Prédiction du nombre de visiteurs de restaurants

Cadmos Kahale-Abdou

26 janvier 2021

Table des matières

1 Introduction **3**

2 Description de la source **3**
2.1 Analyse descriptive 3

3 Modélisation des données **5**
3.1 Saisonnalité 5
3.2 Tendance 7
3.3 Les Parties Aléatoires 9

4 Simulation et Prévisions **10**
4.1 Le Lissage Exponentiel 10
4.2 Le Modèle ARIMA 11
4.3 Étude de la Stationnarité 11
4.3.1 Prédictions du nombre total des visiteurs 13
4.3.2 Prédictions du nombre de visiteurs par restaurant 14
4.3.3 Prédictions du nombre de visiteurs par catégorie de restaurant 16

5 Annexes **17**
5.1 fonction predictVisitors 17

1 Introduction

L'objectif de cette étude est de déterminer le future nombre de consommateurs de restaurants. Nos données viennent de restaurants Japonais. Nous aurons 8 types de données venant d'un site internet japonais qui collecte des données sur ses utilisateurs. Le site"AirREGI / Restaurent Board (air)" similaire au site Yelp, un site qui permet de rechercher des restaurants, réserver et payer en ligne. Les données d'entraînement ont été récoltées de Janvier 2016 à Avril 2017.

2 Description de la source

Nos fichiers sont les suivants :

— air_visit_data.csv : L'historique des visites pour les restaurants. On y retrouve le nombre de visiteurs pour chaque restaurant par jour.

— air_reserve.csv : Le détail des réservations faites avec les systèmes air, tels que la date de la réservation et celle de la visite ainsi que le nombre de visiteurs prévu.

— air_store_info.csv : Les détails sur les restaurants air qui incluent des informations comme le type de restaurant et la localisation.

2.1 Analyse descriptive

Nous commençons par nous familiariser avec nos données, pour ce faire nous procédons dans notre démarche par la manipulation et visualisation des données. On considère la série temporelle du nombre total de visiteurs par date.

Nos observations sont les suivantes :

— Deux structures paraissent être les plus importantes : d'abord on observe un pic aux environs du 1er Janvier 2017, puis une structure sinusoidale qui pourrait s'expliquer par la saisonnalité hebdomadaire des observations. On observe aussi ce qui pourrait être une faible tendance linéaire croissante.

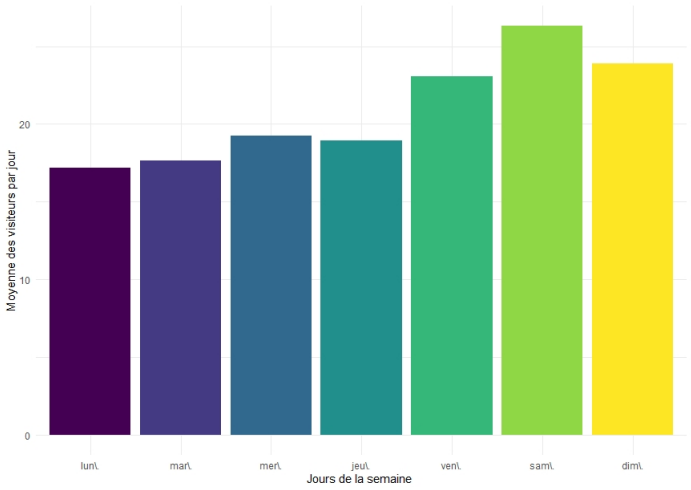
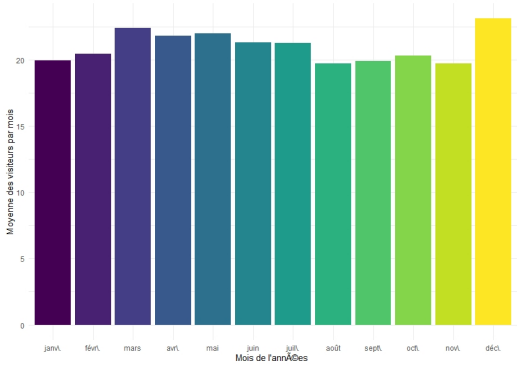
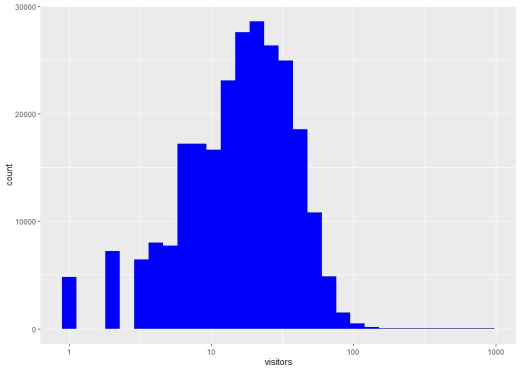


FIGURE 1 – Moyenne des visiteurs par jour

— Le vendredi et le week-end semblent être les jours où il y a le plus de consommateurs, tandis que le le lundi et le mardi sont les jours où il y en a le moins en moyenne. Ceci s'explique naturellement par le fait que les jours ayant le plus de visiteurs correspondent au weekend.



