



Projet d'analyse de données d'une entreprise d'import/export

Foody

[FOODY - Google Drive](#)

[Projet_Foody \(github.com\)](#)

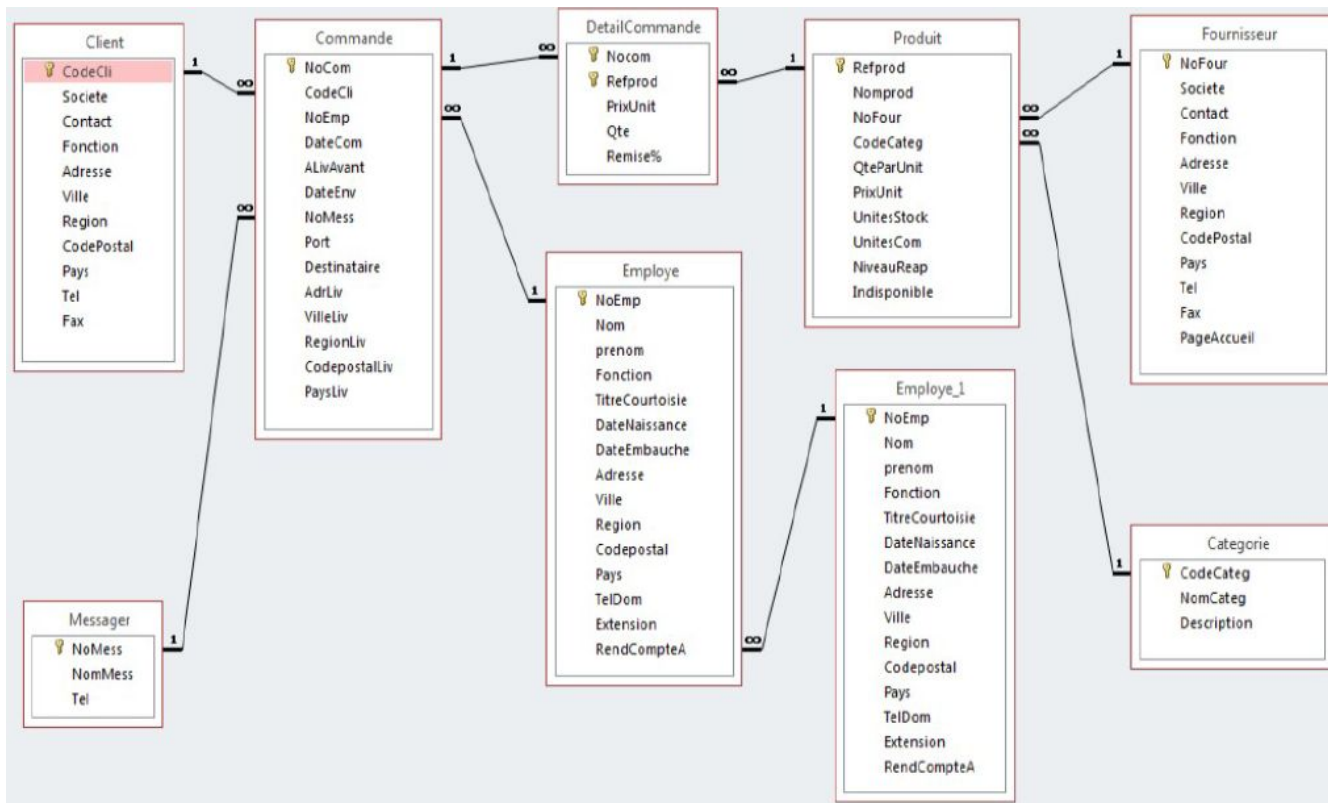


Sommaire

- ❖ Schéma de la Base de données
- ❖ Modèle Conceptuel de données
- ❖ Modèle Logique / dictionnaire des données
- ❖ Modèle physique
 - Restriction sur agrégats
 - Jointures naturelles
- ❖ Visualisation
 - plot
 - plot.pie
 - matplotlib.pyplot
 - avec seaborn.clustermap
 - avec Folium
- ❖ Propositions pour Foody

