NSI 1ère - Tables en csv - Introduction

QK

January 16, 2019

Chercher dans un csv, faire des statistiques

WARNING

Le fichier *nobels_propres.csv* N'EST PAS celui que vous avez obtenu à la fin de la partie précédente !

J'ai dû le modifier légèrement pour faire apparaître chaque lauréat ayant reçu plusieurs prix Nobel.

Ce problème aurait pu être résolu dans la partie précédente mais je préférais que cette difficulté ne fasse pas l'objet d'un exercice pénible et présentant peu d'intérêt pour vous.

Consignes

Etapes à suivre :

1. Afficher un résumé des 10 pays les plus représentés dans le fichiers *nobels2.csv*. On s'aidera d'un dictionnaire
2. Certains lauréats représentent plusieurs pays, d'autres n'en ont pas. Résoudre ce problème ,
    - en créant pour chaque lauréat ayant plusieurs pays une entrée par pays
    - en attribuant à chaque lauréat n'ayant pas de pays un pays "Sans pays d'origine"
3. Enlever la ligne d'entêtes
4. Trier le dictionnaire par odre décroissant des valeurs

5. Compter les genres 'male', 'female' Utiliser from `collections import Counter` pour dénombrer.

6. On s'intéresse aux questions suivante :

   Y-a-t-il une influence du type de prix sur le genre des lauréats ?

   A-t-on plus de femmes en littérature qu'en chimie ? En chimie que dans toutes les catégories réunies ?

   Présenter une liste qui indique en premier la proportion de femmes parmi tous les nobels puis ensuite les catégories par proportion de femmes décroissante

   On fera attention aux *organisations* qui n'ont *pas de genre...*

Correction

Pays les plus représentés

Afficher un résumé des 10 pays les plus représentés dans le fichiers *nobels*propres.csv_

On commence par créer un dictionnaire présentant le nombre de lauréats par pays.

Correction

```
import csv
csv_file = 'nobels_propres.csv'

dict_pays_simple = {}
with open(csv_file, newline='') as csvfile:
    nobels = csv.reader(csvfile, delimiter=',')
    for row in nobels:
        if row[4] in dict_pays_simple:
            dict_pays_simple[row[4]] += 1
        else:
            dict_pays_simple[row[4]] = 1

# print(dict_pays_simple)
```

Probleme : plusieurs pays, pas de pays

Certains lauréats représentent plusieurs pays, d'autres n'en ont pas. Ce sont généralement des organisations internationale comme la Croix Rouge.

Nous allons découper la liste des pays qu'ils représentent et les compter dans chacun d'entre eux. Attention, après le découpage d'un lauréat "France, Germany" on optient une liste : ["France", " Germany"] On doit s'assurer d'enlever l'espace initial du second pays de chaque candidat !

Les lauréats ne représentant aucun pays (pays = '') seront décomptés parmi les "Sans pays d'origine".

Autre probleme : la premiere ligne du fichier csv contient les entetes, on la vire avec next(object, None)

Correction

```
dict_pays = {}
with open(csv_file, newline='') as csvfile:
    nobels = csv.reader(csvfile, delimiter=',')
    # la première ligne du fichier contient les entêtes, on les zappe avec next(objet, None,
    next(nobels, None)
    for row in nobels:
        liste_pays_laureat = row[4].split(",")
        for pays in liste_pays_laureat:
            if pays == '':
                pays = 'Apatride'
            else:
                if pays[0] ==  ' ':
                    pays = pays[1:]
            if pays in dict_pays:
                dict_pays[pays] += 1
            else:
                dict_pays[pays] = 1

# print(dict_pays)
```

Tris

On trie le dictionnaire par ordre décroissant

Les dictionnaires ne sont pas triés en Python. Il existe 1001 façons le faire... ici nous cherchons simplement à afficher les premiers éléments. Le dictionnaire final étant petit, on se contente d'enlever les éléments de trop.

