

# RAPPORT DE GROUPE EN SCIENCES DES DONNÉES 2 ET BASES DE DONNÉES

Katia Djellali (22113493), Elise Commandré(22104606), Alvin Ingabire(22111286),  
Cédric Henry(22105033).



Département MIASHS, UFR 6 Informatique, Mathématique et Statistique  
Université Paul Valéry, Montpellier 3

Mai 2023

SOUMIS COMME CONTRIBUTION PARTIELLE  
POUR LE COURS SCIENCE DES DONNÉES 2 ET BASES DE DONNÉES

---

## Déclaration de non plagiat

---

Nous déclarons que ce rapport est le fruit de notre seul travail, à part lorsque cela est indiqué explicitement.

Nous acceptons que la personne évaluant ce rapport puisse, pour les besoins de cette évaluation:

- la reproduire et en fournir une copie à un autre membre de l'université; et/ou,
- en communiquer une copie à un service en ligne de détection de plagiat (qui pourra en retenir une copie pour les besoins d'évaluation future).

Nous certifions que nous avons lu et compris les règles ci-dessus.

En signant cette déclaration, nous acceptons ce qui précède.

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

---

## Table des matières

---

Chapitre 1	Introduction	1
Chapitre 2	Données	2
2.1	Présentation des données . . . . .	2
2.2	Nettoyage des données . . . . .	2
2.3	Descriptif des jeux de données . . . . .	3
2.4	Descriptif des tables . . . . .	3
Chapitre 3	Modélisation des Données	5
3.1	Modélisation conceptuelle . . . . .	5
3.2	Modélisation organisationnelle . . . . .	5
Chapitre 4	Prétraitement	7
Chapitre 5	Requêtes en SQL et en langage naturel	8
Chapitre 6	Modélisation statistique et analyses	10
Chapitre 7	Conclusion et perspectives	15
Annexes		16
	Bibliographie . . . . .	16
	7.1 <b>Diagramme supplémentaire</b> . . . . .	16
	<b>Codes</b> . . . . .	17

---

# CHAPITRE 1

## Introduction

---

Pour le projet, nous avons choisi le sujet suivant: “Où mange-t-on le mieux?”. Nous nous sommes longtemps interrogés sur comment interpréter ce sujet qui pouvait rapidement devenir assez vaste : devions-nous travailler en fonction du type de restauration (restaurants scolaires, restaurants gastronomiques, etc.), c’est-à-dire en les comparant ou devions-nous plus nous concentrer sur les grandes villes et non les petits villages... Nous en sommes arrivés à la conclusion de nous concentrer sur les restaurants dans les grandes villes et de ne prendre en compte que les restaurants ayant des notes (étoiles, avis, notes sanitaires, etc.) . En approfondissant ainsi les recherches, nous nous sommes tournés vers la problématique :

**Où mange-t-on le mieux dans les grandes villes?**

---

## CHAPITRE 2

### Données

---

#### 2.1 Présentation des données

Pour répondre à cette problématique, nous avons décidé d'utiliser le jeu de données fourni par les enseignants issu de [data.gouv.fr](https://www.data.gouv.fr) ainsi qu'un autre jeu de données trouvé en ligne sur le site Kaggle:

- Jeu de données 1 ( 12 colonnes et 34 013 lignes) : “export\_alimconfiance.xls” fourni par nos enseignants suivant le thème choisi. Il répertorie différents établissements situés en France dans lesquels on a procédé à une inspection sanitaire. Source : <https://www.data.gouv.fr/fr/datasets/resultats-des-controles->
- Jeu de données 2 (12 colonnes et 6 781 lignes) : Il répertorie les restaurants étoilés au guide Michelin dans le monde. Source : <https://www.kaggle.com/datasets/ngshiheng/michelin-guide-restaurants-2021>.

#### 2.2 Nettoyage des données

Les données utilisées étaient très volumineuses et avaient des propriétés (colonnes) qui n'avaient aucune pertinence, ce qui nous aurait ralenti au moment de faire des requêtes. Nous avons donc supprimé les colonnes et lignes non pertinentes et sur certaines colonnes et lignes nous avons choisi de filtrer quelques éléments pour ne garder que ceux qui nous intéressent.

- Jeu de données 1: Pour ce jeu de données, on ne va s'intéresser qu'aux restaurants ; on a donc filtré la colonne libellé pour ne garder que les restaurants et on supprimé les colonnes “ods\_type\_activité” et “Agrément” qui ne donnaient aucune information supplémentaire. “Agrément” n'avait pas de valeur pour la plupart des établissements. On a ainsi obtenu un jeu de données avec 11 658 lignes et 7 colonnes comme le nom du restaurant, son adresse, le code postal et sa commune, l'avis de l'évaluation sanitaire, et les coordonnées du restaurant. Les notes d'hygiène étant caractérisées par des adjectifs allant de satisfaisant à très satisfaisant, nous avons modifié la colonne en remplaçant ces adjectifs par des chiffres allant de un à trois afin de faciliter les calculs de nos futures requêtes : très satisfaisant = 3, satisfaisant = 2 et à améliorer = 1.
- Jeu de données 2: On a filtré les données pour ne garder que les restaurants situés en France (en filtrant la colonne « Location » ) et on a supprimé les colonnes ne nous intéressant pas telles que « Facilities and Services » car non pertinentes pour les requêtes. On obtient alors un nouveau jeu de données composé de 10 colonnes et 242 lignes : il est composé du nom du restaurant, de son adresse, du nom de la ville, du prix (échelle de prix), du type de cuisine,

de sa longitude et latitude, de son numéro de téléphone, de son site web et du nombre d'étoiles. L'échelle de prix étant illustré par un nombre de symboles « € », nous avons modifié la colonne en remplaçant le nombre de symbole par les chiffres correspondant (de un à trois) afin de faciliter les calculs de nos futures requêtes.