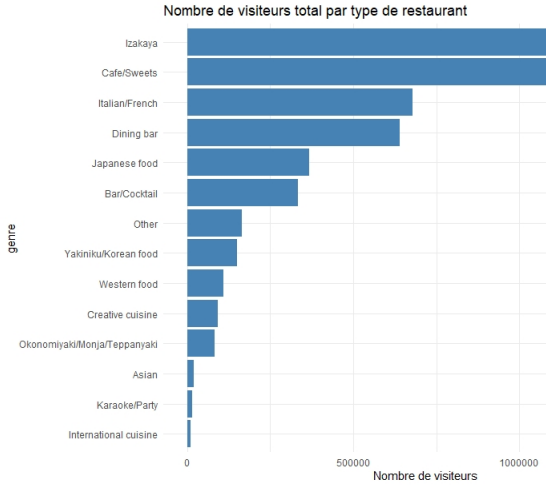
Moyenne des visiteurs par mois



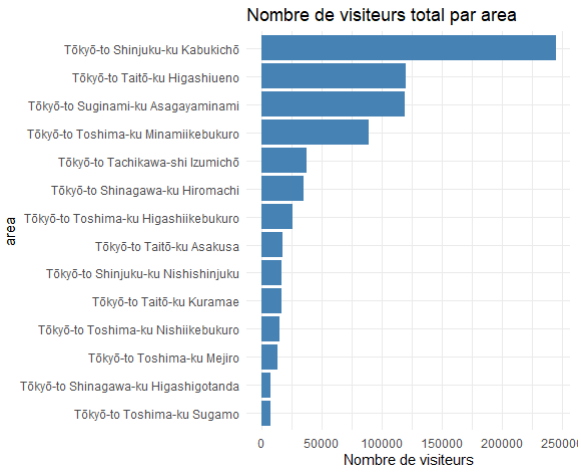
Distribution des visiteurs

— Le nombre de consommateurs par restaurants et par jour atteint les 20 au maximum par jours. Le graphe nous montre quand même que les consommateurs peuvent dans certains restaurants atteindre les 100 par jour.

— A l'échelle de l'année, on observe un certain nombre de variations. Décembre est le mois où qui compte le plus de consommateurs et la période de mars à mai reçoit le même nombre de consommateur à peu près tout au long de la période. On explique cela par le fait que les mois connaissant un plus grand nombre de visiteurs correspondent à des périodes de fête au Japon, le mois décembre étant celui des fêtes de fin d'année et celui de mars le printemps Japonais.



Nombre de visiteurs par type de restaurant



Nombre de visiteurs par localisation

— Il est clair que la catégorie de restauration la plus prisée par les japonais est celle d'izakaya qui occupe au Japon la place du bistrot ou du bar à vin en France, suivie de près par les cafés.

— Pour ce qui est du nombre total de visiteurs, c'est vers le centre de la ville de Tokyo qu'on constate l'affluence la plus importante, ces quartiers correspondraient aux premiers arrondissements de Paris en France. Cette affluence a tendance a fortement diminuer lorsque l'on s'éloigne du centre de la capitale japonaise.

3 Modélisation des données

Nous poursuivons notre démarche par la décomposition de notre série temporelle en ses différentes composantes

3.1 Saisonnalité

La périodicité hebdomadaire de nos observations étant fortement visibles à l'oeil nu, on commencera par l'étudier pour pouvoir plus facilement étudier le comportement des tendances de nos observations.

Dans un premier temps on effectuera une moyenne à fenêtre glissante pour éliminer la période hebdomadaire sur nos données :

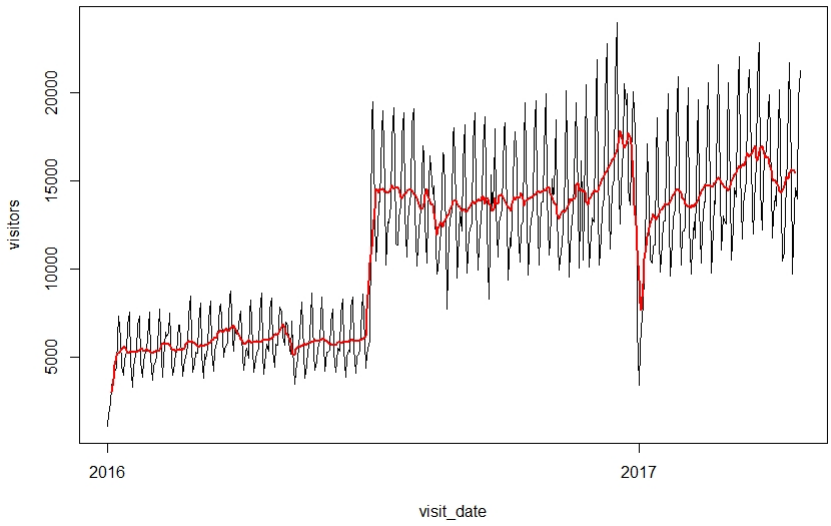


FIGURE 2 – Lissage par convolution : fenêtre de taille K = 7

On voit d'abord que la plupart des oscillations disparaissent, ce qui confirme la présence de la saisonnalité hebdomadaire. Cela dit, on observe toujours un nombre de bosses non-négligeable qui reviennent souvent, en plus d'un saut conséquent aux abords du 1er Juillet 2016 et d'un pic négatif le 1er Janvier 2017. Le graphique n'est pas à proprement dit lisse, mais on arrive à dégager un peu mieux la présence d'une légère tendance linéaire croissante.