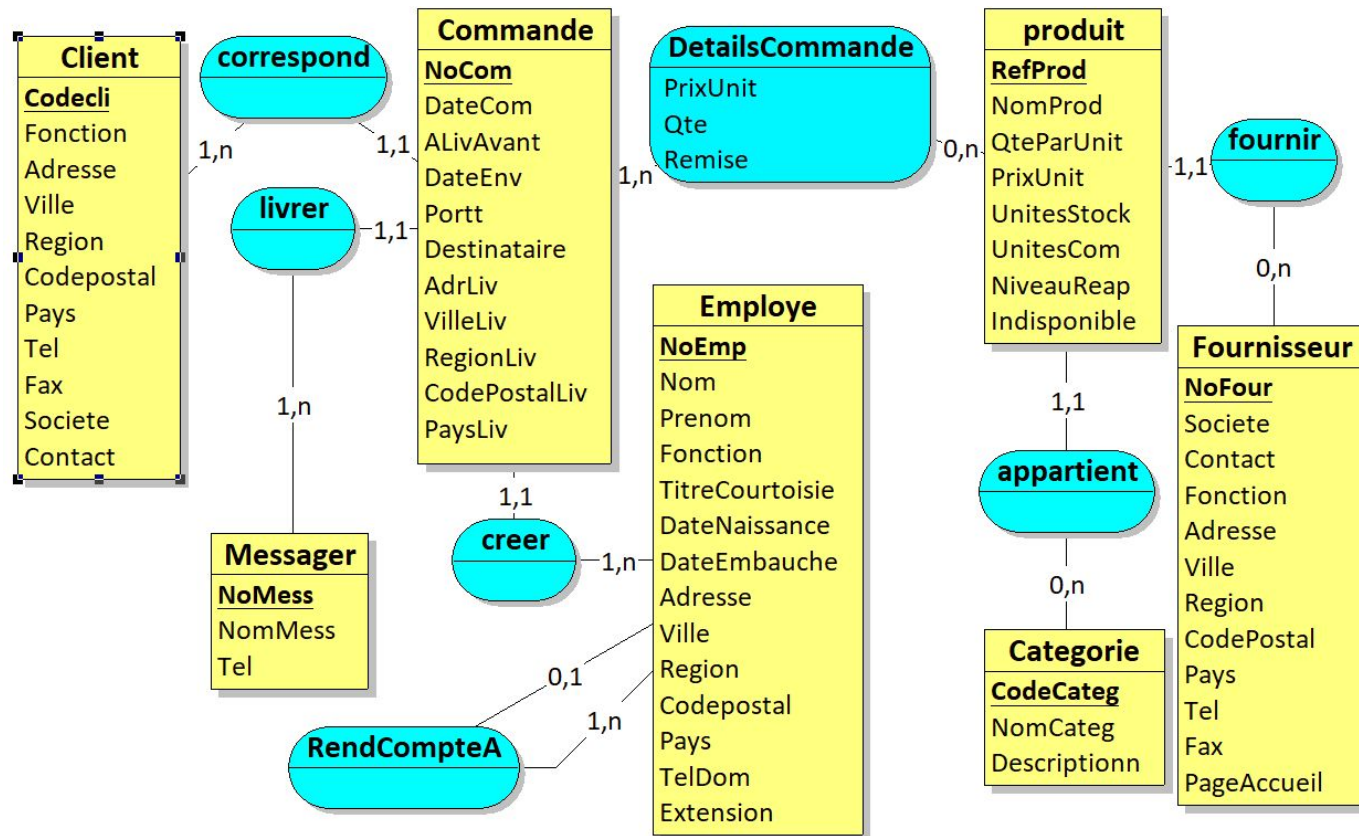


Schéma de la Base de données





Modèle Conceptuel de données





Modèle Logique / dictionnaire des données

Client(Codecli, Societe, Contact, Fonction, Adresse, Ville, Region, Codepostal, Pays, Tel, Fax)

Messenger(NoMess, NomMess, Tel)

Commande(NoCom, #CodeCli, #NoEmp, DateCom, ALivAvant, DateEnv, #NoMess, Port, Destinataire, AdrLiv, VilleLiv, RegionLiv, CodePostalLiv, PaysLiv)

DetailsCommande(#Nocom, # Refprod, PrixUnit, Qte, Remise%)

Employe(NoEmp, Nom, prenom, Fonction, TitreCourtoisie, DateNaissance, DateEmbauche, Adresse, Ville, Region, Codepostal, Pays, TelDom, Extension, #RendCompteA)

