Université de Franche-Comté

Master de Mathématiques Appliquées

Spécialité: *Modélisation statistique*

Rapport de stage de Master I

L'atelier de Francisco

Etudiant:

DOVODJI Kuassi Pierre

Tuteurs:

Ahmed ZAOUI

Andreea TUFFO

Contents

Remerciements **2**

Résumé **3**

Introduction **4**

1 Cadre du stage **4**
1.1 Description de l'entreprise 4
1.2 Problématiques et objectifs du stage 4
1.3 Missions effectuées 5

2 Outils utilisés et Bases de données **5**
2.1 Outils utilisés 5
2.2 Bases de données 6

3 Résultats **8**
3.1 Prétraitement 8
3.2 Analyse des bases de données clients et avis 9
3.3 Analyse de la base de données des réservations 18
3.4 Prévision des réservations 21
3.5 Conception de l'outil de planification 33
3.6 Discussions et Suggestions 36

4 Problèmes rencontrés solutions apportées et bilan des compétences lors du stage **37**
4.1 Problèmes rencontrés 37
4.2 Solutions apportées 37
4.3 Bilan 38

Conclusion **39**

Bibliographie **40**
Sources en ligne 40
Documents 40

Annexe **41**
Annexe code R 41
Annexe code Python 46

Remerciements

Qu'il me soit permis, au terme de ce travail, d'exprimer ma gratitude et vifs remerciements à ma tutrice de stage madame Andreea TUFFO . Qu'elle trouve ici le témoignage de mon estime et de ma profonde reconnaissance pour sa disponibilité, ses directives précieuses, ses conseils et la qualité de son suivi, qu'elle a su me prodiguer tout au long de mon stage malgré ses occupations extrêmes.

Mes remerciements vont également à monsieur Ahmed ZAOUI, pour sa grande disponibilité et les aides qu'ils me fournissent pour accomplir les objectifs de ce stage.

Je tiens à remercier les responsables du master, madame GOGA CAMELIA et monsieur LANDY RABEHASAINA, qui se sont montrés à l'écoute, ainsi pour les aides et les conseils fructueux qu'il nous ont prodigués tout au long de l'année.

Je remercie également tout le corps professoral du Master 1 de l'effort qu'ils fournissent pour mener à bien notre formation.

Mes sincères reconnaissances vont à tous ceux qui, de près ou de loin, ont contribué à l'aboutissement et au bon déroulement de ce modeste travail.

Résumé

Ce rapport de stage présente une analyse approfondie des réservations clients et la mise en œuvre de modèles de prédiction des réservations dans le cadre de mon expérience professionnelle chez L'atelier de Francisco, un établissement gastronomique de renom. L'objectif principal de ce projet était d'explorer en profondeur la base de données des réservations afin de comprendre les tendances de réservation, d'optimiser la gestion des tables et d'améliorer l'expérience globale des clients.

Au début de mon stage, j'ai entrepris une analyse approfondie des données de réservation, en examinant des paramètres tels que le nombre de réservations par client, la répartition démographique des clients, ainsi que leur engagement via les canaux de communication comme les e-mails et les SMS. Cette analyse initiale m'a permis de comprendre le comportement des clients et d'identifier les domaines où des améliorations étaient nécessaires.

En utilisant des techniques d'analyse de données avancées, j'ai développé des modèles de prédiction des réservations pour anticiper la demande future et ajuster la planification en conséquence. Ces modèles ont été essentiels pour optimiser la gestion des tables et garantir une utilisation efficace des ressources du restaurant.

En parallèle, j'ai également travaillé sur la conception et le développement d'un outil de planification innovant, basé sur les modèles de prédiction des réservations. Cet outil a été conçu pour aider l'équipe de gestion à planifier les services de manière plus efficace, en tenant compte des fluctuations de la demande et des tendances saisonnières.

Introduction

Ce rapport de stage détaille mon expérience professionnelle effectuée à *L'atelier de Francisco*, un restaurant situé au cœur de Besançon. Mon stage, d'une durée de deux mois, a pour objectif principal d'analyser en profondeur la base de données des clients et des réservations du restaurant afin d'optimiser plusieurs aspects clés de son fonctionnement.

Dans ce rapport de stage, nous commenceront par présenter l'entreprise, en décrivant son activité et son fonctionnement. Ensuite, nous détaillerons les objectifs de mon stage ainsi que les missions effectuées. Nous passerons ensuite aux résultats obtenus, en mettant en lumière les principales découvertes et analyses réalisées au cours de cette expérience. Par la suite, nous aborderons les difficultés rencontrées et les solutions proposées pour les surmonter. Enfin, nous conclurons en récapitulant les enseignements tirés de ce stage et en partageant mes réflexions personnelles sur cette expérience enrichissante.

1 Cadre du stage

1.1 Description de l'entreprise

L'entreprise L'atelier de Francisco est un groupe de restauration bien établi qui possède deux restaurants renommés: L'atelier de Francisco et Bellagio. Voici une description plus détaillée de chaque restaurant :

- **L'atelier de Francisco** : Situé au 85 Rue Battant, 25000 Besançon, L'atelier de Francisco est un établissement de style pizzeria crée en 01/07/2019 qui propose une cuisine italienne authentique. Le restaurant est réputé pour ses pizzas cuites au feu de bois, ses pâtes fraîches et ses plats de viande grillée. L'ambiance y est conviviale et décontractée, offrant aux clients une expérience gastronomique agréable et accessible.
- **Bellagio** : restaurant Italien situé 226C Rue de Dole, 25000 Besançon, offrant un buffet à volonté avec une grande variété de plats, notamment des spécialités italiennes telles que les pizzas, les pâtes, les salades et les desserts. L'établissement se distingue par son concept de restauration où les clients peuvent déguster une multitude de mets à volonté, dans une ambiance conviviale et familiale.

  L'entreprise est dirigée par le propriétaire Franscic TUFFO et la co-gérante Andréea TUFFO qui a été ma tutrice de stage, une équipe passionnée et expérimentée, qui s'efforce de fournir un service de qualité et de satisfaire les papilles de ses clients. Les deux restaurants bénéficient d'une réputation solide, ce qui en fait des destinations prisées pour les amateurs de bonne cuisine et de convivialité. J'ai été accueilli en tant que stagiaire en statistique (M1), j'étais responsable de l'analyse des données du restaurant, de la modélisation statistique et de la formulation de recommandations pour améliorer ses performances.

1.2 Problématiques et objectifs du stage

- **Problématique**

L'atelier de Francisco, souhaite optimiser plusieurs aspects clé de son fonctionnement grâce à une analyse approfondie de ses données client et de réservation. Cependant, les bases de données présentaientt des défis tels que des données manquantes, ce qui peut entraver une analyse précise. De plus, le restaurant souhaite évaluer la satisfaction globale de sa clientèle, identifier ses points forts et les domaines à améliorer pour offrir une expérience optimale. Enfin, la gestion efficace des réservations est un enjeu majeur pour réduire les temps d'attente et optimiser l'utilisation des ressources.

- **Objectifs**

Pour répondre à ces problématiques, les objectifs principaux de ce stage sont les suivants :

- Identifier et traiter les données manquantes dans les base de données pour permettre une analyse fiable.

- Examiner les données de réservation, les commentaires des clients et les évaluations afin d'évaluer la satisfaction globale des clients et d'identifier les points forts et les domaines à améliorer.

- Développer des modèles prédictifs pour anticiper les réservations futures, permettant ainsi au restaurant de mieux planifier ses ressources, d'optimiser la gestion des tables et de réduire les temps d'attente.

- Développer un outil de planification pour aider l'entreprise à gérer efficacement ses employés.

En résolvant ces problèmes liés aux données manquantes, à la satisfaction clients et à la gestion des réservations, ce stage vise à fournir à L'atelier de Francisco des outils et des analyses pour améliorer son efficacité opérationnelle et offrir une expérience client exceptionnelle.

1.3 Missions effectuées

Pendant mon stage de deux mois chez *L'atelier de Francisco*, j'ai pu découvrir en détail le rôle d'un statisticien dans une entreprise. Mes missions se sont articulées autour de trois points principaux.

J'ai commencé par nettoyer et concaténer deux bases de données, l'une concernant les clients et l'autre les avis. Les premières semaines, j'ai effectué des recherches et étudié différentes méthodes de nettoyage des données, notamment la gestion des valeurs manquantes et aberrantes. Une fois ces connaissances acquises, j'ai appliqué les résultats sur mes bases de données, puis j'ai fusionné les deux bases de données en question. En suite analyser cette base de données pour aider l'entreprise à améliorer la satisfaction des clients, renforcer la fidélité et optimiser les opérations du restaurant.

Ensuite, la deuxième mission principale consistait à analyser les différents aspects des données relatives aux revervations journalières et de développer des modèles pour prédire les réservations journalières.

Enfin, la dernière mission a pour objectif de développer un outil de planification pour aider le restaurant à gérer efficacement ses employés. Cet outil inclut la planification des horaires, la gestion des congés et des jours de repos.

2 Outils utilisés et Bases de données

2.1 Outils utilisés

Lors de cette expérience, j'ai principalement utilisé deux langages de programmation, Python et R, pour réaliser diverses tâches analytiques.

Python a été utilisé pour le nettoyage des données, l'analyse exploratoire et la modélisation prédictive grâce aux bibliothèques *pandas*, *numpy*, *seaborn*, *scipy*, *matplotlib*, *statsmodels* et *gender-guesser* :

- *pandas* : pour la manipulation efficace des données structurées, facilitant le nettoyage et la transformation des données.

- *numpy* : pour le calcul numérique efficace, offrant des fonctionnalités pour la manipulation de tableaux multidimensionnels et la réalisation d'opérations mathématiques sur ces tableaux.

- *seaborn* : pour la création de graphiques statistiques attractifs et informatifs, permettant une visualisation claire des patterns dans les données.

- *scipy* : pour les fonctions statistiques avancées, les tests d'hypothèses et la modélisation numérique, aidant à la compréhension approfondie des données.

- *matplotlib* : pour la création de graphiques et de visualisations personnalisées, permettant une représentation visuelle des résultats.

- *statsmodels* : pour les modèles statistiques et l'analyse prédictive.

- *gender-guesser* : pour la prédiction du genre à partir des prénoms, contribuant à l'analyse démographique des clients.

Pour l'analyse descriptive des réservations, la visualisation des données et une partie de la prediction, j'ai opté pour R, utilisant des packages tels que *dplyr*, *lubridate*, *ggplot2*, *plotly*, *tseries*, *forecast* et *lmtest* :

- *dplyr* : pour la manipulation efficace des données, facilitant le filtrage, le regroupement et la transformation des données.

- *lubridate* : pour la gestion des dates et des heures, permettant le traitement facile des informations temporelles.

- *ggplot2* : pour la création de graphiques élégants et personnalisables, offrant une visualisation détaillée des données.

- *plotly* : pour la création de visualisations interactives et dynamiques, permettant l'exploration approfondie des données.

- *dplyr :* pour la manipulation de données (filtrer, sélectionner, regrouper, etc.).

- *tseries:* pour l'analyse des séries temporelles.

- *forecast :* pour la prévision de séries temporelles.

- *lmtest :* pour les tests de diagnostic des modèles linéaires.

En outre, pour le développement de l'outil de planification, j'ai utilisé :

- *R Shiny* : pour la création d'applications web interactives.

- *RSQLite* : pour la gestion de la base de données intégrée à l'outil.

- *calendar* : pour la gestion des fonctionnalités de calendrier.
- *shinyjs* : pour intégrer du JavaScript dans l'application.
- *timevis* : pour afficher les données dans un calendrier interactif.
- *shinymanager* pour gérer l'authentification, les droits d'accès, et le suivi des utilisateurs.

2.2 Bases de données

Tout mon stage est tourné au tours de trois bases de données. Qui sont :

- La base de données client est la base de données comportant les donnees des clients, elle a 31 variables et 4787 individues. Nous avons entre autre les variables :
  - *First_Name* : Le prénom du client.
  - *Civility* : La civilité du client (Monsieur, Madame, etc.).
  - *Email* : L'adresse e-mail du client.
  - *Phone* : Le numéro de téléphone du client.
  - *Guest_Status* : Le statut du client en tant qu'invité (par exemple, régulier, occasionnel, nouveau).

  - *Email_optin_market* : Indique si le client a accepté de recevoir des e-mails à des fins de marketing.
  - *SMS_optin_market* : Indique si le client a accepté de recevoir des SMS à des fins de marketing.
  - *Mail_optin_review* : Indique si le client a accepté de recevoir des courriers postaux pour des avis ou des enquêtes.
  - *SMS_optin_review* : Indique si le client a accepté de recevoir des SMS pour des avis ou des enquêtes.
  - *Has_no_show* : Indique si le client a eu des réservations sans s'y présenter.
  - *Bookings_number* : Le nombre de réservations effectuées par le client.
- Base de données des avis envoyés par le client qui comporte 1196 individus et 17 variables. Nous avons entre autres :
  - *Prénom* : Le prénom du client.
  - *Nom* : Le nom du client.
  - *Utilisateur désinscrit* : Indique si l'utilisateur s'est désinscrit.
  - *Adresse e-mail* : L'adresse e-mail de l'utilisateur.
  - *Alias Booking* : L'alias de réservation de l'utilisateur.
  - *Langue* : La langue préférée de l'utilisateur.
  - *Tags* : Les balises associées à l'utilisateur.
  - *Messages envoyés* : Le nombre de messages envoyés par l'utilisateur.
  - *Avis déposés* : Le nombre d'avis déposés par l'utilisateur.
  - *Date de création* : La date de création du compte utilisateur.
  - *Nb d'expériences* : Le nombre d'expériences réalisées par l'utilisateur.
  - *Chambre* : La chambre réservée par l'utilisateur.
  - *Catégorie* : La catégorie de réservation de l'utilisateur.
  - *Date de début* : La date de début de la réservation.
  - *Date de fin* : La date de fin de la réservation.
  - *Telephone* : Le numéro de téléphone de l'utilisateur.
- La base de données de reservation est une série temporelle classique qui représente le nombre de reservation par jour dans le restaurant sur 172 jours. La base de données contient les colonnes suivantes :
  - *Jour* : La date (jour, mois et année) de chaque observation.
  - *Couverts* : Le nombre de clients.

– *Service* : Type de service (Déjeuner ou Dîner).

– *Taux de remplissage* : Taux de remplissage du restaurent.

Les données des différents base couvrent la période de novembre 2023 à avril 2024, soit 6 mois de données mensuelles.

3 Résultats

3.1 Prétraitement

Les bases de données des clients et avis après l'inspection comportent beaucoup de valeurs manquant et de variables inutiles pour notre analyse, alors il est criciale de les nettoyer avant de les utiliser pour une analyse afin d'avoir des résultats fiables.

Les figure ci-contres montrent l'état de ses deux bases de données.

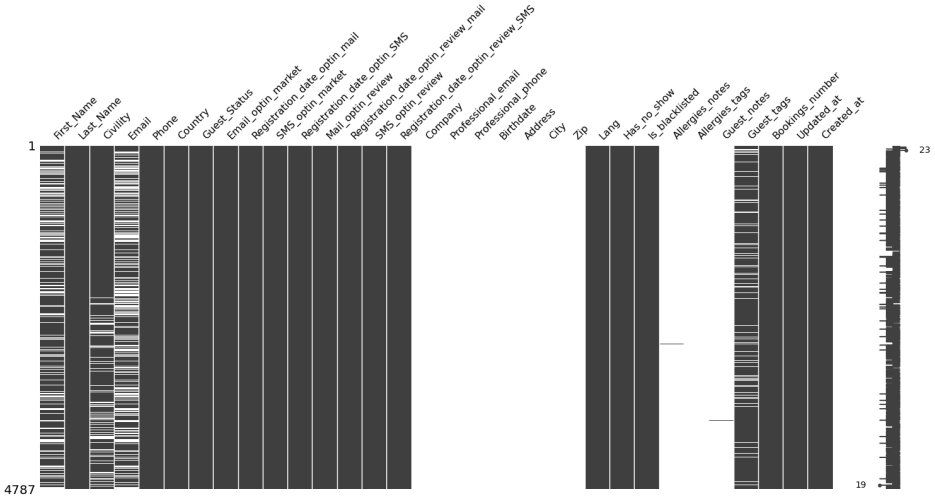

Figure 1: Données manquantes dans la base de données des clients

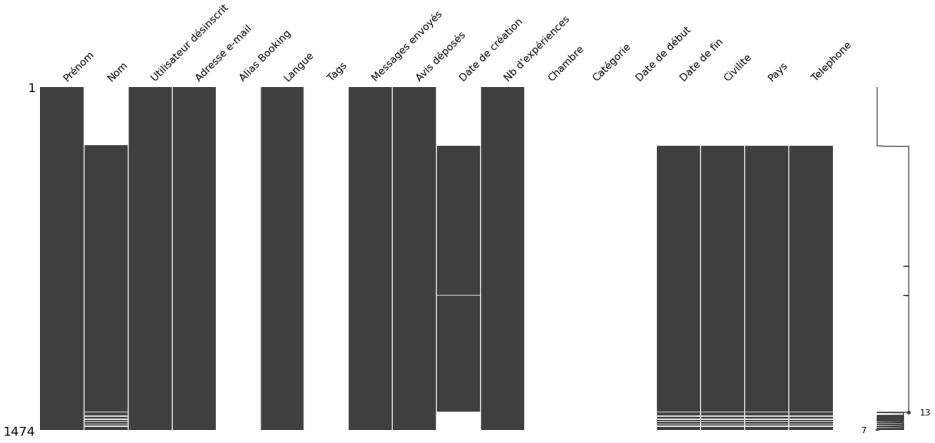

Figure 2: Données manquantes dans la base de données des avis

Sur ses figures les espaces blans représents les valeurs manquants.

