

# DOCUMENTATION : Qui sera l'heureux élu ?

12 mai 2024

---

## Table des matières

<b>1</b>	<b>Description du projet</b>	<b>2</b>
<b>2</b>	<b>Les différents modes de scrutin</b>	<b>3</b>
2.1	Le scrutin uninominal majoritaire à un tour . . . . .	3
2.1.1	Fonctionnement et pays utilisant ce scrutin . . . . .	3
2.1.2	Avantages et inconvénients . . . . .	3
2.2	Le scrutin uninominal majoritaire à deux tours . . . . .	3
2.2.1	Fonctionnement et pays utilisant ce scrutin . . . . .	3
2.2.2	Avantages et inconvénients . . . . .	3
2.3	Le scrutin par éliminations successives . . . . .	4
2.3.1	Fonctionnement et pays utilisant ce scrutin . . . . .	4
2.3.2	Avantages et inconvénients . . . . .	4
2.4	Le scrutin de Condorcet . . . . .	4
2.4.1	Fonctionnement et pays utilisant ce scrutin . . . . .	4
2.4.2	Avantages et inconvénients . . . . .	4
2.5	Le scrutin de Borda . . . . .	5
2.5.1	Fonctionnement et pays utilisant ce scrutin . . . . .	5
2.5.2	Avantages et inconvénients . . . . .	5
<b>3</b>	<b>Implémentation algorithmique en Python 🍄</b>	<b>5</b>
3.1	Les pré-requis . . . . .	5
3.2	Les fonctions . . . . .	5
3.2.1	La fonction élection . . . . .	5
3.2.2	Exemple d'algorithmes de modes de scrutins : Scrutin uninominal majoritaire à deux tours ; Scrutin par éliminations successives	6
3.2.3	La fonction générant des histogrammes . . . . .	7
<b>4</b>	<b>Annexe</b>	<b>9</b>
4.1	Un fichier contenant 50 classements de votants menant à un candidat élu différent pour chaque scrutin, soit 5 candidats élus . . . . .	9
4.2	Quelques histogrammes . . . . .	10
<b>5</b>	<b>Crédits</b>	<b>10</b>

---

# Introduction

## 1 Description du projet

Alors que nous nous rapprochons pas à pas des prochaines **élections** qui auront lieu en avril 2022, beaucoup de lycéens paraissent désintéressés par la question du vote, qui est pourtant essentielle pour nous, futurs électeurs. Au cours des dernières années de nombreux événements ont affecté le monde en prenant un tournant politique nous poussant à remettre en question l'importance de notre rôle en tant que citoyen et celle de nos droits civiques.

« **Qui sera l'heureux élu ?** » est un projet qui a pour objectif, en se basant sur des modes de scrutin existants, de créer des algorithmes copiant ces processus, en **Python** ♣, afin de visualiser les issues des votes pour savoir quel candidat serait élu selon les modes de vote.

**Alors selon vous quel processus électoral est le plus juste ?**



## 2 Les différents modes de scrutin

### 2.1 Le scrutin uninominal majoritaire à un tour

#### 2.1.1 Fonctionnement et pays utilisant ce scrutin

Le **scrutin uninominal majoritaire à un tour** élit le candidat "préféré" de la population, c'est à dire le candidat récoltant **le plus grand nombre de voix**.

Ce mode de scrutin est notamment utilisé en **Islande**, au **Venezuela**, et au **Mexique**.

#### 2.1.2 Avantages et inconvénients

Il est très simple à mettre en place et incite les électeurs à voter pour leur candidat préféré puisqu'il n'y a qu'**un seul tour**. Cependant, le résultat déçoit souvent une majorité d'électeurs car en général, il n'y aura pas **majorité absolue**.

### 2.2 Le scrutin uninominal majoritaire à deux tours

#### 2.2.1 Fonctionnement et pays utilisant ce scrutin

Le **scrutin uninominal majoritaire à deux tours** consiste, comme son nom l'indique, en **plusieurs tours** :

-Au **premier tour**, les électeurs votent pour leur candidat préféré. Si un candidat obtient plus de **50%** des voix, il est directement élu. Sinon, les **deux** candidats ayant **le plus de voix** sont retenus pour un **second tour**.

-Au **second tour**, chaque électeur choisit un des deux candidats. Le candidat obtenant la **majorité** des voix est élu.

Ce mode de scrutin est notamment utilisé en **France**, au **Portugal**, en **Ukraine**, au **Sénégal**, au **Brésil**, et **dans beaucoup d'autres pays** encore !

