TP3 : Traitement Fonctionnel de Données

RÉALISÉ PAR : GAOWEN LI, SARAH MELAIKIA
CODE SOURCE SUR GITHUB : HTTPS://GITHUB.COM/GL0004/TP3
DATE DE RENDU : 13 JUILLET 2025

1. INTRODUCTION :

Ce projet s'inscrit dans le cadre du TP3 et vise à explorer et modéliser des données liées à la production de déchets ménagers à Paris.

Les données, générées de manière synthétique, décrivent différents foyers selon leur code postal, le type de déchet et le nombre de personnes dans le foyer. L'objectif est de prédire la quantité annuelle de déchets à partir de ces caractéristiques.

Pour cela, l'analyse s'appuie sur deux outils complémentaires : Scala (avec la bibliothèque Smile) pour le traitement et la modélisation, et Python (notebook Jupyter) pour la visualisation des résultats.

Les données proviennent d'un fichier CSV (donnees_dechets_paris.csv) utilisé comme jeu de données dans ce projet.
À noter : les données étant fictives, les performances du modèle ne reflètent pas une capacité réelle de prédiction.

2. DESCRIPTION DES DONNÉES

Le jeu de données utilisé dans ce projet est un fichier CSV simulé nommé donnees_dechets_paris.csv. Il contient des informations détaillées sur la production de déchets par foyer à Paris. Chaque ligne représente un foyer unique, avec les variables suivantes :

- id : identifiant unique du foyer.
- adresse : adresse postale.
- code_postal : code postal du foyer.
- commune : nom de la commune.
- nombre_personnes_foyer : nombre de personnes dans le foyer.
- type_dechet : type de déchet (verre, plastique, etc.).
- quantite_kg_par_an : quantité annuelle de déchets produits (en kg).
- mode_de_collecte : méthode de collecte des déchets.
- tri_effectue : indicateur binaire du tri effectué ou non.
- date_collecte : date de la collecte.
- surface_logement : surface du logement en m^2.
- revenu_menage : revenu annuel du ménage.
- age_moyen_menage : âge moyen des membres du foyer.
- nombre_dechets_menager_par_mois : fréquence mensuelle moyenne de production de déchets ménagers.
- densite_population : densité de population dans la zone du foyer.

Ces données ont été générées automatiquement à l'aide d'un script Python, dans le but de simuler un jeu de données réaliste pour un usage pédagogique.

3. TRAITEMENT ET AGRÉGATION

Les données ont d'abord été nettoyées pour ne conserver que les lignes valides. Les enregistrements incomplets ou malformés (par exemple, une date invalide ou une valeur numérique non convertible) ont été automatiquement écartés lors du parsing. De plus, seules les lignes où le tri des déchets a été effectivement réalisé (tri_effectue = true) ont été retenues pour l'analyse.

Le fichier d'origine contient 15 colonnes, décrivant les caractéristiques du foyer (code postal, nombre de personnes, surface, etc.), le type de déchet collecté, ainsi que des variables socio-économiques comme le revenu du ménage ou la densité de population.

Une fois les données nettoyées, nous avons réalisé un traitement d'agrégation, en regroupant les foyers par code postal, type de déchet et nombre de personnes, afin de calculer la quantité moyenne de déchets par an (quantiteKgParAn) pour chaque combinaison. Cette étape permet d'obtenir une vision synthétique du volume de déchets par type et par zone géographique.

Ensuite, des statistiques descriptives ont été calculées sur les quantités annuelles produites : moyenne, médiane, écart-type, minimum et maximum. Ces indicateurs sont ensuite exportés dans un fichier CSV à des fins de visualisation.