### 2.3 Descriptif des jeux de données

Nom colonne	Type	Signification
App_libelle_etablissement	Varchar	Nom du restaurant
Adresse_2_UA	Varchar	Adresse du restaurant
Code_postal	Numeric	Code postal
Libelle_commune	Varchar	Nom de la commune
APP_libelle_acti_etabl	Varchar	Activité de l'établissement
Synthese_eval_sanit	Varchar	Avis de l'évaluateur
Geores	Numeric	Coordonnées du restaurant

Table : Jeu de données 1 ( $8 \times 3$ )

Nom colonne	Type	Signification
Name	Varchar	Nom du restaurant
Adress	Varchar	Adresse du restaurant
Location	Varchar	Nom de la ville
Price	Varchar	Echelle de prix
Cuisine	Varchar	Type de cuisine
Longitude	Numeric	Longitude du restaurant
Latitude	Numeric	Latitude du restaurant
PhoneNumber	Integer	Numéro de téléphone
WebsiteUrl	Varchar	Site web du restaurant
Award	Integer	Nombre d'étoile

Table : Jeu de données 2 ( $11 \times 3$ )

### 2.4 Descriptif des tables

Grâce à ces données, nous avons pu créer différentes tables afin de modéliser au mieux la situation :

- **Table Restaurant\_classique** : dans cette table, nous avons l'id\_resto qui est en integer unique pour identifier les restaurants et qui est la clé primaire. Nous avons également un nom et une adresse qui sont en varchar ainsi que la latitude et la longitude qui sont en decimal.
- **Table Note\_hygiene** : dans cette table, nous avons l'id\_note qui est en integer unique pour identifier les notes d'hygiène et qui est la clé primaire. Nous avons également une note qui est en integer.

- **Table Ville** : dans cette table, nous avons l'id\_ville qui est en integer unique pour identifier les villes et qui est la clé primaire. Nous avons également un nom qui est en varchar et un code postal qui est en integer.
- **Table Restaurant\_etoile** : dans cette table, nous avons l'id\_resto qui est en integer unique pour identifier les restaurants étoilés et qui est la clé primaire. Nous avons également un nom, une adresse et un sit web (url) qui sont en varchar ainsi que la latitude et la longitude qui sont en decimal et un numéro de téléphone qui est en integer.
- **Table Cuisine** : dans cette table nous avons l'id\_cuisine qui est en integer unique pour identifier les types de cuisine des restaurants étoilés et qui est la clé primaire. Nous avons aussi type\_cuisine, correspondant au type de cuisine qui est en varchar.
- **Table Note** : dans cette table, nous avons l'id\_note qui est en integer unique pour identifier les notes et qui est la clé primaire. Nous avons également nombre\_etoile qui correspond au nombre d'étoile d'un restaurant et qui est en integer.

On peut voir sur la Figure 2.1 ci-dessous un aperçu de la structure d'une table de la base de données :

Table ▲	Lignes ⓘ	Type	Interclassement	Taille	Perte
<input type="checkbox"/> appartient	~273	InnoDB	utf8_general_ci	48 Kio	-
<input type="checkbox"/> cuisine	~21	InnoDB	utf8_general_ci	16 Kio	-
<input type="checkbox"/> note	~241	InnoDB	utf8_general_ci	32 Kio	-
<input type="checkbox"/> note_hygiene	~519	InnoDB	utf8_general_ci	64 Kio	-
<input type="checkbox"/> restaurant_classique	~519	InnoDB	utf8_general_ci	80 Kio	-
<input type="checkbox"/> restaurant_etoile	~241	InnoDB	utf8_general_ci	80 Kio	-
<input type="checkbox"/> ville	~116	InnoDB	utf8_general_ci	16 Kio	-
<b>7 tables</b>	<b>1 930</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>336 Kio</b>	<b>0 0</b>

Figure 2.1: Structure de la base de données.

---

## CHAPITRE 3

### Modélisation des Données

---

#### 3.1 Modélisation conceptuelle

Avec nos jeux de données, nous avons construit un Modèle Conceptuel des Données (MCD) avec l'outil en ligne mocodo utilisé lors des Tds comme représenté sur la Figure 3.1 ci-dessous:

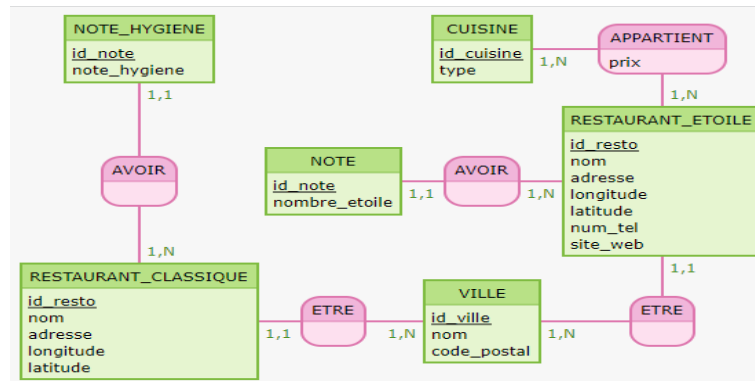


Figure 3.1: Modèle Conceptuel des données.