#### 2.2.2 Avantages et inconvénients

Ce mode de scrutin ouvre les portes à des possibilités de votes stratégiques. Par exemple, un électeur peut voter pour un candidat pour en empêcher un autre de gagner. En général, ce scrutin donne une **meilleure** représentation de qui est le candidat préféré de la population que **le scrutin uninominal majoritaire à un tour**.

(cf 2.1.2)

Cependant, il peut parfois mener à l'élection de candidats assez clivants.

## 2.3 Le scrutin par éliminations successives

### 2.3.1 Fonctionnement et pays utilisant ce scrutin

Le **scrutin par éliminations successives** à été théorisé par **Cao George Andrae** et **Thomas Hare** dans les années **1850**. Ce vote se fait par **étapes**. À chacune d'entre elle , les électeurs choisissent un candidat parmi ceux en lice. Le candidat ayant le **moins de voix** est alors **éliminé** de l'élection. Ce processus se répète jusqu'à ce qu'il ne reste plus qu'un **seul candidat**

Ce mode de scrutin est notamment utilisé en **Irlande**.

### 2.3.2 Avantages et inconvénients

Dans le cas de l'Irlande, les **préférences** des électeurs sont récoltées et la procédure est automatisée. Cependant, si on prend les votes tour par tour comme suggérés par les théoriciens, cela laisserait place à des votes stratégiques. Enfin , le **dépouillement** des votes prend énormément de temps. Par exemple , si il y a 12 candidats , il y aurait **12 tours** !

## 2.4 Le scrutin de Condorcet

### 2.4.1 Fonctionnement et pays utilisant ce scrutin

Le **scrutin de Condorcet** à été théorisé par **Nicolas de Condorcet** dans les années **1780**. Ce vote se fait par **étapes**. Chaque candidat va **affronter** tous les autres en **1 contre 1**. Les électeurs votent pour leurs candidats préférés parmi les deux en combat. Le candidat ayant le **plus de voix** gagne le combat. Le gagnant de ces élections, dit aussi **vainqueur de Condorcet** est le candidat ayant gagné tous ses **combats**.

Il n'y a **aucun** gouvernement actuel utilisant le scrutin de Condorcet.

### 2.4.2 Avantages et inconvénients

Ce mode de scrutin permet d'élire réellement le **candidat préféré** de la population. En effet, si un candidat est **vainqueur de Condorcet**, cela signifie qu'il a gagné contre tous les autres candidats. Cependant, le scrutin de Condorcet ne donne pas toujours un gagnant, c'est le **paradoxe de Condorcet**. De plus, il serait très long à dépouiller car il faudrait soit faire un vote à chaque combat ou prendre un choix parmi un ordre de préférences des électeurs. C'est donc principalement pour ces raisons qu'il ne sera probablement **jamais** utilisé dans un gouvernement.

## 2.5 Le scrutin de Borda

### 2.5.1 Fonctionnement et pays utilisant ce scrutin

Le **scrutin de Condorcet** à été théorisé par **Jean-Charles de Borda** dans les années **1770**, en rivalité avec Condorcet (*cf 2.4*). Premièrement, ce vote demande aux électeurs de classer les  $n$ -candidats par **ordre de préférences**. Ensuite, on attribue des **points** aux candidats en fonction de leur **position** dans le classement du votant. Par exemple, le candidat qui est à la **1<sup>re</sup> position** du classement gagnera  **$n$ -points**, celui à la **2<sup>e</sup> position** gagnera  $n-1$  points, et ainsi de suite. À la fin, c'est le candidat possédant le **plus de points** qui sera élu.

Il n'y a **aucun** gouvernement actuel utilisant le scrutin de Borda. Cependant, ce scrutin est notamment utilisé aux **Etats-Unis** pour attribuer des prix sportifs, à l'**Eurovision** en Europe et pour le **ballon d'or** en **France**.

### 2.5.2 Avantages et inconvénients

Ce mode de scrutin incite à la **participation** des électeurs et il donne en général un résultat **satisfaisant** la plupart des électeurs. Cependant, son seul réel défaut est qu'il ne donne pas toujours un **vainqueur de Condorcet** (*cf 2.4.2*) s'il y en a un .

## 3 Implémentation algorithmique en Python 🍄

### 3.1 Les pré-requis

Tous nos **algorithmes** prennent en entrée la **liste des candidats** de l'élection et une **liste de listes** qui représentent des **classements** des candidats générés **aléatoirement** simulant les classements des électeurs. Tous nos programmes respectent la norme **PEP 8**.