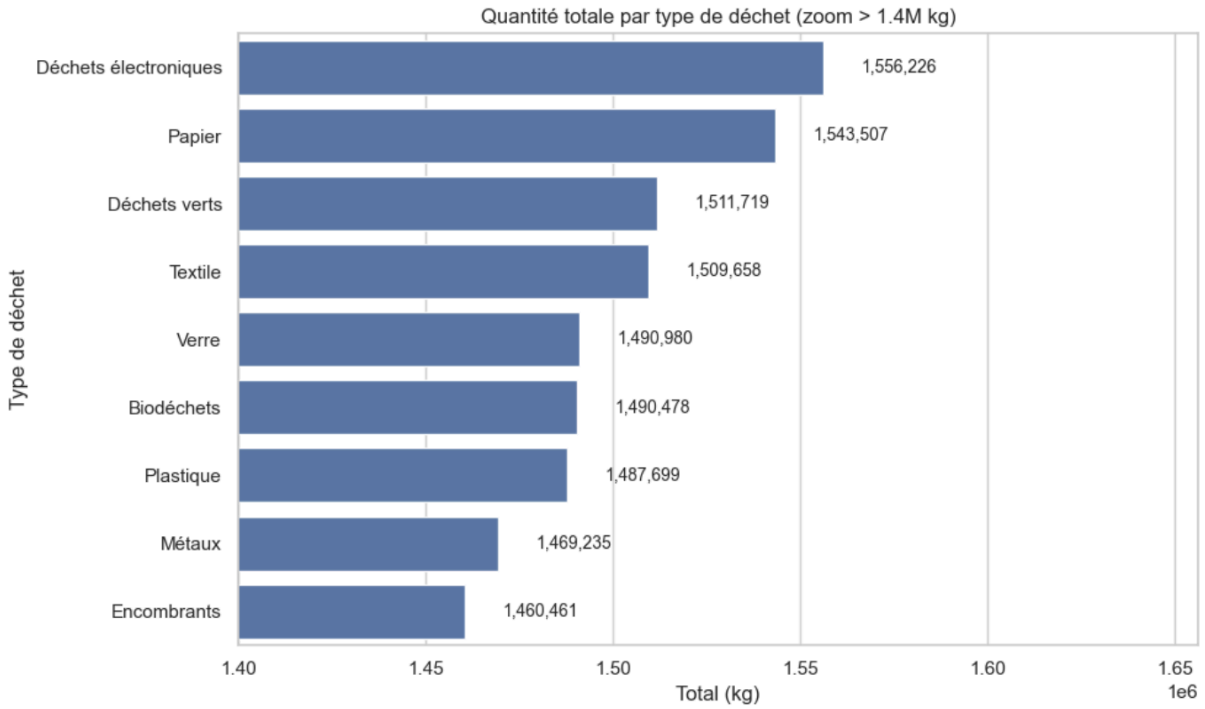
Enfin, un modèle de régression linéaire a été entraîné à l'aide de la bibliothèque Smile, dédiée au machine learning dans l'écosystème Scala.

Toutes les étapes de traitement ont été implémentées en Scala, en appliquant les principes de la programmation fonctionnelle exigés dans le cadre du projet.
Le code repose sur des fonctions pures et immuables, sans effets de bord, et utilise des abstractions telles que map, filter ou Validated.
L'ensemble a été conçu de manière modulaire et testable, en s'appuyant sur la bibliothèque de flux fs2 et les patterns vus en cours.

```scala
override def run: IO[Unit] = {
  Files[IO]
    .readAll(csvPath)
    .through(utf8.decode)
    .through(lines)
    .filter(_.nonEmpty)      // Drop empty lines
    .drop(1)                 // Skip header row
    .map(splitCsv)           // Split CSV line into columns
    .map(parseLine)          // Parse columns into Dechet case class
    .collect { case Some(d) => d }  // Keep only successfully parsed records
    .filter(_.triEffectue)   // Filter for records with recycling enabled
    .compile
    .toList
```

Exemple d'un pipeline fs2 lisant le fichier CSV ligne par ligne, appliquant des transformations fonctionnelles (map, filter, collect) pour parser, nettoyer et filtrer les données avant leur agrégation.

