Explication des scripts

Challenge technique Data & Data, Manon Tessier

1. webscrapper

Script de récupération des données clés du site Panerai.com

La fonction webcrapper prend en entrée une url et renvoie en sortie un dataframe contenant les données sur les montres (nom, prix…) d'un certain pays et d'une certaine collection.

Limitations actuelles :

- le chrome driver ne marche que pour chrome version 114 et antérieur.
- les listes des sites à visiter et des montres à ne pas inclure ne sont pas dynamique → changement manuel nécessaire

Points d'amélioration possibles :

- Utiliser la page d'accueil de Panerai.com pour aller chercher les page d'accueil des pays concernés et descendre dynamiquement pour aller chercher les urls des collections

2. create_files_to_exploit

Script de création des dataframes avec traitement des données pour pouvoir générer des statistiques.

La fonction create_files_to_exploit prend en entrée 2 dataframes, l'un contenant les données des montres en 2021 et l'autre les données de 2023 et renvoit 4 dataframes correspondant au mélange 2021 et 2023 des données mais cette fois-ci répartie sur les 4 marchés (FR, JAP, UK, USA). Elle passe par 4 étapes : enlèvement des données incomplètes (prix manquant…), renommage des collections (le tiret est manquant ou non en fonction des pays), concaténation les 2 dataframes entrants, séparation par pays.

La fonction passe aussi par l'écriture des données des 4 marchés dans des fichiers excel pour une visualisation plus accessible.

Limitations actuelles :

- l'écriture des fichiers excel telle qu'elle existe actuellement écrase les anciens fichiers --> perte des anciennes données

Points d'amélioration possibles :

- les lignes new_prices['price] = ... génèrent des warnings --> syntaxe à changer
- rendre dynamique l'écriture des fichiers (création d'un dossier spécifique à chaque exécution + nom des fichiers qui incluent la date)

3. create_data_analysis

Script de création des statistiques sur les prix des montres.

La fonction create_stats_by_market prend en entrée un dataframe contenant les données sur les montres d'un certain marché et donne en sortie un autre dataframe contenant des statistiques telles que la moyenne des prix, le prix médian, le prix max... en 2021 et en 2023.

La fonction create_stats_by_market_and_collection fait strictement la même chose mais en filtrant en plus sur une collection particulière.

Points d'amélioration possibles :

- Les deux fonctions peuvent être regroupées en 1 seule en mettant une valeur à collection en entrée et un if... pour filtrer le dataframe derrière.

4. main

Script principal d'exécution des fonctions pour créer les stats.

Ce script regroupe tous les scripts précédents et suit 5 étapes:

1) Récupération des données 2021 via le fichier excel
2) Récupération des données 2023 via le scrapper
3) Création des fichiers data à exploiter (un fichier par marché)
4) Création des stats par marché puis par collection
5) Ecriture en excel des stats par marché = stats globale sur le marché + stats par collection (5 feuilles par fichier)

Limitations actuelles :

- Noms des fichiers généré de manière non dynamique --> écrasement des fichiers à

chaque exécution

Points d'amélioration possibles :

- Dynamiser le noms des fichiers avec la date et rangement par dossiers
- Trouver le moyen de faire une boucle pour exécuter les fonctions sur les marchés et les collections. Ici, le nombre de 4 marchés ne rend pas la tâche trop ardue mais cela peut vite devenir un problème si on prend tous les marchés de la marque par exemple

5. data_viz

Script de création des graphiques pour visualiser les données.

Ce script prend les fichiers de données Excel contenant des informations de prix sur les montres, utilise la bibliothèque pandas pour effectuer le nettoyage et la transformation des données, puis utilise pandas pour générer des graphiques avec matplotlib pour visualiser les statistiques de prix.

Limitations actuelles :

- Visualisation "à la main" en commentant / décommentant les lignes.