#### 3.2 Modélisation organisationnelle

A partir de ce modèle conceptuel, nous avons créé la version manuscrite de notre Modèle Organisationnel de Données (MOD) :

RESTAURANT\_CLASSIQUE(id-resto, nom, adresse, longitude, latitude, id\_ville)  
RESTAURANT\_ETOILE (id-resto-etoile, nom, adresse, longitude, latitude, téléphone, id\_ville) VILLE (id\_ville, nom, code postal) CUISINE (id-cuisine, type\_cuisine)  
NOTE (id-note, nombre\_etoiles, id\_resto) NOTE\_HYGIENE(id-note, note\_hygiene, id\_resto) APPARTIENT (id-resto-etoile, id-cuisine, prix)

Nous avons ensuite créé ce modèle organisationnel sur notre base de données grâce à l'outil Concepteur de WAMP - PhpMyAdmin afin de mieux visualiser les liens entre les tables (clés étrangères) comme on peut le voir sur la Figure 3.2 ci-dessous:



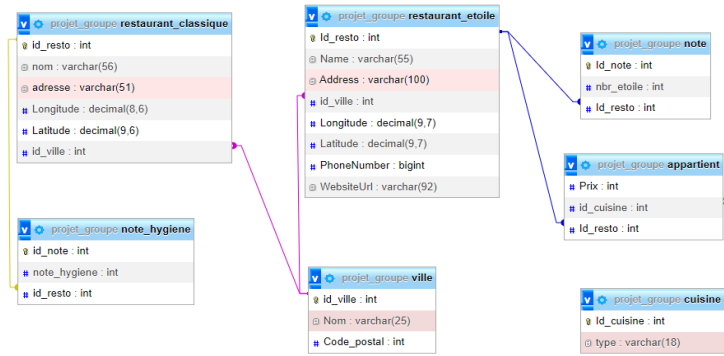


Figure 3.2: Modèle Organisationnel des données.

---

## CHAPITRE 4

### Prétraitement

---

Afin d'importer nos données dans notre base, nous avons séparé les tables dans des fichiers csv différents, en faisant correspondre les données entre elles grâce à nos clés primaires et clés étrangères. Pour ces dernières, nous avons dû ajouter des colonnes dans les tables afin de pouvoir créer nos clés étrangères dans la base de données et donc les liens entre les tables.

Avant de réaliser l'import des données, nous avons remarqué que notre table restaurant était trop volumineuse, nous avons donc dû supprimer des données. Pour cela, nous avons gardé les restaurants dont les villes étaient en commun avec les villes des restaurants étoilés. Nous sommes donc passé de 11 658 lignes à 520 lignes.

Après l'import dans SQL, nous avons modifié le type de certaines de nos variables afin qu'elles aient le bon type et nous avons ensuite créé les clés primaires et étrangères qui nous ont permis de générer le MOD situé plus haut.

---

## CHAPITRE 5

### Requêtes en SQL et en langage naturel

---

Afin de pouvoir répondre à notre problématique, nous avons effectué plusieurs requêtes :

- **Restaurants étoilés qui ne sont pas à Paris (162 résultats) :**

```
SELECT restaurant_etoile.Name, ville.Nom
FROM restaurant_etoile, ville
WHERE ville.id_ville=restaurant_etoile.id_ville
AND ville.Nom!="Paris"
```

- **Moyenne des notes hygiène par ville (39 résultats) :**

```
SELECT AVG(note_hygiene.note_hygiene), ville.Nom
FROM note_hygiene, restaurant_classique, ville
WHERE ville.id_ville=restaurant_classique.id_ville
AND note_hygiene.id_resto=restaurant_classique.id_resto
GROUP BY ville.id_ville
HAVING COUNT(restaurant_classique.nom)>1
```

- **Le nom et la note moyenne de tous les restaurant étoilés (241 résultats) :**

```
SELECT R.Name, AVG(N.nbr_etoile) AS "note_moyenne"
FROM restaurant_etoile AS R
LEFT JOIN note AS N ON R.Id_resto = N.Id_resto
GROUP BY R.Id_resto
```

- **Affiche les restaurants étoilés de 3 étoiles ainsi que les restaurants classiques ayant une note hygiène de 3 (149 résultats) :**

```
SELECT restaurant_etoile.Name
FROM restaurant_etoile, note
WHERE restaurant_etoile.Id_resto=note.Id_resto
AND note.nbr_etoile=3
UNION
SELECT restaurant_classique.nom
FROM restaurant_classique, note_hygiene
WHERE restaurant_classique.id_resto=note_hygiene.id_resto
AND note_hygiene.note_hygiene=3
```

- Nombre d'étoile par ville, rangé par ordre décroissant (21 résultats)  
:

```
SELECT sum(note.nbr_etoile) as "total etoiles par ville", ville.Nom
FROM note, restaurant_etoile, ville
WHERE ville.id_ville=restaurant_etoile.id_ville
AND note.id_resto=restaurant_etoile.id_resto
GROUP BY ville.id_ville
HAVING COUNT(restaurant_etoile.name)>1
order by sum(note.nbr_etoile) DESC
```