Après l'idendification, pour le nettoyage , j'ai principalement utilisé deux approches pour gérer les valeurs manquantes dans les données : la suppression des lignes et des colonnes contenant des valeurs manquantes et l'imputation des valeurs manquantes. La première méthode consiste à supprimer les lignes de données avec des valeurs manquantes dans des colonnes spécifiques et aussi à supprimer des variables comportant de beaucoup de valeurs manquantes, tandis que la deuxième méthode remplace les valeurs manquantes par des estimations basées sur des techniques spécifiques, comme la moyenne, l'utilisation du détecteur de genre pour estimer le sexe des clients à partir de leurs prénoms.

3.2 Analyse des bases de données clients et avis

Après le nettoyage nous avons antamé les analyses qui met en évidence plusieurs tendances significatives qui peuvent guider les décisions stratégiques de l'établissement.

3.2.1 Fondements théoriques des méthodes mathématiques utilisées

Pour cette analyse, plusieurs notions mathématiques ont été utilisées. Nous avons entre autres :

- Pour les analyses univariées :
  - Le mode : la modalité ou la valeur la plus fréquente dans une variable qualitative ou quantitative discrète.
  - La moyenne : la valeur correspondant au centre de gravité de l'ensemble des valeurs d'une variable quantitative. Elle est calculée en utilisant la formule suivante :

$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ où N est le nombre total d'observations et x_i représente chaque valeur.
- L'écart-type : une mesure de la dispersion des valeurs autour de la moyenne, calculée comme :

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

où μ est la moyenne et x_i représente chaque valeur.

- Pour les analyses bivariées plusieurs graphiques ont été utilisées pour permettre de visualiser différente des relations entre les variables : des diagrammes en barres, des diagrammes en barres empilées, des diagrammes en boîte, des nuages de points, des histogrammes et des matrices de corrélation. Pour confirmer si les relations affichées par les graphiques sont réelles ou dû au hasard j'ai utiliser des tests statistiques commes :
  - Test du Chi-deux χ^2 : pour tester si deux variables quantitatives sont indépendantes ou pas.

Statistique du Chi-deux χ^2 : calculée comme :

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

où O_i est la fréquence observée et E_i est la fréquence attendue. Une valeur élevée de χ^2 indique une divergence significative entre les valeurs observées et attendues.

Le degré de liberté : calculés comme $(r-1) \times (c-1)$, où r est le nombre de lignes et c est le nombre de colonnes dans le tableau de contingence.

- Pour tester l'association entre deux variables quantitatives, j'ai utilisé le coefficient de corrélation de Pearson, qui mesure la force et la direction de la relation linéaire entre les deux variables. Ce coefficient est calculé comme suit :

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2}}$$

où x_i et y_i représentent les valeurs des deux variables, $\bar{x}$ et $\bar{y}$ représentent leurs moyennes respectives, et r est le coefficient de corrélation de Pearson.

Ce coefficient varie de -1 à 1. Une valeur de 1 indique une corrélation positive parfaite, -1 indique une corrélation négative parfaite, et 0 indique l'absence de corrélation linéaire entre les deux variables.

- L'interprétation de ses tests statistiques est basé sur la **p-value** : la probabilité d'obtenir un résultat au moins aussi extrême que celui observé, sous l'hypothèse nulle (aucune association entre les variables).
  - si la p-value est inférieure à un niveau alpha choisi (par exemple 0.05), alors l'hypothèse nulle est rejetée (i.e. il est improbable que l'hypothèse nulle soit réalisée).
  - si la p-value est supérieure au niveau alpha choisi (par exemple 0.05), alors on ne doit pas rejeter l'hypothèse nulle. La valeur de la p-value alors obtenue ne présuppose en rien de la nature des données.

En résumé, une faible p-value suggère une forte preuve contre l'hypothèse nulle, tandis qu'une p-value élevée suggère une faible preuve contre l'hypothèse nulle.

3.2.2 Applications

- **Profil des Clients**

La figure suivante présente la distribution des reservations.

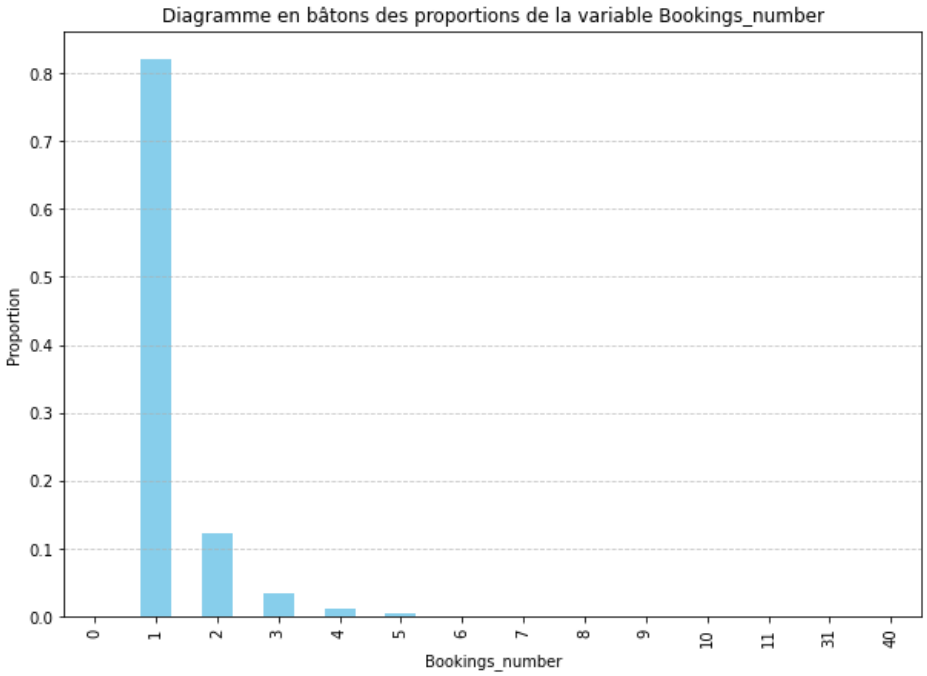

Figure 3: Distribution de nombres de réservations par clients

```
## count    4388.000000
## mean        1.292616
## std         1.056882
## min         0.000000
## 25%         1.000000
## 50%         1.000000
## 75%         1.000000
## max        40.000000
```

```
## Name: Bookings_number, dtype: float64
```

Interprétation :

En examinant le nombre de réservations effectuées par chaque client, il est clair que la majorité des clients(plus de 80%) réalisent une seule réservation. Toutefois, une proportion notable de clients effectue plusieurs réservations, indiquant un potentiel pour la mise en place de programmes de fidélisation.

- **Démographie de la clientèle**

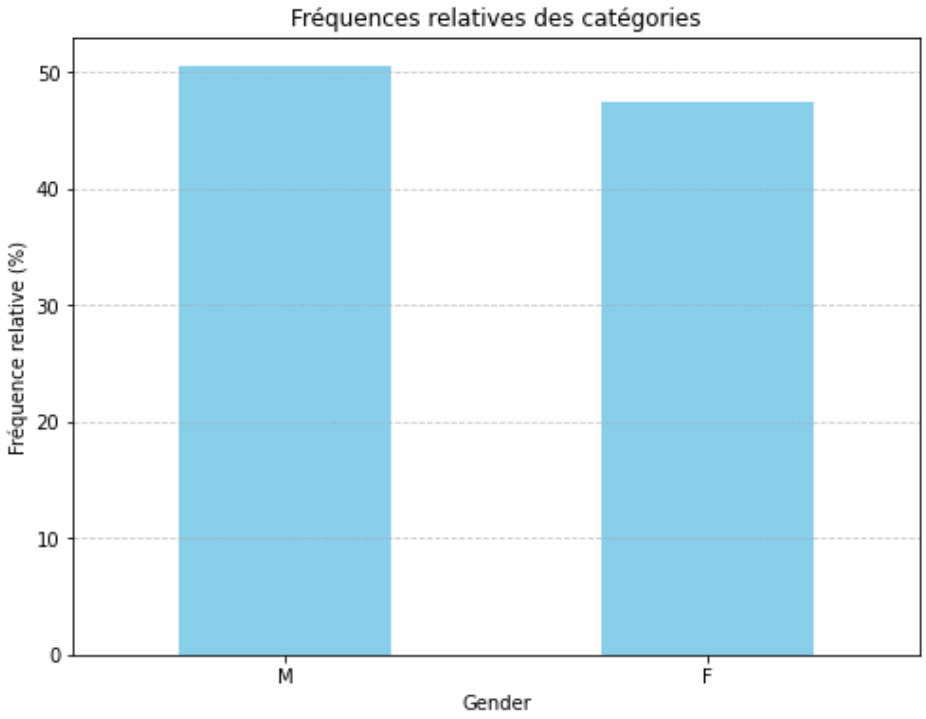

Figure 4: Distribution du sexes

```
## count     4301
## unique       2
## top          M
## freq      2217
## Name: Gender, dtype: object
```

Interprétation : En ce qui concerne la répartition démographique, une parité entre les hommes et les femmes est observée parmi la clientèle du restaurant. Cette diversité démographique peut offrir des opportunités pour développer des campagnes marketing inclusives et ciblées.

- **Engagement client**

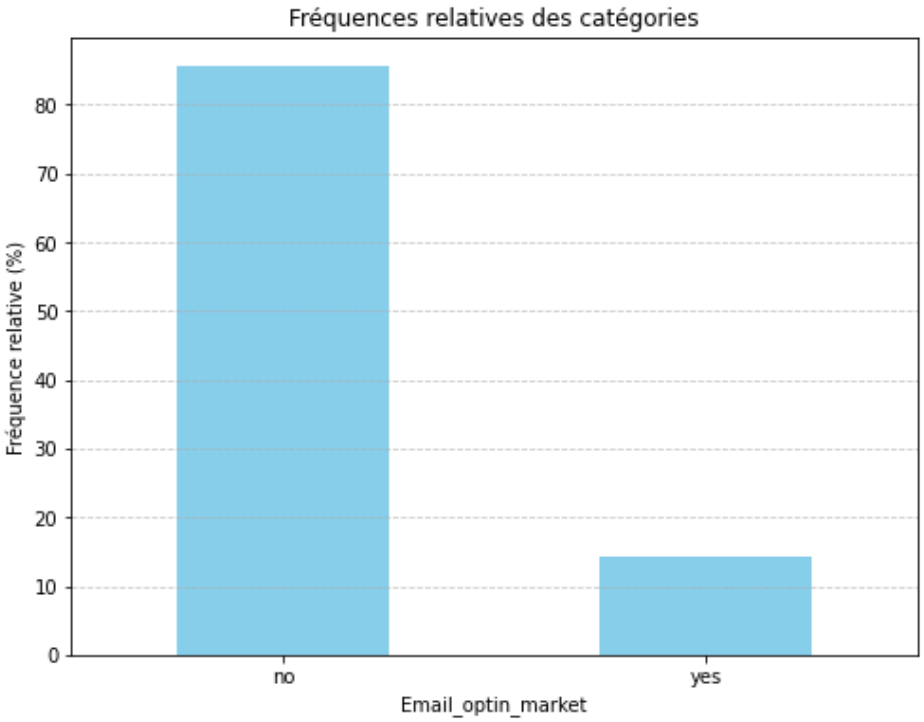

Figure 5: Distribution opt-in marketing mail

```
## count     4388
## unique       2
## top         no
## freq      3755
## Name: Email_optin_market, dtype: object
```

```
## count     4388
## unique       2
## top         no
## freq      3750
## Name: SMS_optin_market, dtype: object
```

Interprétation :

Concernant l'engagement client, la plupart des clients n'utilisent pas activement les canaux de communication proposés par le restaurant, tels que les e-mails ou les SMS. Cependant, une fraction des clients a recours à ces moyens de communication, suggérant une opportunité d'améliorer l'interaction avec la clientèle.

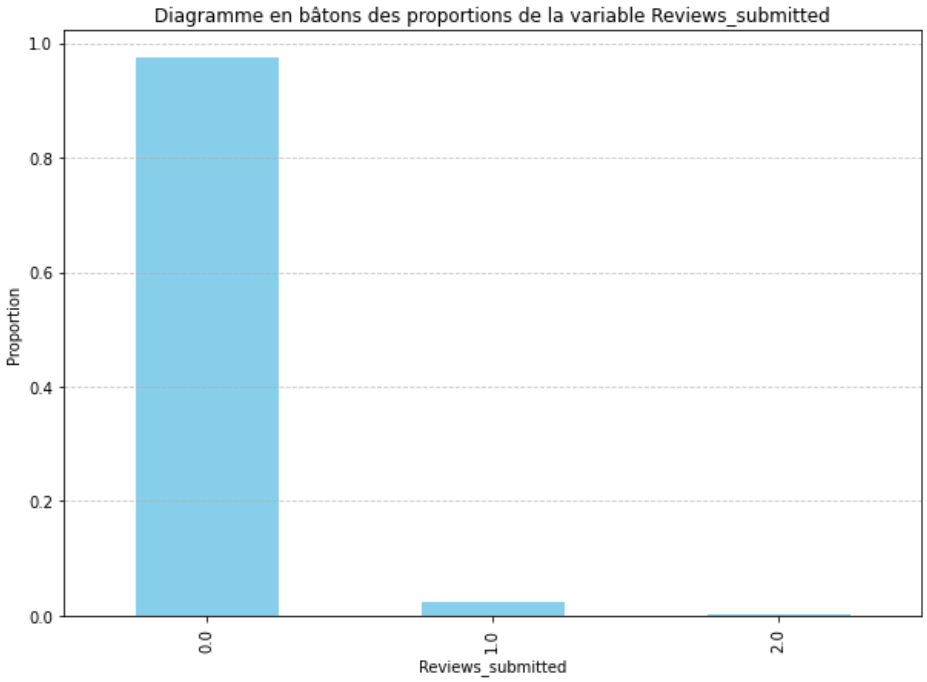

Figure 6: Distribution des avis

```
## count    4388.000000
## mean        0.027803
## std         0.172544
## min         0.000000
## 25%         0.000000
## 50%         0.000000
## 75%         0.000000
## max         2.000000
## Name: Reviews_submitted, dtype: float64
```

Interprétation :

De plus, l'analyse révèle que la majorité des clients n'a pas soumis d'avis, bien que certains aient soumis jusqu'à deux avis. Cette observation met en lumière l'importance d'encourager les clients satisfaits à laisser des avis positifs, ce qui peut influencer la réputation en ligne du restaurant.

- **Analyse bivariée des variables catégorielles**

Dans cette partie, nous avons essayer de voir les différentes relations entre les varianles.

Les tests de Chi-deux ont été utilisés pour évaluer les relations entre certaines variables catégorielles. Les résultats révèlent des associations significatives :

- **Gender vs. Guest_Status**

Interprétation

A partie de ce graphique, on constate que le segment le plus courant pour les deux genres est le statut "Normal". Cependant, le segment "Très bon client" est plus représenté chez les femmes que chez les hommes, de même que le segment "Bon client". Pour confirmé cette hypothèse, le test de χ^2 est realisé avec l'hypothèse nulle: *il n'y a pas de relation significative entre les variables étudiées.*

```
## Guest_Status  Good customer  Normal  Very good customer
## Gender
```

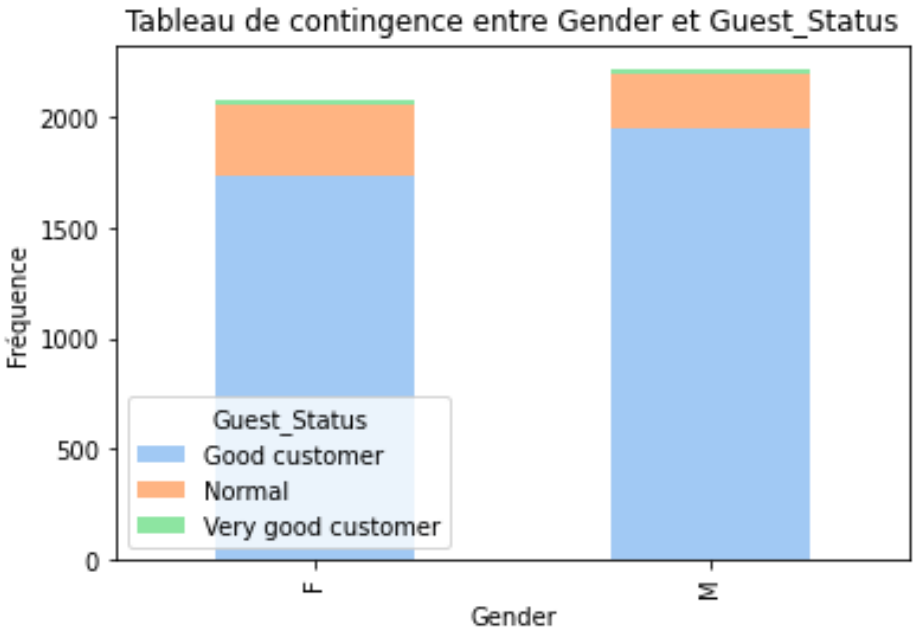

Figure 7: Analyse bivariables entre Gender et Guest_Status

```
## F                      1731     323                  30
## M                      1951     245                  21
##
## Résultats du test du Chi-deux :
## Chi2 : 21.352185997790908
## p-value : 2.3090414139082715e-05
## Degrés de liberté : 2
## Les variables sont dépendantes.
```

Interprétation :

On obtient (Chi2 = 21.35, p < 0.001) donc on rejete l'hypothèse nulle c'est que le genre peut influencer le comportement des clients vis-à-vis du restaurant.

