

Rapport de Projet : API Publique & Visualisation de Données

Cours : Python Avancé

Étudiant : Dadou ASSOGBA

1. Choix de l'API

Pour cet exercice, j'ai sélectionné l'API publique **SpaceX-API (v4)**.

- **URL** : <https://api.spacexdata.com/v4/launches>
- **Justification technique :**
 1. **Accessibilité** : C'est une API ouverte (*Auth: No*), ce qui simplifie le code (pas de gestion de clés API) et facilite la correction de l'exercice.
 2. **Richesse des données** : L'API fournit des objets JSON complexes (imbrications, dates, statuts booléens), ce qui est idéal pour démontrer des compétences en nettoyage de données.

2. Méthodologie et "Best of Code"

J'ai structuré le script selon le modèle **ETL** (Extract, Transform, Load) pour séparer la logique de récupération de celle du nettoyage.

A. Stratégie de sauvegarde (Extract)

J'ai fait le choix de sauvegarder les données brutes (raw.json) dès la réception. **Pourquoi ?** Cela permet de conserver une trace exacte de la réponse API et d'éviter de refaire des requêtes HTTP inutiles lors de la phase de développement du nettoyage.

Python

```
# Sauvegarde des données brutes pour traçabilité
```

```
# Sauvegarde brute
with open(RAW_FILE, "w", encoding="utf-8") as f:
    json.dump(data, f, indent=4)
print(f"✓ {len(data)} lancements récupérés et sauvagardés dans {RAW_FILE}")
return data
except Exception as e:
    print(f"✗ Erreur : {e}")
return None
```

B. Nettoyage des données (Transform)

Le traitement des données était nécessaire car l'API renvoie tous les lancements, y compris ceux futurs (dont le résultat est null). **Choix effectués :**

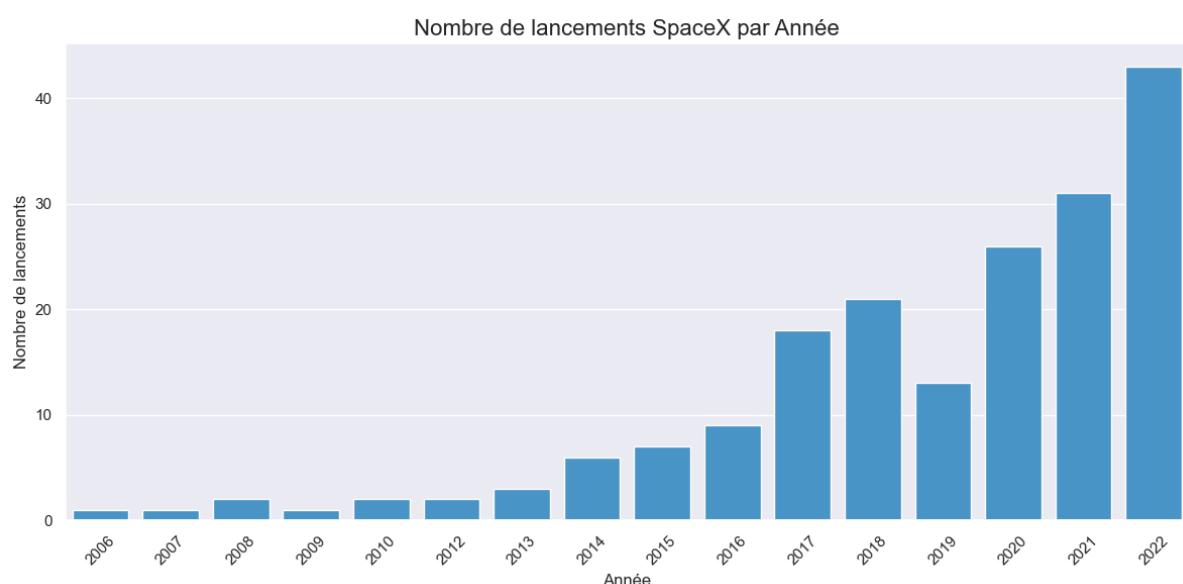
1. **Filtrage** : Suppression des lancements futurs pour ne pas fausser les statistiques de réussite.
2. **Extraction** : Récupération de l'année (slicing sur la chaîne ISO 8601) pour l'analyse temporelle.

```
for launch in raw_data:  
    # On ignore les lancements futurs (où 'success' est null)  
    if launch.get("success") is None:  
        continue  
  
    # Extraction de l'année (la date est au format "2020-10-24T...")  
    date_utc = launch.get("date_utc", "0000")  
    year = date_utc[:4] # On garde juste les 4 premiers caractères
```

3. Visualisations et Analyse

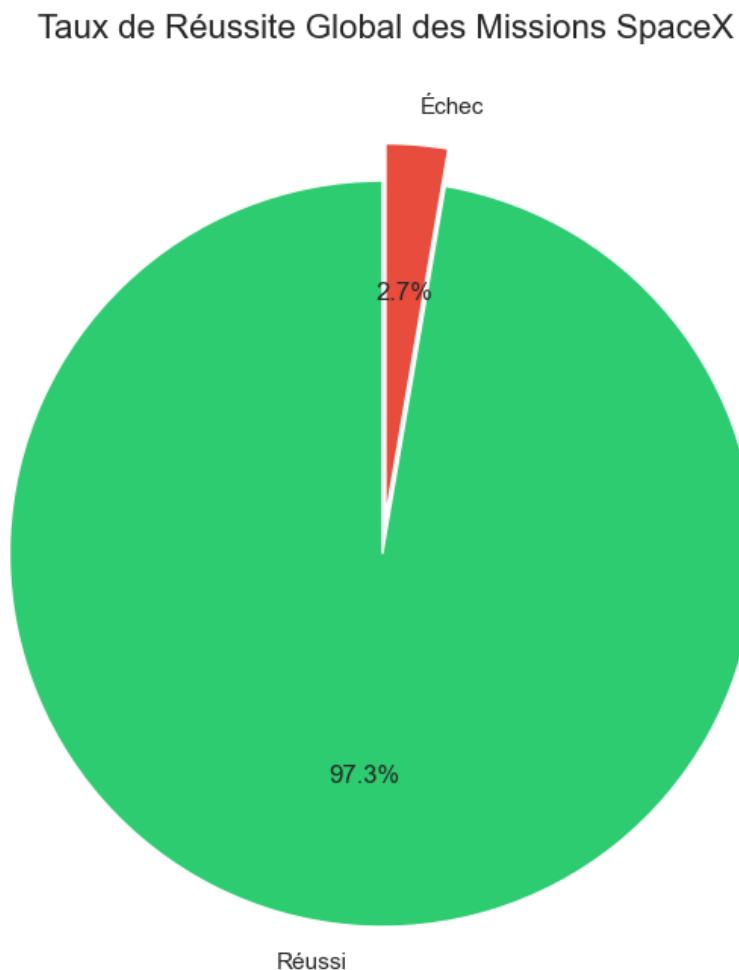
Les graphiques ont été générés avec les librairies **Pandas** (gestion des DataFrames) et **Seaborn** (rendu visuel).

Graphique 1 : Nombre de lancements par année



Observation : Ce graphique permet de visualiser l'évolution temporelle de l'activité. On constate une augmentation significative de la fréquence des tirs à partir de 2017, démontrant une phase d'industrialisation des lancements.

Graphique 2 : Répartition des succès/échecs



Observation : Ce diagramme met en évidence le taux de fiabilité. La part prédominante des succès (en vert) par rapport aux échecs (en rouge) confirme la stabilisation technologique du lanceur sur l'ensemble des données historiques.