---

## CHAPITRE 6

### Modélisation statistique et analyses

---

Diagramme des notes d'hygiènes par villes, Figure 6.1 :

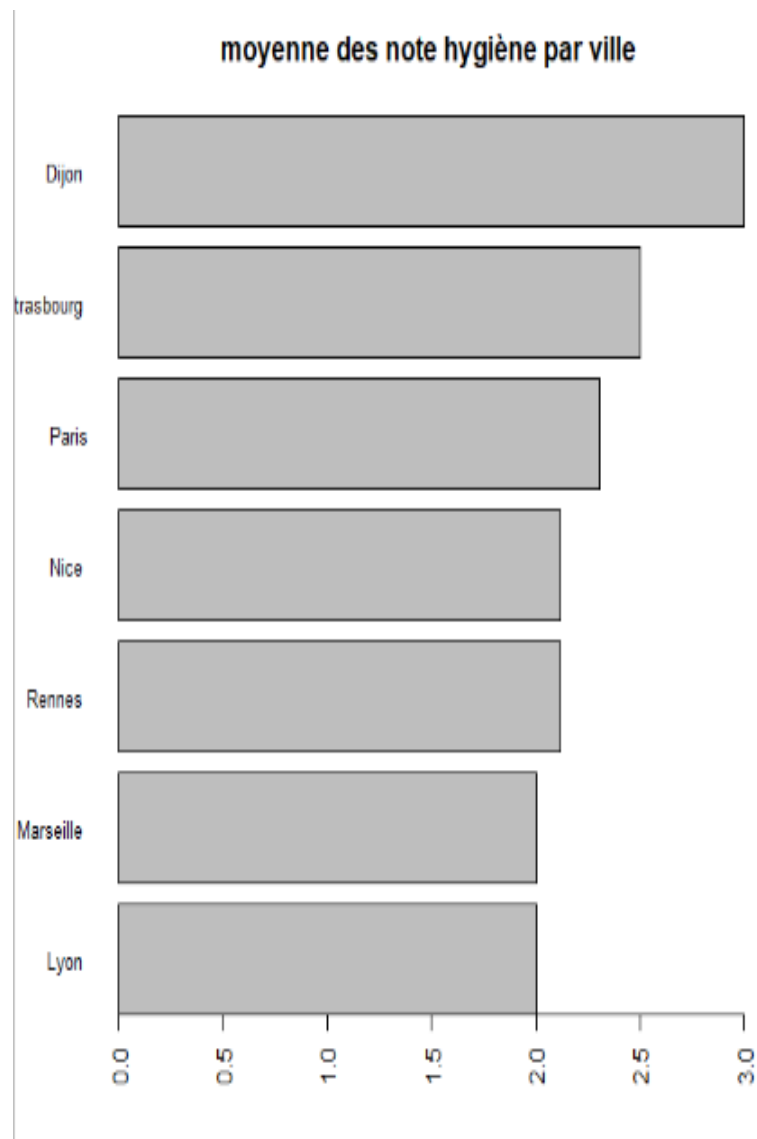


Figure 6.1: Première modélisation

Ce graphique en barre permet de comparer les notes d'hygiène entre sept villes de France : Paris, Marseille, Dijon, Lyon, Nice, Rennes et Strasbourg qui sont

affichées sur l'axe des y, Les notes d'hygiène sont évaluées de 1 à 3 et présentées sur l'axe des x. Le graphique montre que la ville de Dijon à la note d'hygiène la plus élevée suivie de Strasbourg et Paris. Nice et Rennes ont la même note d'hygiène. Tandis que Marseille et Lyon ont les notes d'hygiène les plus basses. Cette analyse indique que la moyenne des notes hygiènes varient entre les villes, et que certaines villes obtiennent de meilleurs résultats que d'autres en termes de qualité d'hygiène dans les restaurants. Ces résultats peuvent être utilisés pour identifier les villes qui ont besoin d'améliorer leurs pratiques d'hygiène et pour encourager les restaurants à respecter des normes élevées en matière d'hygiène alimentaire.

Diagramme du nombre de restaurants étoilés par ville, Figure 6.2 :