L'analyse bivariables entre le genre et les autres revèle également une dépendance significative entre le genre des clients et plusieurs variables étudiées, telles que l'opt-in pour les e-mails et les SMS, ainsi que la participation aux avis. En général, les femmes semblent être plus enclines à participer aux communications marketing et à laisser des avis que les hommes.

- **Email_optin_market vs. SMS_optin_market**

```
## SMS_optin_review      no   yes
## Email_optin_market
## no                  2048  1707
## yes                    0   633
##
## Résultats du test du Chi-deux :
## Chi2 : 645.2131905257993
## p-value : 2.455578264409328e-142
## Degrés de liberté : 1
## Les variables sont dépendantes.
```

Interprétation :

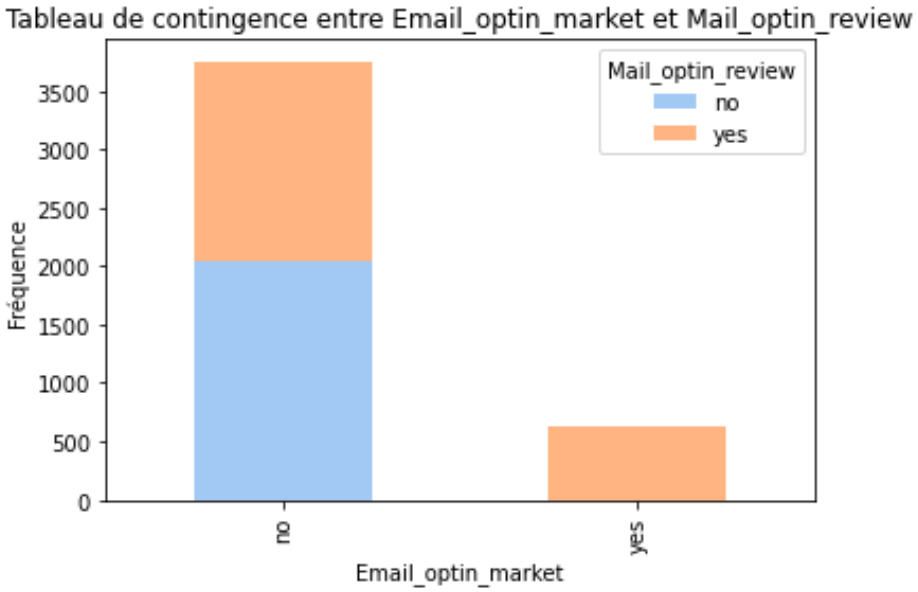

Figure 8: Analyse bivariables entre Email_optin_market et SMS_optin_market

Une forte association a été trouvée entre l'opt-in des clients pour les e-mails et les SMS à des fins de marketing (Chi2 = 1642.19, $p < 0.001$). Cela indique que les clients qui acceptent de recevoir des e-mails sont également plus enclins à recevoir des SMS.

Notons également que le même comportement s'est observé entre les variables **Mail_optin_review vs. SMS_optin_review**: c'est-à-dire une relation significatif a été observé entre l'opt-in des clients pour les avis par e-mail et par SMS (Chi2 = 4383.98, $p < 0.001$), soulignant une corrélation entre ces deux modes de communication.

- **Analyse bivariée des variables quantitatives**

```
## Matrice de corrélation entre les variables quantitatives :

##                     Bookings_number  Sent_messages  Reviews_submitted
## Bookings_number            1.000000       0.054667           0.074125
## Sent_messages              0.054667       1.000000           0.052078
## Reviews_submitted          0.074125       0.052078           1.000000
```

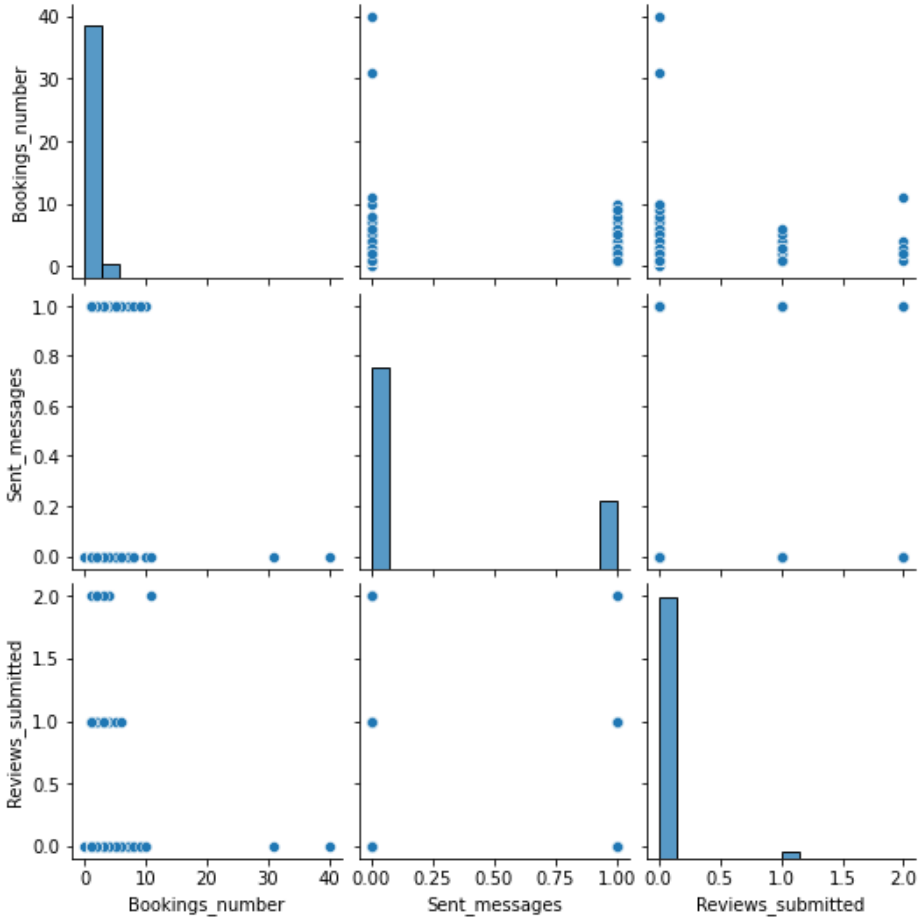

Figure 9: La matrice de dispersion pour les paires de variables quantitatives

Interprétation : selon ces coefficients de corrélation et le graphique, il n'y a pas de relation linéaire significative ou forte entre ces variables.

- **Analyse trivariables**

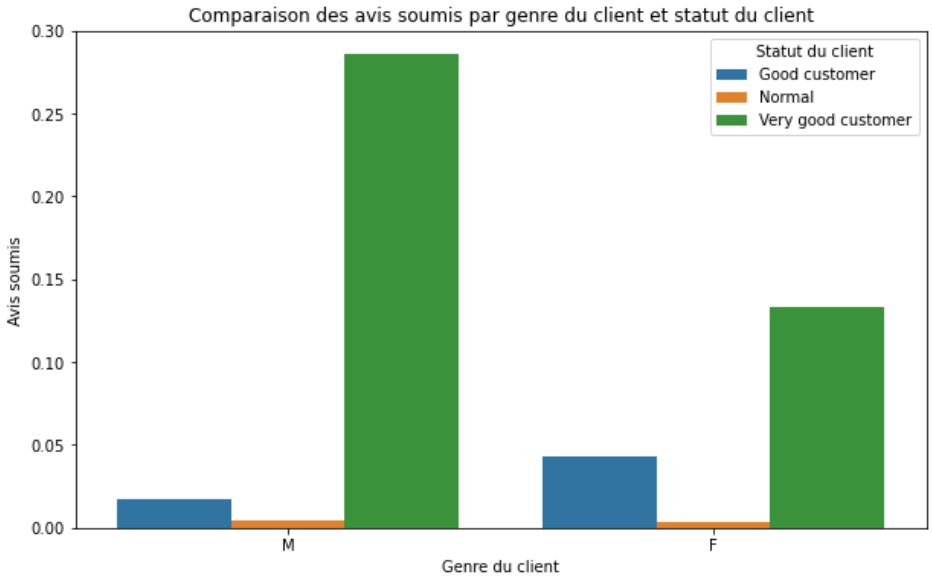

Figure 10: Comparaison des avis soumis par genre du client et statut du client

Interprétation : Les femmes soummettent plus avis que les hommes.

3.3 Analyse de la base de données des réservations

Dans cette partie, il a été question de voir les différentes aspects des reservations ce qui a été fait juste en utilisant des graphiques.

Interprétation :

- Le nombre de couverts est plus élevé au dîner qu'au déjeuner tout au long de la semaine. Cela suggère que le service du soir est plus fréquenté que celui du midi.
- Le samedi semble être le jour le plus fréquenté, avec un nombre de couverts plus élevé au dîner. Cela pourrait être un moment stratégique pour maximiser les revenus.
- Analyse par jour :
  - Lundi : Le déjeuner est plus bas que le dîner, mais il y a une légère augmentation en fin de semaine.
  - Mardi : Le déjeuner est similaire au lundi, mais le dîner est plus élevé.
  - Mercredi : Le déjeuner est stable, mais le dîner est plus élevé.
  - Jeudi : Le déjeuner est légèrement plus bas, mais le dîner est élevé.
  - Vendredi : Le déjeuner est bas, mais le dîner est très élevé.
  - Samedi : Le déjeuner est bas, mais le dîner atteint son maximum.

Interprétation :

Il y a une variabilité dans le nombre moyen de couverts pour les services de déjeuner et de dîner tout au long du mois. Certains jours montrent des différences significatives entre les deux services, tandis que d'autres

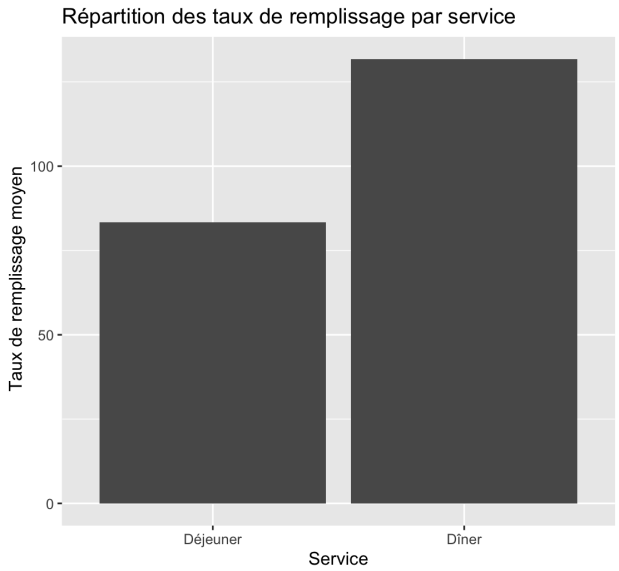

Figure 11: Visualiasation des taux de couveture moyen en focntion des services

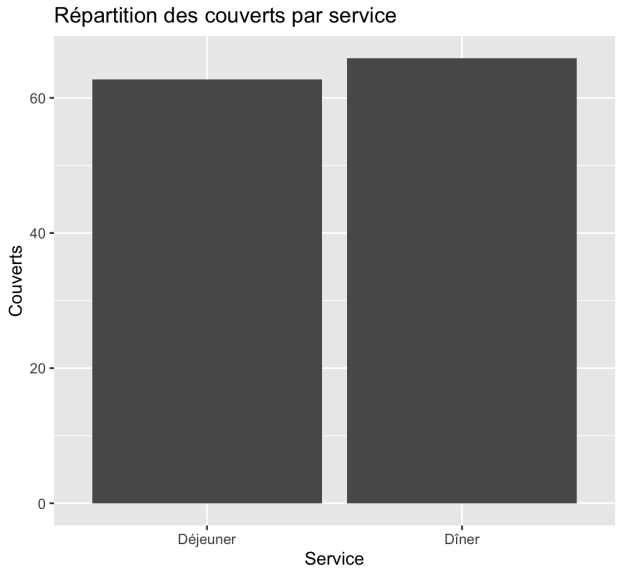

Figure 12: Visualiasation du nombre moyen de réservaion en focntion des services

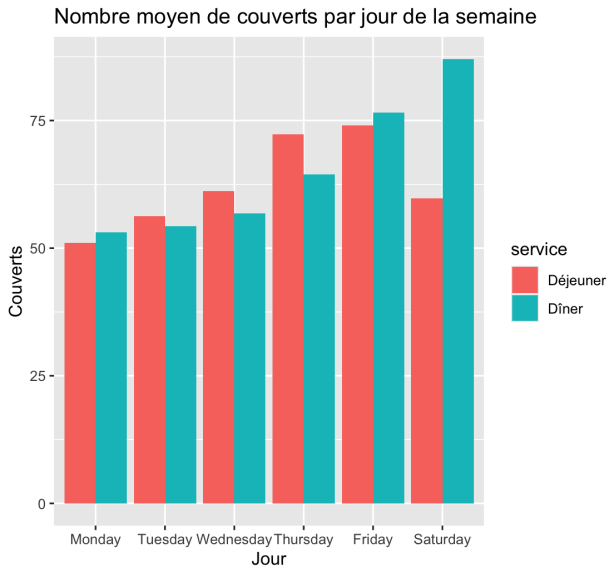

Figure 13: Visualistion de nombre moyens de couverts par jour

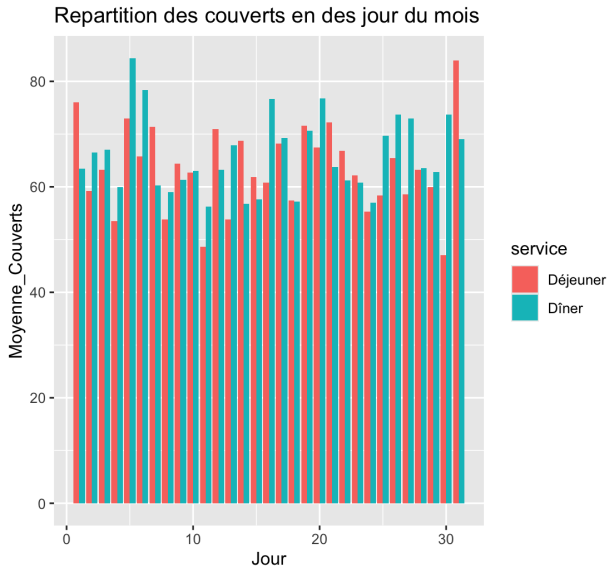

Figure 14: Visualistion de nombre moyens de couverts par jour dans un mois

jours présentent des moyennes similaires.

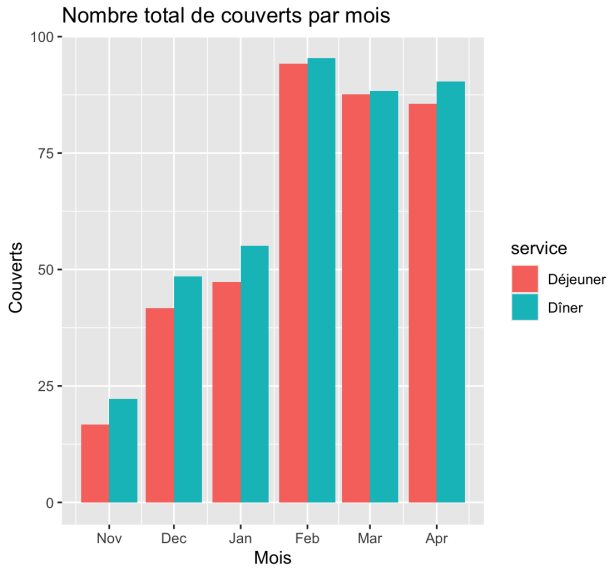

Figure 15: Visualistion de nombre moyens de couverts par mois

Interprétation :

Le nombre de couverts (réservations) augmente progressivement au fil des mois, passant de novembre à février et semble se stabiliser.

3.4 Prévision des réservations

L'objectif principal est de prédire le nombre de réservations quotidiennes. Étant donné que ces données sont temporelles, nous devons aborder ce problème en utilisant des techniques appropriées aux séries temporelles. Ainsi, la prédiction repose sur l'analyse et la modélisation de ces séries. Pour la prédiction initiale, j'ai exploré la possibilité de décomposer la variable temporelle en différentes composantes telles que le mois, le jour et l'année, afin de les utiliser comme variables explicatives dans un modèle de régression traditionnel. Cependant, cette approche n'a pas fourni des résultats satisfaisants, avec un faible pouvoir explicatif du modèle sur les données observées. C'est à ce stade que j'ai décidé de me tourner vers les modèles de séries temporelles, en particulier le modèle ARIMA, pour mieux capturer les structures et les tendances inhérentes aux données temporelles.

