

Vérifier le résultat des élections

Michel Billaud (michel.billaud@laposte.net)

27 avril 2022

Table des matières

1	Résumé	2
2	Récupération des données	2
3	Analyse du fichier	2
3.1	Les colonnes	3
3.2	Les données	3
4	Communes ou bureaux ?	4
5	Le programme de totalisation	4
5.1	Outils de base	4
5.2	Phase 1 : Lecture des données	4
5.2.1	Algorithme	4
5.2.2	Programme Python (phase 1)	5
5.2.3	Exécution du test	5
5.3	Phase 2 : lecture + comptabilisation	5
5.3.1	Algorithme	5
5.3.2	Code Python	6
5.3.3	Exécution	7
5.4	Programme final	7
5.4.1	Code	7
5.4.2	Exécution	8
5.4.3	Comparaison des résultats	8



Ce texte fait partie d'une petite collection de notes mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 2.0 France.

- Les notes sont publiées dans <https://www.mbillaud.fr/notes/>
- Sources dans <https://github.com/MichelBillaud/notes-diverses>

1 Résumé

Sur les réseaux sociaux, on trouve régulièrement des messages promouvant l’usage des machines à voter, ou du vote en ligne.

Pour faire bref, je suis contre les deux, mais ce n’est pas le point. L’objet de cette note c’est de réfuter l’affirmation à propos du vote “papier” :

Ok, on peut surveiller le vote et le dépouillement (en étant présent dans les bureaux) , mais pour la totalisation nationale, on ne peut pas faire confiance.

en montrant

- où on peut obtenir les données de vote des bureaux
- comment vérifier la totalisation.

Su l’exemple de la présidentielle 2022.

Pour ça, on verra comment

- analyser les données du fichier qu’on récupère au format CSV
- écrire un petit programme Python qui fait la totalisation.

Pourquoi Python ? Parce que c’est un langage de programmation

- à portée du non-professionnel (il y a une initiation à Python au Lycée, voire au collège)
- qui va assez bien pour faire ça. C’est un programme “one-shot” dont on ne se resservira pas¹, la vitesse de calcul n’a pas d’importance.

2 Récupération des données

Une recherche Google, “open data vote présidentielle 2022” donne comme second résultat² le lien <https://www.data.gouv.fr/fr/datasets/resultats-du-premier-tour-de-lelection-presidentielle-2022-par-commune-et-par-departement/>

Puisqu’on tient à vérifier les totaux, voyons grand : on part du fichier `presidentielle-2022-t1-resultats-par-commune.csv` qui pèse 104Mo et a été télé-chargé 2905 fois à ce jour.

En cliquant sur le lien <https://www.data.gouv.fr/fr/datasets/r/54782507-e795-4f9d-aa70-ed06feba22e3> (bouton bleu à droite), ça nous ramène un fichier qui s’appelle en fait `04-resultats-par-commune.csv`.

On le place dans un répertoire de travail, où il sera rejoint plus tard par le programme Python.

3 Analyse du fichier

C’est un fichier CSV classique, qui présente les données sous forme de table. On peut l’importer avec un tableau, sans problème.

¹pour comptabiliser d’autres élections, il faudra le modifier un peu, chaque élection donnant lieu à une présentation légèrement différente des données dans le fichier.

²Le premier donne les résultats pour Paris uniquement.

3.1 Les colonnes

La **première ligne** fournit les noms des colonnes, et permet aussi de voir que les données d’une ligne sont séparées par des virgules

C’est une très longue ligne, pour la lisibilité, elle est coupée en lignes de 5 champs ci-dessous

```
url,reg_code,reg_code_3,reg_name,dep_code,
dep_code_min,dep_code_3,dep_name,commune_code,commune_name,
num_tour,inscrits_nb,abstention_nb,abstention_pourc,votants_nb,
blancs_nb,blancs_pourc_inscrits,blancs_pourc_votants,nuls_nb,nuls_pourc_inscrits,
nuls_pourc_votants,exprimés_nb,exprimés_pourc_inscrits,exprimés_pourc_votants,cand_num_panneau,
cand_nom,cand_prenom,cand_civil,cand_nb_voix,cand_rapport_exprim,
cand_rapport_inscrits
```

3.2 Les données

En regardant les **deux premières lignes de données**

```
https://www.resultats-elections.interieur.gouv.fr/telechargements/PR2022/resultatsT1/084/001/001com.xm
Rhône-Alpes,01,
01,001,Ain,001,L’Abergement-Clémenciat,
1,645,108,16.74,537,
16,2.48,2.98,1,0.16,
0.19,520,80.62,96.83,1,
ARTHAUD,Nathalie,Mme,3,0.58,
0.47
```

```
https://www.resultats-elections.interieur.gouv.fr/telechargements/PR2022/resultatsT1/084/001/001com.xm
Rhône-Alpes,01,
01,001,Ain,001,L’Abergement-Clémenciat,
1,645,108,16.74,537,
16,2.48,2.98,1,0.16,
0.19,520,80.62,96.83,2,
ROUSSEL,Fabien,M.,6,1.15,
0.93
```

on constate que chaque ligne montre le nombre de voix obtenues par un candidat dans une commune. Et que les lignes reprennent les informations de la commune (identification, nombre d’inscrits etc.)

