de blancheur des résidus
```

```
Box.test(modele.sarima.3_1_2.0_1_1.study$residuals,
         lag=length(modele.sarima.3_1_2.0_1_1.study$residuals)/4,
         type="Box-Pierce")
```

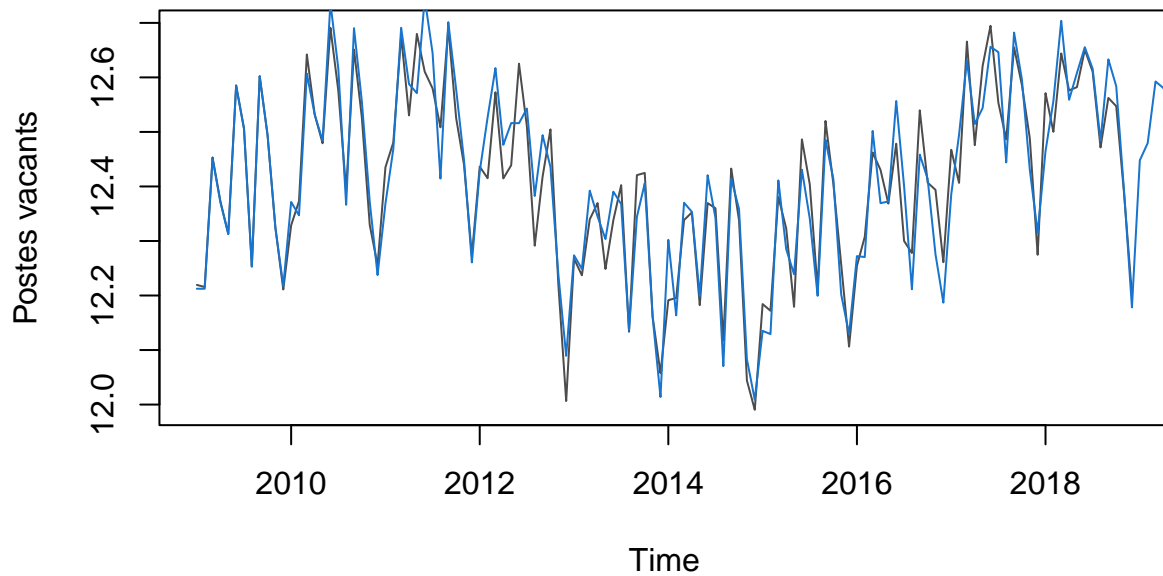
```
##
## Box-Pierce test
##
## data:  modele.sarima.3_1_2.0_1_1.study$residuals
## X-squared = 37.924, df = 33, p-value = 0.2548
```

Après ajustement du modèle **SARIMA(3,1,2)(0,1,1)[12]**,

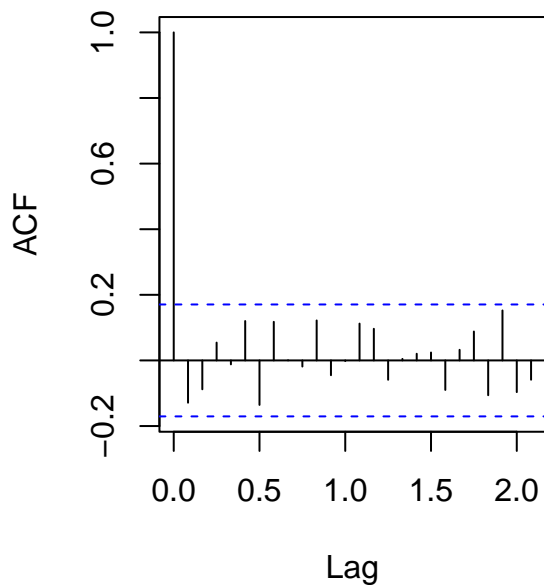
- **tous les coefficients** sont considérés comme **significatifs** (y compris SMA1 dont la p-valeur frôle les 5%),
- **des problèmes de corrélation** sont détectés entre les termes AR (valeurs > 0.9),
- les **résidus** après modélisation sont un **bruit blanc** ;

la **p-valeur du test de blancheur** est > 5%, et on **NE rejette PAS** l'hypothèse que les résidus après modélisation par un processus SARIMA(3,1,2)(0,1,1)[12] soient un **bruit blanc**.