### 3.2 Les fonctions

#### 3.2.1 La fonction élection

Ces élections sont simulées par la fonction `Election()` suivante qui prend en argument **le nombre de candidats et le nombre de votes**, qui génère la **liste de listes** (*cf 3.1*) et qui retourne les résultats de chaque mode de scrutin. Cette fonction se trouve à la page suivante

---

```

def Election(nb_candidats, nb_votes):
    """fonction permettant de
    générer des élections pour les 5 modes de scrutin
    """
    # Liste des lettres de l'alphabet en fonction du nombre de candidats
    liste_candidats = list(string.ascii_uppercase)[:nb_candidats]
    # Génération d'un certain nombre de votes aléatoires
    L = [sample(liste_candidats, len(liste_candidats))
          for i in range(nb_votes)]
    a = Uninominal_majoritaire_a_un_tour(liste_candidats, L)
    b = Uninominal_majoritaire_a_deux_tours(liste_candidats, L)
    c = Borda(liste_candidats, L)
    d = Scrutin_de_condorcet(liste_candidats, L)
    e = Eliminations(liste_candidats, L)
    return a, b, c, d, e # Renvoie le résultat de chaque mode de scrutin

```

---

### 3.2.2 Exemple d'algorithmes de modes de scrutins : Scrutin uninominal majoritaire à deux tours; Scrutin par éliminations successives

Scrutin par **éliminations** successives :

---

```

def Eliminations(liste_candidats, L):
    """fonction simulant une élection
    avec le scrutin par éliminations
    """
    # Liste associant à chaque candidat son nombre de votes initialisé à 0
    C = []
    for i in liste_candidats:
        C.append([0, i])
    while len(C) > 1: # Tant qu'il reste plus d'un candidat
        for i in C: # Parcours le nombre de votes des candidats
            i[0] = 0 # Réinitialise ce nombre de votes à 0
        for j in L: # Boucle parcourant les classements des votants
            for k in range(len(C)):
                # Cherche le candidat préféré du votant
                # dans la liste des candidats
                if j[0] == C[k][1]:
                    # Ajoute 1 voix au premier candidat du classement du votant
                    C[k][0] += 1
        C.sort(reverse=True) # Tri la liste par ordre décroissant
        for i in C: # Parcours chaque liste de votes
            # Supprime de la liste les votes pour le candidat qui est éliminé
            i.remove(C[-1][1])
        del C[-1] # Supprime le candidat éliminé
    return str(C[0][1]) # Renvoie le dernier candidat restant

```

---

Scrutin uninominal majoritaire à **deux** tours :

---

```
def Uninominal_majoritaire_a_deux_tours(liste_candidats, L):
    """fonction simulant une élection avec le
    scrutin Uninominal majoritaire à deux tours
    """
    # Liste associant à chaque candidat son nombre de votes initialisé à 0
    C = []
    for i in liste_candidats:
        C.append([0, i])
    for j in L: # Boucle parcourant les classements des votants
        for k in range(len(liste_candidats)):
            # Cherche le candidat préféré du votant dans la liste des candidats
            if j[0] == liste_candidats[k]:
                # Ajoute 1 voix au premier candidat du classement du votant
                C[k][0] += 1
    C.sort(reverse=True) # Tri la liste par ordre décroissant
    if C[0][0] > len(L)/2: # Vérifie si un candidat à la majorité absolue
        return C[0][0] # Renvoie l' élu à la majorité absolue
    # Procède à présent au second tour de l'élection
    f = C[:2] # Sélectionne les 2 meilleurs candidats
    f[0][0], f[1][0] = 0, 0 # Réinitialise les votes de ces candidats à 0
    for i in L: # Boucle parcourant à nouveau les classements des votants
        # Regarde quel candidat à la meilleure position
        # dans le classement du votant puis lui ajoute une voix
        if i.index(f[0][1]) < i.index(f[1][1]):
            f[0][0] += 1
        else:
            f[1][0] += 1
    f.sort(reverse=True) # Tri la liste par ordre décroissant
    return str(f[0][1]) # Renvoie le candidat avec le plus de voix
```

---

### 3.2.3 La fonction générant des histogrammes

La fonction `Hist()` prend en entrée le **nombre d'élections à générer**, le **nombre de candidats** et le **nombre de votes** et qui retourne un histogramme du nombre de candidats différents élus par chaque mode de scrutin lors d'une **simulation** par la fonction `Election()` (cf 3.2.1). Elle fait appel à une fonction **Autolabel** qui ajoute les valeurs en haut de l'histogramme. Voici cette fonction :