En regardant la fin du fichier, on voit que les données vont jusqu’à la dernière ligne.

Bref, on va simplement faire les totaux de la colonne “**cand_nb_voix**” en les différenciant sur le “**cand_nom**”³, à partir de la seconde ligne. Ce sont les colonnes 26 et 29 respectivement.

³par chance, on n’a pas d’homonymes à la présidentielle. On pourrait aussi utiliser le numéro de panneau, qui est attribué nationalement, pour identifier les candidats.

4 Communes ou bureaux ?

Une autre chose : ce sont bien des résultats **par commune**. Le fichier comporte 420961 lignes. 360000 communes à peu près, multiplié par 12 candidats, ça fait 432000, qui est de cet ordre (les lignes avec 0 voix n'y figurent pas).

Si on sait se servir des commandes, on peut vérifier qu'il n'y a que 12 lignes pour Bordeaux⁴, par exemple

```
$ grep ,Bordeaux, Calculs-Elections/04-resultats-par-commune.csv | wc -l
12
```

donc il s'agit bien de chiffres par commune, **pas par bureau**. Autre moyen de vérifier : regarder avec un tableur.

5 Le programme de totalisation

L'objectif est de totaliser, selon le nom des candidats, les voix qu'ils ont obtenues.

Ce programme va être développé par étapes, pour vérifier plus facilement qu'il n'y a pas d'erreurs. Il sera testé sur un échantillon (un certain nombre de lignes du fichier).

5.1 Outils de base

Pour faire ce travail on va employer

- le module `csv` de Python pour lire les données,
- un **dictionnaire** pour cumuler les votes de chaque candidat.

5.2 Phase 1 : Lecture des données

La première étape consiste à lire le fichier ligne par ligne et isoler, puis afficher, les noms des candidats et leur nombre de voix.

5.2.1 Algorithme

```
limite = 5                # nb de lignes pour le test
pour chaque ligne:
    si c'est la ligne 1:
        rien
    sinon si c'est la ligne limite + 1:
        sortir de la boucle
    sinon
        extraire et afficher le nom et le nombre de voix
```

Normalement, il devrait s'afficher limite-1 résultats puisque la ligne 1 est ignorée.

⁴Ville assez grande pour avoir plusieurs bureaux, et assez de gens qui votent n'importe que quoi pour qu'il soit très probable que chaque candidat a obtenu au moins une voix.

5.2.2 Programme Python (phase 1)

L'exécution de ce code

```
#
# Phase 1 : vérifier qu'on arrive à extraire
# correctement les noms et nombre de voix obtenus

import csv

nom_fichier = '04-resultats-par-commune.csv'
col_nom      = 26-1
col_nb_voix  = 29-1

limite_ligne = 5

with open(nom_fichier) as fichier:
    lecteur = csv.reader(fichier)
    numero_ligne = 0
    for ligne in lecteur:
        numero_ligne += 1
        if numero_ligne > limite_ligne:
            break
        if numero_ligne > 1:
            print ("nom =", ligne[col_nom], "voix =", ligne[col_nb_voix])
```

Notes :

- rappel : le tableau `ligne` est indicé à partir de 0, la 26ième colonne est à l'indice 26-1.
- la présentation des 3 cas dans la boucle a été modifiée pour réduire le code.

5.2.3 Exécution du test

L'exécution produit l'affichage

```
nom = ARTHAUD voix = 3
nom = ROUSSEL voix = 6
nom = MACRON voix = 150
nom = LASSALLE voix = 18
```

dont on peut vérifier (avec un tableur) qu'il correspond bien aux 5 premières lignes du fichier.

5.3 Phase 2 : lecture + comptabilisation

Pour la comptabilisation, on totalise les voix de chaque candidat

5.3.1 Algorithme

Pour tester il faudra avoir au moins deux communes, une vingtaine de lignes devraient suffire

```

limite = 20                # nb de lignes pour le test
totaux_voix = { }         # dictionnaire vide

pour chaque ligne:
    si c'est la ligne 1:
        rien
    sinon si c'est la ligne limite + 1:
        sortir de la boucle
    sinon
        extraire et afficher le nom et le nombre de voix
        ajouter nombre de voix à totaux_voix[nom]
        afficher totaux_voix
afficher total_voix (final)

```

5.3.2 Code Python

Une grande partie du code vient du programme précédent :

```

#
# Phase 2 : extraire les noms et nombre de voix obtenus
# vérifier le cumul

import csv

nom_fichier = '04-resultats-par-commune.csv'
col_nom      = 26-1
col_nb_voix  = 29-1

limite_ligne = 20
totaux_voix = {}

with open(nom_fichier) as fichier:
    lecteur = csv.reader(fichier)
    numero_ligne = 0
    for ligne in lecteur:
        numero_ligne += 1
        if numero_ligne > limite_ligne:
            break
        if numero_ligne > 1:
            nom = ligne[col_nom]
            nb_voix = int(ligne[col_nb_voix])
            print ("nom =", nom, "voix =", nb_voix)
            totaux_voix[nom] = totaux_voix.setdefault(nom, 0) + nb_voix
            print ("=", totaux_voix)
print ("totaux", totaux_voix)

```

Notes :

- les variables intermédiaires `nom` et `nb_voix` sont introduites pour ne pas avoir à répéter des sous-expressions ;
- la lecture du fichier CSV retourne un tableau de chaînes. Le nombre de

voix doit etre converti en entier :

- pour le cumul, on utilise une structure de dictionnaire, tableau de nombre indexé par les noms. La méthode `setdefault` est employée pour retourner la valeur présente pour un nom présent, et 0 si il est absent ;
- les affichages intermédiaires permettent de suivre ce qui se passe.