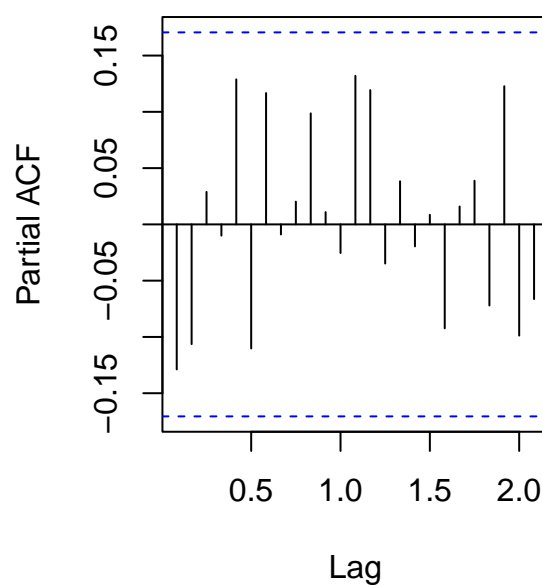
### modélisation SARIMA(2,1,0)(2,1,0)[12]



### SARIMA(3,1,2)(0,1,1)[12] – ACF



### SARIMA(3,1,2)(0,1,1)[12] – PACF



Les valeurs ajustées suivent bien la série temporelle.

ACF et PACF des résidus après **SARIMA(3,1,2)(0,1,1)[12]** présentent des caractéristiques satisfaisantes pour un processus stationnaire :

- décroissance exponentielle,
- des valeurs respectant la bande de significativité de 95%

Bien que des problèmes de corrélations soient constatés entre les termes AR, on décide de les maintenir puisque les termes AR ont été spécifiquement introduits pour lever des problèmes liés à la significativité des termes de la PACF.

On conserve donc le modèle SARIMA(3,1,2)(0,1,1)[12].

### 5.3.5 Choix du modèle

Les modèles SARIMA candidats pour expliquer la série d'étude entière sont les suivants

- SARIMA(2,1,0)(2,1,0)[12] issu de l'identification du modèle sur les séries d'entraînement / test
- SARIMA(0,1,2)(0,1,2)[12] issu de la sélection du meilleur modèle par auto.arima selon le critère AICc (les résidus ne sont pas un bruit blanc)
- SARIMA(3,1,2)(0,1,2)[12] issu de l'amélioration du modèle auto.arima pour effacer les valeurs non significativement nulles de la PACF.
- SARIMA(3,1,2)(0,1,1)[12] issu de l'optimisation du modèle précédent.

Table 3: Résumé des modèles SARIMA sur la série d'étude

	AIC	AICc	BIC
SARIMA(2,1,1)(2,1,1)[12]	-282.12	-281.59	-268.23
SARIMA(0,1,2)(0,1,2)[12]	-285.12	-284.76	-274
SARIMA(3,1,2)(0,1,2)[12]	-306.79	-305.48	-284.56
SARIMA(3,1,2)(0,1,1)[12]	-308.68	-307.67	-289.23

On note que pour l'ensemble des critères, le **modèle SARIMA(3,1,2)(0,1,1)[12]** est le plus apte à la prédiction.

C'est ce modèle qu'on retiendra pour représenter la catégorie SARIMA.

## 6 Résumé des modèles candidats et comparaison

3 types de modélisation ont été entrepris pour expliquer la série temporelle d'étude de Jan 2008 à Décembre 2019.