On essaye alors d'appliquer une convolution avec une fenêtre plus large :

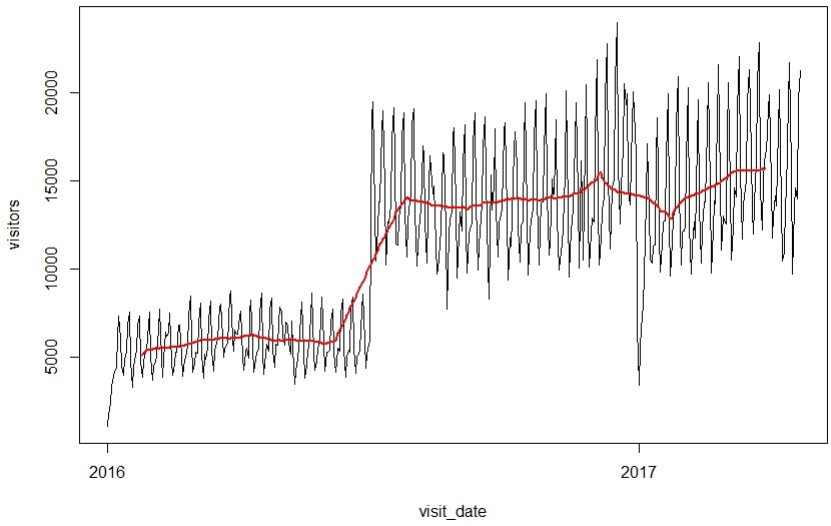


FIGURE 3 – Lissage par convolution : fenêtre de taille K = 49

Ici le phénomène d'oscillations et de bosses disparaît complètement, mais on garde le pic négatif et en regardant les parties que l'on pourrait segmenter, en dehors de la marche du 1er Juillet et le ce pic négatif, on arrive à voir un peu mieux cette tendance linéaire croissante.

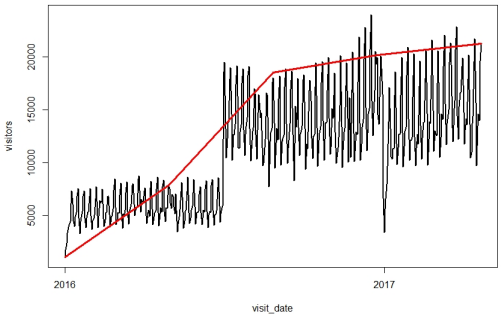
Pour conclure sur la saisonnalité des observation, nous avons une plage de donnée recouvrant un peu plus d'une année, ainsi nous n'avons pas assez de données pour affirmer la saisonnalité mensuelle des observation, mais la saisonnalité hebdomadaire est celle qui semble être la plus prédominante et celle qu'on est capable de mettre en exergue par un lissage à moyenne glissante de taille 7 en plus d'une analyse logique du problème. On ne prendra donc en compte que la saionnalité hebdomadaire.

3.2 Tendance

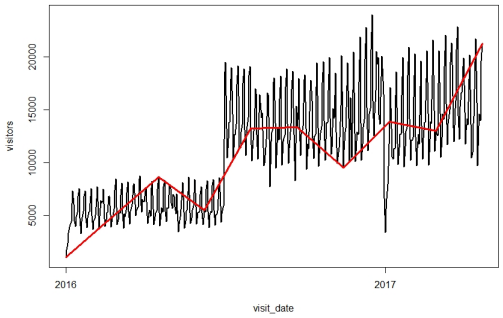
Notre objectif étant de prédire le nombre de visiteurs dans le futur, on commence naturellement par observer la somme des visiteurs de tous les restaurants par jours (c.f. Figure 1). La structure en escalier qui semble être prédominante, on cherchera à mettre en avant les éventuelles autres tendances sous-jacentes au données.

On décide donc de continuer notre étude des tendances de nos observations on utilisant la méthode de fonctions de spline.

Tout d'abord, on sait que l'on cherche à mettre en exergue une tendance linéaire croissante, donc la première étape la plus naturelle serait de choisir des fonctions d'ordre 1 pour pouvoir observer des segments qui nous donnerait des indications sur la tendance générale du graphique :



Approximation du nombre de visiteurs par des fonctions de spline : avec n = 5 noeuds



Approximation du nombre de visiteurs par des fonctions de spline : avec n = 10 noeuds

Dans la figure de gauche, avec n = 5 noeuds, on arrive bien à observer l'accroissement linéaire, tandis qu'avec plus de noeuds, on n'arrive plus très bien à l'observer les fonctions de spline donnant la tendance sur des blocs plus petits. Le problème avec le premier graphique étant que la tendance proposée par les fonctions de spline est trop forte sur les premiers noeuds, sans doute parce que l'approximation commence au point 0 de l'abscisse. On décide donc de placer les noeuds manuellement :