3.4.1 Série temporelle

- Une série temporelle, $y_t, t = 1, 2, \text{ůůů}, T$, est une suite d'observations d'un phé- nomène, indicées par une date, un temps. Elle caractérisée par plusieurs choses a savoir :
- La tendance ou trend mt capte l'orientation à long terme de la série.
- Composante saisonnière : la composante saisonnière st capte un comporte- ment qui se répète avec une certaine périodicité (toutes les douze périodes pour des données mensuelles, toutes les sept périodes pour des données quotidiennes. . .).
- Composante irrégulière: c'est une composante d'erreur, ut. Idéalement, elle est de faible variabilité par rapport aux autres composantes.

A ces trois composantes, on ajoute parfois un cycle.

- On appelle cycle un comportement qui se répète assez régulièrement mais avec une périodicité inconnue et changeante.

3.4.2 Modèle ARIMA

Le modèle ARIMA (AutoRegressive Integrated Moving Average) est une méthode couramment utilisée pour modéliser et prévoir les séries temporelles. Il combine les composantes de l'autorégression (AR) et de la moyenne mobile (MA) avec une différenciation (I) pour prendre en compte les tendances et les comportements saisonniers dans les données.

- **Composantes du modèle ARIMA**
  - **AR (Autorégression)** : L'AR fait référence à la régression linéaire des valeurs actuelles sur les valeurs précédentes de la série temporelle. Il capture les effets de dépendance linéaire à partir des valeurs passées.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

où :

  - y_t : La valeur de la série temporelle à l'instant t.
  - c : Terme constant ou intercepte.
  - $\phi_1, \phi_2, \ldots, \phi_p$: Les coefficients des termes retardés de la série temporelle.
  - $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$: Les valeurs retardées de la série temporelle.
  - ε_t : Terme d'erreur à l'instant t.
  - **MA (Moyenne mobile)** : Le MA utilise la moyenne mobile des erreurs précédentes pour modéliser la relation entre les résidus et les observations actuelles de la série temporelle. Il capture les effets de dépendance entre les résidus.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}$$

où :

  - y_t : La valeur de la série temporelle à l'instant t.
  - c : Terme constant ou intercepte.
  - ε_t : Terme d'erreur à l'instant t.
  - $\theta_1, \theta_2, \ldots, \theta_q$: Les coefficients des termes d'erreur retardés.
  - $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$: Les termes d'erreur retardés.
  - **I (Différenciation)** : La différenciation est utilisée pour rendre les données stationnaires en supprimant les tendances et les comportements saisonniers. Elle consiste à prendre la différence entre les observations consécutives jusqu'à obtenir une série stationnaire.

$$\Delta y_t = y_t - y_{t-1}$$

où :

  - Δy_t : La différence entre la valeur de la série temporelle à l'instant t et celle à l'instant $t-1$.
  - y_t : La valeur de la série temporelle à l'instant t.
  - y_{t-1} : La valeur de la série temporelle à l'instant $t-1$.

- **Ordres du modèle ARIMA**

Le modèle ARIMA est défini par trois ordres : p, d et q.

- **p (Ordre AR)** : L'ordre AR spécifie le nombre de termes autorégressifs à inclure dans le modèle. Il indique combien de valeurs passées sont utilisées pour prédire la valeur actuelle.

- **d (Ordre de différenciation)** : L'ordre de différenciation indique combien de fois la série doit être différenciée pour rendre les données stationnaires.

- **q (Ordre MA)** : L'ordre MA spécifie le nombre de termes de la moyenne mobile à inclure dans le modèle. Il indique combien de résidus passés sont utilisés pour prédire l'observation actuelle.

- **Méthodologie de Box-Jenkins le modèle ARIMA**

La méthodologie de Box-Jenkins est une approche couramment utilisée pour modéliser et prévoir les séries temporelles. Elle comprend les étapes suivantes :

Identification du modèle

- Analyse des données : Examiner les données de la série temporelle pour détecter les tendances, les saisonnalités et les comportements anormaux.
- Différenciation : Si la série temporelle présente une tendance ou une saisonnalité, appliquer une différenciation pour rendre les données stationnaires.
- Identification des ordres : Utiliser les graphiques ACF (fonction d'autocorrélation) et PACF (fonction d'autocorrélation partielle) pour déterminer les ordres p, d et q du modèle ARIMA.

Estimation du modèle

- Estimation des paramètres : Utiliser les méthodes d'estimation (telles que la méthode des moindres carrés) pour estimer les paramètres du modèle ARIMA.

Vérification du modèle

- Diagnostic du modèle : Vérifier si les résidus du modèle ARIMA sont bruit blanc (c'est-à-dire s'ils ne présentent pas de structure ou de corrélation significative).
- Réajustement : Si le modèle ne satisfait pas les critères de bruit blanc, ajuster les ordres du modèle ARIMA et répéter les étapes précédentes.
- Validation : Valider les performances du modèle en effectuant des prédictions sur des données de validation ou en utilisant des mesures d'évaluation telles que l'erreur quadratique moyenne (RMSE) ou le critère d'information d'Akaike (AIC).

ACF et PACF

L'ACF (Autocorrelation Function) et la PACF (Partial Autocorrelation Function) sont deux outils essentiels en analyse de séries temporelles. Elles permettent de comprendre les dépendances temporelles dans une série de données.

- **ACF (Autocorrelation Function)**

L'ACF mesure la corrélation entre une séquence et elle-même à différentes périodes de temps. Autrement dit, elle permet d'évaluer la similitude entre les observations en fonction du décalage de temps entre elles.

Pour un décalage 'k', l'ACF mesure la corrélation entre la série temporelle et elle-même décalée de 'k' périodes.

Par exemple, **une ACF de 0,9 à un décalage de 2 signifie que les données sont très similaires à elles-mêmes il y a deux périodes de temps**.

- **PACF (Partial Autocorrelation Function)**

La PACF est une corrélation qui exclut l'effet des termes intermédiaires. C'est-à-dire qu'elle est une mesure de la corrélation entre des observations à un certain intervalle, en tenant compte des valeurs à des intervalles plus courts.

Pour un décalage 'k', **la PACF est la corrélation entre la série temporelle et elle-même décalée de 'k' périodes, mais après avoir soustrait les effets des décalages 1 à 'k-1'**.

Par exemple, une PACF de 0,5 à un décalage de 3 signifie que les données sont moyennement similaires à elles-mêmes il y a trois périodes de temps, après avoir pris en compte les décalages 1 et 2.

- **Utilisations**

Les graphiques ACF et PACF sont couramment utilisés pour aider à choisir les paramètres d'un modèle ARIMA en analyse de séries temporelles. Par exemple, **le graphique ACF peut être utilisé pour identifier le terme MA (moyenne mobile)** du modèle, tandis que le graphique **PACF peut aider à identifier le terme AR (auto-régressif).**

3.4.3 Application :

Cette partie détaille l'application du modele arima au données de réservation comme indiqué dans la méthodologie de Box-Jenkins le modèle ARIMA précédement, la première étape est d'identifé le modèle.

- **Analyse de la série**

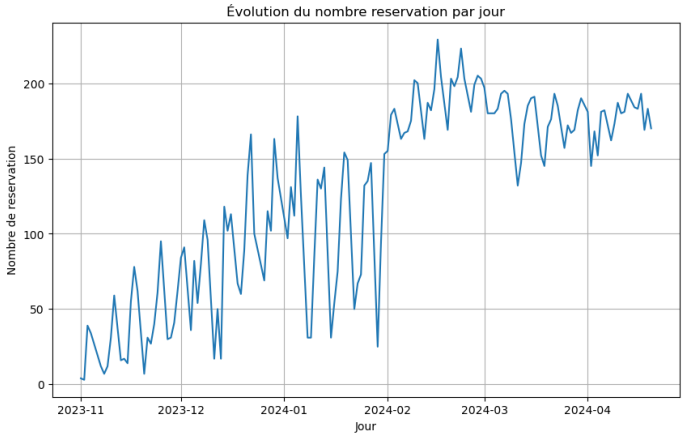

Figure 16: Evolution du nombre de reservation par jour

La figure ci - dessus montre l'évolution des réservations depuis l'ouverture du restaurant.

- **Décomposition**
  - **Série originale** : Cette courbe représente les données brutes au fil du temps. On peut voir des fluctuations significatives, mais il semble y avoir une tendance générale à la hausse.
  - **Tendance** : La deuxième courbe, intitulée "Tendance", montre la composante de tendance de la série. Elle indique une augmentation globale au fil du temps.
  - **Saisonnalité** : La troisième courbe, "Saisonnalité", représente la composante saisonnière. On observe des motifs réguliers qui se répètent à des intervalles spécifiques.
  - **Résidus** : Enfin, la quatrième courbe, "Résidus", montre la composante résiduelle. Cela semble être du bruit aléatoire restant après avoir extrait les composantes de tendance et saisonnières de la série originale.
- **Stationnarité**

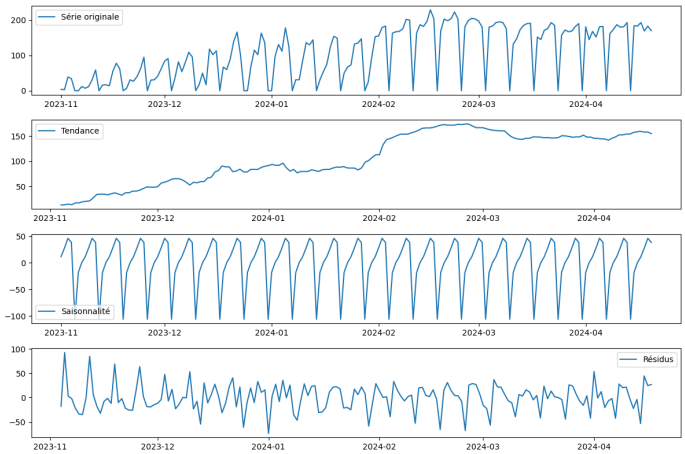

Figure 17: Décomposition de la série temporelle

Pour vérifier la stationarité de la série, nous avons éffectué le test de Dickey-Fuller augmenté ce qui a révelé que la série n'est pas stationnaire.

```
## | Métrique       | Valeur                          |
## |----------------|---------------------------------|
## | Valeur de test | -1.709276980088658              |
## | P-valeur       | 0.4263391477148763              |
## | Conclusion     | La série est non stationnaire |
```

- **Différenciation de la série temporelle**

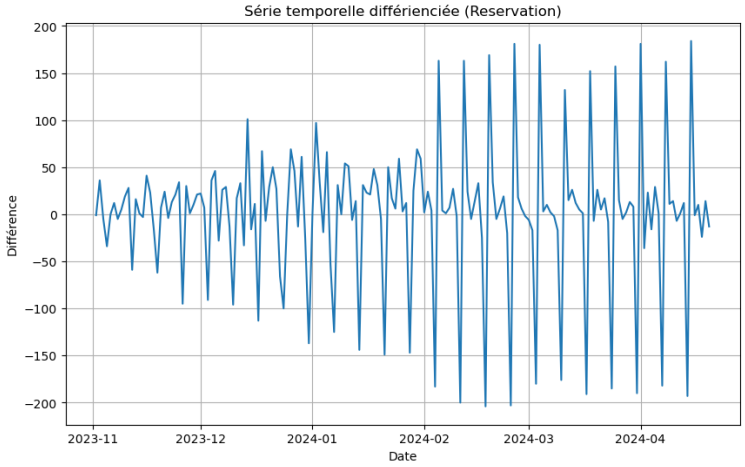

Figure 18: Série temporelle différenciée

```
## | Métrique       | Valeur                      |
## |----------------|-----------------------------|
## | Valeur de test | -3.7982378564247794         |
## | P-valeur       | 0.0029253955695050203       |
## | Conclusion     | La série est stationnaire |
```

En suite, la série a été différentiée et en représentant la série différentié puis en re-effectuant le test de Dickey-Fuller augmenté, on a conclure que ce série différentiée est stationnaire. Ainsi le paramètre **d** du modèles est égal à 1.

- **ACF et PACF de série de réservation**

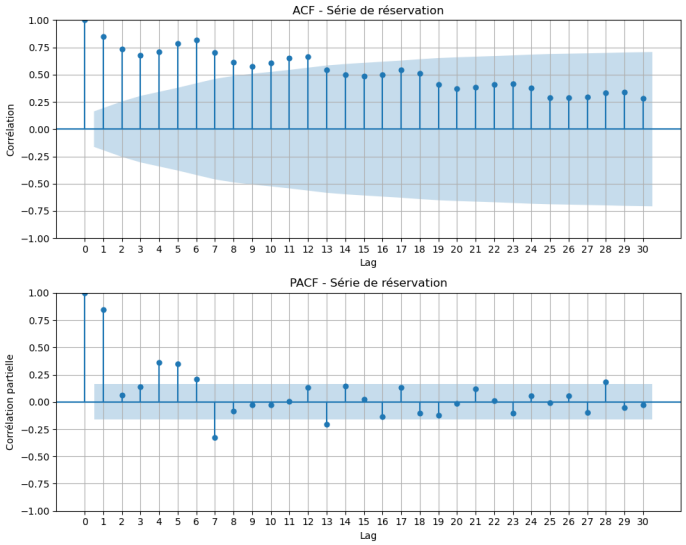

Figure 19: ACF et PACF de série de réservation

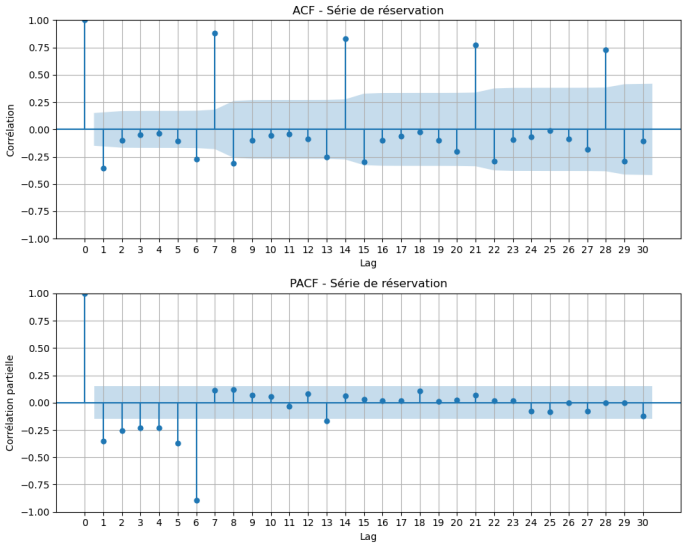

Figure 20: ACF et PACF de série différenciée de réservation

- **Analyse de l'ACF**

Le graphique ACF (AutoCorrelation Function) de la série de passagers aériens montre des valeurs significatives aux décalages 1, 7, 14, 21, etc., suggérant un comportement périodique. Ces premières valeurs significatives indiquent potentiellement la présence d'une composante de moyenne mobile (MA). On recherche le premier décalage où la corrélation tombe sous le seuil de signification, on prend donc q = 2.

- **Analyse du PACF**

Le graphique PACF (Partial AutoCorrelation Function) de la série de réservation montre des valeurs significatives aux décalages 1 et 6, tandis que les valeurs au-delà du lag 1 sont proches de zéro. La PACF est utilisée pour déterminer le nombre de termes autorégressifs (AR). Dans ce cas, le premier lag significatif est à 6 donc p = 6.

En résumé, on a pour paramètre du modèle :

- p = 6
- d = 1
- q = 2

- **Estimation du modèle**

Une fois que les paramètres du modèle sont trouvé nous passons à l'estimation du modèle. Pour cela on divise la base de donnée en deux partie :

- *data_train* pour entrainer le modèle comportant les 152 premières observations.
- *data_test* pour tester le modèle comportant les 20 dernières observations.