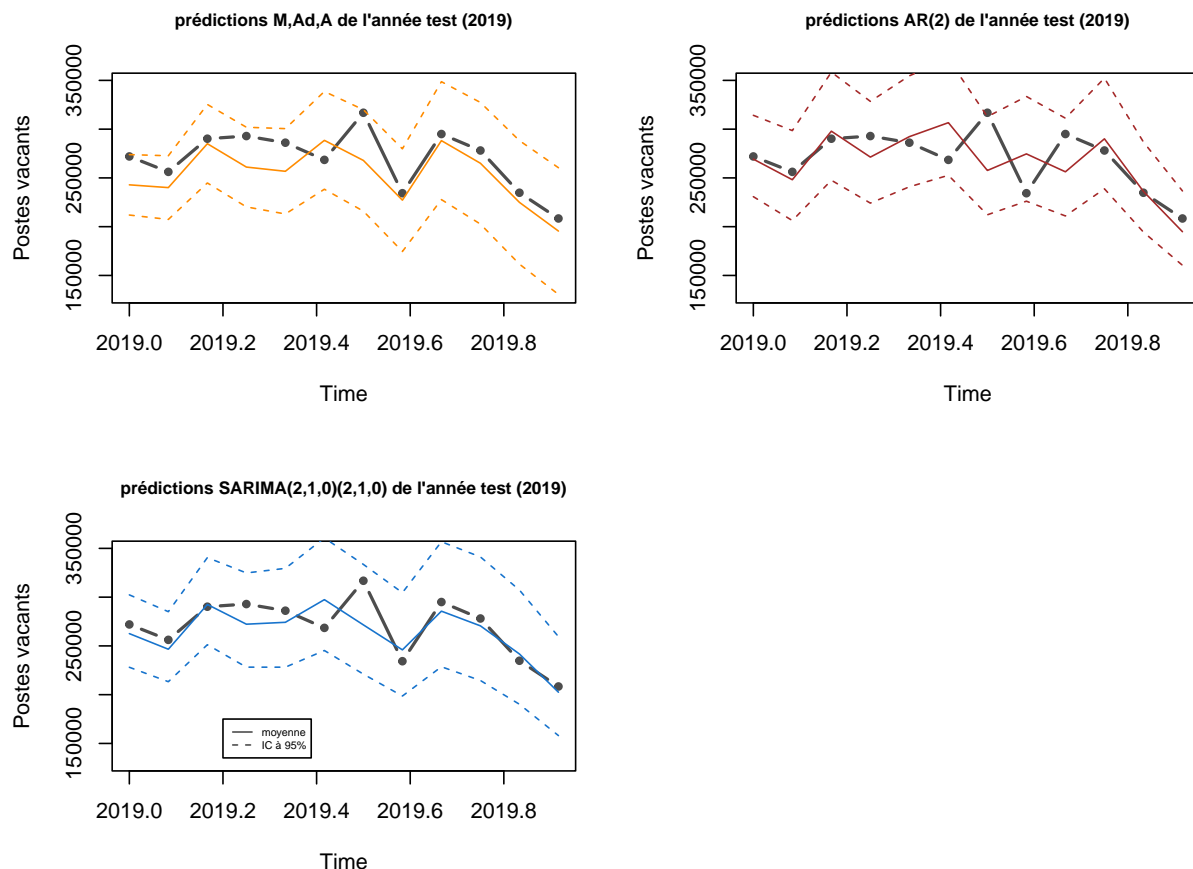
Pour chacun, le meilleur modèle pour la prédiction a été identifié, à savoir

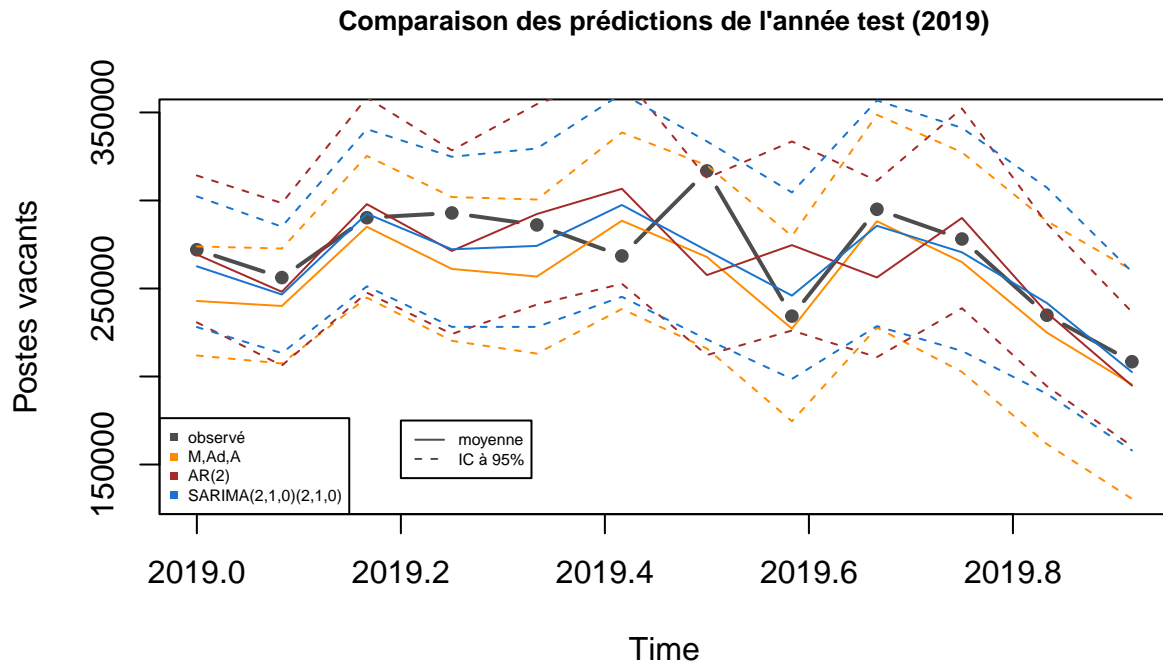
- Pour le lissage exponentiel : (**M**, **Ad**, **A**)
- Pour la catégorie ARMA(p,q) : **AR(2,0)**
- Pour la catégorie SARIMA :
  - **SARIMA(2,1,0)(2,1,0)[12]** lorsqu'il s'agit de prédire la série test 2019 depuis un modèle entraîné sur la série temporelle Jan 2009 - Déc 2018
  - **SARIMA(3,1,2)(0,1,1)[12]** lorsqu'il s'agit de considérer la série d'étude entière

### 6.1 Comparaison des modèles sur la série temporelle de test (2019)

Comparons aux valeurs réellement observées en 2019 les prédictions des modèles :

- Lissage exponentiel (**M**, **Ad**, **A**)
- **AR(2,0)** (les valeurs prédites se sont vues ajouter tendance et saisonnalité)
- **SARIMA(2,1,0)(2,1,0)[12]** (les valeurs “log” prédites ont été transformées par une “exponentielle”)





Quelques constats :

- les valeurs observées entrent dans les intervalles de confiance à 95% des 3 modèles (hormis peut-être un défaut sur le modèle AR(2))
- les intervalles de confiance à 95% sont de largeur similaire pour les 3 modèles
