

TP I

- Les objectifs de ce travail sont :
 - d’explorer une problématique **très simple** de nettoyage de données, soit de déterminer le nombre de lignes et de colonnes dans un tableur en format CSV et de calculer le nombre d’entrées vides,
 - s’initier à l’emploi de bibliothèques externes, principalement les bibliothèque Python `argparse` et `csv`,
 - s’initier au format CSV (Comma-separated values),
 - faire un usage conforme aux conventions pour le traitement des options sur la ligne de commande.
- Vous devez donc développer un programme Python qui remplit les exigences suivantes.
 - Votre programme pourra être appelé de la ligne de commande et son comportement correspondra au message d’usage suivant.


```
usage: tp1.py [-h] [--delimiter DELIMITER] [--print] [--frequency] [--version]
             [infile]
```

Count empty entries in a CSV file.

positional arguments:

`infile` CSV file to process.

optional arguments:

`-h, --help` show this help message and exit
`--delimiter DELIMITER, -d DELIMITER`
 column delimiter. Should be used if autodetect (default) doesn’t work.
`--print, -p` print infile with one field on each line. Useful to understand infile’s layout
`--frequency, -f` print the total number of entries and number of non-empty entries for all columns
`--version, -v` show program’s version number and exit

- Plus précisément :
 - si l’argument `infile` est absent, l’entrée standard sera utilisée,
 - si l’option `--delimiter DELIMITER` est absente, votre programme devra utiliser les fonctions de la bibliothèque `csv` pour tenter de déterminer le format de façon automatique, sinon votre programme devra vérifier que l’argument `DELIMITER` est formé d’un seul caractère, qui sera considéré comme le séparateur des champs,
 - l’option `--print` affichera le contenu du fichier CSV, en indiquant le début d’une ligne et affichera chaque champ sur sa propre ligne.

Par exemple, un fichier contenant

```
a,b,c  
affichera  
row 1  
a  
b  
c
```

- l’option `--frequency` permettra d’afficher le nombre total d’entrées et le nombre d’entrée non-vides pour chaque colonnes. Les colonnes seront numérotée 0, 1, 2, ...
On aura donc, par exemple, qu’un fichier de deux colonnes dont la première ne contient qu’une seule entrée non-vide alors que la second en contient deux sera affiché de la façon suivante :
`no_entries= {0: 2, 1: 2} no_non_empty= {0: 1, 1: 2}`
- Le traitement du fichier CSV devra se faire en utilisant le plus possible bibliothèque `csv`.
- Le traitement des options de la ligne de commande devront se faire à l’aide de la bibliothèque `argparse`, ce qui vous permettra d’effectuer l’affichage d’usage et la lecture des arguments de façon uniforme et sans devoir tout reprogrammer par vous-même.
- Les critères de correction seront les suivants :
 1. Découpage des fonctionnalités en des fonctions simples, courtes et n’effectuant qu’une seule tâche (10 points),
 2. Documentation sous la forme de commentaires, incluant la chaîne de documentation (docstring) des fonctions (5 points),
 3. Les jeux de tests unitaires pour les fonctions que vous allez définir. Il est important de tester les cas limites et d’avoir une bonne couverture des cas représentatifs. Vous devez remettre un court document recopiant l’exécution de vos test dans IDLE où chaque test est brièvement décrit (cas testé et le résultat escompté) (5 points).
 4. La réalisation des fonctionnalités demandées (10 points).
- Quelques tests d’intégration seront fournis pour vous aider dans la réalisation de votre travail.