

# DM1\_ISN\_APB

November 1, 2019

## 1 Lecture de fichier

### 1.1 Lecture des 4 premières lignes du fichier

```
In [2]: SOURCE = 'fr-esr-apb_voeux-et-admissions.csv'
```

```
In [3]: f = open(SOURCE)
        for k in range(4):
            ligne = f.readline()
            print(ligne)
        f.close()
```

```
Session;Code UAI de l'établissement d'accueil;Libellé de l'établissement d'accueil;Code départem
2016;0720048L;Lycée Raphaël ELIZE;72;Sarthe;Nantes;Pays de la Loire;3_BTS;BTS - Services;Assista
2016;0720822C;Lycée Ste Catherine;72;Sarthe;Nantes;Pays de la Loire;Autre;Secteur social;Secteur
2016;07208CFA;CFA E.N. 72 - UFA Lycée Touchard Washington;72;Sarthe;Nantes;Pays de la Loire;3_BT
```

### 1.2 Index du champ *Filières de formations très détaillées* :

C'est l'index 9

```
In [5]: f = open(SOURCE)
        en_tete = f.readline()    #ligne des en-têtes
        champs_en_tete = en_tete.rstrip().split(';')
        print(champs_en_tete.index('Filières de formations très détaillées'))
        f.close()
```

### 1.3 Compter les lignes du fichier

```
In [3]: def nombre_lignes(fichier):  
        c = 0  
        f = open(fichier)  
        for ligne in f:  
            c = c + 1  
        f.close()  
        return c
```

```
In [4]: nombre_lignes(SOURCE)
```

```
Out[4]: 17754
```

## 2 Traitement d'une ligne

### 2.1 Découpage d'une ligne selon le caractère de séparation ';'

```
In [7]: def decoupageLigne(ligne, sep):  
        ligne = ligne.rstrip()  
        return ligne.split(sep)
```

### 2.2 Nombre de champs dans une ligne du fichier SOURCE

```
In [8]: f = open(SOURCE)  
        ligne = f.readline()  
        print(len(decoupageLigne(ligne, ';')))  
        f.close()
```

```
37
```

### 2.3 Nettoyage du caractère de fin de ligne

```
In [6]: f = open(SOURCE)  
        ligne = f.readline()  
        ligne = ligne.rstrip()  
        champs = ligne.split(';')  
        #on peut chaîner les opérations  
        #champs = ligne.rstrip().split(';')  
        print(champs[-1:])  
        f.close()
```

```
['% Candidats admis avec mention T.B au bac']
```

## 2.4 Liste de liste de champs par année

```
In [16]: def listeVoeuxAnnee(fichier, annee, sep):
        voeux = []
        f = open(fichier)
        f.readline() #sauter la ligne des en-tetes
        for ligne in f:
            champs = decoupageLigne(ligne, sep)
            if champs[0] == annee:
                voeux.append(champs)
        f.close()
        return voeux
```

```
In [8]: voeux_tous2016 = listeVoeuxAnnee(SOURCE, '2016', ';')
```

## 2.5 Nombre total de candidats admis

```
In [15]: def nombreAdmis(voeux):
        """Le nombre d'admis est dans le champ d'index 18"""
        s = 0
        for champs in voeux:
            val = champs[18]
            if val != 'inconnu':
                s = s + int(val)
        return s
```

```
In [10]: voeux_tous2016 = listeVoeuxAnnee(SOURCE, '2016', ';')
        len(voeux_tous2016)
```

```
Out[10]: 9004
```

```
In [11]: voeux_tous2017 = listeVoeuxAnnee(SOURCE, '2017', ';')
        len(voeux_tous2017)
```

```
Out[11]: 8749
```

```
In [12]: nombreAdmis(voeux_tous2016)
```

```
Out[12]: 532747
```

```
In [26]: def nombreAdmisFilles(voeux):
        """Le nombre d'admis filles est dans le champ d'index 19"""
        s = 0
        for champs in voeux:
            val = champs[19]
            if val != 'inconnu':
                s = s + int(val)
        return s
```

```
In [14]: nombreAdmisFilles(voeux_tous2016)
```

```
Out[14]: 284859
```

```
In [15]: #Pourcentage de filles parmi les admis  
print(nombreAdmisFilles(voeux_tous2016)/nombreAdmis(voeux_tous2016))
```

```
0.5346984591184933
```