- les 3 modèles proposent des prédictions d'Avril à Juillet plutôt éloignées des valeurs observées
- le lissage exponentiel propose des prédictions proches des valeurs observées d'Août à Décembre, mais est peu précis sur le début d'année de Janvier à Mars
- inversement, le modèle AR(2) propose des prédictions de Janvier à Mars proches des valeurs observées mais est moins juste d'août à décembre
- le modèle SARIMA(2,1,0)(2,1,0)[12] propose des valeurs justes à la fois de Janvier à Mars et d'août à décembre, faisant ainsi mieux que ses 2 concurrents, tout en proposant un intervalle de confiance à 95% aussi resserré que les autres modèles

Au final, le modèle SARIMA(2,1,0)(2,1,0)[12] est le plus efficace pour prédire l'année de test 2019.

## 6.2 Années COVID - prédiction et comparaison des modèles

### 6.2.1 Prédiction de l'année 2020

On note que pour l'étude comparative sur l'année 2020, on retire le modèle AR(2) qui prédit les résidus d'une série dépourvue de tendance et saisonnalité ; la tendance et la saisonnalité n'étant pas non modélisées dans le cas d'un ARMA(p,q), elles ne sont pas prédictibles.

Il reste 2 modèles pertinents pour la prédiction, construits à partir de la série d'étude de Janvier 2009 à Décembre 2019.

