TP 9 – Analyse exploratoire de données avec Python

Pandas est une librairie Python qui permet de rendre le travail avec des données structurées à la fois facile et intuitif. Elle a pour objectif de fournir des outils d'analyse, de manipulation et de visualisation de données réelles en Python.

Pandas est bien adaptée à de nombreux types de données différents :

- Données tabulaires avec des colonnes de type hétérogène, comme dans un tableau SQL ou une feuille de calcul Excel ;
- Données de séries chronologiques ordonnées et non ordonnées (pas nécessairement à fréquence fixe) ;
- Données matricielles arbitraires (typographiées de manière homogène ou hétérogène) avec des étiquettes de lignes et de colonnes ;
- Toute autre forme d'ensemble de données observationnelles/statistiques.

Objectifs du TP : Sur des données expérimentales fournies (base de données de films), il s'agit de comprendre le contenu et la structure de ces données avec l'utilisation de la librairie `Pandas`. Plus précisément, vous pourrez explorer ces données, les transformer et les visualiser. La plupart des exemples en Python sont fournis, il s'agit de bien les comprendre, voire de tester par vous même d'autres opérations d'analyse et de manipulation. Certaines questions sont à chercher en regardant dans la documentation relative à `Pandas`.

Récupération des données

Données

MovieLens 1M Data Set contient les notes attribuées à des films par des utilisateurs du site Movielens. Les données sont fournies, mais peuvent être trouvées, si besoin, à l'adresse: `http://grouplens.org/datasets/movielens/`

Packages

```
import pandas as pd   #pour l'exploration de données
import numpy as np    #pour les opérations numériques
```

Questions

Lecture des données

Dans les questions qui suivent, lire le *readme* pour connaitre les différents fichiers de données et leur structuration (users.dat, ratings.dat et movies.dat).

1. Lire les données "users" dans un DataFrame Pandas et afficher les 5 premières valeurs.

```
unames = ['user_id', 'gender', 'age', 'occupation', 'zip']
users = pd.read_table('ml-1m/users.dat', sep='::', header=None,
names=unames, engine='python')
users.head()
```

2. Lire les données "ratings" dans un DataFrame Pandas et afficher les 10 premières valeurs.
3. Lire les données "movies" dans un DataFrame Pandas et afficher les 10 premières valeurs.
4. Fusionner les données des 3 fichiers dans un seul DataFrame.

```
data = pd.merge((pd.merge(users, ratings), movies)
data.head()
```

Exploration des données

Dans les exercices suivants vous suivrez attentivement le tutoriel qui explique les fonctionnalités de l'opération *groupby*, qui permet de *splitter* des objets, de les combiner ou d'appliquer une fonction. `https://www.tutorialspoint.com/python_pandas/python_pandas_groupby.htm`

1. Combien de films ont une note supérieure à 4.5 ? Existe-t-il une différence entre les hommes et les femmes?

```
np.sum(data.rating > 4.5)
np.sum(data.rating[data.gender == 'F']> 4.5)
np.sum(data.rating[data.gender == 'M'] > 4.5)
```

2. Même question en regardant les proportions : nombre de films notés par les femmes plus de 4.5 sur le nombre total de films notés par les femmes (idem pour les hommes).
3. Quels sont les 10 films dont la moyenne des notes est la plus haute ? L'idée est de regrouper par la fonction `groupby` les films dont la moyenne des notes est la plus haute. Vous irez au préalable chercher la documentation de la fonction `groupby`.

```
best_film = data.groupby('movie_id')['rating'].mean().nlargest(10)
print(best_film)
```

4. Combien de films ont une note médiane au-dessus de 4.5 parmi les hommes de plus de 30 ans ? et parmi les femmes de plus de 30 ans ?
   Indication. Vous pourrez partir de l'expression de la question précédente en filtrant au préalable les données de genre et d'âge, et utiliser la fonction `median`.
5. Dans cette question vous chercherez à produire les données de films dans lesquelles on calcule le nombre de notes et la moyenne des notes.
   (a) Pour cela vous créerez d'abord `d1` : données dans lesquelles on compte le nombre de notes de chacun des films (vous pourrez utiliser la fonction `count`. Vous afficherez les 5 premières valeurs.