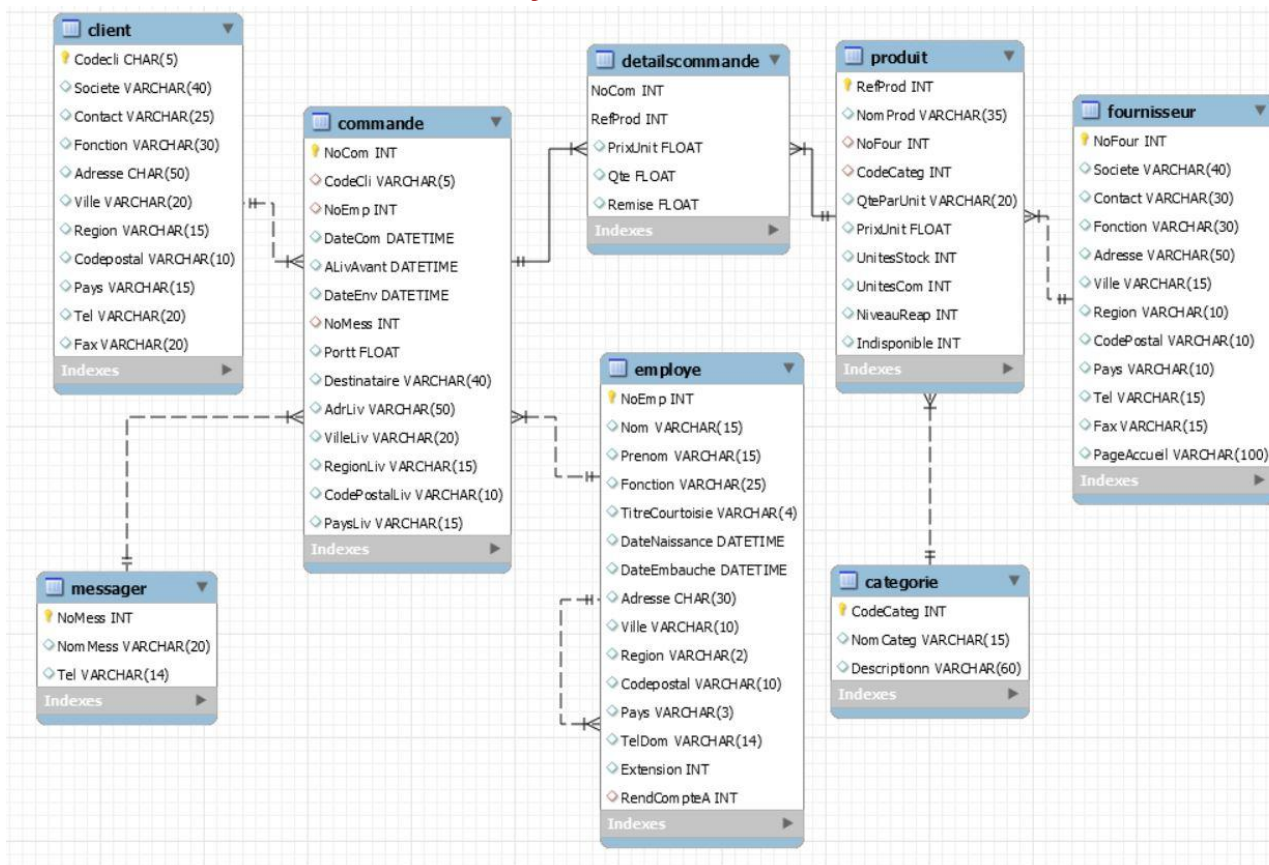
Produit(RefProd, NomProd, #NoFour, #CodeCateg, QteParUnit, PrixUnit, UnitesStock, UnitesCom, NiveauReap, Indisponible)

Fournisseur(NoFour, Societe, Contact, Fonction, Adresse, Ville, Region, CodePostal, Pays, Tel, Fax, PageAccueil)

Categorie(CodeCateg, NomCateg, Description)



Modèle physique



script SQL avec les réponses aux requêtes



III.4- Restriction sur agrégats

```
#1. Lister les fournisseurs ne fournissant qu'un seul produit
select NoFour, RefProd from foody.produit group by NoFour having count(RefProd)<2;

#2. Lister les catégories dont les prix sont en moyenne supérieurs strictement à 50 euros
Select CodeCateg from foody.produit group by CodeCateg having Avg(PrixUnit)>50;

#3. Lister les fournisseurs ne fournissant qu'une seule catégorie de produits
select NoFour, CodeCateg, RefProd, count(*) as 'Nombre de meme produits' from foody.produit
group by NoFour having count(distinct(CodeCateg))=1;

#4. Lister le Products le plus cher pour chaque fournisseur, pour les Products de plus de 50 euro???
select NoFour, RefProd, PrixUnit, max(PrixUnit) from foody.produit group by NoFour having max(PrixUnit)>50;
```