5.3.3 Exécution

Avec 20 lignes, on obtient **au début** :

```
nom = ARTHAUD voix = 3
= {'ARTHAUD': 3}
nom = ROUSSEL voix = 6
= {'ARTHAUD': 3, 'ROUSSEL': 6}
nom = MACRON voix = 150
= {'ARTHAUD': 3, 'ROUSSEL': 6, 'MACRON': 150}
...
```

qui montre le remplissage avec des noms qui n'ont pas encore été rencontrés.

À la fin, les dernières lignes

```
...
= {'ARTHAUD': 5, 'ROUSSEL': 13, 'MACRON': 200, 'LASSALLE': 28, 'LE PEN': 172, 'ZEMMOUR': 5,
  'MÉLENCHON': 66, 'HIDALGO': 5, 'JADOT': 30, 'PÉCRESE': 26, 'POUTOU': 3, 'DUPONT-AIGNAN': 5}
nom = MÉLENCHON voix = 41
= {'ARTHAUD': 5, 'ROUSSEL': 13, 'MACRON': 200, 'LASSALLE': 28, 'LE PEN': 172, 'ZEMMOUR': 5,
  'MÉLENCHON': 107, 'HIDALGO': 5, 'JADOT': 30, 'PÉCRESE': 26, 'POUTOU': 3, 'DUPONT-AIGNAN': 5}
totaux {'ARTHAUD': 5, 'ROUSSEL': 13, 'MACRON': 200, 'LASSALLE': 28, 'LE PEN': 172, 'ZEMMOUR': 5,
  'MÉLENCHON': 107, 'HIDALGO': 5, 'JADOT': 30, 'PÉCRESE': 26, 'POUTOU': 3, 'DUPONT-AIGNAN': 5}
```

font voir l'ajout de 41 voix au score de Mélenchon, qui passe de 66 à 107.

5.4 Programme final

Pour le programme final, on retire

- la limitation du traitement aux premières lignes,
- les affichages intermédiaires,

et on soigne la présentation du résultat.

5.4.1 Code

```
#
# Totalisation des voix au premier tour de l'élection présidentielle 2022
#

import csv

nom_fichier = '04-resultats-par-commune.csv'
col_nom      = 26-1
col_nb_voix  = 29-1
```

```

totaux_voix = {}

print ("= totaux voix premier tour présidentielle 2022")

with open(nom_fichier) as fichier:
    lecteur = csv.reader(fichier)
    numero_ligne = 0
    for ligne in lecteur:
        numero_ligne += 1
        if numero_ligne > 1:
            nom = ligne[col_nom]
            nb_voix = int(ligne[col_nb_voix])
            totaux_voix[nom] = totaux_voix.setdefault(nom, 0) + nb_voix

# affichage final
for (nom, nb_voix) in totaux_voix.items():
    print ('{:10d} {}'.format(nb_voix, nom))

```

5.4.2 Exécution

L'exécution, qui porte maintenant sur l'ensemble des données, ne prend que quelques secondes. On voit s'afficher :

```

= totaux voix premier tour présidentielle 2022
  200662 ARTHAUD
  827497 ROUSSEL
10087245 MACRON
 1116933 LASSALLE
  8261595 LE PEN
 2585422 ZEMMOUR
 8104784 MÉLENCHON
  635705 HIDALGO
 1700458 JADOT
 1751714 PÉCRESSE
 274938 POUTOU
 735698 DUPONT-AIGNAN

```

Avec un effort, on pourrait classer par ordre alphabétique, ou par score, selon les besoins. Le besoin n'étant pas précisé, il est jugé inutile de s'en occuper.

5.4.3 Comparaison des résultats

Ces chiffres diffèrent un petit peu https://fr.wikipedia.org/wiki/%C3%89lection_pr%C3%A9sidentielle_fran%C3%A7aise_de_2022#R%C3%A9sultats_nationaux_des_résultats_nationaux_affichés_sur_Wikipedia,_par_exemple.

Possibilités

- erreur dans le programme
- données incorrectes
- résultats invalidés partiellement
- complot international

- ...

Il est assez probable que les chiffres de Wikipédia, ainsi que l'annonce officielle des résultats au soir du premier tour, aient été établis à partir de données incomplètes.

Rien d'anormal : il faut en réalité quelques jours pour que le Conseil Constitutionnel se prononce sur la validité des résultats dans certains bureaux litigieux.