TP 7 - Mini-projet

Objectifs :

— s'initier à la gestion de projet
— découvrir une situation professionnelle réelle
— manipuler des données textuelles
— manipuler un tableau bidimensionnel

Le fichier `MSM.ods` contient 278 réponses à la question :

— Pouvez-vous citer des villes situées dans la Baie du Mont-Saint-Michel ?

Ces données ont été recueillies dans le cadre d'une enquête réalisée sur les réseaux sociaux par un service de développement local de tourisme (stage 2016). Le questionnaire comportait 17 questions. Chaque réponse est numérotée à l'aide d'un identifiant `id`. Il est important de conserver cet identifiant.

Une réponse peut contenir aucune, une ou plusieurs villes (question ouverte) ou des caractères inappropriés. Le séparateur n'est pas précisé et peut être une virgule, un espace ou tout autre caractère choisi par l'utilisateur au moment de la saisie. L'orthographe des noms de ville n'est pas forcément respectée :

id	villes
30	Granville Cancale Jullouville
61	Avranches - Pontorson - Ducey - Granville - Bréhal
169	saint malo cancale granville juliouville

1 Cahier des charges

Vous devez fournir un fichier dont les colonnes sont les villes et sur chaque ligne la valeur 1 ou 0 suivant que la ville a été citée ou pas dans la réponse :

id	Avranches	Bréhal	Cancale	Ducey	Granville	Jullouville	Pontorson	Saint-Malo
30	0	0	1	0	1	1	0	0
61	1	1	0	1	1	0	1	0
169	0	0	1	0	1	1	0	1

Ce type de fichier essentiellement constitué de 0 et de 1 est appelé fichier **bitmap**. La mise au format bitmap facilite certains traitements statistiques.

— Outils disponibles : le tableur **LibreOffice Calc** et **python3**.
— Pour l'exercice, **il est interdit de modifier le contenu du fichier manuellement**. Le traitement doit être automatisé au maximum et vous devrez décrire chacune des étapes.
— Travail en binôme.

Ne vous attendez pas à un résultat parfait. Par exemple la commune de Saint-Pair-sur-Mer ne figure pas dans le fichier source mais est pourtant citée 23 fois !

2 Ligne directrice

2.1 Constitution d'un dictionnaire

Dans un premier temps, on ne considère que les villes citées dont l'orthographe est exacte.

1. Récupérer la liste des 35885 communes françaises sur le site de l'INSEE
2. Construire un dictionnaire du nom des communes des deux départements concernés 35 et 50 : un nom de commune par ligne. Ceci a pour effet de limiter le nombre de recherches dans le dictionnaire (862 communes).

```
# le dictionnaire est vide
pour chaque ligne faire
    si le département est 35 ou 50 alors
        ajouter le nom officiel au dictionnaire
    fin si
fin pour
```

Remarque. L'encodage des fichiers peut poser un problème. Dans ce cas, utiliser le module **codecs**.

2.2 Les algorithmes

Il faut décider d'une méthode générale à adopter pour la constitution du fichier final. Par exemple :

```
pour chaque réponse faire
    mettre la valeur des colonnes à 0
    pour chaque ville citée faire
        mettre la valeur de la colonne correspondante à 1
    fin pour
fin pour
```

Cet algorithme suppose d'avoir fait une première lecture des données pour construire la liste des villes citées :

```
# la liste des villes citées est vide
pour chaque réponse faire
    pour chaque ville citée faire
        si la ville n'est pas dans la liste alors
            ajouter la ville à la liste
        fin si
    fin pour
fin pour
```

2.3 Séparateurs

Pour une réponse donnée, on obtient la liste des villes par décomposition suivant un séparateur (fonction **split**). Il y a plusieurs séparateurs possibles. Pour déterminer une liste des séparateurs candidats, on peut parcourir le fichier pour afficher les caractères qui ne sont pas des lettres. Il faut tenir compte du fait que le format d'une réponse peut contenir plusieurs caractères qui peuvent être un séparateur :

```
pour chaque séparateur s faire
    si la ligne contient s alors
```

```
        décomposer suivant s
        traiter
    fin si
fin si
```

2.4 Échantillon de test

Dans un premier temps, vous devez travailler avec un échantillon du fichier, par exemple seulement les dix premières lignes ou dix lignes au hasard.

