Informatique - TP 15 20 - 24 janvier 2025

Analyse de données - bibliothèque pandas - introduction

Code de partage avec Capytale : 54e6-1233372 **Corrigé**

1 Préparation

Dans un premier temps, nous allons importer un jeu de données.
Pour cela il faut copier le fichiers films.csv et le placer dans le même dossier que le fichier Python avec lequel vous travaillerez.
Puis dans le fichier Python (ou de même dans l'activité sur Capytale), on écrira et on exécutera :

```
import pandas
films=pandas.read_csv('films.csv', delimiter=';' ,encoding='utf8')
```

2 Exercice

1. Avec ce jeu de données, on pourra tester les commandes suivantes et donner leur fonction :

`films.head()`	Cette commande renvoie un en-tête du jeu de données : les noms de colonnes et les cinq premières lignes.
`films.sample(7)`	Renvoie un échantillon de n lignes (ici $n = 7$). Ces lignes sont choisies de manière aléatoire parmi les lignes du tableau.
`films.columns`	Renvoie les noms de toutes les colonnes.
`films.dtypes`	Renvoie le type de chaque donnée. Cela permet de connaitre les données quantitatives qui peuvent être de deux types : `int` (entiers) ou `float` (réels). Ici on en trouve trois : l'année, le code de région, le tonnage. Certaines variable ne le sont pas vraiment : ici on ne fera pas de traitement statistique sur les années ou les codes de région.
`films.describe()`	Renvoie une description des variables quantitatives en donnant les indicateurs statistiques suivants : nombre de lignes, moyenne, écart-type (std), minimum, maximum, médiane, premier et troisième quartiles.
`films.shape`	Renvoie le nombre de lignes et de colonnes sous la forme d'une liste de deux nombres. En tapant `dechet.shape[0]`, on obtient donc le nombre de lignes.
`films.nlargest(9,'revenue')`	Cette commande s'applique à une colonne qui contient des données quantitatives (il faut spécifier le nom de la colonne), et renvoie les n lignes (ici $n = 9$) contenant les plus grandes valeurs pour la variable choisie.
`films['release_year']`	Renvoie l'extrait du jeu de données constitué de la colonne `release_year`, sans l'en-tête de colonne mais toujours avec les indices de lignes (qui commencent à 0).
`films[['title','release_year']]`	Renvoie, comme la précédente, l'extrait constitué des deux colonnes, mais le double crochet permet de garder le format « table » de données en incluant les en-têtes de colonnes.
`films[(films['release_year'] ==2000) & (films['revenue'] >100000)]`	Ce type de commande illustre la possibilité d'extraire une partie du jeu de données en appliquant des conditions. Cette commande renvoie l'extrait de la table pour l'année 2000 et les revenus des films de plus de 1 000 000 de dollars. L'extrait contient toutes les colonnes (dont leurs noms en en-tête), il s'agit donc d'un filtre sur les lignes.

2. Trouver des commandes qui permettent de répondre aux questions suivantes
   - De quelle année date le film le plus ancien et quelle était sa langue originale ?

La commande `films.describe()` donne les valeurs minimales de chaque variable quantitative (dont l'année), mais ne donne pas le reste de la ligne.
On peut utiliser ici `films.nsmallest(1,'release_ year')`.

On trouve l'année 1916 pour laquelle un seul film est répertorié et dont la durée est de 197 minutes.

- Quel est le film de l'année 2011 le plus long ?

  De manière analogue à la dernière commande du tableau, on peut créer un extrait contenant les données uniquement pour l'année 2011 avec la commande :

  ```
  films_2011=films[films['release_year']==2011]
  ```

  puis on conclut avec `films_ 2011.nlargest(1,'runtime')`.
  On trouve le film *The Girl with the Dragon Tattoo*.

- Quel a été le revenu cumulé des films des années 1990 à 1999 (incluses) ? de 2000 à 2009 (incluses) ?

  On adapte la condition pour avoir les années souhaitées, et on peut également ajouter `['revenue']` pour ne garder que la colonne qui nous intéresse. Ensuite un `.sum()` à la fin ou `sum(...)` « autour » permet d'effectuer le cumul

  ```
  films[(films['release_year']<2000) & (films['release_year']>=1990)]['
      revenue'].sum()
  ```

  On trouve alors un résultat de 61 301 228 838 (dollars ?).
  De même, pour les années 2 000 à 2 009, on exécutera :

  ```
  films[(films['release_year']<2009) & (films['release_year']>=2000)]['
      revenue'].sum()
  ```

  On trouve cette fois un résultat de 135 527 862 403 (dollars ?).

- Combien de langues sont représentées ?

  On peut cette fois sélectionner uniquement la colonne des langues avec `films['original_language']` et en complétant avec .drop_duplicates(), on supprime les doublons. On peut créer une nouvelle table qui contient cet extrait et il ne reste alors plus qu'à compter le nombre de lignes, ce que l'on peut faire avec `.count()`

  ```
  extrait=films['original_language'].drop_duplicates()
  extrait.count()
  ```

  On trouve finalement 37 langues différentes.