On compareras ce modèle avec le modèle trouver automatiquement avec la fonction *auto_arima*.

```
##                               SARIMAX Results
## ==============================================================================
## Dep. Variable:                Couverts   No. Observations:                  152
## Model:                 ARIMA(6, 1, 2)   Log Likelihood                -703.715
## Date:                Fri, 31 May 2024   AIC                           1425.429
## Time:                        23:45:48   BIC                           1452.585
## Sample:                  11-01-2023   HQIC                          1436.461
##                        - 03-31-2024
## Covariance Type:                 opg
## ==============================================================================
##                  coef    std err          z      P>|z|      [0.025      0.975]
## ------------------------------------------------------------------------------
## ar.L1         -0.9857      0.039    -25.174      0.000      -1.062      -0.909
## ar.L2         -0.9989      0.043    -23.217      0.000      -1.083      -0.915
## ar.L3         -0.9693      0.052    -18.804      0.000      -1.070      -0.868
## ar.L4         -0.9579      0.058    -16.598      0.000      -1.071      -0.845
## ar.L5         -0.9504      0.050    -18.896      0.000      -1.049      -0.852
## ar.L6         -0.9122      0.045    -20.125      0.000      -1.001      -0.823
## ma.L1          0.1878      0.096      1.965      0.049       0.000       0.375
## ma.L2          0.1678      0.089      1.894      0.058      -0.006       0.341
## sigma2       670.4361     58.905     11.382      0.000     554.985     785.888
## ===================================================================================
## Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):               154.71
## Prob(Q):                              0.86   Prob(JB):                         0.00
## Heteroskedasticity (H):               0.47   Skew:                             0.86
## Prob(H) (two-sided):                  0.01   Kurtosis:                         7.65
## ===================================================================================
##
## Warnings:
## [1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

- Les coefficients AR (AutoRegressive) semblent tous négatifs et significatifs, ce qui suggère une forte corrélation négative entre les valeurs passées et actuelles de la série temporelle.
- Les coefficients MA (Moving Average) ne sont significatifs que pour les deux premiers retards, ce qui signifie que les erreurs de prédiction dépendent principalement des erreurs récentes de prédiction.
- Le paramètre de variance sigma2 est estimé à environ 670, indiquant la variance des erreurs de prédiction.
- Les tests de Ljung-Box et de Jarque-Bera sont utilisés pour vérifier les hypothèses du modèle. Dans ce cas, le test de Ljung-Box montre qu'il n'y a pas d'autocorrélation significative dans les erreurs du modèle, tandis que le test de Jarque-Bera suggère que les résidus ne suivent pas parfaitement une distribution normale.

- **Vérication du modèle**
  - **Q-Q Plot**

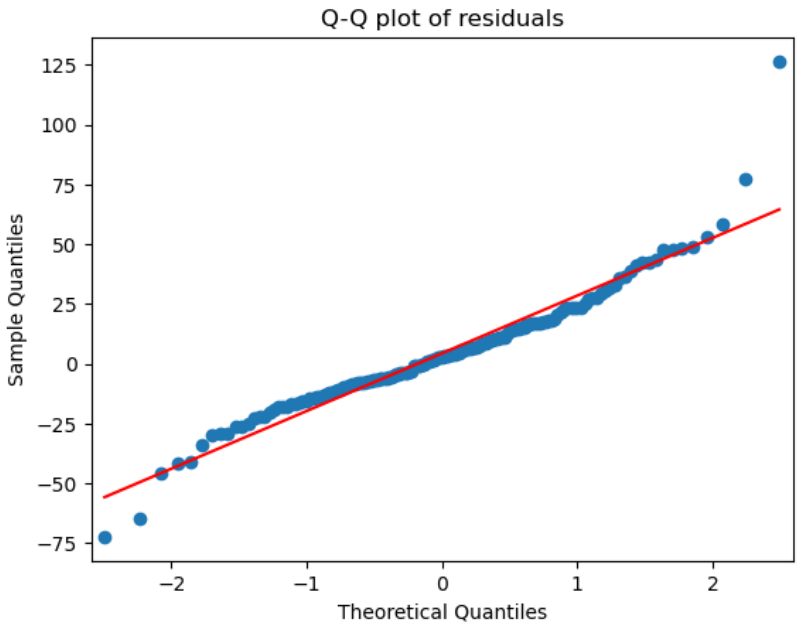

Figure 21: ACF des résidus

La plupart des résidus semblent suivre une distribution normale, mais quatre résidus s'éloignent de la droite, on pourrait s'intéresser à eux pour voir leurs influencent.

- **Résidus ACF et PACF**

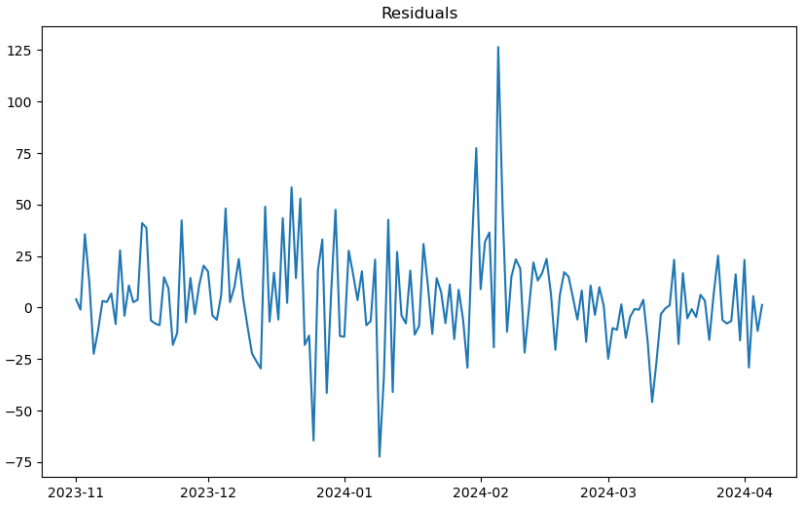

Figure 22: Résidus

**Interprétation*

Les points bleus sont dispersés aléatoirement autour de la ligne zéro (sans motif clair), cela suggère que les résidus sont indépendants dans le temps. Cela signifie que les erreurs du modèle ne sont pas corrélées d'un pas de temps à l'autre.

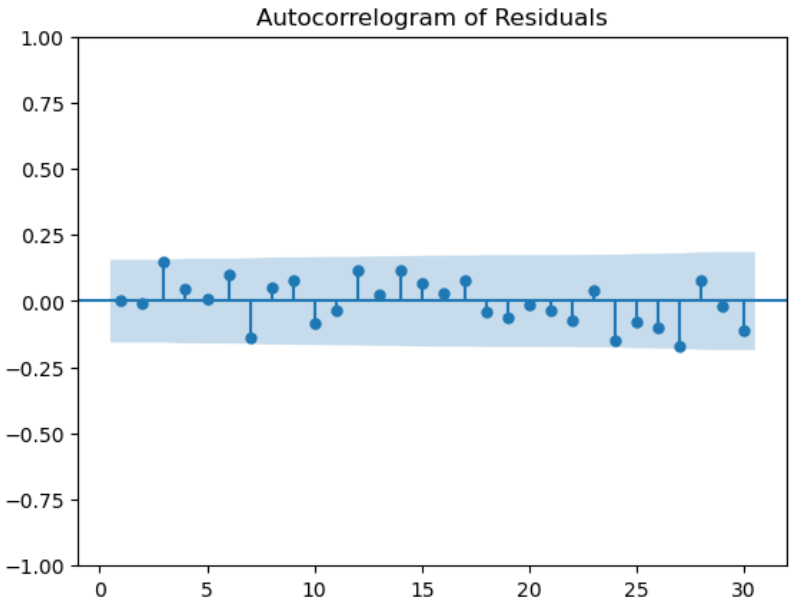

Figure 23: ACF des résidus

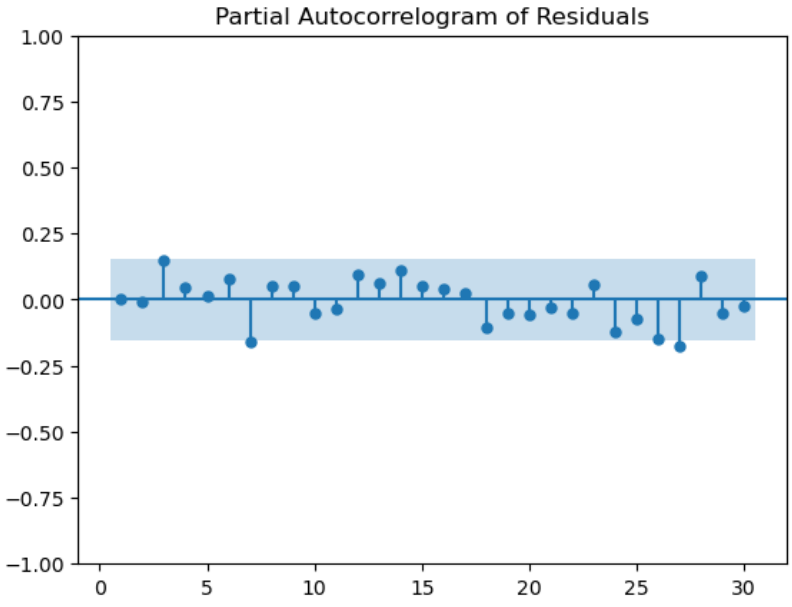

Figure 24: PACF des résidus

- **Prédiction et performance**
- Analysons ici la perfomance du modèle.

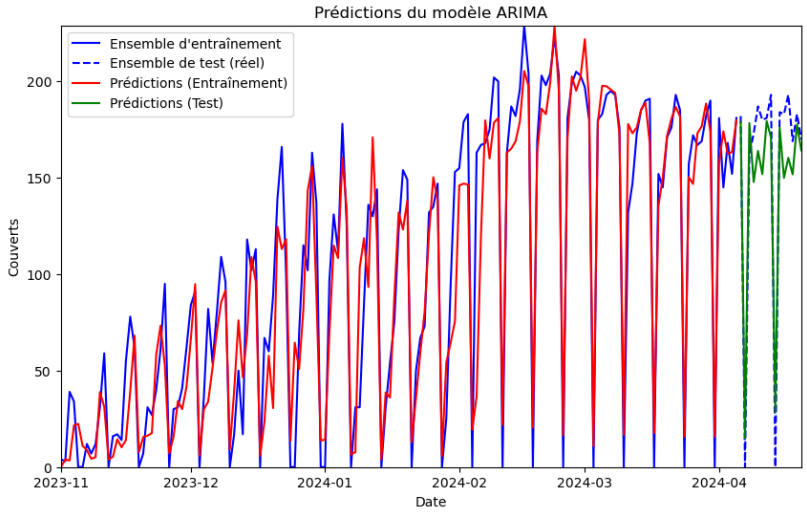

Figure 25: Pédictions du modèle

```
## /Users/peter/.virtualenvs/r-reticulate/lib/python3.9/site-packages/sklearn/metrics/_regression.py:49
##   warnings.warn(
```

```
## /Users/peter/.virtualenvs/r-reticulate/lib/python3.9/site-packages/sklearn/metrics/_regression.py:49
##   warnings.warn(
```

```
##   Métrique  Ensemble d'entraînement  Ensemble de test
## 0      MAE                17.625284         13.714066
## 1      MSE               609.737118        317.855881
## 2     RMSE                24.692856         17.828513
## 3       R²                 0.888291          0.891518
```

Les métriques montrent que le modèle de prédiction des réservations performe bien, tant sur l'ensemble d'entraînement que sur l'ensemble de test :

- Les erreurs de prédiction (MAE et RMSE) sont raisonnablement basses, surtout sur l'ensemble de test, ce qui indique que le modèle est capable de faire des prédictions précises sur des données nouvelles.

- Les valeurs du MSE et du RMSE sont plus basses pour l'ensemble de test que pour l'ensemble d'entraînement, ce qui suggère que le modèle ne souffre pas de surapprentissage (overfitting) et qu'il généralise bien.

- Le coefficient de détermination (R^2) est élevé pour les deux ensembles, ce qui montre que le modèle explique bien la variabilité des réservations.

- **Modèle obtenu aveac la fonction arima(modèle automatique)**

Cette partie explore le modèle trouver a partie de la fonction *auto_arima* de la librairy *pmdarima*.

- **Modèle**

```
## /Users/peter/.virtualenvs/r-reticulate/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSL
##   warnings.warn(
```

```
##                                SARIMAX Results
## ==============================================================================
## Dep. Variable:                      y   No. Observations:                  152
## Model:               SARIMAX(5, 1, 3)   Log Likelihood                -757.036
## Date:                Fri, 31 May 2024   AIC                           1534.072
## Time:                        23:45:52   BIC                           1564.245
## Sample:                  11-01-2023   HQIC                          1546.330
##                        - 03-31-2024
## Covariance Type:                 opg
## ==============================================================================
##                  coef    std err          z      P>|z|      [0.025      0.975]
## ------------------------------------------------------------------------------
## intercept      2.5032      1.360      1.841      0.066      -0.162       5.168
## ar.L1          0.0646      0.098      0.662      0.508      -0.127       0.256
## ar.L2         -0.7043      0.065    -10.809      0.000      -0.832      -0.577
## ar.L3         -0.3555      0.109     -3.260      0.001      -0.569      -0.142
## ar.L4         -0.4031      0.082     -4.911      0.000      -0.564      -0.242
## ar.L5         -0.6037      0.109     -5.523      0.000      -0.818      -0.389
## ma.L1         -1.3143      0.108    -12.152      0.000      -1.526      -1.102
## ma.L2          1.1105      0.144      7.717      0.000       0.828       1.393
## ma.L3         -0.3634      0.098     -3.722      0.000      -0.555      -0.172
## sigma2      1223.1314    139.757      8.752      0.000     949.212    1497.051
## ===================================================================================
## Ljung-Box (L1) (Q):                   1.63   Jarque-Bera (JB):                 3.93
## Prob(Q):                              0.20   Prob(JB):                         0.14
## Heteroskedasticity (H):               2.88   Skew:                             0.27
## Prob(H) (two-sided):                  0.00   Kurtosis:                         3.57
## ===================================================================================
##
## Warnings:
## [1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Interprétation

- Test de Ljung-Box (Q) : La probabilité associée au test de Ljung-Box (Prob(Q) = 0.20) indique qu'il n'y a pas d'autocorrélation significative dans les résidus. Cela suggère que le modèle a bien capturé la structure temporelle des données. Test de Jarque-Bera (JB) :
- La probabilité associée au test de Jarque-Bera (Prob(JB) = 0.14) n'indique pas de non-normalité significative dans les résidus. Cela suggère que les résidus suivent une distribution normale, ce qui est un bon signe pour la validité du modèle. Hétéroscédasticité :
- Le test d'hétéroscédasticité (Prob(H) = 0.00) indique une présence significative d'hétéroscédasticité, suggérant que la variance des erreurs n'est pas constante. Cela peut indiquer que les erreurs sont plus grandes à certains points de la série temporelle.
- Analyse de la variance des résidus Variance des résidus (sigma2 = 1223.1314) : La valeur élevée de sigma2 indique une grande variabilité dans les erreurs résiduelles. Cela pourrait suggérer que le modèle ne capture pas complètement la variabilité des données.

- **Prédiction et perfomance du model automatique**

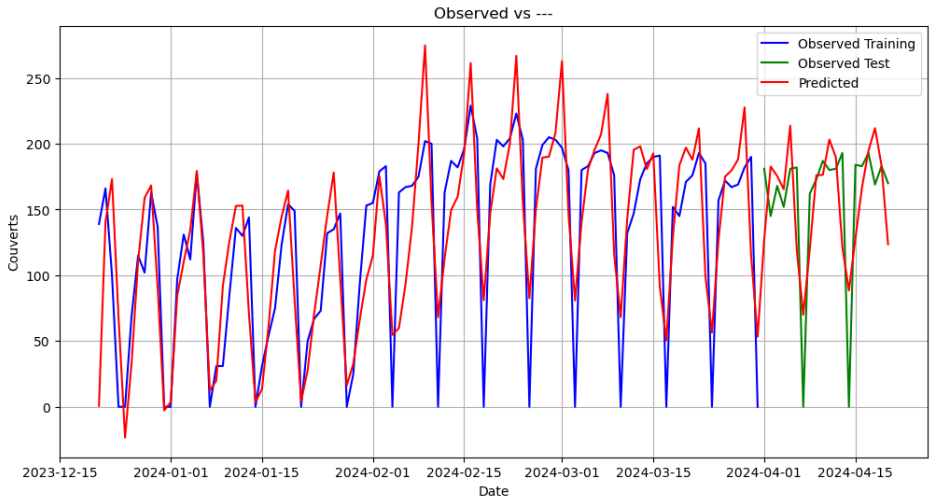

Figure 26: Pédictions du modèle automatique

```
## ARIMA(order=(5, 1, 3), scoring_args={}, suppress_warnings=True)
```

```
## /Users/peter/.virtualenvs/r-reticulate/lib/python3.9/site-packages/sklearn/metrics/_regression.py:49
##   warnings.warn(
```

```
## /Users/peter/.virtualenvs/r-reticulate/lib/python3.9/site-packages/sklearn/metrics/_regression.py:49
##   warnings.warn(
```

```
##   Métrique  Ensemble d'entraînement  Ensemble de test
## 0      MAE                26.761868         32.559989
## 1      MSE              1203.051280       1632.254857
## 2     RMSE                34.685030         40.401174
## 3       R²                 0.779590          0.442922
```

Les métriques montrent que le modèle de prédiction des réservations a des performances inférieures, surtout sur l'ensemble de test :

- MAE et RMSE : Les erreurs de prédiction sont significativement plus élevées sur l'ensemble de test que sur l'ensemble d'entraînement, indiquant une baisse de précision sur les nouvelles données.
- MSE : Une augmentation du MSE sur l'ensemble de test par rapport à l'ensemble d'entraînement suggère que les prédictions comportent des erreurs importantes sur les nouvelles données.
- R^2 : Un R^2 plus faible sur l'ensemble de test indique que le modèle n'explique pas bien la variance des données de test.

3.4.4 Comparaison des deux modèles

En analysant les perfomance des deux modèles, on retient que :

```
##   Métrique  Ensemble de test  Ensemble de test df
## 0      MAE         32.559989            13.714066
## 1      MSE       1632.254857           317.855881
## 2     RMSE         40.401174            17.828513
## 3       R²          0.442922             0.891518
```

- Le premier modèle montre de meilleures performances sur toutes les métriques par rapport au deuxième modèle(Modele automatique), tant sur l'ensemble d'entraînement que sur l'ensemble de test.