4. TABLEAUX DE BORD ET VISUALISATIONS

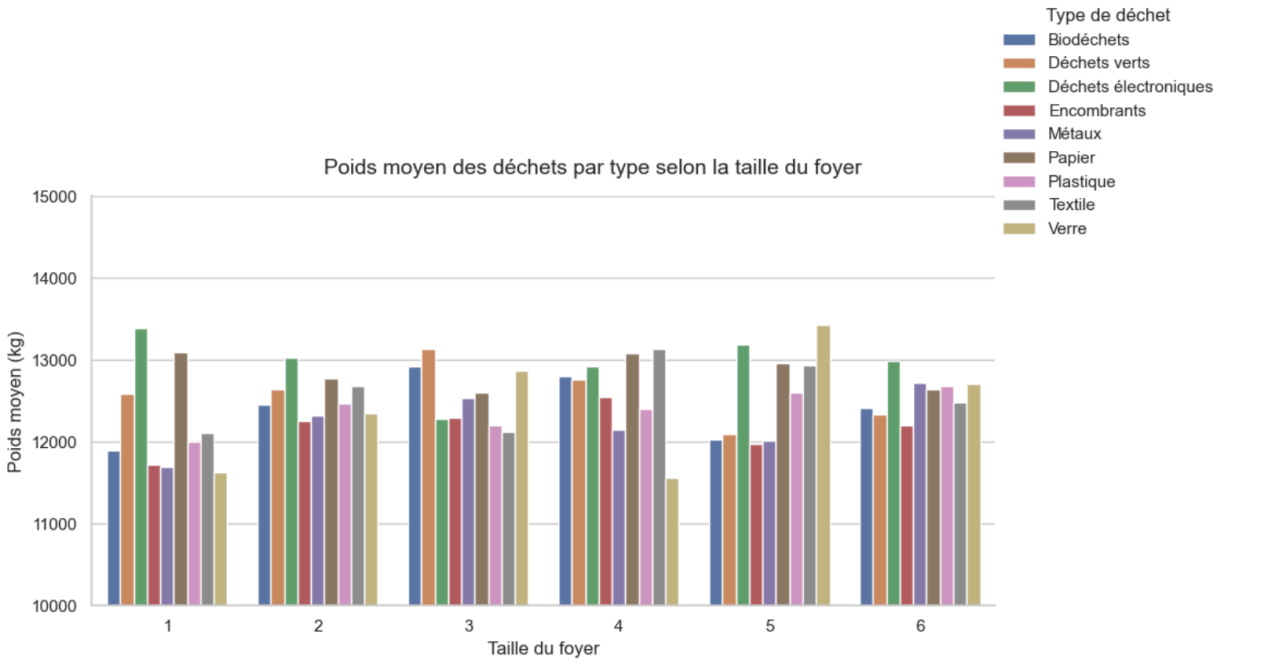


VISUALISATION 1 — QUANTITÉ TOTALE PAR TYPE DE DÉCHET (ZOOM > 1.4M KG)

Cette première visualisation montre la quantité totale de déchets collectés, regroupée par type. Il s'agit d'un graphique en barres horizontales, limité aux valeurs supérieures à 1,4 million de kilogrammes afin de mieux faire apparaître les différences.

On constate que les Déchets électroniques sont les plus abondants, suivis de près par le Papier et les Déchets verts. Les écarts entre catégories sont relativement faibles à ce niveau d'agrégation, ce qui justifie l'usage d'un zoom pour mieux visualiser les variations.

Ce graphique permet d'identifier les types de déchets les plus générés à l'échelle de la ville, ce qui peut guider les politiques de gestion des déchets et de sensibilisation au tri.