### 3 Filtrage des lignes par champs

#### 3.1 Filtrage des lignes selon une condition

On récupère toutes les lignes concernant l'établissement dont le Code UAI (ou RNE) est passé en paramètre

```
In [13]: def filtreUai(voeux, codeUAI):  
        voeux_filtre = []  
        for v in voeux:  
            if v[1] == codeUAI:  
                voeux_filtre.append(v)  
        return voeux_filtre
```

```
In [17]: voeux_2016 = listeVoeuxAnnee(SOURCE, '2016',';')  
        voeux_Parc_2016 = filtreUai(voeux_2016, '0690026D')
```

```
In [18]: voeux_Parc_2016[0]
```

```
Out[18]: ['2016',  
          '0690026D',  
          'Lycée du Parc',  
          '69',  
          'Rhône',  
          'Lyon',  
          'Auvergne-Rhône-Alpes',  
          '4_CPGE',  
          'Classe préparatoire littéraire',  
          'Lettres',  
          '94',  
          '394',  
          '1110',  
          '864',  
          '235',  
          '188',  
          '110',  
          '87',  
          '93',  
          '74',
```

```
'49',  
'38',  
'8',  
'0',  
'1',  
'14',  
'78',  
'1',  
'1',  
'17',  
'18.2795698925',  
'1.0752688172',  
'8.60215053763',  
'0',  
'1.0752688172',  
'15.0537634409',  
'83.8709677419']
```

```
In [19]: len(voeux_Parc_2016)
```

```
Out[19]: 6
```

On récupère toutes les lignes concernant un voeu de valeur '4\_CPGE' dans le champ d'index 7 (8ème champ)

```
In [20]: def filtreCpge(voeux):  
        voeux_filtre = []  
        for v in voeux:  
            if v[7] == '4_CPGE':  
                voeux_filtre.append(v)  
        return voeux_filtre
```

```
In [21]: voeux_2016 = listeVoeuxAnnee(SOURCE, '2016',';')  
        voeux_cpge_2016 = filtreCpge(voeux_2016)
```

```
In [22]: len(voeux_cpge_2016)
```

```
Out[22]: 831
```

Pourcentage de filles parmi les admis en CPGE

```
In [23]: print(nombreAdmisFilles(voeux_cpge_2016)/nombreAdmis(voeux_cpge_2016))
```

```
0.46234610802223985
```

### 3.2 Filtrage des lignes selon deux conditions

```
In [19]: def filtreCpgeDepartement(voeux, departement):
        voeux_filtre = []
        for v in voeux:
            if v[7] == '4_CPGE' and v[3] == departement:
                voeux_filtre.append(v)
        return voeux_filtre

In [22]: voeux_cpge_rhone_2016 = filtreCpgeDepartement(voeux_2016, '69')

In [23]: voeux_cpge_rhone_2016[0]

Out[23]: ['2016',
          '0690522T',
          'Lycée Aux Lazaristes',
          '69',
          'Rhône',
          'Lyon',
          'Auvergne-Rhône-Alpes',
          '4_CPGE',
          'Classe préparatoire scientifique',
          'MPSI',
          '45',
          '168',
          '1075',
          '302',
          '228',
          '49',
          '47',
          '12',
          '45',
          '11',
          '25',
          '4',
          '0',
          '0',
          '0',
          '0',
          '45',
          '2',
          '0',
          '17',
          '37.77777777778',
          '4.444444444444',
          '0',
          '0',
          '0',
          '0',
          '100']
```

```
In [24]: voeux_cpge_2016 = filtreCpge(voeux_2016)
```

```
In [27]: print(nombreAdmisFilles(voeux_cpge_2016)/nombreAdmis(voeux_cpge_2016))
```

```
0.46234610802223985
```

```
In [29]: voeux_Parc_2016 = filtreUai(voeux_cpge_2016, '0690026D')
```

```
In [30]: #Pourcentage des voeux du lycée du Parc par rapport à l'ensemble des voeux CPGE du Rhon  
print(nombreAdmis(voeux_Parc_2016 )/nombreAdmis(voeux_cpge_rhone_2016 ))
```

```
0.27071072883657765
```