Correction

```
def denombrer_dict(dic, n):
    '''
```

```
    renvoie une liste de n elements [def, 20] comportant les elements les plus fréquents et
    param dic: un dictionnaire {"abc":10, "def":20} avec clé en string et valeur en entier
    param n: la longueur désirée de la liste
    return liste_classement: une liste
    '''
    liste_classement = []
    for w in sorted(dic, key=dic.get, reverse=True):
      liste_classement.append([w, dic[w] ])

    del liste_classement[n:]
    return liste_classement


print("\nNobels par pays : \n")
for position in denombrer_dict(dict_pays, 10):
    print(position)
```

Dénombrer avec *counter*

On reprend le fichier *nobels.csv*

Nous allons dénombrer en utilisant la méthode counter.

Les entêtes du fichier : *“annee”, “prix”, “nom”, “genre”, “pays”*

Correction

```
from collections import Counter

multiples = {}
with open(csv_file, newline='') as csvfile:
    nobels = csv.reader(csvfile, delimiter=',')
    # la première ligne du fichier contient les entêtes, on les zappe avec next(objet, None)
    next(nobels, None)
    multiples = [nobel[4] for nobel in nobels] # generator

result = Counter(multiples)
# print(result) # on retrouve les valeurs obtenues avant le traitement des lauréats ayant d

# print("\nNobels par pays : \n")
# for position in denombrer_dict(result, 10):
    # print(position)
```

Proportion de femmes par type de nobel.

On s'intéresse aux questions suivante : *y-a-t-il une influence du type de prix sur le genre des lauréats ?*

A-t-on plus de femmes en littérature qu'en chimie ? En chimie que dans toutes les catégories réunies ?

Consigne : présenter une liste qui indique en premier la proportion de femmes parmi tous les nobels puis ensuite les catégories par proportion de femmes décroissante

correction

```
csv_file = 'nobels_propres.csv'

with open(csv_file, newline='') as csvfile:
    reader = csv.reader(csvfile, delimiter=',')
    next(reader, None) # virer les headers
    genres = [nobel[3] for nobel in reader]

nobels_par_genre = Counter(genres)

print("\nNobels par genre :\n")
for position in denombrer_dict(nobels_par_genre, 3):
    print(position)
# les 22 prix nobels sans genre ont été attribués à des organisations (croix rouge, ligue i
```

Par catégorie

Recommençons avec les nobels de chaque catégorie

```
categories = ["Chemestry", "Economics", "Literature", "Medicine",
"Peace", 'Physics']
```

On sait maintenant qu'il n'y a que trois genres : "Male", "Female" et ""

Correction

```
genre_par_categories = {
                    "Chemestry":{"Male":0, "Female":0, "":0},
                    "Economics":{"Male":0,"Female":0,"":0},
                    "Literature":{"Male":0, "Female":0, "":0},
                    "Medicine":{"Male":0, "Female":0, "":0},
                    "Peace":{"Male":0, "Female":0, "":0},
```

```
                    "Physics":{"Male":0, "Female":0, "":0}
                      }

csv_file = "nobels_propres.csv"

with open(csv_file, newline='') as csvfile:
    reader = csv.reader(csvfile, delimiter=",")
    next(reader, None)
    for nobel in reader:
        # le mot "Physics" est encodé bizarrement et
        # 1H plus tard j'ai trouvé une solution de gangster
        # complètement tordue...
        if nobel[1] in categories:
            genre_par_categories [nobel[1]] [nobel[3]] += 1
        else:
            genre_par_categories[nobel[3]] = {"Male":0, "Female":0, "":0}
            genre_par_categories [nobel[1]] [nobel[3]] += 1
            # print(nobel)

total_nobels = nobels_par_genre["Male"] + nobels_par_genre[""] + nobels_par_genre["M"]
proportion_femme_total = (0.0+nobels_par_genre["Female"]) / total_nobels
proportion_femme_par_categorie = {"Total":proportion_femme_total}

for categorie in categories:
    somme = genre_par_categories[categorie]["Male"] +
     genre_par_categories[categorie][""] +
     genre_par_categories[categorie]["Female"]

    proportion_femme_par_categorie [categorie] = (0.0 +
     genre_par_categories[categorie]["Female"]) / somme

print("\nProportion de femmes par prix\n")
for position in denombrer_dict(proportion_femme_par_categorie, 10):
    print(position)
```