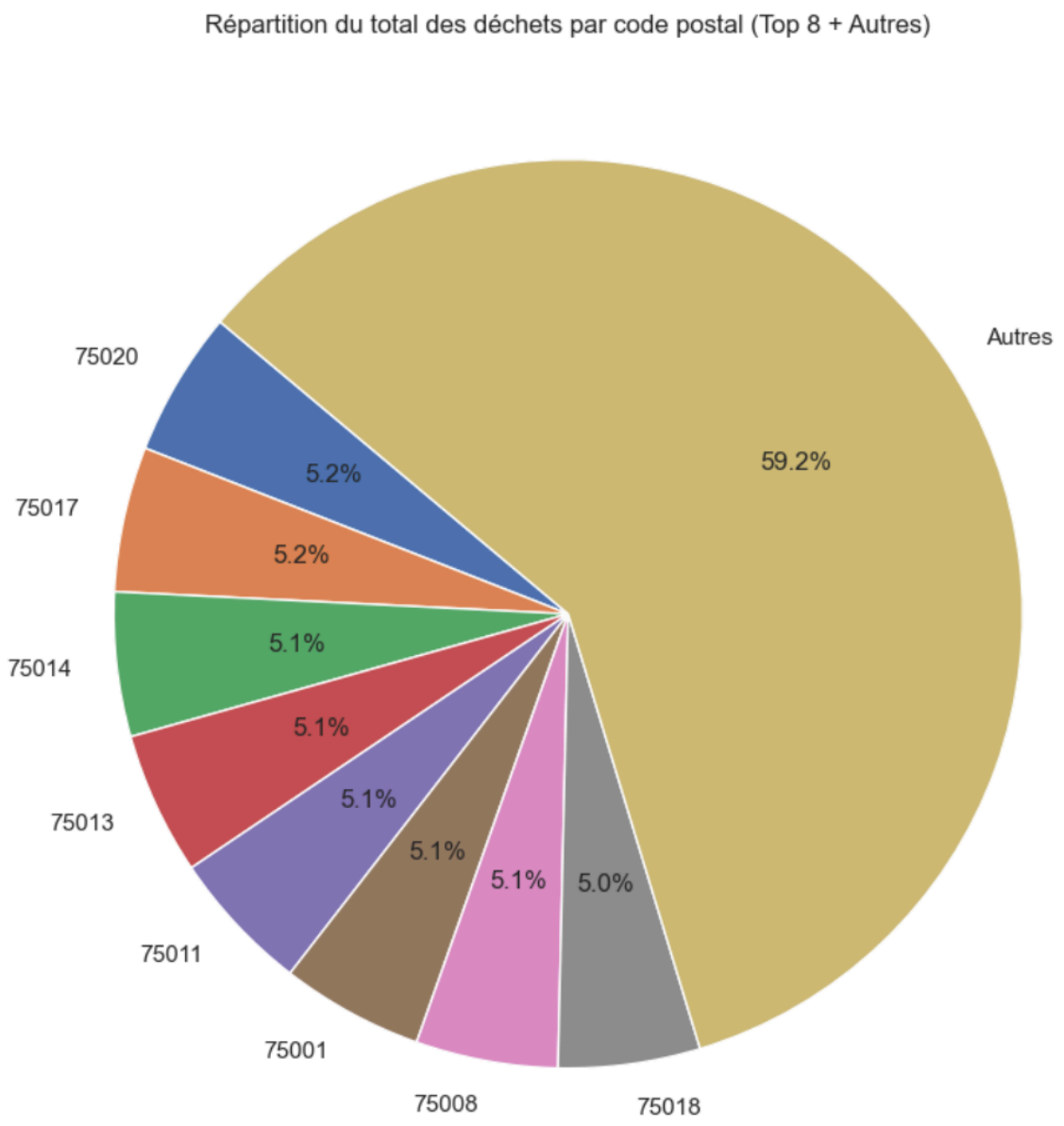


VISUALISATION 2 — POIDS MOYEN DES DÉCHETS PAR TYPE SELON LA TAILLE DU FOYER

Ce graphique présente le poids moyen des déchets (en kg) pour chaque type de déchet, en fonction de la taille du foyer. On observe ici une distribution comparée entre les foyers allant d'une personne à six personnes. Chaque barre colorée représente un type de déchet.

On constate que la production de déchets ne suit pas nécessairement une progression linéaire avec la taille du foyer. Par exemple, certains types de déchets comme le verre ou les déchets électroniques présentent une variation significative selon la taille du foyer, tandis que d'autres restent relativement stables.

Cette visualisation permet d'explorer l'impact de la composition du foyer sur les comportements de consommation et de production de déchets, ce qui peut être utile pour des campagnes ciblées ou des services de collecte adaptés.