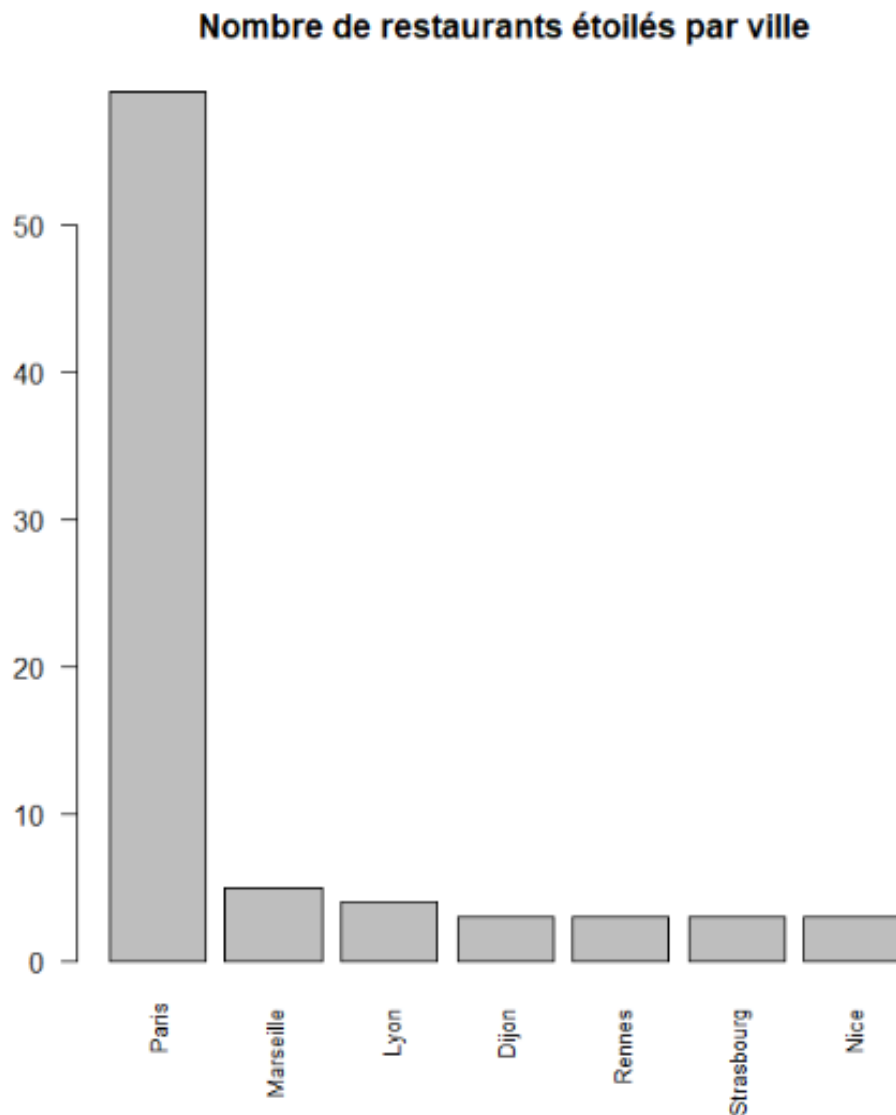


Figure 6.2: Deuxième modélisation

Ce graphique montre le nombre de restaurants étoilés dans les sept villes françaises. Le nombre de restaurants est affiché sur l'axe des y, tandis que les villes sont indiquées sur l'axe des x. On observe que Paris est la ville qui possède le plus grand nombre de restaurants étoilés. Marseille arrive en deuxième position suivie de Lyon et Dijon. Les villes restantes (Rennes, Strasbourg, Nice) ont un nombre de restaurants bien inférieur et presque équivalent. Il est important de noter que ces données ne prennent en compte que les restaurants qui ont reçu au moins une étoile. Les restaurants qui n'ont pas d'étoile ne sont pas pris en compte dans ce graphique. En conclusion, ce graphique montre clairement que Paris est la ville française qui possède le plus grand nombre de restaurants étoilés. Marseille et Lyon sont également des villes importantes pour la gastronomie française mais leur nombre de restaurants étoilés est bien inférieur à celui de Paris.

Diagramme des prix moyens des restaurants étoilés par villes, Figure 6.3:

