

```

[1] using Queryverse
    using Plots
    using LinearAlgebra
    using Swarm
[2] dados = load("dados.xls", "Planilha1") |> DataFrame
[3] dados = dados |> @filter( _[:3]!=0) |> DataFrame
[4] estaca = unique(dados[:1])
[5] betas_vector = zeros(length(estaca));
[6] for j = 1:length(estaca)
[7]     temp = dados |> @filter( _[:Cód] == estaca[j]) |> DataFrame
[8]     corte = findmin(abs.(temp[:3] .- temp[:4]))[2]
[9]     sigma_vec = zeros(size(temp,1))
[10]    for i = 1:length(sigma_vec)
[11]        sigma_vec[i] = temp[:7][i] * temp[:5][1]
[12]    end
[13]    function custo(beta)
[14]        sum(abs.((sigma_vec[1:corte] .* beta[1]) .- temp[:3][1:corte]))
[15]    end
[16]    function mycons(beta)
[17]        [ temp[6][1] - maximum(sigma_vec .* beta[1]) ]
[18]    end
[19]    lb = [0.0]
[20]    ub = [3.0]
[21]    particles = Particles(80,lb,ub)
[22]    pso(particles, custo, minstep=1e-3, constraints=mycons)
[23]    global betas_vector[j] = particles.best_position[1]
[24] end
[25] DataFrame( Betas = betas_vector) |> save(valores_beta.xls)

```

Essa rotina computacional automatiza os seguintes procedimentos:

- Supor valor para β ;
- Calcular a carga lateral verdadeira e comparar com os valores obtidos através dos dados instrumentados (carga lateral aparente);
- Calcular o somatório dos erros absolutos e aplicar a restrição (a carga lateral total não deve ultrapassar a carga máxima aplicada na prova de carga).

Após 80 iterações (valor que pode ser modificado), a rotina mostra o valor de β que apresentou o menor somatório dos erros absolutos. Para utilizar a rotina o usuário deve elaborar uma planilha com as seguintes colunas: códigos das estacas; profundidades - m; carga lateral aparente - kN; carga aparente (instrumentação) - kN; perímetro das estacas - m; carga máxima da prova de carga - P_{últ} ; parcela da carga lateral referente a tensão efetiva ($\int U(\sigma'_z)dz$) - kN.