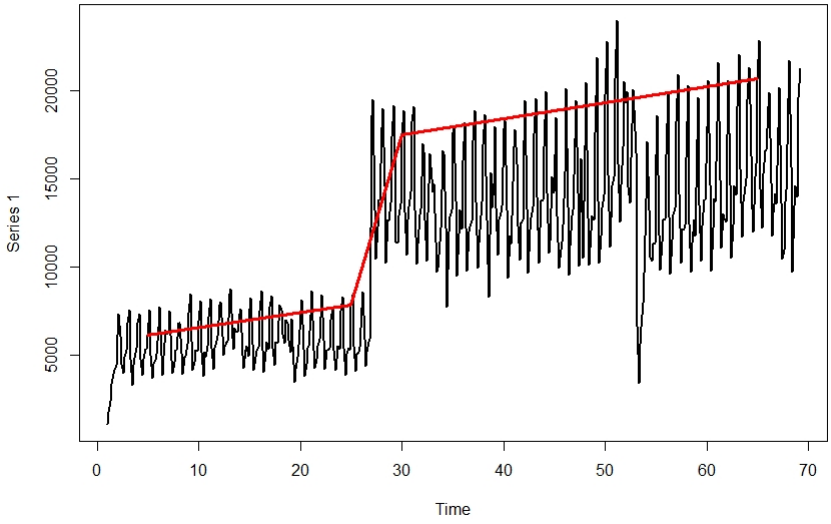


FIGURE 4 – Approximation par des fonctions de spline au noeuds t = 5,25,30,65 (freq = 7)

C'est ici que l'on voit le plus clairement la légère tendance de nos observations en ignorant le saut aux environs de t=30, et qui nous permet donc de conclure cette section

3.3 Les Parties Aléatoires

Après avoir dégagé la tendance et la saisonnalité de no données, il nous reste encore des phénomènes isolés à expliquer :

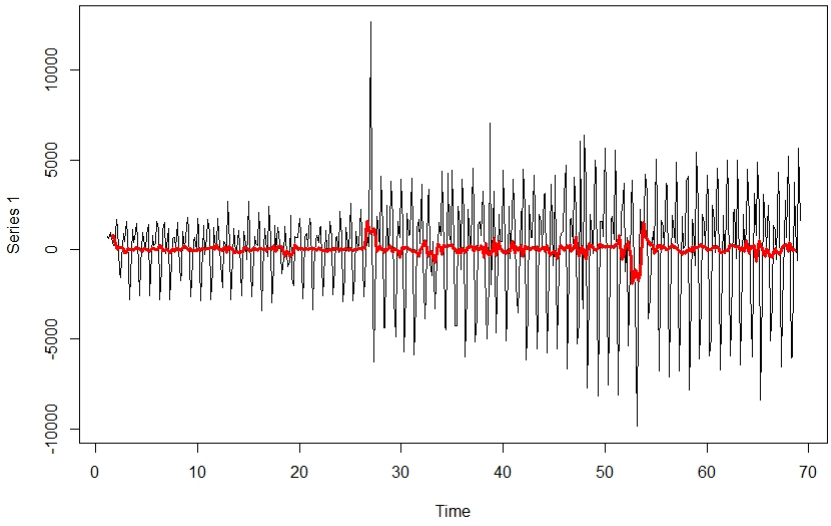


FIGURE 5 – Nombre de visiteurs sans la tendance (noir) sans la tendance et la saisonnalité (rouge)

On se concentre notre étude sur l'explication de deux phénomènes : d'une part le saut aux environs de t = 30, et d'autre part le pic négatifs aux environs de t =55. Le premier saut est expliqué par l'augmentation du nombre de restaurants qui se sont inscrit sur le site Air. La plate-forme recensant plus de restaurants, à vu son nombre de visiteurs être translaté vers le haut à une date précise. Mais 2 phénomènes notables sont rassurants dans au sein de notre étude. D'abord, le saut est suffisamment visible pour pouvoir le repérer au premier coup d'oeil. Ensuite on conserve la saisonnalité hebdomadaire, et mis à part le pic du 1er Janvier 2017, les restaurants semblent se comporter de manière similaire avant et après cette translation.

Quand au pic négatif du 1er Janvier, on peut l'expliquer par la présence de vacances bien spéciales au Japons qu'on pourrait prendre en compte dans une étude plus extensive. Mais nous pouvons, de manière réaliste, supposer que les jours à prédire se ne se comporteront pas comme ça, étant donné que ces vacances ne se reproduisent pas dans la période temporelle que l'on cherche à simuler.

Une fois ces phénomène enlevés, nous nous retrouvons avec un segment relativement lisse centré autour de 0 en ordonnée, ce qui nous permet de conclure l'étude sur la saisonnalité, les tendances et les parties aléatoires de nos observations.

4 Simulation et Prévisions

4.1 Le Lissage Exponentiel

Cette partie est consacrée à la prévision du nombre total de visiteurs par la méthode de Holt-Winters : lissage exponentiel pour série avec saisonnalité additive. Cette méthode semble être pertinente à utiliser de par la partie précédente. On choisit arbitrairement un nombre de jours N à prédire. On décide de diviser notre jeu de données en deux jeux train et test afin de pouvoir comparer le résultat obtenu et celui attendu et ainsi tester l'efficacité de notre prédiction.