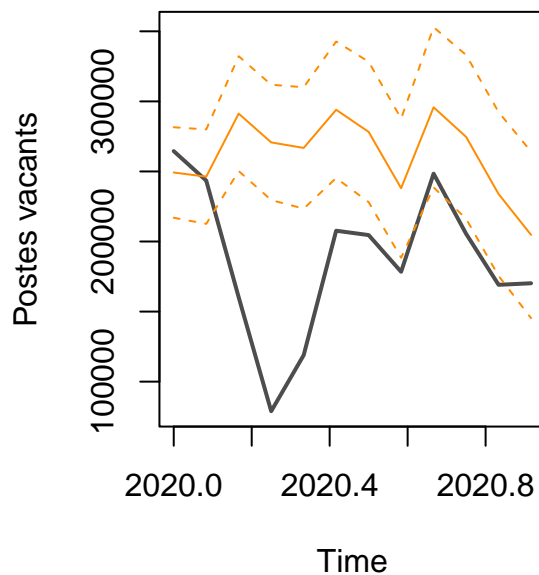
A savoir

- Lissage exponentiel (M, Ad, A)
- SARIMA(3,1,2)(0,1,1)[12] (le plus pertinent sur la série entière)

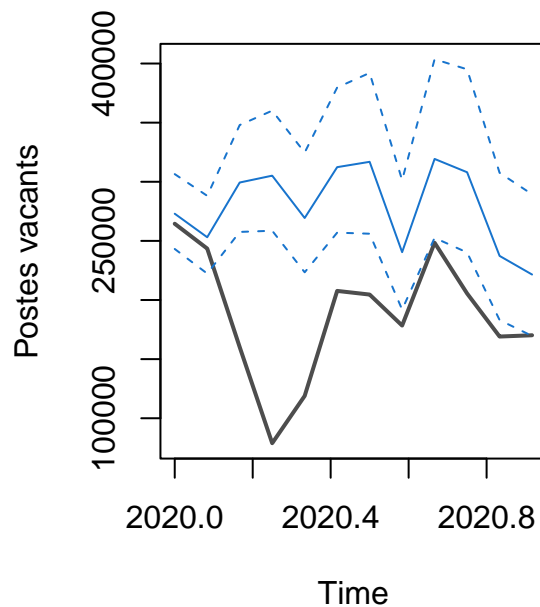
```
horizon=12
```

```
predictions.2020.le.MAdA <- forecast(modele.le.MAdA.study, h=horizon)
predictions.2020.sarima.3_1_2.0_1_1 <- forecast(modele.sarima.3_1_2.0_1_2.study,
                                                h=horizon)
```

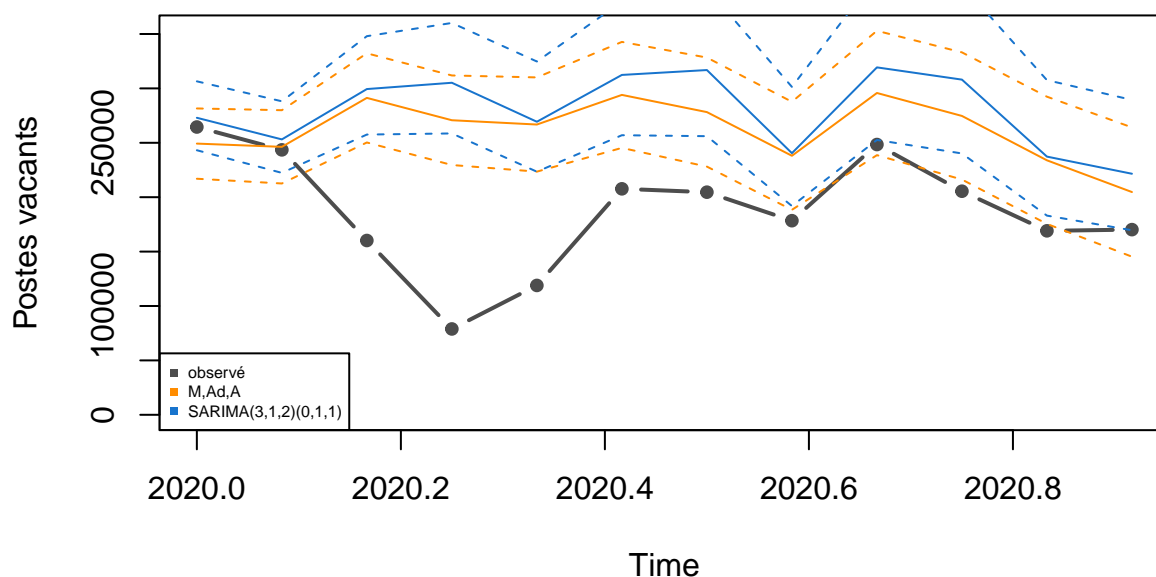
prédictions M,Ad,A de l'année 2020



prédictions SARIMA(3,1,2)(0,1,1) de l'année 20



### Comparaison des prédictions sur l'année 2020



Comme on peut s'y attendre, les prédictions de Janvier et Février 2020 sont correctes.