IV.1- Jointures naturelles

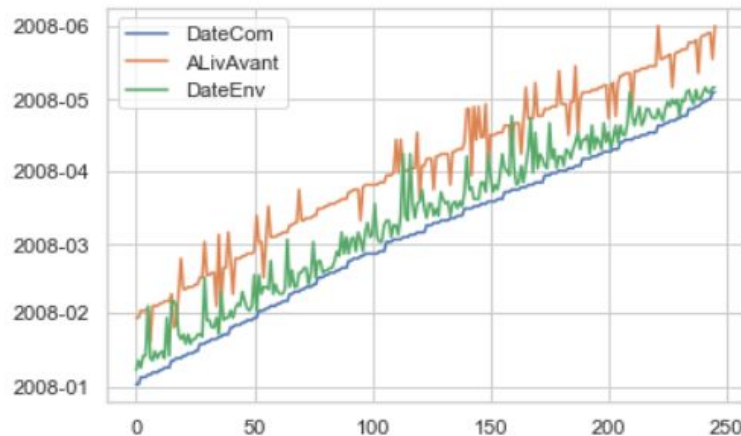
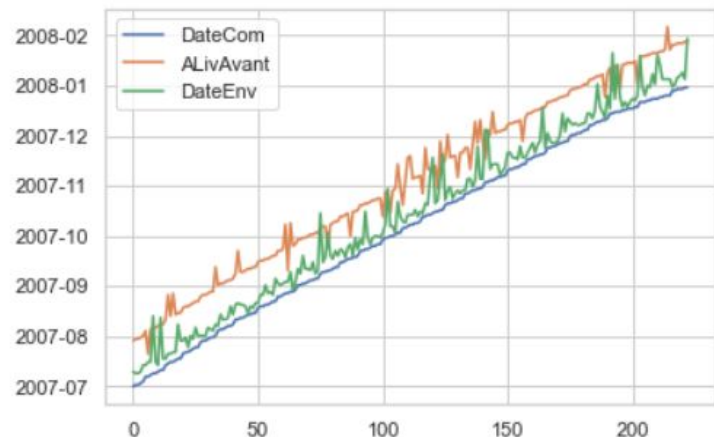
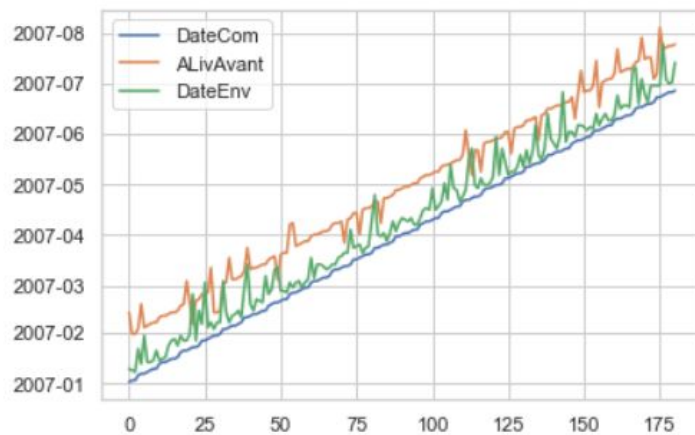
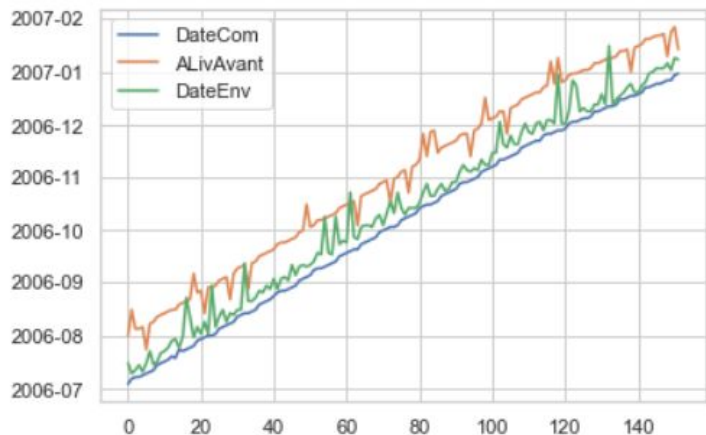
```
#1. Récupérer les informations des fournisseurs pour chaque produit
Select p.RefProd, p.NomProd, f.* from foody.produit as p natural join foody.fournisseur as f;

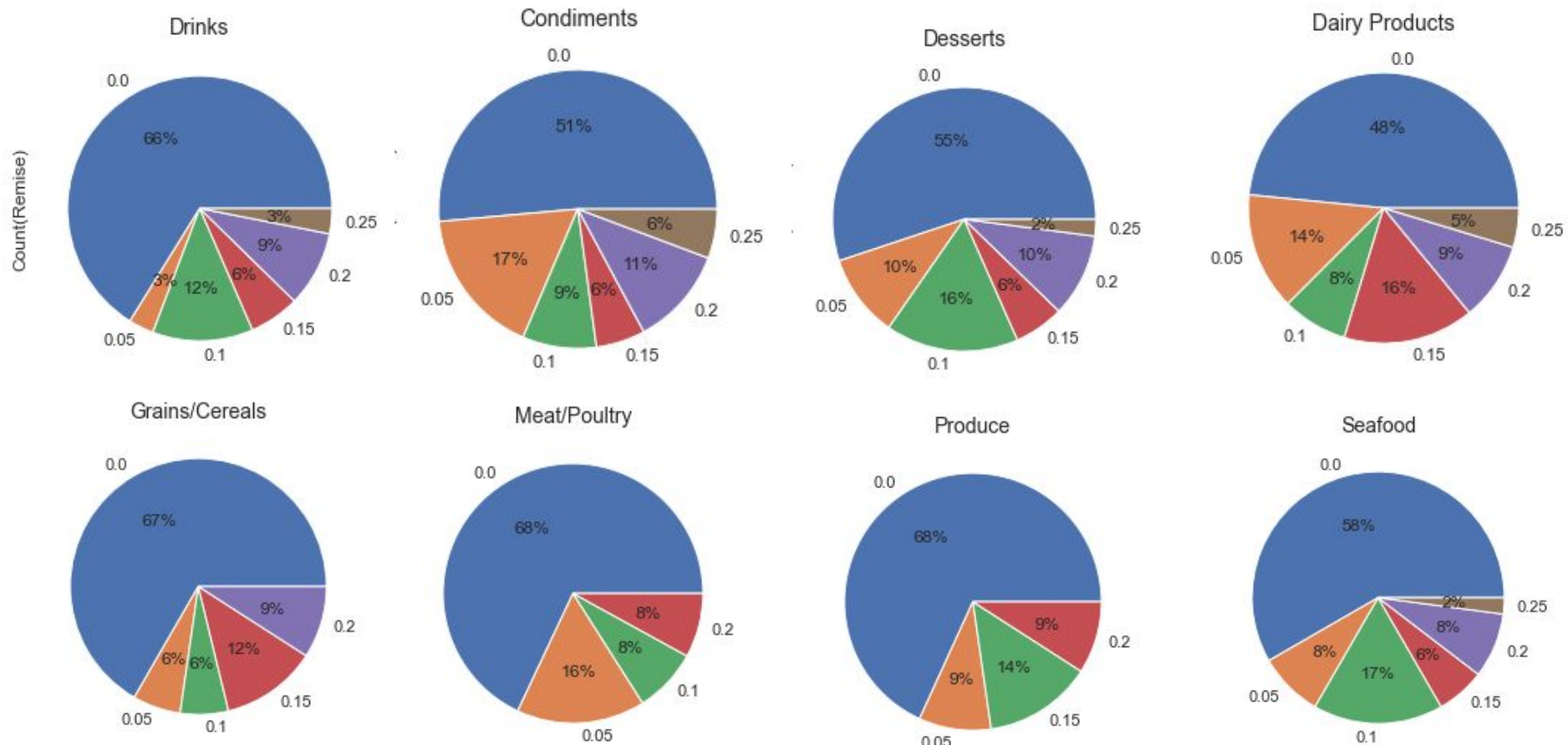
#2. Afficher les informations des commandes du client "Lazy K Kountry Store"
Select cl.Societe, co.* from foody.commande as co natural join foody.client as cl where cl.Societe= "Lazy K Kountry Store"
;
Select * from foody.commande where Destinataire= "Lazy K Kountry Store" ;

#3. Afficher le nombre de commande pour chaque messenger (en indiquant son nom)
Select m.NomMess, Count(co.NoCom) as 'le nombre de commande' from foody.commande as co NATURAL JOIN foody.messenger as m group
by NoMess;
```