---

```

def Hist(nb_elections, nb_candidats, nb_votes):
    """Fonction générant un histogramme
    en fonction du nombre de candidats
    élu différents lors d'une
    exécution de la fonction Elections
    """

    def Autolabel(rects):
        """Fonction affichant les valeurs
        de chaque colonne à leur sommet
        """

        for rect in rects:
            height = rect.get_height()
            ax.annotate('{}'.format(height),
                        xy=(rect.get_x() + rect.get_width() / 2, height),
                        xytext=(0, 3),
                        textcoords="offset points",
                        ha='center', va='bottom')

    # Liste les effectifs des colones de l'histogramme
    valeurs_hist = [0 for i in range(nb_candidats)]
    for i in range(nb_elections):
        # Donne la liste des gagnants de l'élection
        L = (election(nb_candidats, nb_votes))
        if "Pas de gagnant" in L: # Ne prend pas en compte si "pas de gagnant"
            x = len(set(L))-1
        else:
            x = len(set(L))
        # Ajoute 1 à la colone représentant de
        # nombre d'élus différents de l'élection
        valeurs_hist[x-1] += 1
    print(valeurs_hist)
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1])
    titres_colones = ["1 élu",
                      "2 élus \n différents",
                      "3 élus \n différents",
                      "4 élus \n différents",
                      "5 élus \n différents"]
    # Crée les colones de l'histogramme
    v = ax.bar(titres_colones, valeurs_hist)
    # Titre du graphique
    plt.title("Histogramme sur " + str(nb_elections) + " élections")
    plt.ylabel("Nombres d'élections") # Titre de l'axe y
    plt.xlabel("Nombres d'élus différents") # Titre de l'axe x
    # Génère les valeurs de l'axe y
    plt.yticks((np.arange(0, (nb_elections+1)//1.6, nb_elections//20)))
    Autolabel(v) # Affiche la valeur de chaque colonne à leur sommet
    plt.show() # Affichage d'un histogramme

```

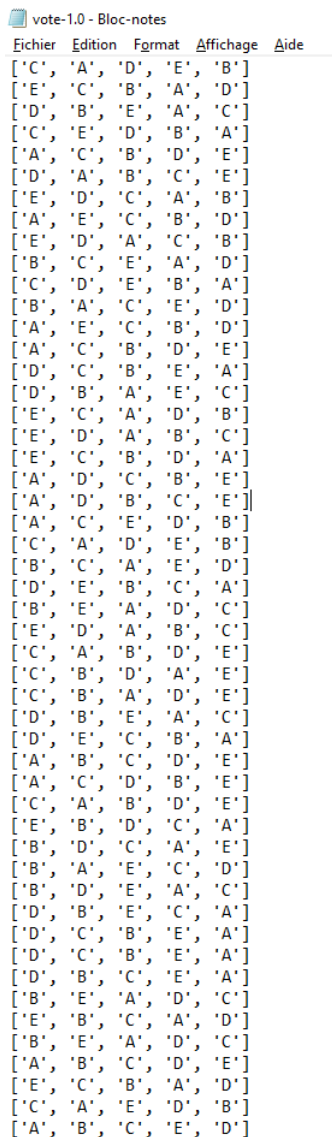
---



## 4 Annexe

### 4.1 Un fichier contenant 50 classements de votants menant à un candidat élu différent pour chaque scrutin, soit 5 candidats élus

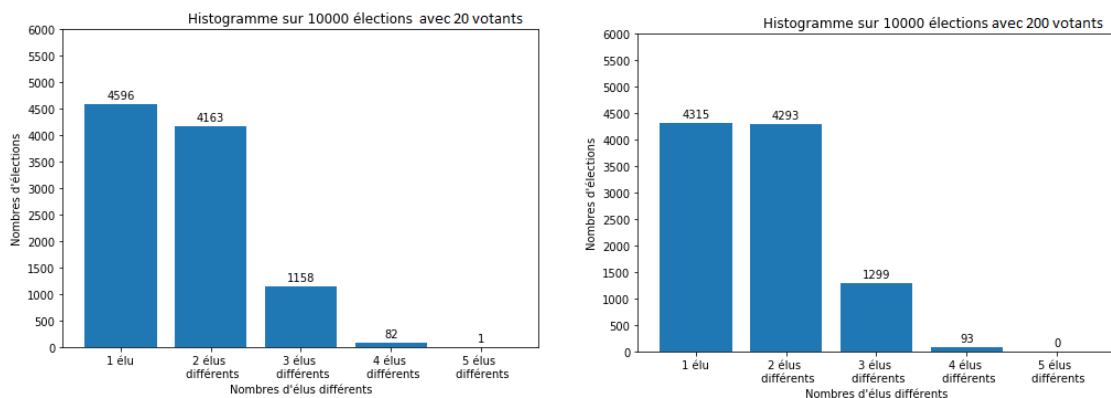
Pour générer ce fichier, nous avons fait une boucle qui enregistrerait le classement des votants **si et seulement si** 5 candidats différents sont élus dans chaque mode de scrutin. Voici un exemple de fichier généré avec **50 classements** :