2.5 Un premier résultat

L'idée est d'obtenir rapidement un résultat même s'il n'est pas parfait (méthode agile). Vous allez considérer un seul séparateur candidat (par exemple la virgule) :

1. constituer la liste des villes citées
2. coder 1 ou 0 suivant que la ville est citée ou pas

Reprendre les étapes précédentes avec l'ensemble des séparateurs candidats puis l'ensemble des données.

2.6 Recherche approximative

Dans le fichier source, les expressions 'saint malo', 'Saint Malo', 'St Malo' et 'st Malo' désigne la même ville dont l'orthographe officielle est Saint-Malo. Nous allons augmenter le taux de réponses traitées en acceptant les erreurs d'orthographe et en cherchant dans le dictionnaire des noms de communes le « meilleur » mot de substitution. C'est une méthode de recherche approximative (fuzzy search). Pour mesurer la distance entre deux mots, vous utiliserez la **distance de Levenshtein** : c'est le nombre minimum de lettres qu'il faut ajouter, supprimer ou substituer pour transformer un mot en un autre. Deux mots sont donc identiques si et seulement si leur distance est nulle. La distance entre chat et chien est 3. La distance entre chat et CHAT est 4 (sensibilité à la casse). Travailler avec des mots de même casse (tout en majuscules ou minuscules) améliorera la qualité de la recherche. On trouvera le code Python de la distance de Levenshtein ici :

https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings

Enfin la multiplicité des séparateurs possibles risque d'engendrer des substitutions totalement éloignées du contenu de la réponse. Par exemple :

— séparateur : espace
— réponse à traiter : `Pontorson,Beauvoir,Precey,Saint-James,`
— une seule ville : `['Pontorson,Beauvoir,Precey,Saint-James,']`
— substitution : `['Montjoie-Saint-Martin']`

alors que :

— séparateur : virgule
— réponse à traiter : `Pontorson,Beauvoir,Precey,Saint-James,`
— les villes : `['Pontorson','Beauvoir','Precey','Saint-James','']`
— substitutions : `['Pontorson','Beauvoir','Précey','Saint-James']`

Il faut donc établir un ordre de priorité. Par exemple si la réponse contient une virgule alors il est très probable que l'espace ne soit pas le séparateur :

```
pour chaque séparateur s faire
    si la ligne contient s alors
        décomposer suivant s
```

```
        traiter
        sortir # on ne teste pas d'autres séparateurs
    fin si
fin si
```

Il est fortement conseillé de contrôler la qualité du remplacement en faisant quelques tests (affichage de la ville et de sa substitution) pour définir les priorités avant d'appliquer le traitement.

3 Calendrier et évaluation

Remise sur Moodle avant le mercredi 14 décembre 12h d'une archive ZIP (`nom1_nom2.zip`) contenant

1. un **seul** fichier de script python :
   (a) nommage correct des variables
   (b) utilisation des fonctions
   (c) avec commentaires
2. le fichier bitmap (format CSV)
3. un rapport (format PDF)
   (a) précisant la qualité de vos données :
   — nombre de villes considérées (nombre de colonnes)
   — nombre de lignes ne contenant que des 0
   — nombre de 1 dans le fichier (compteur de villes)
   (b) contenant un diagramme en barre des effectifs pour les dix villes les plus citées
   (c) avec la réponse à la question : quelles sont les deux villes le plus souvent citées dans une même réponse (association en fouille de données) ?

La note de mini-projet compte pour moitié de la note du module.

