**Analyse exploratoire des données Navitia**

**Objectif : Comprendre la gestion des flux ferroviaires et les impacts sur la livraison client**

**Méthodologie :**

Afin de mieux appréhender les données liées à l'infrastructure ferroviaire et anticiper les retards de livraison, je me suis fixé comme objectif de me familiariser avec l'API **Navitia**.

Pour cela, j’envisage une approche modulaire reposant sur l’exploration successive de plusieurs endpoints clés :

* **disruptions** : pour lister les perturbations ou incidents sur le réseau.
* **vehicle\_journeys** : pour suivre les trajets planifiés et effectués des trains.
* **routes** : pour comprendre les itinéraires empruntés.
* **stop\_areas** : pour localiser les points d’arrêt, potentiellement corrélés à des zones de chargement ou de dépôt de marchandises.
* **commercial\_modes** : pour distinguer les différents types de transport (fret, voyageurs, etc.).

L’objectif à terme est d’associer chaque **trainId** ou **disruptionId** à des données temporelles et géographiques, afin de détecter les causes et impacts des retards dans la chaîne logistique.

**Hypothèse de travail sur les perturbations (disruptions)**

À la lecture des données, deux champs retiennent mon attention :

* **baseArrivalTime** : heure théorique d’arrivée.
* **amendedArrivalTime** : heure d’arrivée révisée (en cas de perturbation).

Premiers constats :

* Lorsqu’un train prend du retard (par exemple +5 minutes sur un arrêt), ce même retard semble se répercuter automatiquement sur les arrêts suivants.
* Cela suggère que Navitia repose sur un modèle d’horaires fixes, où les retards ne recalculent pas dynamiquement l’ensemble du parcours.

🔍 **Conséquence analytique** :  
Pour évaluer le **retard effectif d’un train**, il est plus pertinent de comparer **le dernier arrêt planifié avec son heure réelle d’arrivée**, plutôt que de cumuler les retards intermédiaires. Cela permet d’éviter les doublons de comptage et d'obtenir une mesure précise de l’impact d’une perturbation.