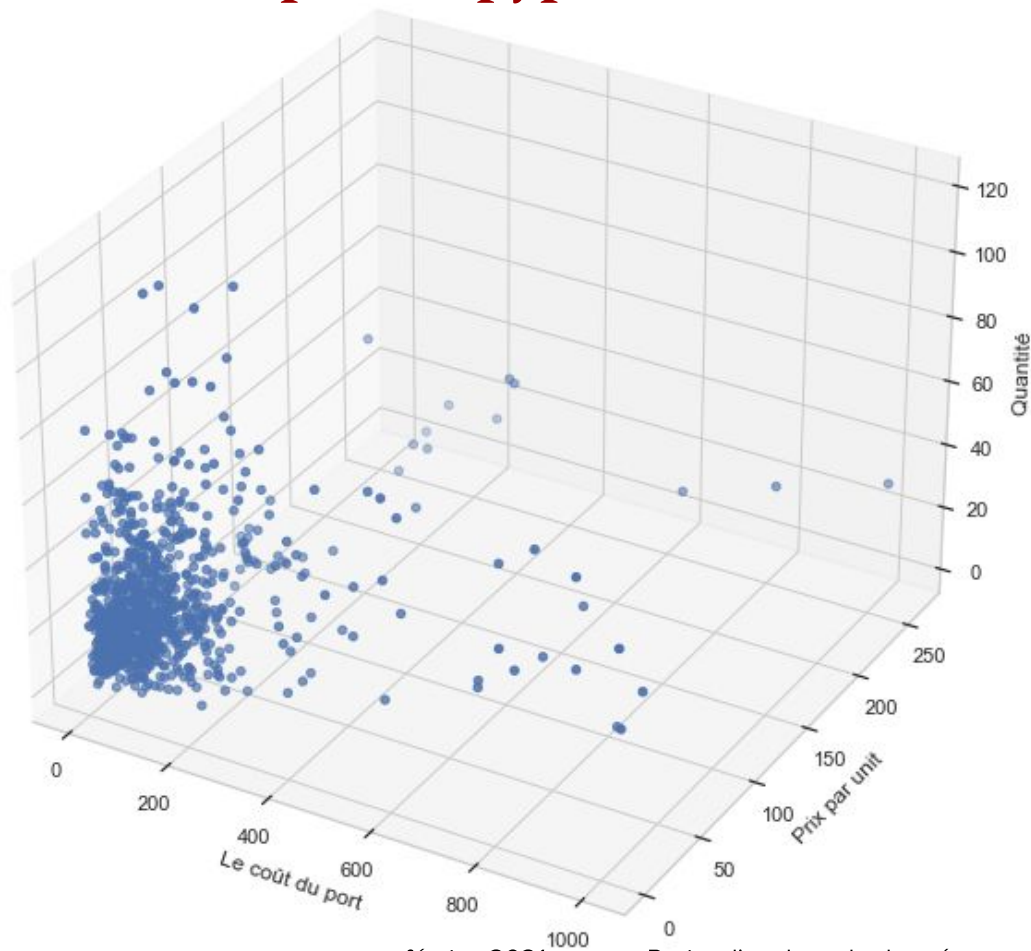

Visualisation- avec plot



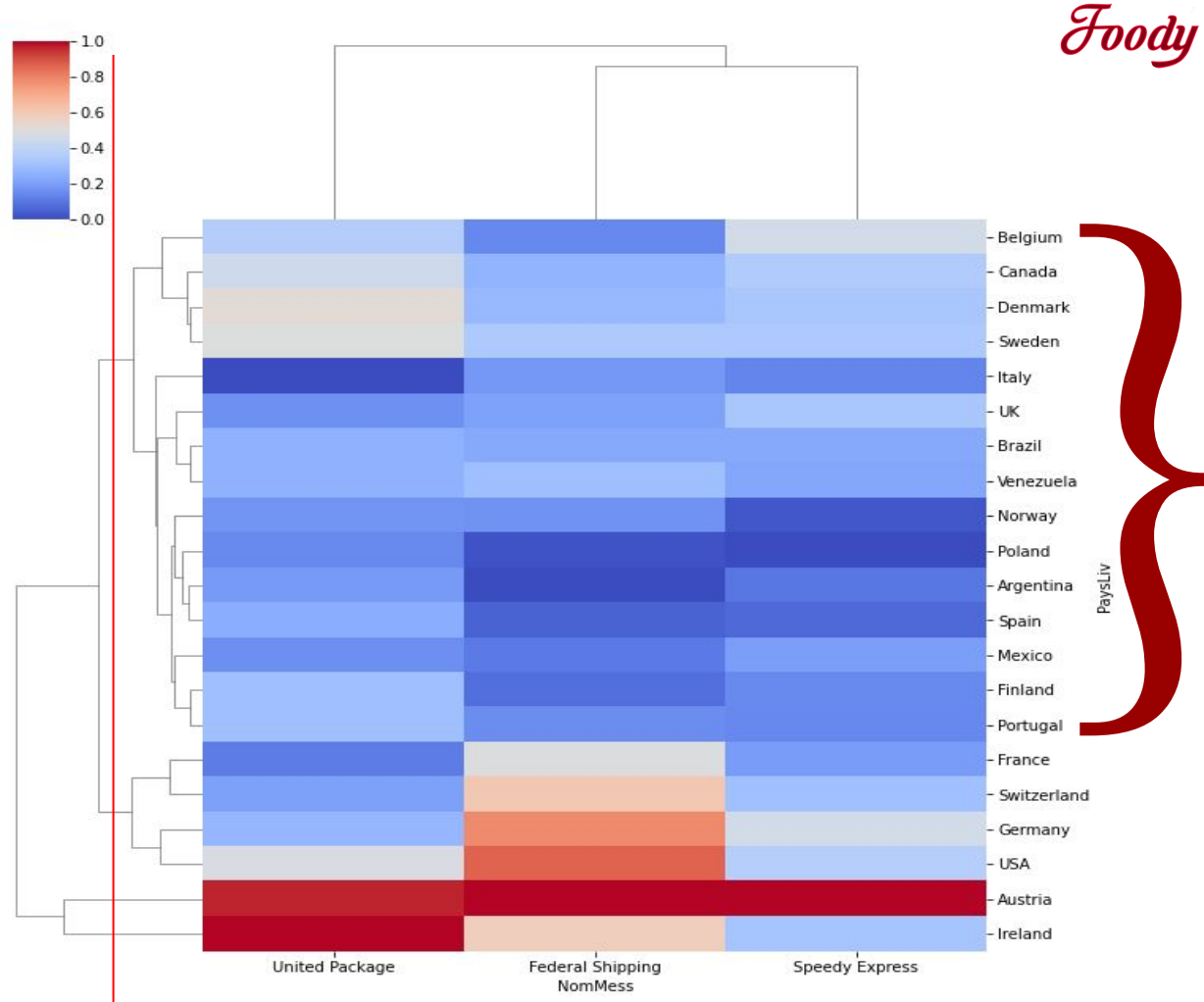




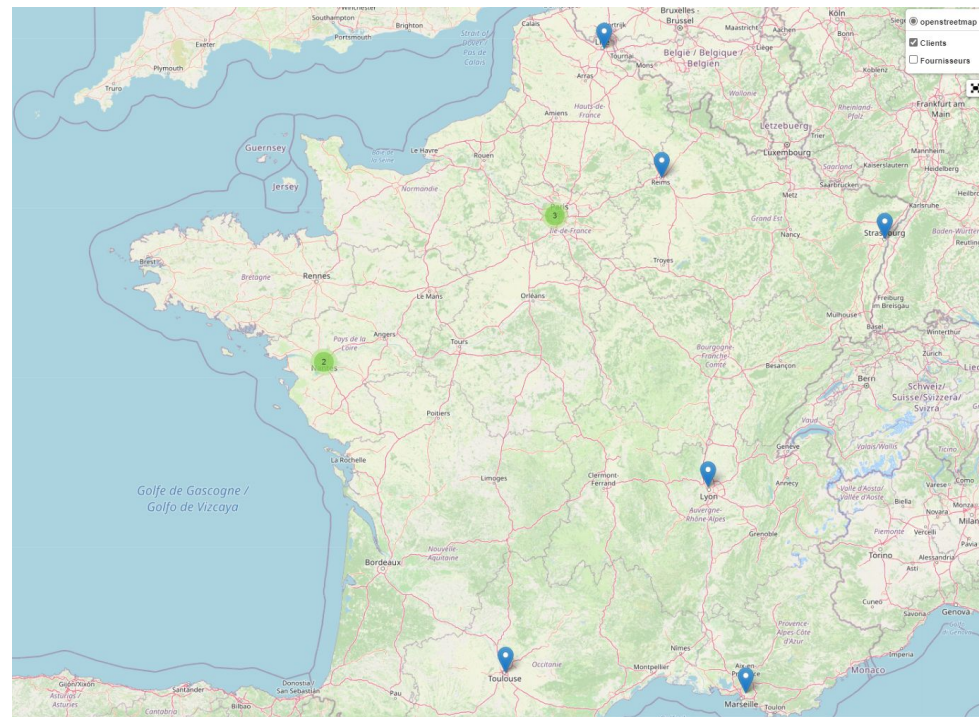
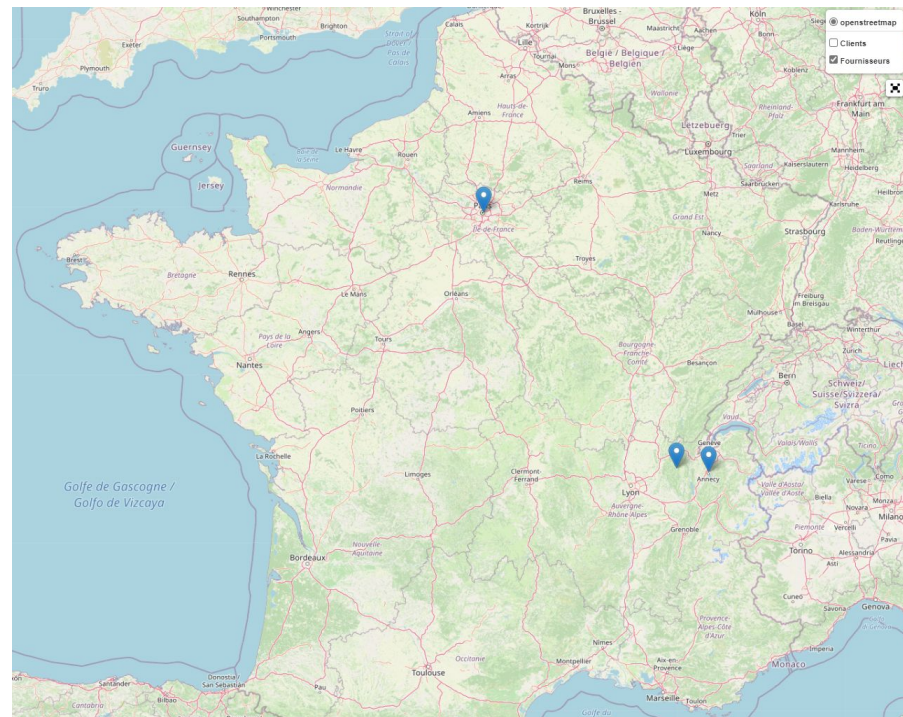
Visualisation- avec matplotlib.pyplot



Visualisation- avec seaborn.clustermap



Visualisation- avec Folium



ⓘ Propositions:

- ❖ Négocier le prix du produit “Côtes-de-blaye” du fournisseur (*Aux joyeux ecclésiastiques*) car c’est le produit le plus cher.
- ❖ Reconsidérer les remises, pour éventuellement augmenter le profit(par exemple “un acheté, un gratuit”).