- Le premier modèle a des valeurs de MAE, MSE, RMSE inférieures, ce qui indique une meilleure précision et une meilleure performance générale.
- De plus, le premier modèle a un R^2 plus élevé, ce qui signifie qu'il explique une plus grande proportion de la variance des données.

En conclusion premier modèle est supérieur en termes de précision et de capacité à expliquer la variance des données par rapport au deuxième modèle.

3.4.5 Autre modèle

Bien que le premier modèle de prédiction des réservations ait montré une performance impressionnante, il présente des limitations significatives. En effet, ce modèle repose uniquement sur la variable représentant le nombre de réservations, sans tenir compte de nombreux autres facteurs potentiellement influents. Pour développer un modèle plus robuste et précis, nous explorons l'idée de créer et d'incorporer de nouvelles variables. Parmi celles-ci, nous incluons :

- *Jour de la semaine (day_of_week)* : Influence les habitudes de réservation des clients.
- *Heure de la journée (time_of_day)* : Permet de capter les variations horaires des réservations.
- *Mois (month)* : Prend en compte les tendances mensuelles et saisonnières.
- *Saison (season)* : Influence les comportements de réservation en fonction des saisons.
- *Événements locaux (local_event)* : Les événements spéciaux peuvent augmenter les réservations.
- *Jours fériés (holiday)* : Les jours fériés peuvent voir une fluctuation des réservations.
- *Conditions météorologiques (weather)* : Le temps peut affecter la volonté des clients de sortir.
- *Promotions (promotion)* : Les offres spéciales peuvent stimuler les réservations.
- *Marketing (marketing_campaign)* : Les campagnes publicitaires peuvent attirer plus de clients.
- *Disponibilité des tables (table_availability)* : La capacité de réservation peut varier.

En intégrant ces nouvelles variables dans notre modèle, nous visons à mieux comprendre et à capturer les motivations des clients pour effectuer des réservations. Cette approche holistique permet de créer un modèle de prédiction plus précis et fiable, optimisant ainsi la planification des tables et améliorant l'expérience globale des clients au restaurant

3.5 Conception de l'outil de planification

L'application de planification développée repose sur le framework Shiny, qui permet la création d'interfaces utilisateur interactives en utilisant le langage de programmation R. Nous allons examiner les différents éléments qui composent l'application, comme sa structure générale et les technologies utilisées pour la développer.

- **Architecture Globale**

L'application est conçue pour être conviviale et intuitive, avec une disposition en onglets facilitant la navigation entre les différentes fonctionnalités. Elle se compose de trois onglets principaux :

- *Systhème d'authentification* : Permettant la connexion et la deconnection des utilisateurs.
- *Calendrier* : Permet aux utilisateurs de visualiser les services planifiés dans un format de calendrier interactif.
- *Gestion des Services* : Offre aux gestionnaires la possibilité d'ajouter de nouveaux services, de supprimer des services existants et de consulter la liste complète des services planifiés.
- *Calcule des services* : Permet au gestionnaires et employers de calculer le nombre de service effectués entre deux dates.

- **Justification des Choix de Conception**

Le choix de Shiny comme framework pour le développement de l'application était motivé par sa simplicité d'utilisation et sa capacité à créer des interfaces utilisateur interactives avec peu de code.

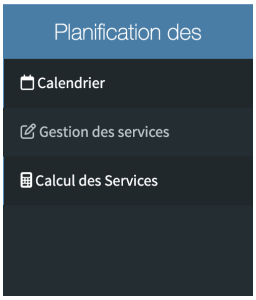

Figure 27: Les differentes parties de l'application

La base de données SQLite a été choisie pour stocker les données des services en de sa simplicité d'utilisation et de sa compatibilité avec R.

- **Fonctionnalités**

L'application a plusieurs fonctionalité telsque :

- *Ajout de Services* : les gestionnaires peuvent ajouter de nouveaux services via un formulaire. Les informations nécessaires incluent la date, le type de service et nom de l'employer.

Code permettant d'executer cette fonctionnalité :

```
# Ajouter un service
  observeEvent(input$add_service, {
    for (employee in input$employees) {
      dbExecute(conn, "INSERT INTO services (date, content, type, employee_name) VALUES (?, ?, ?, ?)",
                params = list(as.character(input$date), input$service, input$type, employee))
    }
    output$service_table <- renderDT(dbGetQuery(conn, "SELECT * FROM services WHERE date >= ?", params =
  })
  })
```

- *Suppression de Services* : les gestonnaires peuvent supprimer des services existants en sélectionnant le type de service et l'employé associé, puis en cliquant sur le bouton "Supprimer".

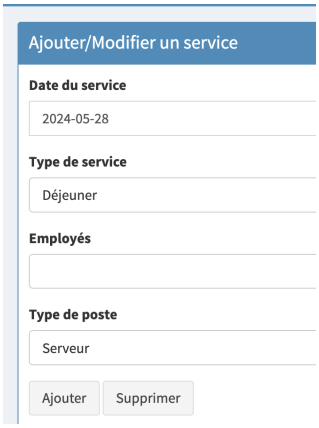

Figure 28: Ajout/Suppression

Code permettant d'executer cette fonctionnalité :

```r
# Supprimer un service
  observeEvent(input$delete_service, {
    for (employee in input$employees) {
      dbExecute(conn, "DELETE FROM services WHERE date = ? AND content = ? AND type = ? AND employee_nam
                params = list(as.character(input$date), input$service, input$type, employee))
    }
    output$service_table <- renderDT(dbGetQuery(conn, "SELECT * FROM services WHERE date >= ?", params =
  })
```

- *Affichage de la liste des services* : l'application affiche la liste des services planifiés dans un tableau interactif, récupéré dynamiquement depuis la base de données SQLite.

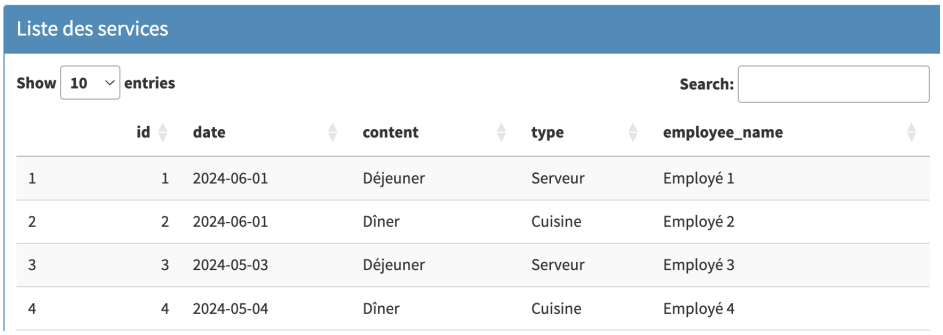

Figure 29: Affichage des liste des services

- *Calcule des services* : cette fonctionnalité permet aux employés et aux gestionnaires de suivre facilement le nombre de services effectués entre deux dates.

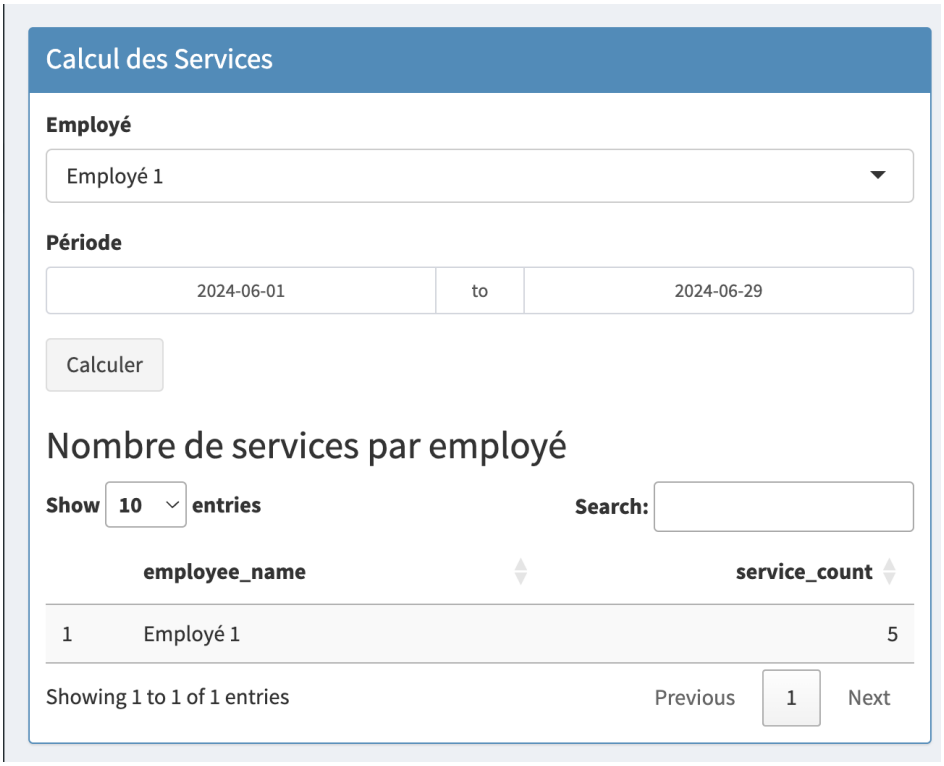

Figure 30: Calcule des services

Code permettant d'executer cette fonctionnalité :

```
# Calculer le nombre de services
observeEvent(input$calc_services, {
  output$services_count <- renderDT({
    dbGetQuery(conn, "
      SELECT employee_name, COUNT(*) AS service_count
      FROM services
      WHERE employee_name = ? AND date BETWEEN ? AND ?
      GROUP BY employee_name",
               params = list(input$calc_employee, as.character(input$date_range[1]), as.character(inpu
  })
})
```

- *Visualisation des Services dans un Calendrier* : l'application offre une visualisation des services planifiés dans un calendrier interactif, permettant aux utilisateurs de naviguer facilement entre les jours, les semaines et les mois.

3.6 Discussions et Suggestions

- Un résumé des principaux résultats obtenus à partir de l'analyse effectuée :
  - La majorité des clients (plus de 80%) réalisent une seule réservation, mais une proportion notable effectue plusieurs réservations, suggérant un potentiel pour des programmes de fidélisation.
  - L'engagement client via les canaux de communication tels que les e-mails et les SMS est plutôt faible, mais non négligeable, indiquant une opportunité d'amélioration de l'interaction avec la clientèle.
  - La plupart des clients n'ont pas soumis d'avis, soulignant l'importance d'encourager les clients satisfaits à laisser des avis positifs pour influencer la réputation en ligne du restaurant.
  - L'analyse bivariée révèle des associations significatives entre le genre des clients et leur comportement vis-à-vis du restaurant, ainsi qu'entre l'opt-in pour les e-mails et les SMS.
  - Les analyses des réservations fournissent des informations précieuses sur les tendances de fréquentation, les variations de couverture selon les jours et les services, ainsi que la saisonnalité des réservations.
  - Les modèles de prévision des réservations permettent de prédire les réservations futures, offrant un outil précieux pour la gestion opérationnelle.
- Suggestions pour l'Amélioration
  - Optimiser l'engagement client en développant des stratégies de marketing ciblées et en mettant en place des programmes de fidélité.
  - Améliorer la gestion des avis en mettant en œuvre un système de suivi en temps réel et en encourageant activement les clients satisfaits à laisser des avis positifs.
  - Optimiser les services opérationnels en utilisant les modèles de prévision des réservations pour ajuster les horaires du personnel et les approvisionnements en fonction de la demande prévue.
  - Améliorer la base de donné des avis en utilisant un format similaire au suivant :

    Table des Avis :

* ID Avis (Clé primaire) : Identifiant unique de l'avis.
* ID Client (Clé étrangère) : Identifiant du client associé à cet avis (lié à la table des clients).
* Note Dessert : Note attribuée au dessert (sur 10).
* Note Plat : Note attribuée au plat entré (sur 10).
* Note Plat : Note attribuée au plat principal (sur 10).
* Note Boisson : Note attribuée à la boisson (sur 10).
* Note Service : Note attribuée au service (sur 10).
* Commentaire : Commentaire laissé par le client.

4 Problèmes rencontrés solutions apportées et bilan des compétences lors du stage

4.1 Problèmes rencontrés

Durant ce stage, j'ai dû faire face à plusieurs défis en tant que seul statisticien au sein de l'équipe.

- Tout d'abord, certaines missions impliquaient des notions que je n'avais pas encore étudiées en cours, ce qui représentait un véritable défi. Ne pouvant partager mes problématiques statistiques avec d'autres spécialistes, je devais souvent me débrouiller seul pour résoudre les questions complexes.

- De plus, une barrière de communication s'est installée avec le reste de l'équipe, qui ne possédait que peu de connaissances en statistiques. Il était parfois difficile d'expliquer des concepts techniques sans les perdre.

- Ensuite, les limitations des données disponibles, qui ne permettaient pas d'analyser en profondeur certains aspects comme la satisfaction client, en raison du non-stockage de certaines informations lors de l'envoi des avis.

- Aussi, j'étais sous une certaine pression pour obtenir des résultats rapides et précis. Cette attente pouvait parfois être stressante et mettre la pression sur mes épaules.

- Enfin, j'ai dû faire face à un apprentissage intensif constant, jonglant entre les concepts théoriques et leur application pratique en entreprise. Cette transition n'était pas toujours évidente.

4.2 Solutions apportées

Pour surmonter ces obstacles, voici les principales solutions que j'ai mises en œuvre :

- J'ai élaboré un planning détaillé, une sorte de feuille de route, pour m'aider à respecter les délais. Ce planning m'a permis d'organiser mon temps de manière plus efficace, de prioriser mes tâches et de suivre mes progrès. Grâce à cette approche structurée, j'ai pu mieux gérer la pression et rester concentré sur mes objectifs.

- Pour pallier les limitations des données, j'ai proposé à l'entreprise de mettre en place un modèle relationnel pour la base de données. Cette solution permettrait de structurer les données de manière plus efficace et de faciliter l'accès aux informations nécessaires pour les analyses statistiques.

- Afin de surmonter la barrière de communication avec le reste de l'équipe, je répétais souvent les mêmes explications en utilisant le moins possible de termes techniques.

- Pour les nouvelles notions que je devais acquérir, je me suis tourné vers des ressources en ligne comme des vidéos YouTube expliquant les concepts avec des exemples pratiques, ainsi que des cours de pratique. Cela m'a permis d'entrer rapidement dans le vif du sujet, sans perdre de temps avec trop de théorie.

4.3 Bilan

Après ces deux mois de stage, je peux faire un point sur toutes les compétences aussi bien techniques que relationnelles et organisationnelles que j'ai eu la chance de développer au sein de L'atelier de Francisco. Le cadre et l'ambiance de travail étaient idéaux pour grandir dans ce métier et dans le secteur de la Data Science.

- **Compétences techniques** :

Data Science : J'ai pu mettre en œuvre plusieurs aspects du métier de Data Scientist, de l'extraction et la récupération de données, au nettoyage, en passant par la mise en place d'algorithmes de machine learning, jusqu'à la Data Visualisation.

Python : j'ai pu consolider et approfondir mes connaissances en Python, le langage de programmation incontournable pour un data scientist.

R : de plus, j'ai renforcé mes compétences en R, un autre langage essentiel pour la data science.

SQL : j'ai également renforcé mes compétences en sql.

- **Compétences relationnelles et organisationnelles**

En effet, étant la seule personne spécialisée en data science au sein d'une équipe de 15 personnes travaillant dans la restauration, dont la plupart ne comprenaient pas mon travail au départ, j'ai dû faire preuve de pédagogie et de capacités de communication pour expliquer clairement mes analyses et résultats. J'ai également développé mon esprit d'équipe, mon adaptabilité et ma capacité à travailler dans un environnement différent du mien. De plus, la nécessité de simplifier et clarifier les informations pour des non-experts a renforcé mes compétences organisationnelles, comme la gestion du temps, la priorisation des tâches et le respect des délais. J'ai également appris à être plus objective et critique dans l'évaluation des données. Enfin, le cadre et l'ambiance de travail idéaux à *L'atelier de Francisco* m'ont permis de grandir dans ce métier passionnant de la data science, tout en contribuant au transfert de connaissances au sein de l'équipe.

Conclusion

Ce stage de deux mois à *L'atelier de Francisco* fut une expérience des plus enrichissantes. J'ai pu mettre en pratique mes connaissances théoriques en data science et développer de nouvelles compétences techniques à travers l'utilisation de librairies Python et R dédiées à l'analyse de données. Les différentes missions qui m'ont été confiées, du nettoyage et de la fusion des bases de données à l'analyse de la satisfaction client en passant par le développement de modèles prédictifs, m'ont permis d'appliquer l'ensemble du processus de la data science. Cette immersion dans un environnement professionnel réel a renforcé mes capacités d'analyse, de résolution de problèmes et de gestion de projet. Je tiens à remercier chaleureusement toute l'équipe de L'atelier de Francisco pour leur accueil, leur confiance et les opportunités d'apprentissage qu'ils m'ont offertes. Cette expérience restera une étape marquante dans mon parcours professionnel et personnel. Elle a confirmé mon intérêt pour le domaine passionnant de la data science et m'a donné l'envie de continuer à approfondir mes compétences.

