TP3 – Fouille de texte

Expressions régulières - Pandas

Pour chaque fiche de TP, créer un notebook jupyter (un notebook par binôme)

- La consigne est à insérer dans un cell de type texte
- Le codage de la réponse dans un cell de type code
- Ajouter un commentaire avant chaque ligne de code

Notebook à envoyer par mail : TP3-NomPrenom1-NomPrenom2.ipynb

1- Extraire des adresses emails

Le fichier 'mailInput.txt' contient des adresses emails. Il s'agit d'extraire ces adresses à l'aide d'expressions régulières (les plus simples possibles) et de les insérer dans un fichier 'mailOutput.txt', une par ligne.

a. Commencer par définir une fonction 'extractMail' qui prend en entrée un texte et retourne une liste d'adresses (tip : utiliser re.findall()).
b. Lire le contenu de mailInput.txt dans une variable 'inputContent'.
c. Ecrire le résultat de l'appel de la fonction extractMail sur inputContent dans le fichier résultat mailOutput.
d. Afficher le résultat.
e. Une adresse valide ne doit pas contenir plus d'un point '.' après le caractère '@' et au plus deux ou trois caractères après le '.' (ex. attachement.bonjour@monecole.tif'). A partir de la liste des résultats retournés par la fonction extractMail, créer une nouvelle liste 'valid' qui contient les adresses valides (tip : utiliser une liste comprehension et re.match).
f. Ecrire le résultat dans un fichier 'mailOutputValid.txt'.
g. Afficher la liste.

2- Tutoriel : manipuler des dataframes avec pandas de python

Suivre le tutoriel « Pandas Tutorial: DataFrames in Python » (https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python).

3- Extraire des dates

Le fichier « dates.xlsx » contient des textes avec des variations d'une date du mois de mars de la forme :

dd-03-yyyy	dd/03/yyyy	dd/03/yy
dd Mar yyyy	dd March yyyy	
Mar dd, yyyy	March dd, yyyy	

a. Utiliser la bibliothèque Pandas pour charger le fichier dates.xlsx dans un dataframe df :
```
import pandas as pd
df = pd.read_excel (r'dates.xlsx', header=None, names=['text'])
```

b. Trouver le nombre de caractères dans chaque texte de df['text'] (tip : df['text'].str. ..).
c. Trouver le nombre de tokens dans chaque texte de df['text'].
d. Trouver le lignes qui contiennent le mot 'Emacs' (tip : contains()).
e. Trouver combien de fois un chiffre (digit) apparaît dans chaque chaîne de caractère (tip : count(r' …')).
f. Trouver toutes les occurrences des chiffres (tip : findall(r' … ')).
g. Remplacer dans toutes les chaines 'Mar' ou 'March' par ' !!!' (tip : replace()).
h. Définir les expressions régulières nécessaires pour extraire l'ensemble des dates dans le texte. Stocker chaque regex dans une variable de type string.
i. Créer un nouveau dataframe à partir des dates extraites (tip : extractall()).