Le confinement ayant été décidé en Mars 2020,

- le nombre de postes vacants chute fortement en Mars et Avril (et se reprend au mois de mai),
- les modèles “sur-estiment” les valeurs et les prédictions sont sans surprise bien au-dessus des valeurs observées en 2020.

Après le mois de février 2020, les valeurs observées sont en dehors des bandes de prévision à 95%.

Afin de quantifier à quel point la “cassure” éloigne les observations des prédictions, et donc à quel point les modèles sont devenus obsolètes, on va s'intéresser aux probabilités des observations de 2020 connaissant les modèles prédictifs.

Soit  $y_t$  l'observation à un mois donné ; la loi de probabilité d'observation est pour ce mois est  $N(\mu, \sigma^2)$  où

- $\mu$  est la valeur moyenne prédite par le modèle (sortie de la fonction *forecast*)
- $\sigma^2$  est la variance de la prédiction du modèle

La variance  $\sigma^2$  n'est pas fournie en l'état par la sortie de la fonction *forecast* ; néanmoins on peut l'obtenir à partir de la borne supérieure de l'intervalle de prédiction à 80% qui s'écrit dans l'hypothèse de normalité  $BorneSup_{80} = \mu + quantile(0.9)\sigma$ .

Ainsi, pour un mois donné,

- l'écart-type  $\sigma$  de la prédiction vaut  $\frac{BorneSup_{80} - \mu}{q(0.9)}$

- et la probabilité d'observer la valeur réelle ou pire sachant la loi  $N(\mu, \sigma^2)$  est l'ordre-quantile  $F(\frac{y_t - \mu}{\sigma})$  où  $F$  est la fonction de répartition de la loi normale  $N(0, 1)$ .

Ainsi, on peut calculer la probabilité d'occurrence des observations de 2020 sous nos 2 modèles de prédiction:  
**probabilités des observations pour le modèle (M,Ad,A) (en pourcentage %)**

```
# ordres quantiles des observations pour le modèle (M,Ad,A)

sigma.2020.MAdA <-
  (predictions.2020.le.MAdA$upper[,1] - predictions.2020.le.MAdA$mean) / qnorm(0.9)

ordres.2020.MAdA <-
  pnorm((job.ts.2020-predictions.2020.le.MAdA$mean)/sigma.2020.MAdA)
```

```
##          Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 2020 82.25 43.59  0.00  0.00  0.00  0.03  0.20  0.93  5.20  1.02  1.49 12.73
```

En prenant en considération le modèle de lissage exponentiel (M,Ad,A), les probabilités d'observer les valeurs de Mars à Décembre 2020 sont faibles (< 12.73%) voire nulles (de Mars à Août).

**probabilités des observations pour le modèle SARIMA(3,1,2)(0,1,1) (en pourcentage %)**

```
# ordres quantiles des observations pour le modèle SARIMA(3,1,2)(0,1,1)

sigma.2020.sarima.3_1_2.0_1_1 <-
  (exp(predictions.2020.sarima.3_1_2.0_1_1$upper[,1])
   - exp(predictions.2020.sarima.3_1_2.0_1_1$mean)) / qnorm(0.9)

ordres.2020.sarima.3_1_2.0_1_1 <-
  pnorm((job.ts.2020-exp(predictions.2020.sarima.3_1_2.0_1_1$mean))
        /sigma.2020.sarima.3_1_2.0_1_1)
```

```
##          Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 2020 30.60 29.11  0.00  0.00  0.00  0.08  0.12  1.88  4.29  0.79  2.32  5.98
```

De la même façon, en prenant en référence le modèle SARIMA(3,1,2)(0,1,1)[12], les probabilités d'observer les valeurs de 2020 sont faibles (< 6%).

La série observée est très inférieure aux prédictions des modèles de référence ; les modèles ajustés avant 2020 ne sont plus valables dès Mars 2020 et on conclut qu'il s'agirait de procéder à une nouvelle modélisation après la "cassure".

