import matplotlib.pyplot as plt  
import numpy as np  
  
# -> Definicion de funciones  
# --------------------------  
def FFN(years, I0, ing, egr, imp, am):  
 """  
 Función de calculo de flujo de fondos en un determinado rango  
 de años  
 """  
 amort = am[0]  
 amort\_total = am[1]  
 FFs = [-I0]  
 for year in years:  
 ubt = ing - egr - amort  
 uat = ubt - ubt\*imp  
 uat += amort  
 FFs.append(uat)  
 amort\_total -= amort  
 if amort\_total < 0:  
 amort\_total = 0  
  
 FFs[-1] += amort\_total  
 return FFs, amort\_total  
  
def VANeo(FF, d):  
 """  
 A partir del flujo de fondos de un año y una tasa de interes se  
 calcula el VAN  
 """  
 VANs = []  
 VAN = 0  
 for i, FFi in enumerate(FF):  
 VAN += FFi/((1+d)\*\*i)  
 VANs.append(VAN)  
 return VANs  
  
def TIR(FF):  
 """  
 Calcula el valor de la TIR para un Flujo de Fondos determinado  
 iterando a partir de una tasa de descuento de 0.05 hasta la  
 tasa correspondiente para obtener un VAN negativo, e interpola  
 para obtener que resulta en un VAN igual a cero.  
 """  
 previous\_d = 0  
 previous\_VAN = 1  
 d = -0.05  
 while previous\_VAN > 0:  
 VAN = 0   
 for i, FFi in enumerate(FF):  
 VAN += FFi/((1+d)\*\*i)  
 if VAN < 0:   
 # Interpolar con el valor de VAN = 0  
 return previous\_d - ((d-previous\_d)/(VAN-previous\_VAN))  
 previous\_d = d  
 previous\_VAN = VAN  
 d += 0.01  
  
# -> Definicion de variables  
# --------------------------  
years = range(5) # Años a calcular el flujo de fondos  
I0 = 5675.20 # Valor de inversión inicial  
egr = 7709\*3 # Valor de egresos anuales  
ing = 10000 # Valor de ingresos anuales  
imp = 0.35 # Impuesto a las ganancias  
am = [I0/10,I0] # [tasa de amortizacion, valor total de amortizacion]  
d = 0.20 # Tasa de descuento  
  
ings = []  
VANs = []  
TIRs = []  
FFs = []  
  
# -> Calculos iterativos  
# ----------------------  
# Se realiza una iteración entre valores de ingresos y se  
# calcula el VAN y TIR correspondiente para ese valor de  
# ingreso determinado, los cuales son agregados a una lista.  
for ing in range(egr,int(egr\*1.3),10):  
 # Iteración sobre múltiples ingresos   
 FF, VR = FFN(years, I0, ing, egr, imp, am)  
 VANs.append(VANeo(FF,d)[-1])  
 TIRs.append(TIR(FF))  
 ings.append(ing)  
  
# -> Graficar  
# -----------  
x = ings  
y = VANs  
  
f = np.polyfit(x,y,deg=1)  
zero = -f[1]/f[0]  
  
  
plt.rcParams['font.family'] = 'Times New Roman'  
plt.rcParams['font.size'] = '13'  
plt.scatter(  
 x,y,  
 color=['red' if i < 0 else 'green' for i in y],  
 s=1, alpha=0.5)  
  
plt.axvline(zero,c='darkgray',ls='--')  
plt.xlabel('Ingresos anuales (U$D)')  
plt.ylabel('VAN (U$D)')  
plt.savefig('rent.svg')  
print('Ingresos anuales para un VAN positivo: ',zero)