

Explication ligne par ligne du script `script1.py`

```
# @ Auteur: Basri, Miah  
# @ 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
```

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

```
# définition des variables  
N = 1000 # population totale  
I0 = 1 #personnes infectées  
R0 = 0 #personnes rétablis  
V0 = 0 #personnes vaccinées  
tf = 150 #jours  
S0 = N - I0 - R0 - V0 #personnes susceptibles  
 $\beta$  = 0.4 #taux de contact  
 $\gamma$  = 0.05 #taux de rétablissement  
 $\alpha$  = 0.02 #taux de vaccination
```

Lignes 9-17 : Initialisation des paramètres du modèle :

- `N` : population totale.
- `I0`, `R0`, `V0` : conditions initiales pour infectés, rétablis, vaccinés.
- `tf` : durée totale de la simulation.
- `S0` : 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 18-23 : 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 24-27 : Affectation des conditions initiales aux premiers éléments des tableaux.

```
#variations
for i in range(len(t)-1):
    dS = -β*S[i]*I[i]/N - α*S[i] #variation de S
    dI = β*S[i]*I[i]/N - γ*I[i] #variation de I
    dR = γ*I[i] #variation de R
    dV = α*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 28-36 : 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 37-38 : 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')
plt.plot(t, I, label='I')
plt.plot(t, R, label='R')
plt.plot(t, V, label='V')
plt.xlabel('temps')
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.show()
```

Lignes 39-47 : 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.