### 3.3 Filtrage générique sur 1 champ

```
In [31]: def filtre1champ(voeux, indexchamp, valchamp):  
    t = []  
    for v in voeux:  
        if v[indexchamp] == valchamp:  
            t.append(v)  
    return t
```

```
In [32]: filtre1champ(voeux_2016, 1, '0690026D') == filtreUai(voeux_2016, '0690026D')
```

```
Out[32]: True
```

### 3.4 Filtrage générique sur n champs

```
In [33]: def filtreNchamps(voeux, liste_indexchamp, liste_valchamp):  
    t = []  
    for v in voeux:  
        n = len(liste_indexchamp)  
        match = True  
        k = 0  
        while k < n and match:  
            indexchamp = liste_indexchamp[k]  
            valchamp = liste_valchamp[k]  
            if v[indexchamp] != valchamp:  
                match = False  
            else:  
                k = k + 1  
        if k == n:  
            t.append(v)  
    return t
```

```
In [34]: filtreNchamps(voeux_2016, [3,7], ['69','4_CPGE']) == filtreCpgeDepartement(voeux_2016,
```

```
Out[34]: True
```

## 4 Tri

### 4.1 Classement des vœux CPGE 2016 dans l'ordre croissant du taux de boursiers

```
In [28]: voeux_cpge_2016 = filtreCpge(voeux_2016)
```

```
def clefTri(v):  
    taux_boursier = v[32]  
    if taux_boursier == '':  
        taux_boursier = 0  
    return float(taux_boursier)
```

```
tri_boursier_croissant_2016 = sorted(voeux_cpge_2016, key = clefTri)
```

```
In [29]: tri_boursier_croissant_2016[:1]
```

```
Out[29]: [['2016',  
            '0920904H',  
            'Lycée Sainte Marie',  
            '92',  
            'Hauts-de-Seine',  
            'Versailles',  
            'Île-de-France',  
            '4_CPGE',  
            'Classe préparatoire scientifique',  
            'PCSI',  
            '48',  
            '201',  
            '269',  
            '64',  
            '13',  
            '6',  
            '49',  
            '10',  
            '44',  
            '9',  
            '7',  
            '4',  
            '0',  
            '1',  
            '10',  
            '21',  
            '12',  
            '5',  
            '2',  
            '22',  
            '50',  
            '11.3636363636',  
            '0',
```



```
'2.27272727273',
'22.7272727273',
'47.7272727273',
'27.2727272727']]
```

## 4.2 Classement des vœux CPGE 2016 dans l'ordre décroissant du taux de boursiers

```
In [33]: tri_boursier_decroissant_2016 = sorted(voeux_cpge_2016, key = clefTri, reverse = True)

def affichage_tri(voeux, nb lignes = 5):
    print('{:<5} {:<40} {:<10} {:<40} {:<20}'.format('Rang', 'Lycée', 'Département', 'Filière', 'Taux de boursiers'))
    nb lignes = min(len(voeux), nb lignes)
    for k in range(nb lignes):
        v = voeux[k]
        print('{:<5} {:<40} {:<10} {:<40} {:<20}'.format(k + 1, v[2][:40], v[3], v[9][:40], v[10]))

In [34]: affichage_tri(tri_boursier_decroissant_2016)
```