On retrouve ci-dessous le résultat obtenu pour $N = 39$ jours

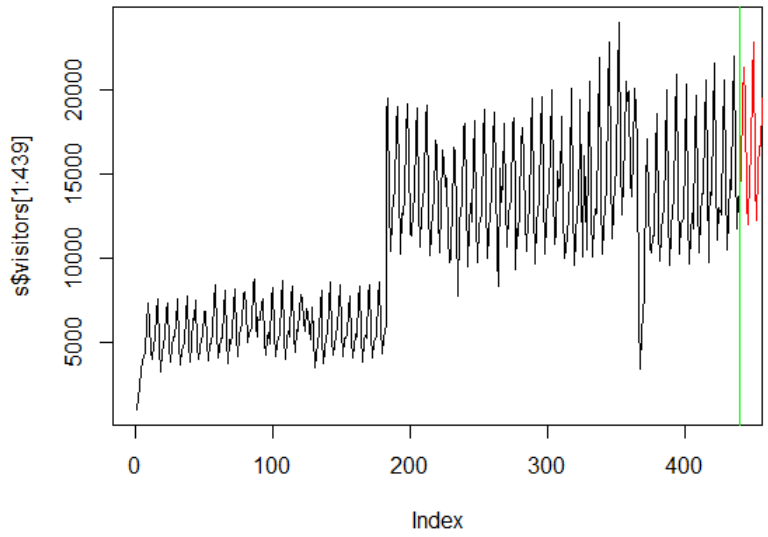


FIGURE 6 – Prévision du nombre total de visiteurs par lissage de Holt-Winters

On remarque que nos prédictions, ici affichées en rouge, semblent avoir une tendance linéaire croissante. Cela est dû au comportement du lissage de Holt-Winters qui effectue une prédictions en donnant un poids plus fort aux observations les plus récentes. Le fait que celles-ci aient, ce qu'on pourrait décrire comme une légère tendance croissante, se retranscrit donc dans nos résultats. On observe également un phénomène d'oscillation, dont l'amplitude se rapproche assez bien de celle de notre série temporelle.

Nous appliquons ensuite notre méthode à la série temporelle privée de sa saisonnalité en utilisant la moyenne mobile, et obtenons ce qui suit :

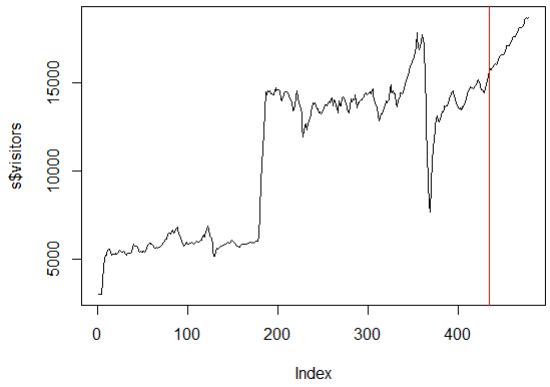


FIGURE 7 – Prévision du nombre total de visiteurs par lissage de Holt-Winters

Les résultats sont en cohérence avec ceux obtenus précédemment, on observe toujours cette même tendance linéaire croissante(quoique un peu plus prononcée). On perd cependant nos oscillations, en effet la série temporelle étant privée de sa saisonnalité, le phénomène de périodicité n'apparaît plus.

4.2 Le Modèle ARIMA

Nous poursuivons notre démarche, cette fois en utilisant le modèle ARIMA pour prédire dans un premier temps le nombre total de visiteurs. Nous avons trouvé intéressant d'affiner nos résultats en nous penchant sur les prédictions du nombre total de visiteurs pour un restaurant donné puis pour une catégorie de restaurant donnée *(cf fonction predictVisitors)*.
Notons que nous préférons dans la suite prédire nos résultats en nous basant sur nos données brutes.

4.3 Étude de la Stationnarité

Dans un premier temps, on cherche à établir une étude suffisante de la stationnarité des observations pour pouvoir choisir des paramètres avisés, et établir un modèle ARIMA qui nous fournirait des prédictions réalistes.

Notre premier réflexe est d'abord de transformer nos données dans le format `xts` en indiquant une fréquence de 7, puis d'y appliquer la fonction `acf()`. On obtient le graphique d'autocorrélation suivant :