(b) Puis vous créerez `d2` : données dans lesquelles on calcule la moyenne des notes de chacun des films.

(c) Enfin vous créerez `d3` en concaténant les deux informations : comptage des notes et moyenne des notes. Vous utiliserez la fonction `concat`, et `columns` pour afficher le type de valeurs affichées.

6. Dans cette question on définit les films "populaires" comme étant les films les mieux notés en moyenne parmi ceux qui ont été notés un certain nombre de fois. Ce nombre de fois est déterminé par un seuil donné. Vous créerez un ensemble de films "suffisamment" populaires. Pour cela vous repartirez de l'ensemble `d3` de la question précédente et garderez uniquement l'ensemble `d4` des films qui sont au-dessus d'un seuil de popularité donné (par exemple seuil = 30). Vous pourrez utiliser la fonction `sort_values` ainsi que le paramètre `ascending`.
Vous afficherez le titre des 2 films les plus populaires dans ce nouvel ensemble.

Visualisation des données

1. Afficher l'histogramme des notes de tous les films. Vous pourrez utiliser la fonction `hist`.

```
data.rating.hist(bins=5, align='left', range=[0, 6])
```

2. Afficher l'histogramme du nombre de notes reçues par chaque film (vous prendrez `bins = 10`.

3. Afficher l'histogramme des notes moyennes des films.

4. La distribution des notes dépend-elle du sexe?

Pour aller plus loin

Les deux questions qui suivent ont pour objet de comprendre les dictionnaires et fonctions `map` sous Pandas.

1. Utilisation des dictionnaires sous Pandas. Choisir un exemple de votre choix, du type :

```
df = pd.DataFrame({'col0': [1, 2, 3],
                   'col1': ['chat', 'chien','lapin'],

                   'col2': ['cat', 'dog','rabbit']},

                  index=['row1', 'row2', 'row3'])

dico = df.to_dict()
```

Affichez ce que représentent `df` et `dico`.

2. Application de `map` sur une série ou un dictionnaire sous Pandas. Soit la Série suivante :

```
s = pd.Series(['chat', 'chaton', 'chien', 'chiot', np.nan, 'lapin','rat'])
```

Appliquez les fonctions map de la façon suivante :

```
s1 = s.map({'chat': 'cat','chaton': 'kitten', 'chien': 'dog','chiot': 'puppy',
'lapin': 'rabbit'})

s2 = s.map('Je suis un {}'.format, na_action='ignore')
```

Affichez ce que représentent `s1` et `s2`.

3. En vous appuyant sur vos connaissances précédentes, expliquez la suite d'instructions suivantes en Pandas qui permet de tracer deux courbes.

```
map_id_to_count = data.groupby('movie_id')['rating'].count().to_dict()
data['movie_count'] = data['movie_id'].map(map_id_to_count)

data[data.movie_count >= seuil_pop].groupby('movie_id', axis=0)['rating'].
mean().plot(kind='kde', color='b')
data[data.movie_count <= seuil_pop].groupby('movie_id', axis=0)['rating'].
mean().plot(kind='kde', color='g')
```

4. Affichez un `scatter plot` des notes moyennes pour les hommes contre les notes des femmes pour chaque film (notés plus de 100 fois). Vous regarderez la documentation des opérations `pivot_table`, et l'affichage en mode `scatter` (`plot`(..., `kind='scatter',` ...))

5. Affichez un `scatter plot` des notes moyennes des hommes versus les femmes pour chaque film noté moins de 100 fois.

6. Quelle disparité de comportement observe-t-on entre les hommes et les femmes d'après la dernière figure ?