Rang	Lycée	Département	Filière
1	Lycée Paul Eluard	93	PCSI
2	Lycée Baimbridge	971	ECT - Option technologique
3	Lycée Amiral Pierre Bouvet	974	TSI
4	Lycée Le Corbusier	93	PTSI
5	Lycée Olympe De Gouges	93	ECE - Option économique

## 4.3 Tri des vœux APB de l'année 2016 par ordre décroissant du taux de boursier admis puis par ordre alphabétique croissant de la filière (champ d'index 9)

```
In [35]: #Stabilité du tri

def clefTri2(v):
    return v[9]

tri_boursier_decroissant_2016_filiere = sorted(tri_boursier_decroissant_2016, key = clefTri2)
affichage_tri(tri_boursier_decroissant_2016_filiere)
```

Rang	Lycée	Département	Filière
1	Lycée Alphonse DAUDET	30	B/L - Lettres et sciences sociales
2	Lycée Faiderbe	59	B/L - Lettres et sciences sociales
3	Lycée Albert Chatelet	59	B/L - Lettres et sciences sociales
4	Lycée HENRI IV	34	B/L - Lettres et sciences sociales
5	Lycée Pothier	45	B/L - Lettres et sciences sociales

## 4.4 Fonction filtreFiliere

```
In [37]: def filtreFiliere(voeux, filiere):
    voeux_filtre = []
```

```

for v in voeux:
    if v[7] == filiere:
        voeux_filtre.append(v)
return voeux_filtre

```

```
In [38]: tri_boursier_decroissant_MPSI = filtreFiliere(tri_boursier_decroissant_2016, 'MPSI')
```

#### 4.5 Séparation des tris par taux décroissant d'admis boursiers par filière puis écriture de chaque liste triée dans un fichier

```
In [39]: def tri_par_filiere(t):
    """Retourne une liste de liste, chaque sous-liste contient les voeux d'une filière
    triée par taux d'admis boursiers décroissants.
    La liste t prise en argument est la liste de tous les voeux CPGE triés par
    taux d'admis boursiers décroissants puis par filière."""
    filiere_courante = t[0][9]
    tri_filiere = [t[0]]
    tous_tris_filieres = [tri_filiere]
    for k in range(1, len(t)):
        filiere = t[k][9]
        if filiere == filiere_courante:
            tri_filiere.append(t[k])
        else:
            tous_tris_filieres.append(tri_filiere)
            filiere_courante = filiere
            tri_filiere = [t[k]]
    tous_tris_filieres.append(tri_filiere)
    return tous_tris_filieres

```

```
In [41]: len(tri_par_filiere(tri_boursier_decroissant_2016_filiere))
```

```
Out[41]: 19
```

```
In [44]: tri_par_filiere(tri_boursier_decroissant_2016_filiere)[0][:1]
```

```
Out[44]: [['2016',
            '0300021K',
            'Lycée Alphonse DAUDET',
            '30',
            'Gard',
            'Montpellier',
            'Occitanie',
            '4_CPGE',
            'Classe préparatoire littéraire',
            'B/L - Lettres et sciences sociales',
            '48',
            '207',
            '478',
            '315',
```

```

'48',
'26',
'59',
'39',
'46',
'29',
'11',
'7',
'14',
'0',
'2',
'22',
'22',
'4',
'3',
'26',
'56.5217391304',
'8.69565217391',
'30.4347826087',
'0',
'4.34782608696',
'47.8260869565',
'47.8260869565']]

```

```

In [42]: def ecrire_fichier(voeux, fichier):
          f = open(fichier, 'w')
          f.write('Lycée;Département;Filière;Taux de boursiers admis\n')
          for k in range(len(voeux)):
              v = voeux[k]
              f.write(';'.join([v[2], v[3], v[9], v[32]]) + '\n')
          f.close()

In [43]: for tri_filiere in tri_par_filiere(tri_boursier_decroissant_2016_filiere):
          filiere = tri_filiere[0][9]
          ecrire_fichier(tri_filiere, 'tri_boursiers_'+filiere.replace("/", "-")+'.csv')

```