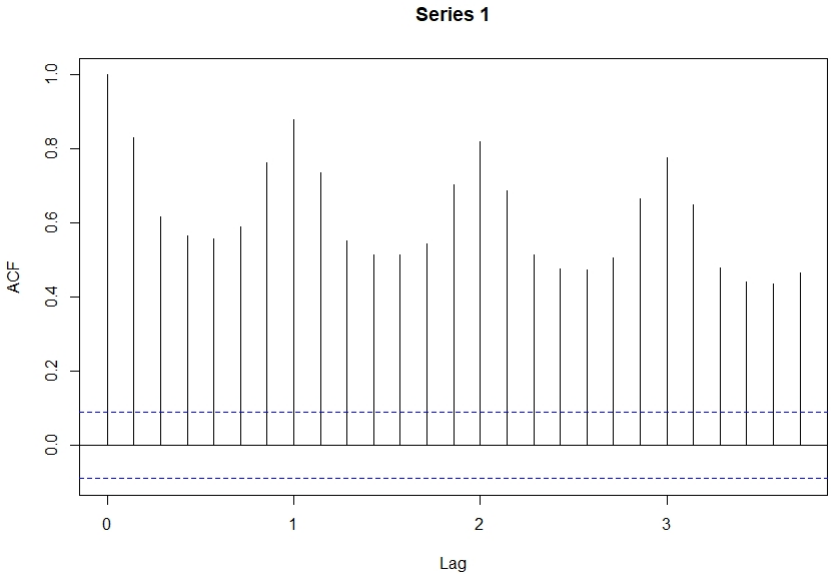


FIGURE 8 – Autocorrélation temporelle du nombre de visiteurs en fonction du temps

On voit une forte corrélation des données au niveau de tous les lags accessibles. Cela s'explique par le fait que la fonction `acf()` prend en compte les corrélations inférieures à chaque fois. Or nous avons mis en exergue la présence d'une saisonnalité hebdomadaire, il est donc logique de retrouver une périodicité dans les auto-corrélations.

On décide donc de traiter chacun des lags indépendamment des précédents. On applique la fonction `pacf()` qui nous donne le graphique suivant :

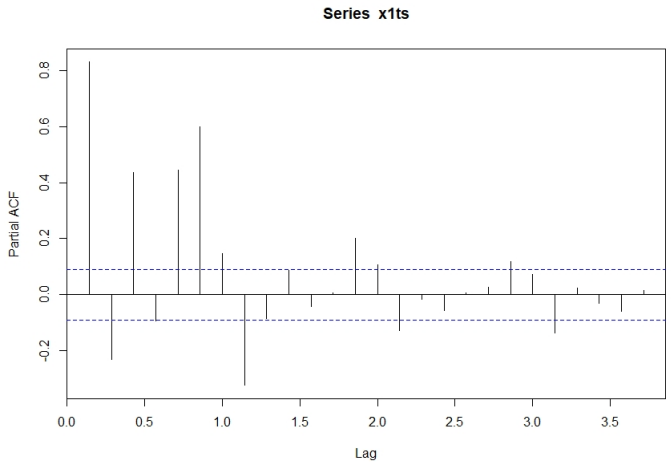


FIGURE 9 – Auto-corrélation temporelle partielle du nombre de visiteurs en fonction du temps

Ici, nous avons perdu le caractère périodique des valeurs de corrélation. Mais on voit quand même que les premières valeurs de lag ont une forte valeur de corrélation. On observe que ces valeurs sont essentiellement présentes pour les lags entre 0 et 2. Malgré le fait que l'on retrouve des corrélations relativement fortes après, on comprend en fait qu'une étude de stationnarité sur ce jeux d'informations n'est en fait pas la chose la plus pertinente, les observations étant corrélées. Cela étant dit, pour appliquer au mieux notre modèle ARIMA on fera le choix d'établir la prédiction de deux jours dans le futur à partir de deux jours dans le passé pour les observations et l'erreur, et la prédiction d'un jour en fonction du même jour de la semaine d'avant (de par la saisonnalité) pour les observations et l'erreur. On justifie davantage ce choix en prenant les données de visite brutes(sans les sommer), puis en y appliquant la fonction `diff()` qui supprime la tendance au sein des données. Ensuite on applique la fonction d'auto-corrélation qui nous donne le graphique suivant :

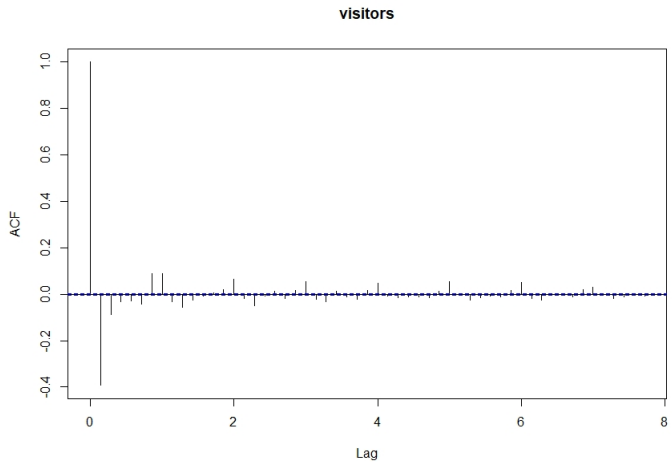


FIGURE 10 – Auto-corrélation temporelle du nombre de visiteurs (brute)

Dans ce graphique, on observe encore une fois que la densité de lag les plus corrélés se trouvent entre 0 et 2, ce qui nous confortera dans le choix de nos paramètres pour notre modèle ARIMA.