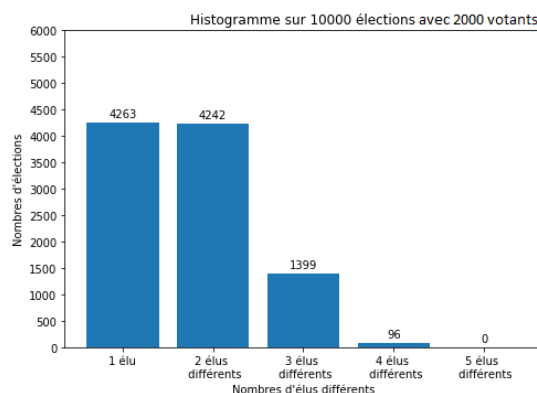
```
vote-1.0 - Bloc-notes
Fichier Edition Format Affichage Aide
['C', 'A', 'D', 'E', 'B']
['E', 'C', 'B', 'A', 'D']
['D', 'B', 'E', 'A', 'C']
['C', 'E', 'D', 'B', 'A']
['A', 'C', 'B', 'D', 'E']
['D', 'A', 'B', 'C', 'E']
['E', 'D', 'C', 'A', 'B']
['A', 'E', 'C', 'B', 'D']
['E', 'D', 'A', 'C', 'B']
['B', 'C', 'E', 'A', 'D']
['C', 'D', 'E', 'B', 'A']
['B', 'A', 'C', 'E', 'D']
['A', 'E', 'C', 'B', 'D']
['A', 'C', 'B', 'D', 'E']
['D', 'C', 'B', 'E', 'A']
['D', 'B', 'A', 'E', 'C']
['E', 'C', 'A', 'D', 'B']
['E', 'D', 'A', 'B', 'C']
['E', 'C', 'B', 'D', 'A']
['A', 'D', 'C', 'B', 'E']
['A', 'D', 'B', 'C', 'E']
['A', 'C', 'E', 'D', 'B']
['C', 'A', 'D', 'E', 'B']
['B', 'C', 'A', 'E', 'D']
['D', 'E', 'B', 'C', 'A']
['B', 'E', 'A', 'D', 'C']
['E', 'D', 'A', 'B', 'C']
['C', 'A', 'B', 'D', 'E']
['C', 'B', 'D', 'A', 'E']
['C', 'B', 'A', 'D', 'E']
['D', 'B', 'E', 'A', 'C']
['D', 'E', 'C', 'B', 'A']
['A', 'B', 'C', 'D', 'E']
['A', 'C', 'D', 'B', 'E']
['C', 'A', 'B', 'D', 'E']
['E', 'B', 'D', 'C', 'A']
['B', 'D', 'C', 'A', 'E']
['B', 'A', 'E', 'C', 'D']
['B', 'D', 'E', 'A', 'C']
['D', 'B', 'E', 'C', 'A']
['D', 'C', 'B', 'E', 'A']
['D', 'C', 'B', 'E', 'A']
['D', 'B', 'C', 'E', 'A']
['B', 'E', 'A', 'D', 'C']
['E', 'B', 'C', 'A', 'D']
['B', 'E', 'A', 'D', 'C']
['A', 'B', 'C', 'D', 'E']
['E', 'C', 'B', 'A', 'D']
['C', 'A', 'E', 'D', 'B']
['A', 'B', 'C', 'E', 'D']
```

## 4.2 Quelques histogrammes

À travers les **histogrammes** suivants, nous avons cherché à observer si le mode de scrutin influe beaucoup sur le résultat des élections :



(1)



(2)

Comme nous pouvons le constater, les cas où 5 candidats différents sont élus est très rare. Cependant, il n'est pas si rare d'obtenir 2,3 voir 4 candidats élu différents. Ce résultat pose donc la question "**Quel est le mode de scrutin le plus juste ?**"

## 5 Crédits

Ce projet a été réalisé, puis vous a été présenté, par **LIN Yann, SUON Léo, BLANCAL Aurélien, ECONOMOU Emmanuel, RINCLA Jade.**

**Merci d'avoir lu cette documentation jusqu'au bout !**