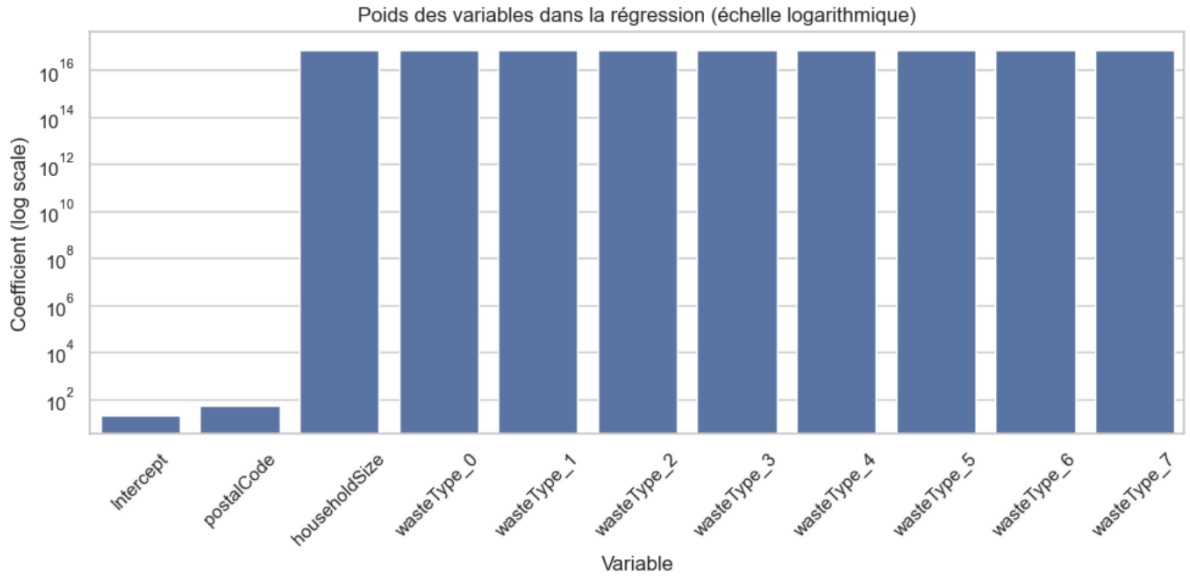


VISUALISATION 3 — RÉPARTITION DU TOTAL DES DÉCHETS PAR CODE POSTAL (TOP 8 + AUTRES)

Ce graphique en camembert illustre la répartition du volume total de déchets par code postal à Paris. Les huit arrondissements les plus générateurs de déchets sont affichés individuellement, tandis que le reste des codes postaux est regroupé dans la catégorie « Autres ».

On remarque que la catégorie « Autres » représente à elle seule près de 60 % du total, ce qui indique une répartition relativement dispersée de la production de déchets sur l'ensemble du territoire parisien. Les codes postaux comme 75020, 75017 ou 75014 apparaissent parmi les plus contributeurs individuels.

Ce type de visualisation est utile pour identifier les zones à fort volume de déchets, ce qui peut orienter des actions de sensibilisation ou des ajustements logistiques pour les services de collecte.



VISUALISATION 4 — POIDS DES VARIABLES DANS LA RÉGRESSION (ÉCHELLE LOGARITHMIQUE)

Ce graphique présente l'importance relative des différentes variables explicatives dans le modèle de régression linéaire, exprimée ici en valeur absolue et sur une échelle logarithmique.

On observe que certaines variables, notamment la taille du foyer (householdSize) et les variables dérivées du type de déchet (wasteType_0, wasteType_1, etc.), ont un poids très élevé dans le modèle. À l'inverse, des variables comme le code postal (postalCode) ou l'ordonnée à l'origine (Intercept) ont un impact bien plus limité.

L'utilisation d'une échelle logarithmique permet ici de mieux visualiser les écarts d'ordre de grandeur entre les coefficients, facilitant ainsi l'interprétation des facteurs les plus influents.

À noter que ces résultats sont à interpréter avec prudence, car le modèle a été entraîné sur des données simulées, ce qui limite sa validité prédictive.

CONCLUSION

Ce projet présente une chaîne de traitement complète des données simulées sur les déchets des foyers parisiens : de la lecture et du nettoyage des données, à l'agrégation, l'analyse statistique, la modélisation prédictive et la visualisation des résultats.

L'ensemble du traitement a été réalisé en Scala en respectant rigoureusement le paradigme fonctionnel. Les données ont été nettoyées et filtrées dès l'ingestion afin de garantir la validité des traitements suivants.

Bien que les données soient simulées, et donc que les résultats de la régression n'aient pas de valeur prédictive réelle, la structure mise en place est réutilisable et pourrait être appliquée à des données réelles sans modification majeure.

Les visualisations ont été générées dans un notebook Jupyter, permettant d'explorer efficacement les tendances par type de déchet, taille du foyer ou localisation géographique. Ce travail constitue une base solide pour de futures améliorations, telles que l'ajout de nouvelles sources de données ou l'utilisation d'autres modèles de machine learning.