4.3.1 Prédictions du nombre total des visiteurs

Nous observons les résultats ci-dessous pour le même N

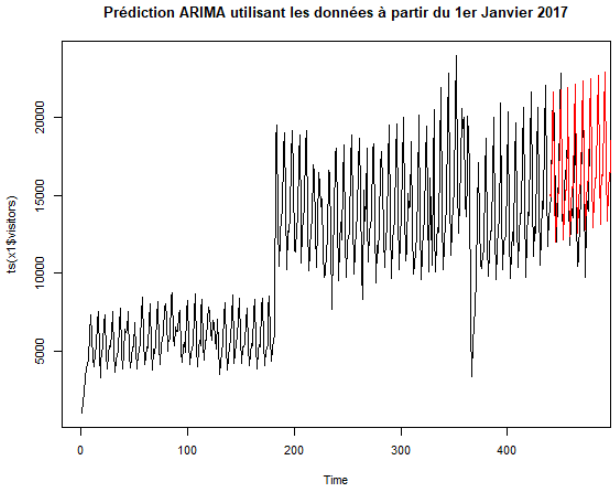


FIGURE 11 – Prévision du nombre total de visiteurs par ARIMA

On remarque que les prédictions obtenues sur le nombre total des visiteurs par la méthode d'ARIMA ont une tendance linéaire croissante, et qu'elles conservent le phénomène d'oscillations attendu. Cela dit, on voit aussi que notre modèle prend fortement en compte la tendance sur la période après le 1er Janvier (fortement croissante), et fournit un résultat qui prédit une continuité dans la croissance des visiteurs, alors que ce n'est pas ce qu'il s'est passé en réalité. De plus on voit que le modèle de prédiction à du mal à s'adapter à l'amplitude des oscillations. En effet, toutes les oscillations en rouge semblent avoir la même amplitude ce qui ne semble pas être en accord avec nos données de comparaison.

4.3.2 Prédictions du nombre de visiteurs par restaurant

Nous observons ci-dessous les prédictions sur le nombre de visiteurs de 3 restaurants de catégories différentes :

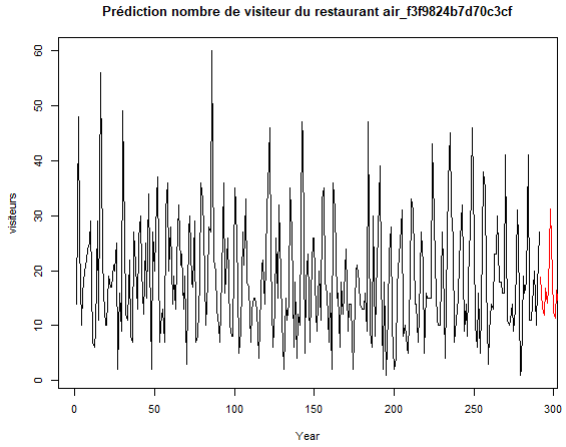
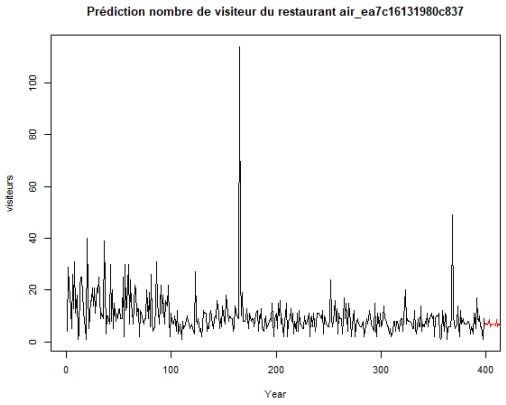
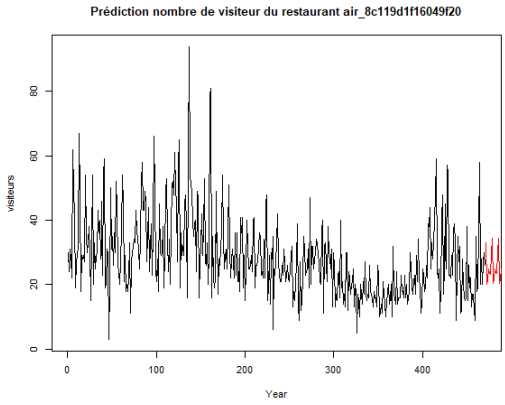


FIGURE 12 – Hokkaidō Sapporo-shi Minami 3 Jōnishi Bar/Cocktail



Prévision pour Tokyo-to Chuo-ku Ginza french/italian



Prévision pour Tokyo-to Shinjuku-ku Kabukicho Café/Sweet

Ici nos résultats sont plus adaptées à ce qu'il se passe dans des restaurants au cas par cas. En effectuant nos prévisions sur tous les visiteurs de tous les restaurants on peut prédire la tendance du marché mais on ne peut pas pour autant remonter à ce qu'il se passe dans chacun des secteurs, qui a priori, sont confrontés à des problématiques et des évolutions différentes. Ensuite, on tombe assez facilement dans du sur-apprentissage, si la base de donnée était amenée à prendre en compte plus de restaurants encore cela serait amené à empirer. Regarder chacun des restaurants pourrait offrir une prédiction plus réaliste, on pourrait par exemple faire une prédiction sur chacun des restaurants et sommer les prédictions, ce qui nous permettrai peut être d'obtenir un résultat sur le marché global (d'autant plus que l'on pourrait vendre un service spécifique à chaque restaurant). Cela étant dit, encore une fois, l'amplitude des pics observée pour les prédiction ne semble pas s'adapter à celle de notre jeu de données.