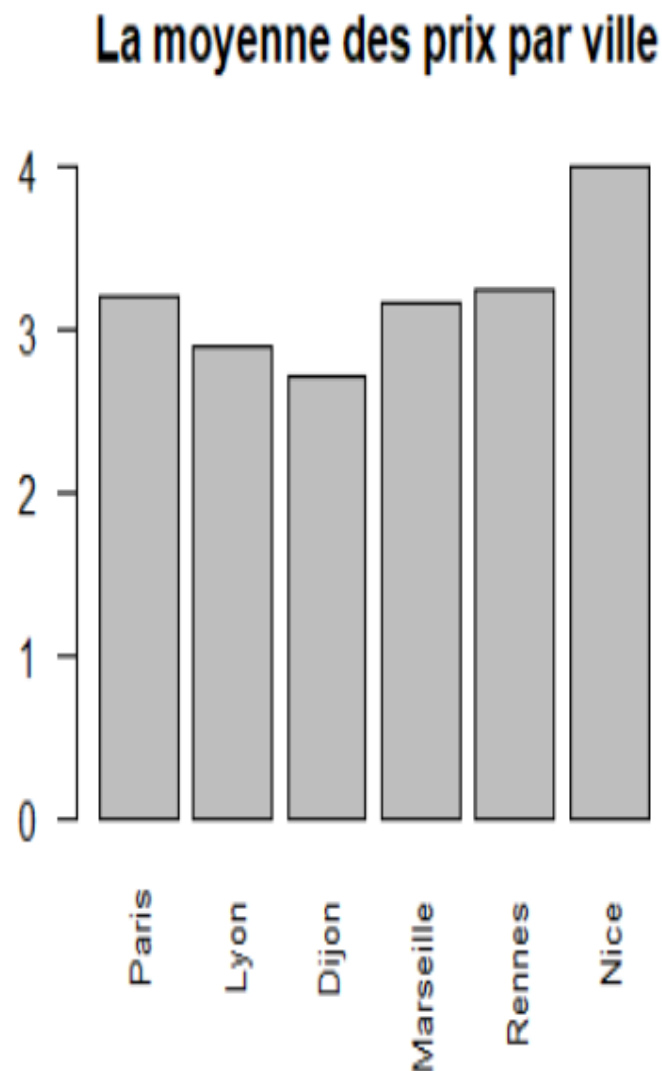


Figure 6.3: Troisième modélisation

Le graphe représente la moyenne des prix de restaurants de six villes différentes. Les barres représentent la moyenne des prix de chaque ville, avec la plus haute barre indiquant la ville avec la moyenne des prix la plus élevée. L'analyse des résultats montre que Nice a la moyenne de prix la plus élevée, suivie de Rennes et Paris puis de Marseille, Lyon et Dijon. Cela signifie que les restaurants de la ville de Nice sont en moyenne plus chers que ceux des autres villes. Cette information peut être utilisée pour diverses analyses. Par exemple, pour une personne qui planifie un voyage gastronomique et qui a un budget limité, il pourrait être judicieux de choisir une ville autre que Paris ou Nice. Les villes ayant une moyenne de prix plus basse (Marseille, Lyon et Dijon) pourraient être de bonnes options pour cette personne, car elle pourrait profiter d'une expérience culinaire similaire à un prix plus abordable. De même, pour les entreprises du secteur de la restauration, les villes ayant une moyenne de prix plus basse pourraient être plus avantageuse pour minimiser les coûts d'exploitation.

Carte de France représentant les villes ayant des restaurant étoilés, Figure 6.4 :

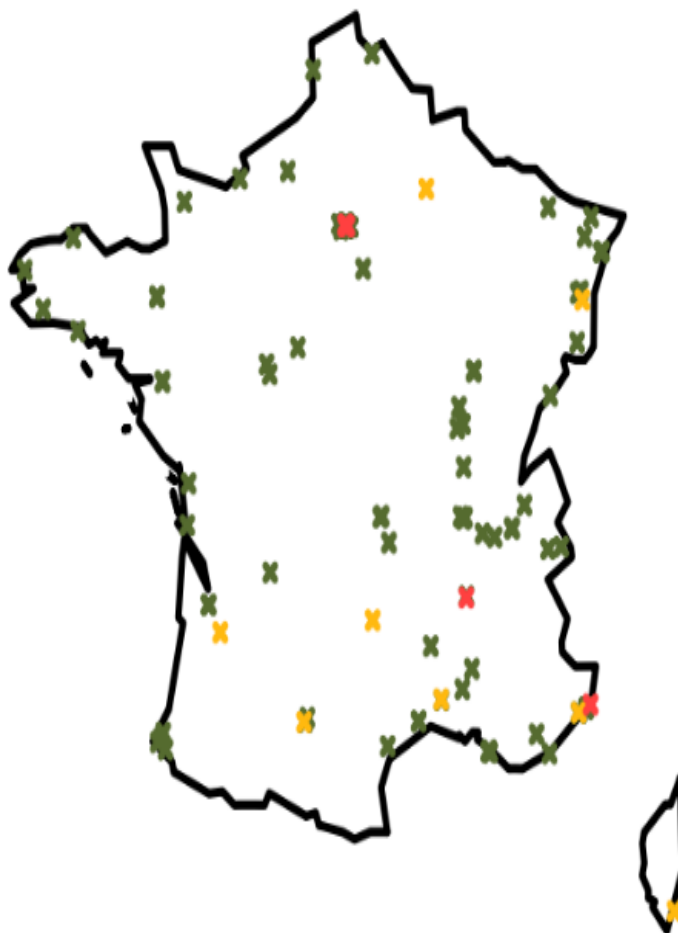


Figure 6.4: Cinquième modélisation

La carte montre la répartition spatiale des restaurants étoilés en France, en distinguant ceux qui ont une étoile, deux étoiles, et trois étoiles. Les points vert



foncé représentent les restaurants avec une étoile, les points jaunes représentent ceux avec deux étoiles et les points marrons représentent les restaurants avec trois étoiles. On peut observer que les restaurants étoilés sont concentrés dans certaines régions notamment en Île-de-France, dans le Sud-Est de la France et dans la région de la Nouvelle Aquitaine. On observe également que Paris est la ville avec le plus grand nombre de restaurants étoilés en France, avec une concentration élevée de restaurants avec trois étoiles. En revanche, d'autres régions de la France semblent moins bien représentées en termes de restaurants étoilés, comme la région du Grand-Est ou de la Normandie. Cela pourrait être dû à des facteurs tels que le manque de tradition gastronomique dans ces régions ou à une concurrence plus forte pour des restaurants gastronomiques dans les régions moins touristiques. On peut également noter que la densité de restaurants étoilés diminue à mesure que l'on s'éloigne des grandes villes ou des régions les plus touristiques. Enfin, la différence de densité entre les restaurants étoilés avec une, deux, ou trois étoiles peut être expliquée par la difficulté croissante d'obtenir une étoile supplémentaire, ce qui peut limiter le nombre de restaurants étoilés avec deux ou trois étoiles par rapport aux restaurants étoilés avec une étoile.

---

## CHAPITRE 7

### Conclusion et perspectives

---

Après avoir étudié et analysé les graphiques des parties “Modélisations statistiques et analyses” et “Annexes”, on peut conclure que les grandes villes où l’on mange le mieux en tenant compte uniquement des restaurants étoilés et des notes d’hygiène de restaurants non-étoilés, sont Paris, en tête du classement, avec Marseille, Lyon et Nice à la suite. En effet, ces trois dernières sont souvent classées dans le top trois des différents graphiques.