## 6.2.2 et au-delà... Prédiction à horizon 36 mois

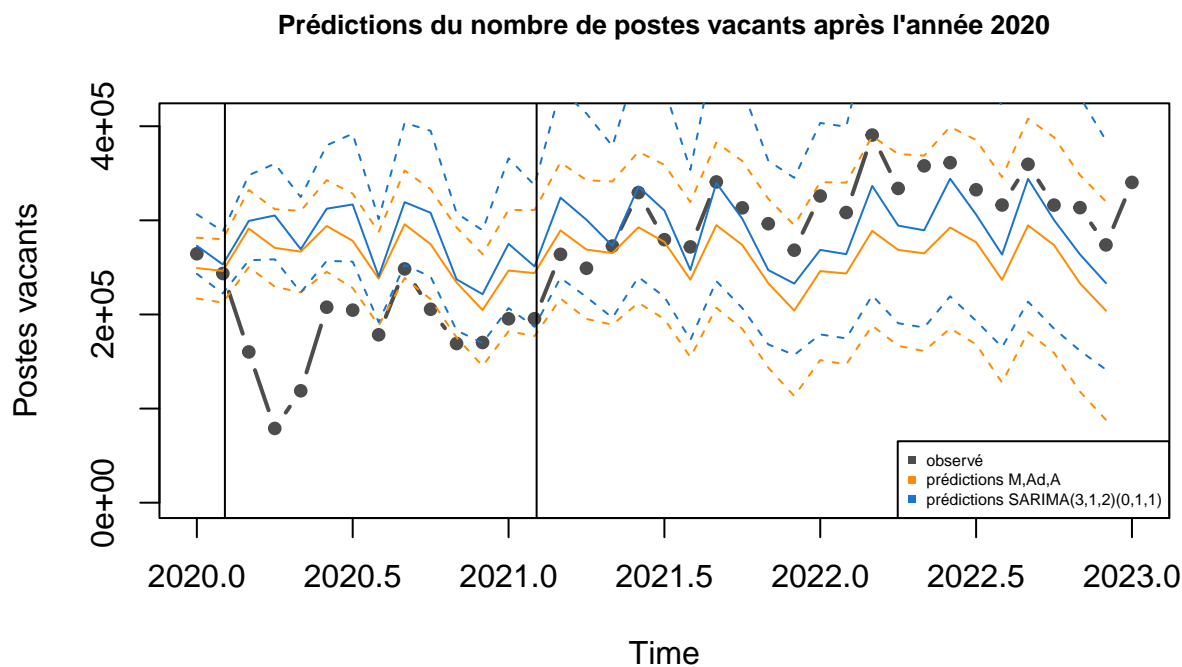
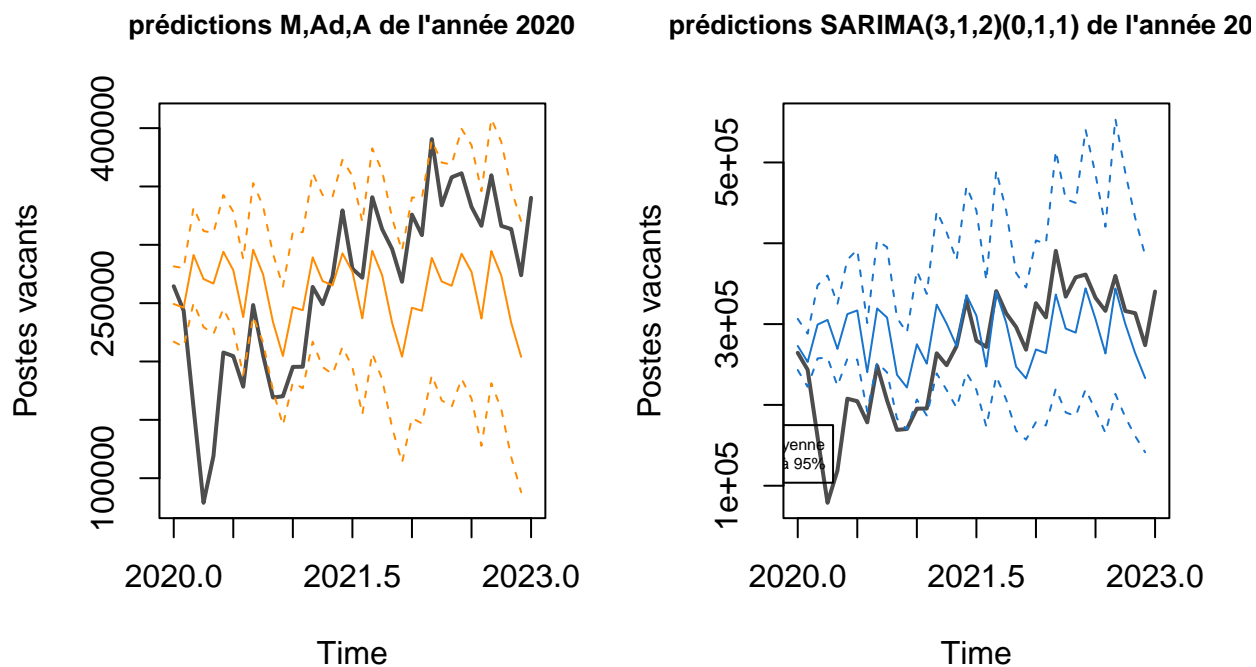
Voyons comment se comportent les modèles prédictifs au-delà de l'année 2020.

