```
%% Task 4.3
clc, clear all

% Settings
a = 1.4;
b = 0.3;
nrOfInitConditions = 1000;
nrOfIterations = 2000;
transient = 10;
nTot = nrOfInitConditions*(nrOfIterations-transient);

% Create initial
[xList, yList] = CreateHenonMap(a,b,nrOfInitConditions,nrOfIterations,transient);

nEpsilons = 10;
Qlist = [0,1,2,linspace(0.1,4,nEpsilons-3)];
Iq = zeros(nEpsilons,nEpsilons);
epsilon = linspace(1e-3, 1e-2,nEpsilons);
for i = 1:nEpsilons

    [H, nTot] = CreateHistogram(epsilon(i), xList, yList);

    % Iq 0
    Iq(1,i) = size(H(H>0),1);

    % Iq 2
    tmpH = H./nTot .* log(1./(H/nTot));
    Iq(2,i) = sum(tmpH(~isnan(tmpH)));

    % Iq 3
    Iq(3,i) = sum(sum((H/nTot).^2));

    % Iq
    for j = 1:nEpsilons-3
        Iq(j+3,i) = sum(sum((H/nTot).^Qlist(3+j)));
    end
end
[D, DqOfq] = FitPolynomial(Iq, epsilon, Qlist);

% c) Boxcounting, information and correlation
D(1,1)
disp("D1: " + D(1,1) + "D two: " + D(2,1) + "D three: " + D(3,1))


% e) Lyaponov exponents
clc
lambda = zeros(2,1);
Q = eye(2);
for k=1:length(xList(end,:))
    M = [-2*a*xList(end,k), 1; b, 0];
    [Q,R]=qr(M*Q);
    lambda= lambda + log(abs(diag(R)));
end
lambda = sort(lambda /length(xList(end,:)), 'descend');
```

```
DL = 1 - lambda(1)/lambda(2)
Lyapunov_exponents =sprintf('lambda1: %f, $lambda2$: %f',lambda(1),lambda(2))



%% Plots
% Plots Heinon attractor
figure(1)
clf
plot(xList, yList, '.')
xlabel("x")
ylabel("y")
axis equal

% Plots Heinon attractor but with histo
figure(2)
clf
colormap(flipud(gray))
imagesc(H'>1)
set(gca, 'YDir', 'normal')
xlabel("x")
ylabel("y")

% Plots D
figure(3)
clf
q = linspace(-20,20,1000);
Dq = (1./(1-q).*log( (1-2/3).^q + (2/3).^q ) /log(3));
plot(q,Dq,'.')
xlabel("q")
ylabel("D_q")

% b) plots
figure(4)
clf
hold on
plot(log(1./epsilon), log(Iq(1,:)), '-x')
plot(log(1./epsilon), Iq(2,:), '-o')
plot(log(1./epsilon), -log(Iq(3,:)), '-*')
legend('q = 0', 'q = 1', 'q = 2', 'Location', 'northwest')
xlabel('ln(1 / \epsilon)')
title('(1 - q)^{-1} ln[I(q, \epsilon)] / \Sigma_1^{N_{boxes}} [p_k ln(1 / p_k)]')
hold off

% d)
figure(5)
clf
hold on
plot(Qlist,D(:,1),'x')
plot(Qlist, DqOfq(:,1)*Qlist+DqOfq(:,2))
xlabel("q")
ylabel("D_q")
title("All Dq points and a line fitted")
```

```
%%%%
%%% Functions
%%%%

function [xList, yList] = CreateHenonMap(a,b,nrOfInitConditions,nrOfIterations,↙
transient)
x0 = linspace(-0.1, 0.1, nrOfInitConditions);
y0 = linspace(-0.1, 0.1, nrOfInitConditions);
x = zeros(1, nrOfIterations);
y = zeros(1, nrOfIterations);
xList = zeros(nrOfInitConditions, nrOfIterations-transient);
yList = zeros(nrOfInitConditions, nrOfIterations-transient);
    for i=1:nrOfInitConditions
        x(1) = x0(i);
        y(1) = y0(i);
        for j=1:nrOfIterations-1
            x(j+1) = y(j) + 1 - a * x(j)^2;
            y(j+1) = b*x(j);
        end
        xList(i,:) = x(transient+1:end);
        yList(i,:) = y(transient+1:end);
    end
end


function [H, nTot] = CreateHistogram(epsilon, xList, yList)
xEdges = min(xList,[], 'all'):epsilon:max(xList,[],'all');
yEdges = min(yList,[], 'all'):epsilon:max(yList,[],'all');
[H, ~, ~]=histcounts2(xList,yList,xEdges,yEdges);
nTot= sum(sum(H));
end


function [D,DqOfq] = FitPolynomial(Iq, epsilon,Q)
D = zeros(10,2);
D(1,:) = polyfit(log(1./epsilon),log(Iq(1,:)),1);
D(2,:) = polyfit(log(1./epsilon),Iq(2,:),1);
D(3,:) = polyfit(log(1./epsilon),-log(Iq(3,:)),1);

for i = 1:7
    D(i+3,:) = polyfit(log(1./epsilon), log(Iq(i+3,:))/(1-Q(i+3)),1);
end
DqOfq = polyfit(Q,D(:,1)',1);
end
```