- ❖ Foody pourrait reconsidérer les coûts de port pour l’Autriche et l’Irlande car il y a peut être des risques de perte de client.
- ❖ Pour que Foody fasse plus de profit , on peut lui conseiller de se focaliser sur son optimisation des processus liés à ce genre de commande (les coûts du port sont de moins de 200, la quantité de moins de 80 unités et le prix d'une unité est de moins de 150).
- ❖ Il faudra aussi se concentrer sur la livraison rapide des commandes pour qu’aucune ne dépasse la date prévue de livraison. Il faudrait encore essayer de réduire les délais de livraison .
- ❖ Peut-être, selon cette base, on peut envisager de délocaliser ou de trouver de nouveaux fournisseurs plus proches des clients afin de réduire les coûts de livraison.



Sources:

1. Matériel de discord
2. <https://www.kaggle.com/>
3. <https://seaborn.pydata.org/>
4. [Pandas & Seaborn - A guide to handle & visualize data in Python | Tryolabs Blog](#)
5. <https://stackoverflow.com/>
6. [2.3. Clustering — scikit-learn 0.24.1 documentation \(scikit-learn.org\)](#)
7. [Create pie charts with Matplotlib, Seaborn and Pandas \(dataforeverybody.com\)](#)
8. <https://www.google.com/maps/>
9. [Mapping Points with Folium | Data EconoScientist \(georgetsilva.github.io\)](#)

Retour personnel