```
horizon=36
```

```
predictions.postcovid.le.MAdA <-
  forecast(modele.le.MAdA.study, h=horizon)

predictions.postcovid.sarima.3_1_2.0_1_1.2020 <-
  forecast(modele.sarima.3_1_2.0_1_2.study, h=horizon)
```





Quelques constats:

- L'accident COVID aura duré un an au vu du nombre de postes vacants ; ainsi on retrouve à partir de Mars 2021 un nombre de postes proches des prédictions,

- Sur cette période post-covid, les intervalles de confiance à 95% sont certes plus larges, mais ils sont adaptés pour contenir les valeurs réellement observées.
- Ainsi les modèles entraînés sur les années 2008-2019 prédisent relativement bien les années 2021-2023
- Le modèle SARIMA(3,1,2)(0,1,1) tire son épingle du jeu en proposant les prédictions les plus proches des valeurs observées
- A mesure que le temps passe, les modèles deviennent moins performants et sous-estiment les valeurs réelles ; on peut vraisemblablement imputer la sous-évaluation à une sur-performance compensatoire du nombre de postes vacants.

### 6.3 Conclusion

Face à la période COVID, les modèles entraînés sur la série temporelle de Janvier 2009 à Décembre 2019 se comportent bien dès la sortie de crise.

Certes l'accident COVID en 2020 met à mal les prédictions de l'ensemble des modèles, mais la résilience de l'économie qui transparaît au travers du nombre de postes vacants remet en selle les modèles ; notamment le modèle SARIMA(3,1,2)(0,1,1)[12] dont les prédictions dès Avril 2021 sont plus qu'honorables.

Dans un cadre professionnel où il se serait agi de suivre et prédire l'évolution du nombre de postes vacants en "temps réel" dès Mars 2020, une stratégie de suivi aurait pu être :

- de conserver en tant que référence les modèles en vigueur qui proposaient de bonnes estimations jusqu'en février 2020 (Lissage exponentiel (M,Ad,A) et SARIMA(3,1,2)(0,1,1) en sont de bons exemples)
- de suivre les valeurs mensuelles à partir de Mars 2020,
  - en les comparant aux valeurs prédites par les modèles précédents : il se serait agi de déterminer le moment à partir duquel le nombre de postes vacants revient à une norme "sans-covid" (en l'occurrence Avril 2021)
  - en ajustant mensuellement de nouveaux modèles à partir de mars 2020 pour obtenir des prédictions au plus juste pendant la période "accidentée".

En ce qui concerne la période post-covid (au-delà de Mars 2021), bien que les 1ers mois sont plutôt bien prédits, les modèles entraînés jusqu'en 2019 tendent à sous-évaluer de plus en plus au fil du temps les valeurs observées. Il s'agirait de les mettre à jour en ajustant de nouveaux modèles prenant en considération l'effet post-covid.

## 7 Références

Organization for Economic Co-operation and Development, Total New Job Vacancies, Flow for France [LMJVTTNVFRM647N], retrieved from FRED, Federal Reserve Bank of St. Louis; <https://fred.stlouisfed.org/series/LMJVTTNVFRM647N>, May 31, 2023

Séries Temporelles avec R, Méthodes et cas. Yves Aragon (Springer, 2011)