Cette conclusion ainsi que les analyses précédente sont valables dans le cadre de notre problématique ciblée. En effet, le sujet “Où mange-t-on le mieux” est vaste et peut-être interprété différemment selon les besoins. Ainsi, d’autres jeux de données et modélisations seront plus adaptés à d’autres besoins, par exemple si l’on se concentre sur le type de restauration (restauration collective, restaurants de ville, etc.).

**Problèmes rencontrés:**

---

## Annexes

---

### Bibliographie

#### 7.1 Diagramme supplémentaire

Diagramme du nombre d'étoiles par ville , Figure 7.1 :

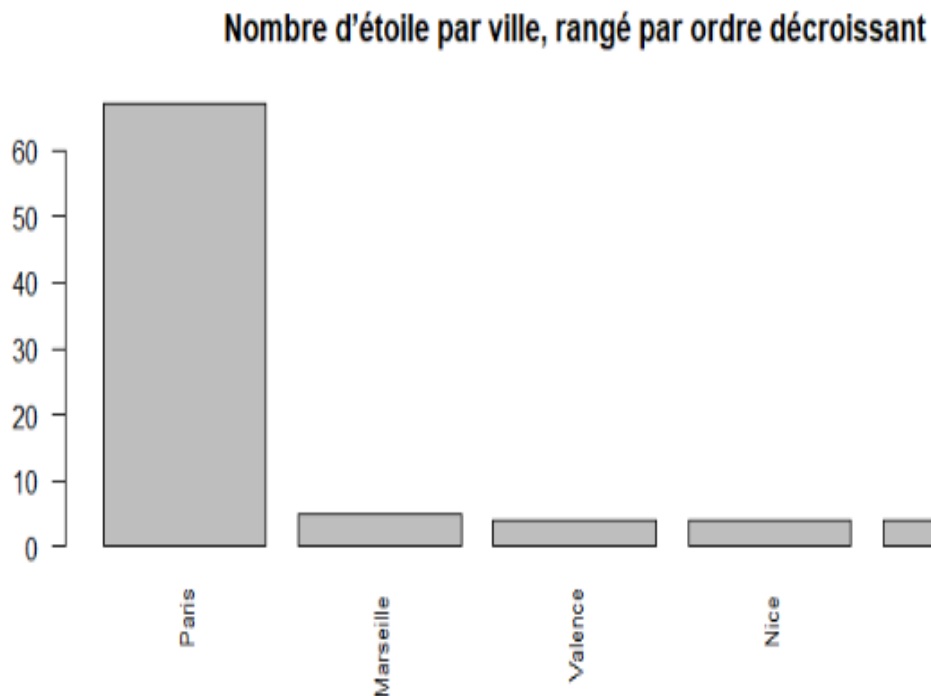


Figure 7.1: Quatrième modélisation

Ce graphique représente le nombre d'étoiles attribuées aux restaurants de chaque ville. Les cinq villes ayant le plus grand nombre d'étoiles sont affichées sur le graphique, classées par ordre décroissant. On peut voir que Paris a le plus grand nombre d'étoiles attribuées, suivi de Marseille, Valence, Nice et Lyon qui ont toutes un nombre d'étoiles assez similaire. Cela suggère que ces villes ont un grand nombre de restaurants renommés, qui ont reçu des étoiles Michelin en récompense de leur cuisine de haute qualité. En général, ce graphique peut être utilisé pour aider à identifier les tendances en matière de gastronomie et les centres de restauration renommés dans les différentes villes.

## Codes

Code de la carte de France (Figure 6.5) :

```
install.packages("RMySQL")

install.packages("DBI")
library(DBI)

bd <- DBI::dbConnect(RMySQL::MySQL(),

                     host = "9ha.h.files.io", port = 3307,
                     username = "ProjetBD_believedbe",
                     password = "83c50259163a65e8ca715f92e1bbb9c15d0ca7a9",
                     dbname = "ProjetBD_believedbe")

dbListTables(bd)

requete3 <- "SELECT restaurant_etoile.latitude, restaurant_etoile.longitude
FROM restaurant_etoile, note
where note.id_resto=restaurant_etoile.id_resto
and note.nbr_etoile=3;
"

requete2 <- "SELECT restaurant_etoile.latitude, restaurant_etoile.longitude
FROM restaurant_etoile, note
where note.id_resto=restaurant_etoile.id_resto
and note.nbr_etoile=2;
"

requete <- "SELECT restaurant_etoile.latitude, restaurant_etoile.longitude
FROM restaurant_etoile, note
where note.id_resto=restaurant_etoile.id_resto
and note.nbr_etoile=1;
"

resultatetoile3 <- dbSendQuery(bd, requete3)

resetoile3<-fetch(resultatetoile3)

resultatetoile2 <- dbSendQuery(bd, requete2)

resetoile2<-fetch(resultatetoile2)

resultatetoile <- dbSendQuery(bd, requete)

resetoile<-fetch(resultatetoile)

dbDisconnect(bd)

install.packages("maps")
install.packages("mapdata")

library(maps)
library(mapdata)
map("worldHires", "france", xlim=c(-5,10), ylim=c(35,55))
etoile1lat<-subset(resetoile, select = latitude)
etoile1long<-subset(resetoile, select = longitude)
etoile2lat<-subset(resetoile2, select = latitude)
etoile2long<-subset(resetoile2, select = longitude)
etoile3lat<-subset(resetoile3, select = latitude)
etoile3long<-subset(resetoile3, select = longitude)

points(unlist(etoile1long),unlist(etoile1lat),pch=4,cex=0.2, col="darkolivegreen")
points(unlist(etoile2long), unlist(etoile2lat), pch=4, cex=0.2, col="darkgoldenrod1")
points(unlist(etoile3long),unlist(etoile3lat),pch=4,cex=0.2,col="brown1")
```

