```
using PyPlot

function em_alorithm(y::Vector, θ_0::Vector, nbr_iter::Int)

    # set obs data
    y_A, y_B, y_AB, y_00 = y
    N = sum(y)

    # pre-allocate matrixm to store param ests
    θ_matrix = zeros(length(θ_0), nbr_iter)

    # set start values
    θ_matrix[:,1] = θ_0

    # loop over nbr_iter
    for i in 2:nbr_iter

        # set old parameter values
        p_i, q_i, r_i = θ_matrix[:,i-1]

        # E step
        E_AA = y_A*p_i^2/(p_i^2 + 2*p_i*r_i)
        E_A0 = y_A*2*p_i*r_i/(p_i^2 + 2*p_i*r_i)
        E_AB = y_AB
        E_00 = y_00
        E_B0 = y_B*2*q_i*r_i/(2*q_i*r_i + q_i^2)
        E_BB = y_B*q_i^2/(2*q_i*r_i + q_i^2)

        E_A = 2*E_AA + E_A0 + E_AB
        E_B = 2*E_BB + E_B0 + E_AB
        E_0 = 2*E_00 + E_A0 + E_B0

        # M step
        p_hat = E_A/(2*N)
        q_hat = E_B/(2*N)
        r_hat = E_0/(2*N)

        # store new param ests

        θ_matrix[:,i] = [p_hat;q_hat;r_hat]


    end

    return θ_matrix[:,end], θ_matrix


end
```

```
# run em algorithm
y = [40; 27; 24; 9] # data
θ_0 = [1/3; 1/3; 1/3] # start value for parameters
N = 10 # number iterations of the EM algorithm

θ_hat, θ_matrix = em_alorithm(y, θ_0, N)


round.(θ_hat, digits= 2)

# plot em trace

θ_1 = [1/3; 1/3; 1/3]
θ_2 = [0.2; 0.2; 0.6]
θ_3 = [0.8; 0.1; 0.1]

θ_hat_1, θ_matrix_1 = em_alorithm(y, θ_1, N)
θ_hat_2, θ_matrix_2 = em_alorithm(y, θ_2, N)
θ_hat_3, θ_matrix_3 = em_alorithm(y, θ_3, N)

PyPlot.figure()
PyPlot.scatter3D(θ_matrix_1[1,:], θ_matrix_1[2,:],θ_matrix_1[3,:], color="blue", "*")
PyPlot.scatter3D(θ_matrix_2[1,:], θ_matrix_2[2,:],θ_matrix_2[3,:], color="red", "*")
PyPlot.scatter3D(θ_matrix_3[1,:], θ_matrix_3[2,:],θ_matrix_3[3,:], color="green", "*")
PyPlot.xlabel("p", fontsize=12)
PyPlot.ylabel("q", fontsize=12)
PyPlot.zlabel("r", fontsize=12)
PyPlot.savefig("fig_em_trace.pdf")
```