Bibliographie

Sources en ligne

- Le Coin Stat “https://www.youtube.com/c/LeCoinStat”
- DataScientest “https://datascientest.com/”
- Analysez et modélisez des séries temporelles sur OpenClassrooms “https://openclassrooms.com/fr/courses/4525371-analysez-et-modelisez-des-series-temporelles”
- Nettoyez et analysez votre jeu de données sur OpenClassrooms “https://openclassrooms.com/fr/courses/7410486-nettoyez-et-analysez-votre-jeu-de-donnees”
- Analysez et nettoyez vos données avec R sur OpenClassrooms “https://openclassrooms.com/fr/courses/8282361-analysez-et-nettoyez-vos-donnees-avec-r/8387635-analysez-les-besoins”
- Wikipedia “https://fr.wikipedia.org/”
- Kaggle “https://www.kaggle.com/”

Documents

- Aragon, Yves. 2016. *Série Temporelle Avec R.*
- Grus, Joel. *Data Science par la Pratique, Fondamentaux avec Python.*

Annexe

Annexe code R

- **Code de la partie analyse des descriptives des réverationss**

```r
# Les bibliothèques nécessaires
library(dplyr)
library(readxl)
library(ggplot2)
library(plotly)
library(lubridate)

# Importer les données
df <- read_excel("reservation.xlsx")

# Afficher les premières lignes de la base de données pour vérifier
head(df)

# Changer les noms des variables
colnames(df) <- c("jour", "service", "couverts", "taux")

# Afficher la structure du dataframe pour voir les types de données
str(df)

# Conversion des données
df$jour <- as.Date(df$jour)
df$service <- as.factor(df$service)
df$couverts <- as.numeric(df$couverts)
df$taux <- as.numeric(df$taux)

# Statistique descriptive

# Résumé statistique
summary(df)

# Analyse des services

# Afficher un résumé statistique par service
summary_by_service <- df %>%
  group_by(service) %>%
  summarise(
    Moyenne_Couverts = mean(couverts),
    Moyenne_Taux_de_remplissage = mean(taux)
  )
print(summary_by_service)

# Tracer un graphique de répartition des taux de remplissage par service
ggplot(df, aes(x = service, y = taux)) +
  geom_bar(stat = "identity") +
  labs(title = "Répartition des taux de remplissage par service", x = "Service", y = "Taux de remplissa

# Tracer un graphique de répartition des couverts par service
p2 <- ggplot(df, aes(x = service, y = couverts)) +
```

```r
  geom_bar(stat = "identity") +
  labs(title = "Répartition des couverts par service", x = "Service", y = "Couverts")
ggplotly(p2)

# Analyse des réservations par jour de la semaine

# Regrouper les données par jour et calculer la somme des autres variables
summary_by_week <- df %>%
  mutate(jour_week = wday(jour, label = TRUE, abbr = FALSE)) %>%
  group_by(jour_week, service) %>%
  summarise(
    couverts_moyen = mean(couverts, na.rm = TRUE),
    taux_remplissage_moyen = mean(taux, na.rm = TRUE)
  )

# Afficher les premières lignes du résumé
summary(summary_by_week)
print(summary_by_week)

p0 <- ggplot(summary_by_week, aes(x = jour_week, y = couverts_moyen, fill = service)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Nombre moyen de couverts par jour de la semaine", x = "Jour", y = "Couverts")
ggplotly(p0)

# Analyse des réservations par mois

summary_by_month <- df %>%
  group_by(mois = floor_date(jour, "month"), service) %>%
  summarise(
    couverts = mean(couverts),
    taux_remplissage_moyen = mean(taux)
  )
print(summary_by_month)

# Tracer les couverts par mois
ggplot(summary_by_month, aes(x = mois, y = couverts, fill = service)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Nombre total de couverts par mois", x = "Mois", y = "Couverts")

# Regrouper les données par jour du mois
summary_by_month_day <- df %>%
  mutate(jour_month_day = day(jour)) %>%
  group_by(jour_month_day, service) %>%
  summarise(
    couverts_moyen = mean(couverts, na.rm = TRUE),
    taux_remplissage_moyen = mean(taux, na.rm = TRUE)
  )
# Afficher les premières lignes du résumé quotidien
summary(summary_by_month_day)

p4 <- ggplot(summary_by_month_day, aes(x = jour_month_day, couverts_moyen, fill = service)) +
  geom_bar(stat = 'identity', position = "dodge") +
  labs(title = "Répartition des couverts en des jours du mois", x = "Jour", y = "Total Couverts")
```

```r
# Afficher le graphique
ggplotly(p4)

# Modèle de régression linéaire

df_reg <- df[,-4]
df_reg$service <- as.factor(df_reg$service)

# Tracer un graphique de répartition des couverts par service
p5 <- ggplot(df_reg, aes(x = service, y = couverts)) +
  geom_boxplot() +
  labs(title = "Répartition des couverts par service") +
  theme(plot.title = element_text(hjust = 0.5))

ggplotly(p5)
```

- **Code consernant l'outil de planification**

```r
# Chargement des packages nécessaires
# if (!require(pacman)) install.packages("pacman")
# pacman::p_load(pacman, shiny, dplyr, shinydashboard, DT, lubridate, RSQLite, shinyjs, timevis)
#
# library(shiny)
# library(shinydashboard)
# library(DT)
# library(lubridate)
# library(dplyr)
# library(RSQLite)
# library(shinyjs)
# library(timevis)

# # Créaction d'une base de données SQLite pour stocker les données des services et des employés
# conn <- dbConnect(SQLite(), "bellagio.db")
# dbExecute(conn, "CREATE TABLE IF NOT EXISTS services (id INTEGER PRIMARY KEY, date DATE, content TEXT
# dbExecute(conn, "CREATE TABLE IF NOT EXISTS employees (id INTEGER PRIMARY KEY, first_name TEXT, last_
#
# # Insertion des données de test dans la table "employees"
# dbExecute(conn, "
#   INSERT INTO employees (first_name, last_name) VALUES
#     ('Employé', '1'),
#     ('Employé', '2'),
#     ('Employé', '3'),
#     ('Employé', '4')
# ")
#
# # Insérer des données de test dans la table "services"
# dbExecute(conn, "
#   INSERT INTO services (date, content, type, employee_name) VALUES
#     ('2024-06-01', 'Déjeuner', 'Serveur', 'Employé 1'),
#     ('2024-06-01', 'Dîner', 'Cuisine', 'Employé 2'),
#     ('2024-05-03', 'Déjeuner', 'Serveur', 'Employé 3'),
#     ('2024-05-04', 'Dîner', 'Cuisine', 'Employé 4')
# ")
```

```r
ui <- dashboardPage(
  dashboardHeader(title = "Planification des services"),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Calendrier", tabName = "calendar", icon = icon("calendar")),
      menuItem("Gestion des services", tabName = "manage", icon = icon("edit")),
      #menuItem("Compte Employé", tabName = "employee", icon = icon("user")),
      menuItem("Calcul des Services", tabName = "calculate", icon = icon("calculator"))
    )
  ),
  dashboardBody(
      tabItem(tabName = "manage",
              fluidRow(
                box(title = "Ajouter/Modifier un service", status = "primary", solidHeader = TRUE, widt
                    dateInput("date", "Date du service", value = Sys.Date()),
                    selectInput("service", "Type de service", choices = c("Déjeuner", "Dîner")),
                    selectizeInput("employees", "Employés", choices = NULL, multiple = TRUE),
                    selectInput("type", "Type de poste", choices = c("Serveur", "Cuisine")),
                    actionButton("add_service", "Ajouter"),
                    actionButton("delete_service", "Supprimer")
                ),
                box(title = "Liste des services", status = "primary", solidHeader = TRUE, width = 12,
                    DTOutput("service_table"))
              )),
      tabItem(tabName = "calculate",
              fluidRow(
                box(title = "Calcul des Services", status = "primary", solidHeader = TRUE, width = 12,
                    selectizeInput("calc_employee", "Employé", choices = NULL, multiple = FALSE),
                    dateRangeInput("date_range", "Période", start = Sys.Date() - 30, end = Sys.Date()),
                    actionButton("calc_services", "Calculer"),
                    h3("Nombre de services par employé"),
                    DTOutput("services_count")
                )
              ))
    )
  )


)

server <- function(input, output, session) {
  # Générer la liste des employés pour les sélecteurs
  observe({
    employees <- dbGetQuery(conn, "SELECT first_name || ' ' || last_name AS name FROM employees")
    updateSelectizeInput(session, "employees", choices = employees$name, server = TRUE)
    updateSelectizeInput(session, "calc_employee", choices = employees$name, server = TRUE)
  })

  # Ajouter un service
  observeEvent(input$add_service, {
    for (employee in input$employees) {
      dbExecute(conn, "INSERT INTO services (date, content, type, employee_name) VALUES (?, ?, ?, ?)",
                params = list(as.character(input$date), input$service, input$type, employee))
```

```
  }
  output$service_table <- renderDT(dbGetQuery(conn, "SELECT * FROM services WHERE date >= ?", params =
})

# Supprimer un service
observeEvent(input$delete_service, {
  for (employee in input$employees) {
    dbExecute(conn, "DELETE FROM services WHERE date = ? AND content = ? AND type = ? AND employee_nam
              params = list(as.character(input$date), input$service, input$type, employee))
  }
  output$service_table <- renderDT(dbGetQuery(conn, "SELECT * FROM services WHERE date >= ?", params =
})

# Afficher la liste des services
output$service_table <- renderDT({
  dbGetQuery(conn, "SELECT * FROM services WHERE date >= ?", params = list(Sys.Date()))
})

# Gestion du compte employé
observeEvent(input$login, {
  if (input$username != "" && input$password != "") {
    shinyjs::show("employee_schedule")
    output$employee_table <- renderDT({
      dbGetQuery(conn, "SELECT * FROM services WHERE employee_name = ?", params = list(input$username)
    })

    output$weekly_summary <- renderDT({
      dbGetQuery(conn, "
        SELECT employee_name, strftime('%W', date) AS week, COUNT(*) AS service_count
        FROM services
        WHERE employee_name = ?
        GROUP BY employee_name, week",
                 params = list(input$username))
    })

    output$monthly_summary <- renderDT({
      dbGetQuery(conn, "
        SELECT employee_name, strftime('%m', date) AS month, COUNT(*) AS service_count
        FROM services
        WHERE employee_name = ?
        GROUP BY employee_name, month",
                 params = list(input$username))
    })
  }
})

# Calculer le nombre de services
observeEvent(input$calc_services, {
  output$services_count <- renderDT({
    dbGetQuery(conn, "
      SELECT employee_name, COUNT(*) AS service_count
      FROM services
      WHERE employee_name = ? AND date BETWEEN ? AND ?
```

```
        GROUP BY employee_name",
                params = list(input$calc_employee, as.character(input$date_range[1]), as.character(inpu
    })
  })

}

shinyApp(ui = ui, server = server)
```

