

Explication ligne par ligne du script `script1.py`

```
# @ Auteur: Basri, Miahhy
# @ Créé le : 05/05/2025 13:54:25
# @ Description: fichier de script pour le projet en outils numériques
```

Lignes 1-3 : Commentaires d'en-tête indiquant l'auteur, la date de création et une brève description du script.

```
###

# Importation des modules
import numpy as np
import matplotlib.pyplot as plt
import os
```

Lignes 4-8 : Définition d'une section (souvent pour l'organisation dans certains IDE) et importation des modules `numpy` (pour les calculs numériques), `matplotlib.pyplot` (pour la visualisation) et `os` (pour la gestion des chemins de fichiers).

```
# chemin pour images générées
path = os.pardir + "/img/"
os.makedirs(path, exist_ok=True)
```

Lignes 9-11 : Définition du chemin pour les images générées et création du répertoire s'il n'existe pas déjà.

```
# définition des variables
N = 1000 # population totale
I0 = 1 # personnes infectées
R0 = 0 # personnes rétablies
V0 = 0 # personnes vaccinées
tf = 150 # jours
S0 = N - I0 - R0 - V0 # personnes susceptibles
β = 0.4 # taux de contact
γ = 0.05 # taux de rétablissement
α = 0.02 # taux de vaccination
```

Lignes 12-20 : Initialisation des paramètres du modèle :

- N : population totale.
- I_0, R_0, V_0 : conditions initiales pour infectés, rétablis, vaccinés.
- t_f : durée totale de la simulation.
- S_0 : susceptibles initiales, calculées par différence.
- β, γ, α : taux de transmission, de rétablissement, de vaccination.

```
# tableaux
t = np.arange(0, tf+1, 0.1) # temps
S = np.zeros(len(t)) # personnes susceptibles
I = np.zeros(len(t)) # personnes infectées
R = np.zeros(len(t)) # personnes rétablies
V = np.zeros(len(t)) # personnes vaccinées
```

Lignes 21-26 : Création des tableaux pour le temps (t) et pour stocker l'évolution des populations (S, I, R, V) initialisés à zéro.

```
# conditions initiales
I[0] = I0
R[0] = R0
S[0] = S0
V[0] = V0
```

Lignes 27-31 : Affectation des conditions initiales aux premiers éléments des tableaux.

```
# variations
for i in range(len(t)-1):
    dS = - $\beta$ *S[i]*I[i]/N -  $\alpha$ *S[i] # variation de S
    dI =  $\beta$ *S[i]*I[i]/N -  $\gamma$ *I[i] # variation de I
    dR =  $\gamma$ *I[i] # variation de R
    dV =  $\alpha$ *S[i] # variation de V

    S[i+1] = S[i] + dS * 0.1
    I[i+1] = I[i] + dI * 0.1
    R[i+1] = R[i] + dR * 0.1
    V[i+1] = V[i] + dV * 0.1
```

Lignes 32-41 : Boucle pour l'intégration numérique :

- Calcul des variations (dS , dI , dR , dV) à chaque pas, selon les équations différentielles.
- Mise à jour des populations pour l'étape suivante en utilisant la méthode d'Euler avec un pas de 0.1.

```
X = V + R
Y = np.abs(X - 0.75*N).argmin()
```

Lignes 42-43 : Calcul du moment où la somme des vaccinés et rétablis ($V + R$) est la plus proche de 75% de la population ($0.75 * N$) :

- X : somme à chaque instant.
- Y : indice du tableau où cette somme est la plus proche de 75%.

```
# affichage
plt.figure(figsize=[10,6])
plt.plot(t, S, label='S (t) : personnes susceptibles')
plt.plot(t, I, label='I (t) : personnes infectées')
plt.plot(t, R, label='R (t) : personnes rétablies')
plt.plot(t, V, label='V (t) : personnes vaccinées')
plt.xlabel('temps en jours')
plt.ylabel('population')
plt.axvline(t[Y], color='k', linestyle='--', label=f'75% atteint à t = {t[Y]:.1f} jours')
plt.title('Modèle SIRV')
plt.legend()
plt.grid(ls='dashed')
plt.savefig(path + 'SIRV.png')
plt.show()
```

Lignes 44-56 : Visualisation graphique :

- Création d'une figure.
- Tracé des courbes pour chaque population (S , I , R , V) en fonction du temps.
- Ajout d'une ligne verticale indiquant le moment où 75% de la population est vaccinée ou rétablie.
- Ajout de légendes, titre, axes, grille, puis affichage.

Résumé :

Ce script modélise la dynamique d'une épidémie avec vaccination en utilisant la méthode d'Euler. Il calcule l'évolution des populations susceptibles, infectés, rétablis et vaccinés, puis identifie le moment où 75% de la population est protégée, en visualisant le tout dans un graphique.

Annexe :