Code du graphique moyenne des notes hygiènes par ville (Figure 6.1):

```
install.packages("RMySQL")
install.packages("DBI")

lbdDisconnect(bd)
library(DBI)
bd <- DBI::dbConnect(RMySQL::MySQL(),

                     host = "9ha.h.files.io", port = 3307,
                     username = "ProjetBD_believedbe",
                     password = "83c50259163a65e8ca715f92e1bbb9c15d0ca7a9",
                     dbname = "ProjetBD_believedbe")

dbListTables(bd)

requete <- "SELECT AVG(note_hygiene.note_hygiene), ville.Nom
FROM note_hygiene, restaurant_classique, ville
WHERE ville.id_ville=restaurant_classique.id_ville
AND note_hygiene.id_resto=restaurant_classique.id_resto
AND ville.Nom in ('Paris','Marseille','dijon','Lyon','Nice','Rennes','Strasbourg')
GROUP BY ville.Nom
ORDER BY AVG(note_hygiene.note_hygiene) ASC
"

resultatetoile <- dbSendQuery(bd, requete)

resetoile<-fetch(resultatetoile)
```

```
attach(resetoile)
head(resetoile)

barplot(`AVG(note_hygiene.note_hygiene)`, names.arg = Nom, las= 2, horiz= TRUE, cex.names = 0.8)
title("moyenne des note hygiène par ville")
```

Code du graphique nombre de restaurants étoilés par ville (Figure 6.2) :

```
dbDisconnect(bd)
library(DBI)
bd <- DBI::dbConnect(RMySQL::MySQL(),

                      host = "9ha.h.filess.io", port = 3307,
                      username = "ProjetBD_believedbe",
                      password = "83c50259163a65e8ca715f92e1bbb9c15d0ca7a9",
                      dbname = "ProjetBD_believedbe")

dbListTables(bd)

requete3 <- "SELECT COUNT(restaurant_etoile.id_resto), ville.Nom
FROM note, ville, restaurant_etoile
WHERE ville.id_ville=restaurant_etoile.id_ville
AND note.id_resto=restaurant_etoile.id_resto
and note.nbr_etoile>0
GROUP BY ville.id_ville
order by COUNT(restaurant_etoile.id_resto) desc
limit 7
"

resultatetoile3 <- dbSendQuery(bd, requete3)

resetoile3<-fetch(resultatetoile3)

attach(resetoile3)
head(resetoile3)

barplot(`COUNT(restaurant_etoile.id_resto)`, names.arg = Nom, cex.names = 0.8, las=2)
title("Nombre de restaurants étoilés par ville")
```

Code du graphique moyenne des prix des restaurants étoilés par ville (Figure 6.3) :

```
dbDisconnect(bd)
library(DBI)
bd <- DBI::dbConnect(RMySQL::MySQL(),

                      host = "9ha.h.filess.io", port = 3307,
                      username = "ProjetBD_believedbe",
                      password = "83c50259163a65e8ca715f92e1bbb9c15d0ca7a9",
                      dbname = "ProjetBD_believedbe")

dbListTables(bd)

requete4 <- "SELECT V.Nom, avg(A.Prix) prix_moyen
FROM restaurant_etoile R, ville V, appartient A, note
WHERE A.Id_resto=R.Id_resto
and V.Nom in ('Paris', 'Marseille', 'dijon', 'Lyon', 'Nice', 'Rennes')
AND R.id_ville=V.id_ville
and note.id_resto=R.id_resto
GROUP BY V.Nom
ORDER BY count(R.id_resto) desc
"

resultatetoile4 <- dbSendQuery(bd, requete4)

resetoile4<-fetch(resultatetoile4)

attach(resetoile4)
head(resetoile4)

barplot(`prix_moyen`, names.arg = Nom, cex.names = 0.8, las=2)
title(" La moyenne des prix par ville")
```

Code du graphique du nombre d'étoile par ville rangé par ordre décroissant (Figure 6.4) :

```
library(DBI)
bd <- DBI::dbConnect(RMySQL::MySQL(),

                      host = "9ha.h.filess.io", port = 3307,
                      username = "ProjetBD_believedbe",
                      password = "83c50259163a65e8ca715f92e1bbb9c15d0ca7a9",
                      dbname = "ProjetBD_believedbe")

dbListTables(bd)
```

```

requete2 <- "SELECT sum(note.nbr_etoile), ville.Nom
FROM note, restaurant_etoile, ville
WHERE ville.id_ville=restaurant_etoile.id_ville
AND note.id_resto=restaurant_etoile.id_resto
GROUP BY ville.id_ville
HAVING COUNT(restaurant_etoile.name)>1
order by sum(note.nbr_etoile) desc
limit 5
"

resultatetoile2 <- dbSendQuery(bd, requete2)

resetoile2<-fetch(resultatetoile2)

attach(resetoile2)

head(resetoile2)

barplot(`sum(note.nbr_etoile)`, names.arg = Nom,las= 2,cex.names = 0.8)
title("Nombre d'etoile par ville, rangé par ordre décroissant")

```