# Rapport ACCIPREDICT

# Scripts Python pour le Nettoyage des Données CSV

Cette partie regroupe tous les scripts Python utilisés pour analyser et nettoyer un fichier CSV en supprimant les lignes contenant au moins une cellule vide tout en conservant l'intégrité du format d'origine.

## 1️⃣ Script - Comptage des cellules vides par colonne

**Ce script charge le fichier CSV et compte le nombre de cellules vides dans chaque colonne.**

import pandas as pd  
def count\_empty\_cells(csv\_file):  
 try:  
 # Lecture du fichier CSV avec encodage UTF-8 et séparation par point-virgule  
 df = pd.read\_csv(csv\_file, encoding="utf-8-sig", sep=";")  
   
 # Comptage des cellules vides par colonne  
 empty\_cells\_count = df.isnull().sum()  
   
 print("Nombre de cellules vides par colonne :")  
 print(empty\_cells\_count)  
 except Exception as e:  
 print(f"Erreur lors de la lecture du fichier : {e}")  
  
# Exemple d'utilisation  
csv\_file = r"C:\Users\ayoub\Documents\PSID\accidents-corporels-de-la-circulation-millesime.csv"  
  
count\_empty\_cells(csv\_file)

## 2️⃣ Script - Comptage des lignes contenant au moins une cellule vide

**Ce script compte le nombre de lignes où au moins une cellule est vide.**

import pandas as pd  
def count\_rows\_with\_empty\_cells(csv\_file):  
 try:  
 # Lecture du fichier CSV avec encodage et séparation correcte  
 df = pd.read\_csv(csv\_file, encoding="utf-8-sig", sep=";")  
   
 # Comptage des lignes ayant au moins une cellule vide  
 empty\_rows\_count = df.isnull().any(axis=1).sum()  
   
 print(f"Nombre de lignes contenant au moins une cellule vide : {empty\_rows\_count}")  
 except Exception as e:  
 print(f"Erreur lors de la lecture du fichier : {e}")  
  
# Exemple d'utilisation  
csv\_file = r"C:\Users\ayoub\Documents\PSID\accidents-corporels-de-la-circulation-millesime.csv"  
  
count\_rows\_with\_empty\_cells(csv\_file)

## 3️⃣ Script - Suppression des lignes contenant des cellules vides

**Ce script supprime toutes les lignes contenant au moins une cellule vide et sauvegarde un nouveau fichier propre.**

import pandas as pd  
def remove\_empty\_rows(csv\_file, output\_file):  
 try:  
 # Lecture du fichier CSV en forçant le séparateur et l'encodage UTF-8  
 df = pd.read\_csv(csv\_file, encoding="utf-8-sig", sep=";")  
   
 # Suppression des lignes ayant au moins une cellule vide  
 df\_cleaned = df.dropna()  
   
 # Sauvegarde du fichier nettoyé en conservant le séparateur et l'encodage correct  
 df\_cleaned.to\_csv(output\_file, index=False, sep=";", encoding="utf-8-sig")  
   
 print(f"Fichier nettoyé enregistré dans : {output\_file}")  
 except Exception as e:  
 print(f"Erreur lors de la lecture du fichier : {e}")  
  
# Exemple d'utilisation  
csv\_file = r"C:\Users\ayoub\Documents\PSID\accidents-corporels-de-la-circulation-millesime.csv"  
output\_file = r"C:\Users\ayoub\Documents\PSID\accidents-corporels-de-la-circulation-millesime \_clean.csv"  
  
remove\_empty\_rows(csv\_file, output\_file)

## 📌 Conclusion

Avec ces scripts, il est possible d'analyser un fichier CSV, d'identifier les cellules vides, de compter les lignes concernées, puis de supprimer celles qui contiennent des valeurs manquantes. Cela permet de générer un fichier propre et exploitable sans perte d'informations importantes.