

```

% This code will calculate the dfactal dimension of an image from
some
% given data.
%
% Using ImageJ and the box-grid method of calculating fractal
% dimension, the log of the box size of the grid and the log of the
number
% of boxes that contain the aerosol has been exported to a text
% file. They are arrays of the same size that have been combined into
one
% array. This code will graph that data (box size is the x-axis and
box
% count is the y-axis) and calculate the fslope of the linear
regression of
% that data which is the fractal dimension.
%
% Input: None
%
% Output: None
%
% Created on October 1st, 2020 CE by Timothy Honablew

%% Importing the Data and Separating It

data = importdata('imageDataDensity.txt');
boxSize = zeros(length(data)/2, 1);
nonEmpty = zeros(length(data)/2, 1);

for ii = 1:length(data)/2
    boxSize(ii) = data(ii);
    nonEmpty(ii) = data(ii + length(data)/2);
end

% disp(boxSize);
% disp(nonEmpty);

%% Calculating the Slope of the Linear Regression

% From the curve fitting toolbox (comes with MATLAB but must be
manually
% installed

Df = fit(boxSize, nonEmpty, 'poly1');

% , 'Exclude', nonEmpty < 0.01

disp(Df);

%% Graphing the Data and the Linear Regression

```

```

hFig = figure();
set(hFig, 'units', 'pixels', 'Position', [10 10 500 255]);

hold on;
box on;
grid on;

dataPlot = plot(boxSize, nonEmpty, 'o');
linearRegressionPlot = plot(boxSize, Df(boxSize), 'Color', 'k');

title('Box Grid Fractal Dimension Calculation');
xlabel('Log of Box Length');
ylabel('Log of Box Count');
xlim([min(boxSize) - 1 max(boxSize) + 2]);
ylim([min(nonEmpty) - 1 max(nonEmpty) + 1]);

leg = legend([dataPlot, linearRegressionPlot],...
    {'Data', 'Linear Regression'},...
    'interpreter', 'latex', 'location', 'northeast');

set(leg, 'FontSize', 12.5);

save2pdf('BoxGridMethod.pdf', hFig, 600);

```