4.3.3 Prédictions du nombre de visiteurs par catégorie de restaurant

Effectuer une étude approfondie sur chacun des restaurants pourrait devenir une lourde tâche trop avide en temps et en puissance de calcul. Le bon compromis serait peut être d'étudier les restaurants par catégorie :

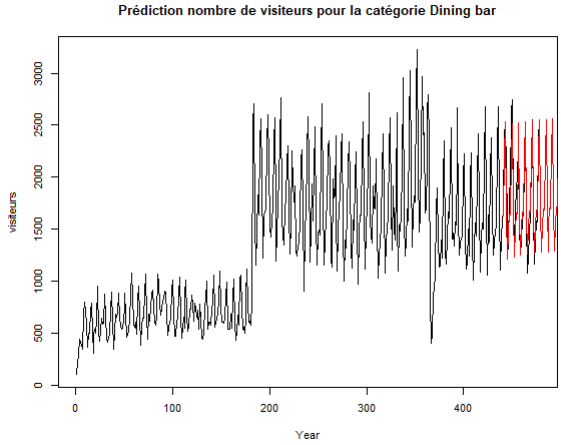
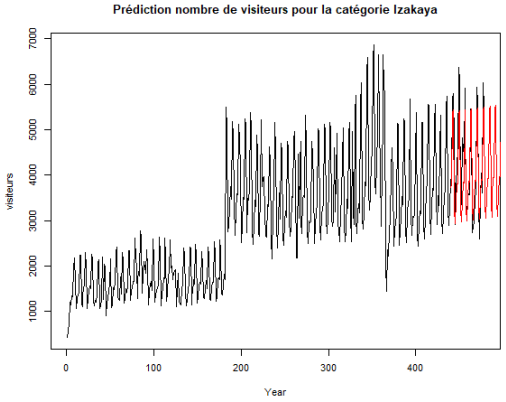
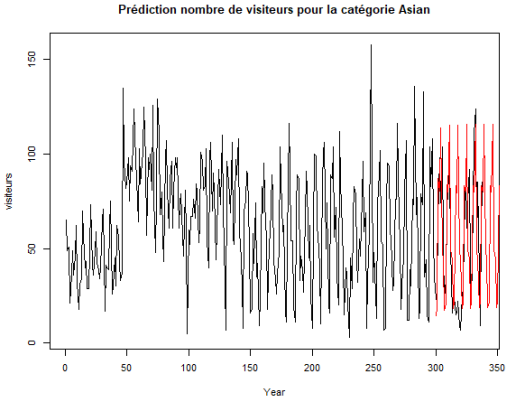


FIGURE 13 – Prévision du nombre total de visiteurs par catégorie de restaurant



Prévision pour la catégorie Izakaya



Prévision pour la catégorie Asian

Les prédictions semblent recouvrir un peu plus les données réelles ce qui nous conforte dans le choix de notre étude. Cela dit, on observe un réel décalage entre les données et les prédictions effectuées. On en conclut que le modèle ARIMA n'est pas la méthode la plus efficace pour faire des prédictions sur notre jeu de donnée, de par la corrélation évidente des observations.

5 Annexes

5.1 fonction predictVisitors

On propose une fonction qui prédit le nombre de visiteurs d'un restaurant étant donnée son identifiant ou son type. Le comportement de la fonction sera défini par la valeur de la variable *type* passée en entrée

```
predictVisitors  <- function(id, N, type) {
  max_date <- max(air_visits$visit_date)
  split_date <- max_date - N

  if(type == 0){
  # Prédiction du nombre de visiteurs pour un restaurant donné

    air_visits.id <- air_visits %>% filter(air_store_id == id)
    train <- air_visits.id %>% filter(visit_date <= split_date)
    test <- air_visits.id %>% filter(visit_date > split_date)

    m <- arima(air_visits.id$visitors, order=c(2,1,2), seasonal= list(order=c(1,1,1), period=7))
    y_pred <- forecast::forecast(m, h=80)

    plot(air_visits.id$visitors, xlab='Year', ylab='visiteurs', type = 'l')

    lines(y_pred$mean, col='red')
    title(str_c( "Prédiction nombre de visiteur du restaurant ",id))
  }

  if(type == 1){
  # Prédiction du nombre de visiteurs pour un genre de restaurant donné

    air_visits.new <- air_visits %>%
      dplyr::left_join(air_store, by='air_store_id', how='left')

    air_visits.id <- air_visits.new  %>%
      group_by(visit_date, air_genre_name) %>%
      summarize(visitors=sum(visitors))

    air_visits.id <- air_visits.id %>% filter(air_genre_name == id)
    train <- air_visits.id %>% filter(visit_date <= split_date)
    test <- air_visits.id %>% filter(visit_date > split_date)

    m <- arima(train$visitors, order=c(2,1,2), seasonal= list(order=c(1,1,1), period=7))
    y_pred <- forecast::forecast(m, h=80)

    plot(air_visits.id$visitors, xlab='Year', ylab='visiteurs', type = 'l')
    lines(y_pred$mean, col='red')
    title(str_c( "Prédiction nombre de visiteurs pour la catégorie ",id))
  }
}
```

FIGURE 14 –