Annexe code Python

- **Code consernant le prétraitement et le traitement des données relatives au client et avis.**

```
import pandas as pd

# -*- coding: utf-8 -*-
"""
Created on Fri Apr 12 03:34:17 2024

Auteur : Kuassi Pierre DOVODJI
"""

import numpy as np
import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt
import gender_guesser.detector as gender
from scipy.stats import f_oneway, pearsonr, spearmanr, chi2_contingency
import seaborn as sns
import logging

# Configuration du logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

url = 'data/Fichier_clients_BELLAGIO.csv'
url1 ='data/app-emails.csv'

def load_data(url):
    """
    Charge les données depuis un fichier CSV.

    Arguments :
    url : str, chemin vers le fichier CSV

    Retourne :
    DataFrame pandas contenant les données
    """
    try:
        df = pd.read_csv(url, sep=";")
        logging.info("Données chargées avec succès.")
        return df
    except Exception as e:
        logging.error(f"Erreur lors du chargement des données : {e}")
        return None
```

```python
def clean_data(df):
    """
    Prépare les données en renommant les colonnes, gérant les valeurs manquantes et les doublons.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données

    Retourne :
    DataFrame pandas préparé
    """
    try:
        # Renommer la colonne 'Bookings number'
        df.rename(columns={df.columns[29]: 'Bookings_number'}, inplace=True)
        df.columns = df.columns.str.replace(' ', '_')

        # Affichage des valeurs manquantes
        logging.info(df.isnull().sum())

        # Visualisation des valeurs manquantes
        msno.bar(df)
        plt.show()
        msno.matrix(df)

        # Conserver les variables à utiliser
        variables = ['First_Name', 'Civility', 'Email', 'Phone', 'Guest_Status', 'Email_optin_market',
                     'SMS_optin_market', 'Mail_optin_review', 'SMS_optin_review', 'Has_no_show',
                     'Bookings_number']
        df = df[variables]

        # Traitement des numéros de téléphone manquants
        df0 = df.drop(df.loc[df['Phone'].isnull()].index[0])
        df0 = df0.sort_values(by='Phone')
        df0 = df0.drop_duplicates('Phone')
        df0 = df0.drop(columns=['Bookings_number'])

        # Fusionner les DataFrames
        df1 = df.groupby('Phone')['Bookings_number'].sum().reset_index()
        df = pd.merge(df0, df1, on='Phone', how='left')
        df.rename(columns={'Civility': 'Gender'}, inplace=True)

        # Gestion des valeurs manquantes du sexe
        df = estimate_gender(df)

        return df
    except Exception as e:
        logging.error(f"Erreur lors du prétraitement des données : {e}")
        return None

def estimate_gender(df):
    """
    Estime le sexe des individus à partir de leur prénom.

    Arguments :
```

```
    df : DataFrame pandas, le DataFrame contenant les données

    Retourne :
    DataFrame pandas avec les sexes estimés
    """
    prenoms = pd.Series(df[df['Gender'].isnull()]['First_Name'])
    prenoms = prenoms.str.lower().str.capitalize()

    detector = gender.Detector()
    df.loc[df['Gender'].isnull(), 'Gender'] = [detector.get_gender(row) for row in prenoms]

    homme = ['mr', 'male', 'mostly_male', 'andy']
    femme = ['mrs', 'female', 'mostly_female']
    df['Gender'] = np.where(df['Gender'].isin(homme), 'M',
                            np.where(df['Gender'].isin(femme), 'F', df['Gender']))
    df.loc[df['Gender'] == 'unknown', 'Gender'] = np.nan

    return df

def load_and_merge_review_data(df, url):
    """
    Charge les données d'avis et les fusionne avec les données principales.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données principales
    url : str, chemin vers le fichier CSV des avis

    Retourne :
    DataFrame pandas fusionné
    """
    try:
        df_avis = pd.read_csv(url, sep=';')
        df_avis.columns = df_avis.columns.str.replace(' ', '_')
        df_avis = df_avis[['Adresse_e-mail', 'Messages_envoyés', 'Avis_déposés', 'Telephone']]
        df_avis.columns = ['Email', 'Sent_messages', 'Reviews_submitted', 'Phone']

        liste_mails = df_avis.loc[df_avis['Phone'].isnull(), 'Email']
        liste_phone = df.loc[df['Email'].isin(liste_mails), ['Phone', 'Email']]
        df_avis.loc[df_avis['Email'].isin(liste_mails), 'Phone'] = liste_phone.loc[liste_phone['Email']

        df_client = pd.merge(df, df_avis, on='Phone', how='left')
        df_client = df_client.drop(columns=['Phone', 'Email_x', 'First_Name', 'Email_y'])
        df_client.loc[df_client['Reviews_submitted'].isnull(), 'Reviews_submitted'] = 0
        df_client.loc[df_client['Sent_messages'].isnull(), 'Sent_messages'] = 0

        logging.info("Données fusionnées avec succès.")
        return df_client
    except Exception as e:
        logging.error(f"Erreur lors du chargement ou de la fusion des données d'avis : {e}")
        return None

def analyse_univarie_quantitative(df, label):
    """
```

```
    Réalise une analyse univariée d'une variable quantitative avec un diagramme en bâtons
    représentant les proportions de chaque catégorie, et un résumé des statistiques descriptives.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données
    label : str, le nom de la variable quantitative
    """
    print(df[label].describe())
    proportions = df[label].value_counts(normalize=True).sort_index()

    plt.figure(figsize=(10, 7))
    proportions.plot(kind='bar', color='skyblue')
    plt.xlabel(label)
    plt.ylabel('Proportion')
    plt.title(f'Diagramme en bâtons des proportions de la variable {label}')
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.show()

def analyse_univarie_categorielle(df, label):
    """
    Réalise une analyse univariée d'une variable catégorielle avec un diagramme en barres
    représentant les fréquences relatives de chaque catégorie.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données
    label : str, le nom de la variable catégorielle
    """
    print(df[label].describe())
    comptage_categories = df[label].value_counts()
    freq_relatives = comptage_categories / len(df) * 100

    plt.figure(figsize=(8, 6))
    freq_relatives.plot(kind='bar', color='skyblue')
    plt.xlabel(label)
    plt.ylabel('Fréquence relative (%)')
    plt.title('Fréquences relatives des catégories')
    plt.xticks(rotation=0)
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.show()

def analyse_bivarier_cat(df, label1, label2, test_hypotheses=True):
    """
    Réalise une analyse bivariable entre deux variables catégorielles avec un tableau de contingence
    et un test du Chi-deux pour évaluer l'indépendance entre les variables.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données
    label1 : str, le nom de la première variable catégorielle
    label2 : str, le nom de la deuxième variable catégorielle
    test_hypotheses : bool, indique si le test du Chi-deux doit être effectué (par défaut True)
    """
    tableau_contingence = pd.crosstab(df[label1], df[label2])
    print(tableau_contingence)
```

```
    if test_hypotheses:
        chi2, p_value, dof, expected = chi2_contingency(tableau_contingence)
        print("\nRésultats du test du Chi-deux :")
        print(f"Chi2 : {chi2}")
        print(f"p-value : {p_value}")
        print(f"Degrés de liberté : {dof}")
        if p_value < 0.05:
            print("Les variables sont dépendantes.")
        else:
            print("Les variables sont indépendantes.")

    tableau_contingence.plot(kind='bar', stacked=True, color=sns.color_palette("pastel"))
    plt.xlabel(label1)
    plt.ylabel('Fréquence')
    plt.title(f'Tableau de contingence entre {label1} et {label2}')
    plt.show()

def analyse_bivarier_quant(df, label1, label2, method='pearson'):
    """
    Calcule et affiche la corrélation entre deux variables quantitatives,
    et affiche un nuage de points représentant cette corrélation.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données
    label1 : str, le nom de la première variable quantitative
    label2 : str, le nom de la deuxième variable quantitative
    method : str, méthode de calcul de la corrélation ('pearson' ou 'spearman')
    """
    if method == 'pearson':
        corr, p_value = pearsonr(df[label1], df[label2])
    elif method == 'spearman':
        corr, p_value = spearmanr(df[label1], df[label2])
    else:
        raise ValueError("Méthode de corrélation non valide. Utilisez 'pearson' ou 'spearman'.")

    print(f"Corrélation ({method}) : {corr}")
    print(f"Valeur de p : {p_value}")

    if p_value < 0.05:
        print("La valeur de p   est inférieure à 0.05, donc les variables sont significativement corrél
    else:
        print("La valeur de p est supérieure à 0.05, donc les variables ne sont pas significativement c

    sns.scatterplot(data=df, x=label1, y=label2)
    plt.xlabel(label1)
    plt.ylabel(label2)
    plt.title(f'Nuage de points entre {label1} et {label2} avec corrélation ({method})')
    plt.show()

def analyse_bivarie(df, var_quantitative, var_qualitative):
    """
    Réalise une analyse bivariable entre une variable quantitative et une variable qualitative
    en utilisant une analyse de variance (ANOVA) pour évaluer s'il existe des différences significative
```

```
    entre les groupes définis par la variable qualitative.

    Arguments :
    df : DataFrame pandas, le DataFrame contenant les données
    var_quantitative : str, le nom de la variable quantitative
    var_qualitative : str, le nom de la variable qualitative
    """
    # Affichage de la boîte à moustaches
    plt.figure(figsize=(10, 7))
    sns.boxplot(x=var_qualitative, y=var_quantitative, data=df)
    plt.xlabel(var_qualitative)
    plt.ylabel(var_quantitative)
    plt.title(f'Analyse bivariable entre {var_quantitative} et {var_qualitative}')
    plt.grid(True)
    plt.show()

    # Effectuer le test ANOVA
    groups = df.groupby(var_qualitative)[var_quantitative].apply(list)
    f_statistic, p_value = f_oneway(*groups)
    print("Statistique F :", f_statistic)
    print("Valeur de p :", p_value)

    # Interprétation de la valeur de p
    if p_value < 0.05:
        print("L'ANOVA indique qu'il existe des différences significatives entre les groupes.")
    else:
        print("L'ANOVA n'a pas détecté de différences significatives entre les groupes.")

df = load_data(url)
df = clean_data(df)
df = estimate_gender(df)
df_client = load_and_merge_review_data(df, url1)



 # Appel des fonctions d'analyse univariée pour chaque variable d'intérêt
analyse_univarie_quantitative(df_client, 'Bookings_number')
analyse_univarie_quantitative(df_client, 'Sent_messages')
analyse_univarie_quantitative(df_client, 'Reviews_submitted')
analyse_univarie_categorielle(df_client, 'Gender')
analyse_univarie_categorielle(df_client, 'Guest_Status')
analyse_univarie_categorielle(df_client, 'Email_optin_market')
analyse_univarie_categorielle(df_client, 'SMS_optin_market')
analyse_univarie_categorielle(df_client, 'Mail_optin_review')
analyse_univarie_categorielle(df_client, 'SMS_optin_review')
analyse_univarie_categorielle(df_client, 'Has_no_show')

# Appel des fonctions d'analyse bivariable pour chaque paire de variables d'intérêt
analyse_bivarier_cat(df_client, 'Gender', 'Guest_Status')
analyse_bivarier_cat(df_client, 'Gender', 'Email_optin_market')
analyse_bivarier_cat(df_client, 'Gender', 'Email_optin_market')
analyse_bivarier_cat(df_client, 'Gender', 'SMS_optin_market')
analyse_bivarier_cat(df_client, 'Gender', 'Mail_optin_review')
```

```
analyse_bivarier_cat(df_client, 'Gender', 'SMS_optin_review')
analyse_bivarier_cat(df_client, 'Guest_Status', 'Email_optin_market')
analyse_bivarier_cat(df_client, 'Guest_Status', 'SMS_optin_market')
analyse_bivarier_cat(df_client, 'Guest_Status', 'Mail_optin_review')
analyse_bivarier_cat(df_client, 'Guest_Status', 'SMS_optin_review')
analyse_bivarier_cat(df_client, 'Guest_Status', 'Has_no_show')
analyse_bivarier_cat(df_client, 'Email_optin_market', 'SMS_optin_market')
analyse_bivarier_cat(df_client, 'Email_optin_market', 'Mail_optin_review')
analyse_bivarier_cat(df_client, 'Email_optin_market', 'SMS_optin_review')
analyse_bivarier_cat(df_client, 'Email_optin_market', 'Has_no_show')
analyse_bivarier_cat(df_client, 'SMS_optin_market', 'Mail_optin_review')
analyse_bivarier_cat(df_client, 'SMS_optin_market', 'SMS_optin_review')
analyse_bivarier_cat(df_client, 'SMS_optin_market', 'Has_no_show')
analyse_bivarier_cat(df_client, 'Mail_optin_review', 'SMS_optin_review')
analyse_bivarier_cat(df_client, 'Mail_optin_review', 'Has_no_show')
analyse_bivarier_cat(df_client, 'SMS_optin_review', 'Has_no_show')

# Affichage de la matrice de corrélation pour les variables quantitatives
correlation_matrix = df_client[['Bookings_number', 'Sent_messages', 'Reviews_submitted']].corr()
print("Matrice de corrélation entre les variables quantitatives :")
print(correlation_matrix)

# Analyse de la corrélation entre les variables quantitatives
analyse_bivarier_quant(df_client, 'Bookings_number', 'Sent_messages', method='pearson')
analyse_bivarier_quant(df_client, 'Bookings_number', 'Reviews_submitted', method='pearson')
analyse_bivarier_quant(df_client, 'Sent_messages', 'Reviews_submitted', method='pearson')

# Affichage de la matrice de dispersion pour les paires de variables quantitatives
sns.pairplot(df_client[['Bookings_number', 'Sent_messages', 'Reviews_submitted']])
plt.show()

# Utilisation de la fonction pour analyser la relation entre 'Bookings_number' et 'Gender'
analyse_bivarie(df_client, 'Bookings_number', 'Gender')

# Analyse 1 : Répartition du nombre de réservations par statut de client et opt-in pour les e-mails mar
plt.figure(figsize=(10, 6))
sns.boxplot(x='Guest_Status', y='Bookings_number', hue='Email_optin_market', data=df_client)
plt.title('Répartition du nombre de réservations par statut de client et opt-in pour les e-mails market
plt.xlabel('Statut du client')
plt.ylabel('Nombre de réservations')
plt.legend(title='Opt-in pour les e-mails marketing')
plt.show()

# Analyse 2 : Relation entre le nombre de réservations, le genre du client et l'envoi de SMS marketing
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Bookings_number', y='Gender', hue='SMS_optin_market', data=df_client)
plt.title('Relation entre le nombre de réservations, le genre du client et l\'envoi de SMS marketing')
plt.xlabel('Nombre de réservations')
plt.ylabel('Genre du client')
plt.legend(title='Opt-in pour les SMS marketing')
plt.show()

# Analyse 3 : Comparaison des avis soumis par genre du client et statut du client
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x='Gender', y='Reviews_submitted', hue='Guest_Status', data=df_client, ci=None)
plt.title('Comparaison des avis soumis par genre du client et statut du client')
plt.xlabel('Genre du client')
plt.ylabel('Avis soumis')
plt.legend(title='Statut du client')
plt.show()

# Analyse 4 : Moyenne des avis soumis pour chaque combinaison d'opt-in pour les e-mails marketing et op
pivot_table = df_client.pivot_table(index='Email_optin_market', columns='Mail_optin_review', values='Re
plt.figure(figsize=(10, 6))
sns.heatmap(pivot_table, cmap='viridis', annot=True, fmt='.2f')
plt.title('Moyenne des avis soumis pour chaque combinaison d\'option pour les e-mails marketing et opti
plt.xlabel('Option pour les avis par e-mail')
plt.ylabel('Option pour les e-mails marketing')
plt.show()
```

- **Code consernant la prévision des réservations journaliers**

```
## Importion de packages

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')

from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import statsmodels.api as sm
from scipy.stats import shapiro
import pmdarima as pm

## Données

# Importation des données
df = pd.read_excel("reservation.xlsx", index_col='Jour', parse_dates=True)

# Copie de la base de données
data = df.copy()

# Grouper les données par jour
data = data.groupby(level='Jour').sum()

# Supprimer les colonnes 'service' et 'Taux de remplissage'
data = data.drop(['service', 'Taux de remplissage'], axis=1)
```

```
# Information sur la base de données
data.info()

# Vérification des valeurs manquantes
data.isnull().sum()

# Type des variables
data.dtypes

# Représentation graphique
plt.figure(figsize=(10, 6))
plt.plot(data.index, data['Couverts'])
plt.title('Évolution du nombre de réservations par jour')
plt.xlabel('Jour')
plt.ylabel('Nombre de réservations')
plt.grid(True)
plt.show()

# Graphique ACF et PACF
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))

# Tracer l'ACF
plot_acf(data['Couverts'], lags=30, zero=True, ax=ax1)
ax1.set_title('ACF - Série de réservation')
ax1.set_xlabel('Lag')
ax1.set_ylabel('Corrélation')
ax1.grid(True)
ax1.set_xticks(np.arange(0, 31, 1))

# Tracer le PACF
plot_pacf(data['Couverts'], lags=30, zero=True, ax=ax2)
ax2.set_title('PACF - Série de réservation')
ax2.set_xlabel('Lag')
ax2.set_ylabel('Corrélation partielle')
ax2.grid(True)
ax2.set_xticks(np.arange(0, 31, 1))

plt.tight_layout()
plt.show()

# Convertir l'index en DatetimeIndex
data.index = pd.to_datetime(data.index)

# Analyse de la stationnarité de la variable
result = adfuller(data['Couverts'])

# Formater les résultats dans un tableau
table = [
    ['Valeur de test', result[0]],
    ['P-valeur', result[1]],
    ['Conclusion', 'La série est stationnaire' if result[1] < 0.05 else 'La série est non stationnaire']
]
```

```
# Afficher les résultats sous forme de tableau
print(tabulate(table, headers=['Métrique', 'Valeur'], tablefmt='github'))

# Définir la fréquence quotidienne
data = data.asfreq('D')

# Remplacer les valeurs manquantes (dimanches manquants) par des zéros
data.fillna(0, inplace=True)

# Effectuer la décomposition saisonnière
decomposition = seasonal_decompose(data['Couverts'], model='additive')

# Extraire les composantes de la décomposition
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

# Afficher les composantes de la décomposition
plt.figure(figsize=(12, 8))

plt.subplot(411)
plt.plot(data['Couverts'], label='Série originale')
plt.legend(loc='best')

plt.subplot(412)
plt.plot(trend, label='Tendance')
plt.legend(loc='best')

plt.subplot(413)
plt.plot(seasonal, label='Saisonnalité')
plt.legend(loc='best')

plt.subplot(414)
plt.plot(residual, label='Résidus')
plt.legend(loc='best')

plt.tight_layout()
plt.show()

# Différenciation pour rendre la série stationnaire
differenced = data['Couverts'].diff().dropna()

# Afficher la série différenciée
plt.figure(figsize=(10, 6))
plt.plot(differenced)
plt.title('Série temporelle différenciée (Réservations)')
plt.xlabel('Date')
plt.ylabel('Différence')
plt.grid(True)
plt.show()

# Créer les subplots pour ACF et PACF de la série différenciée
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))
```

```
# Tracer l'ACF
plot_acf(differenced, lags=30, zero=True, ax=ax1)
ax1.set_title('ACF - Série différenciée')
ax1.set_xlabel('Lag')
ax1.set_ylabel('Corrélation')
ax1.grid(True)
ax1.set_xticks(np.arange(0, 31, 1))

# Tracer le PACF
plot_pacf(differenced, lags=30, zero=True, ax=ax2)
ax2.set_title('PACF - Série différenciée')
ax2.set_xlabel('Lag')
ax2.set_ylabel('Corrélation partielle')
ax2.grid(True)
ax2.set_xticks(np.arange(0, 31, 1))

plt.tight_layout()
plt.show()

# Analyse de la stationnarité de la série différenciée
result = adfuller(differenced)

# Formater les résultats dans un tableau
table = [
    ['Valeur de test', result[0]],
    ['P-valeur', result[1]],
    ['Conclusion', 'La série est stationnaire' if result[1] < 0.05 else 'La série est non stationnaire']
]

# Afficher les résultats sous forme de tableau
print(tabulate(table, headers=['Métrique', 'Valeur'], tablefmt='github'))

# Identification de l'ordre p,d,q
p = 2
d = 1
q = (1, 3)

# Séparer les données en ensemble d'entraînement et ensemble de test
train_data = data['Couverts'][:-15]
test_data = data['Couverts'][-15:]

# Estimation du modèle ARIMA(6,1,2)
model = ARIMA(train_data, order=(6,1,2))
model_fit = model.fit()

# Afficher le résumé du modèle
print(model_fit.summary())

# Vérification du modèle
# Calculer les résidus
residuals = model_fit.resid

# Test de Ljung-Box
```

```
ljung_box_results = pd.DataFrame(columns=["Order", "Test_Statistic", "P_Value"])

for i in range(1, 13):
    test_result = sm.stats.acorr_ljungbox(residuals, lags=[i], return_df=True)
    ljung_box_results = pd.concat([ljung_box_results, pd.DataFrame({"Order": [i],
                                                                    "Test_Statistic": test_result["lb_s
                                                                    "P_Value": test_result["lb_pvalue"]
print(ljung_box_results)

# Test de normalité (Shapiro-Wilk)
plt.figure(figsize=(10, 6))
sm.qqplot(residuals, line='s')
plt.title('Q-Q plot of residuals')
plt.show()

shapiro_test = shapiro(residuals)
print(f"Shapiro-Wilk Test: Statistic={shapiro_test.statistic}, P-Value={shapiro_test.pvalue}")

# Tracer les résidus
plt.figure(figsize=(10, 6))
plt.plot(residuals)
plt.title('Residuals')
plt.show()

# Tracer l'ACF des résidus
plt.figure(figsize=(10, 5))
plot_acf(residuals, lags=30, zero=False)
plt.title("Autocorrelogram of Residuals")
plt.show()

# Tracer la PACF des résidus
plt.figure(figsize=(10, 5))
plot_pacf(residuals, lags=30, zero=False)
plt.title("Partial Autocorrelogram of Residuals")
plt.show()

# Prédictions
forecast_steps = len(test_data)
forecast = model_fit.forecast(steps=forecast_steps)

# Afficher les valeurs prédites et réelles
print(forecast)
print(test_data)

# Évaluer la performance
mae = mean_absolute_error(test_data, forecast)
mse = mean_squared_error(test_data, forecast)
rmse = np.sqrt(mse)
r2 = r2_score(test_data, forecast)

performance_table = [
    ['MAE', mae],
    ['MSE', mse],
```

```
    ['RMSE', rmse],
    ['R²', r2]
]

print(tabulate(performance_table, headers=['Métrique', 'Valeur'], tablefmt='github'))

# Comparaison graphique des valeurs réelles et prédites
plt.figure(figsize=(12, 8))
plt.plot(test_data, label='Valeurs réelles', marker='o')
plt.plot(test_data.index, forecast, label='Prédictions', marker='x')
plt.fill_between(test_data.index, forecast - 1.96*forecast.std(), forecast + 1.96*forecast.std(), alpha=
plt.title('Comparaison des valeurs réelles et prédites')
plt.xlabel('Date')
plt.ylabel('Nombre de réservations')
plt.legend()
plt.grid(True)
plt.show()
```