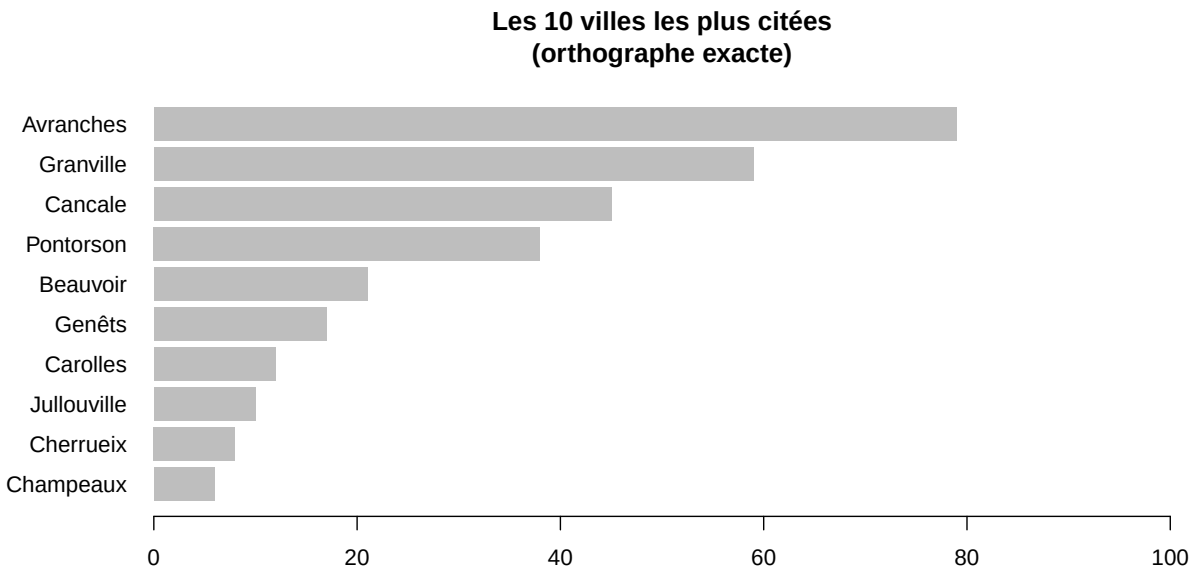

FIGURE 1 – 71% de réponses fiables

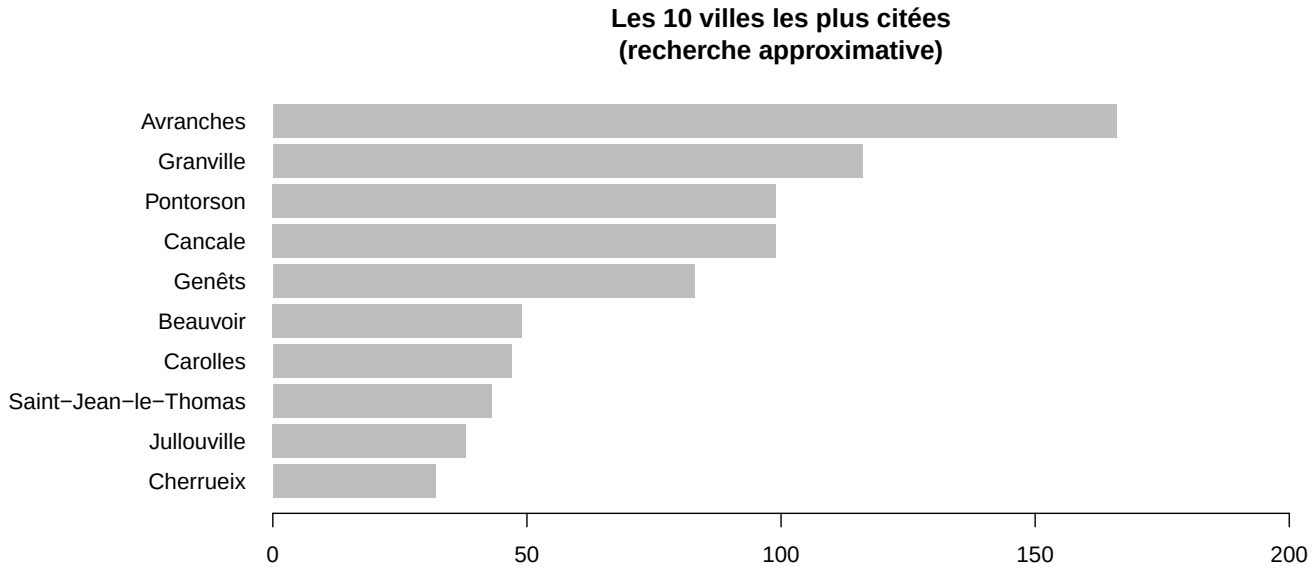

FIGURE 2 – 100% des réponses apportent une information mais des erreurs sont possibles