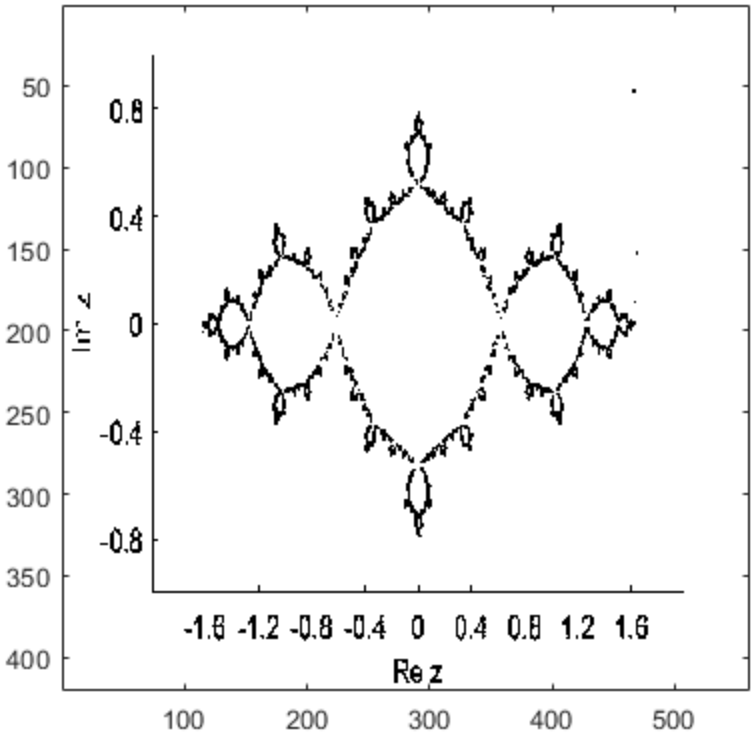
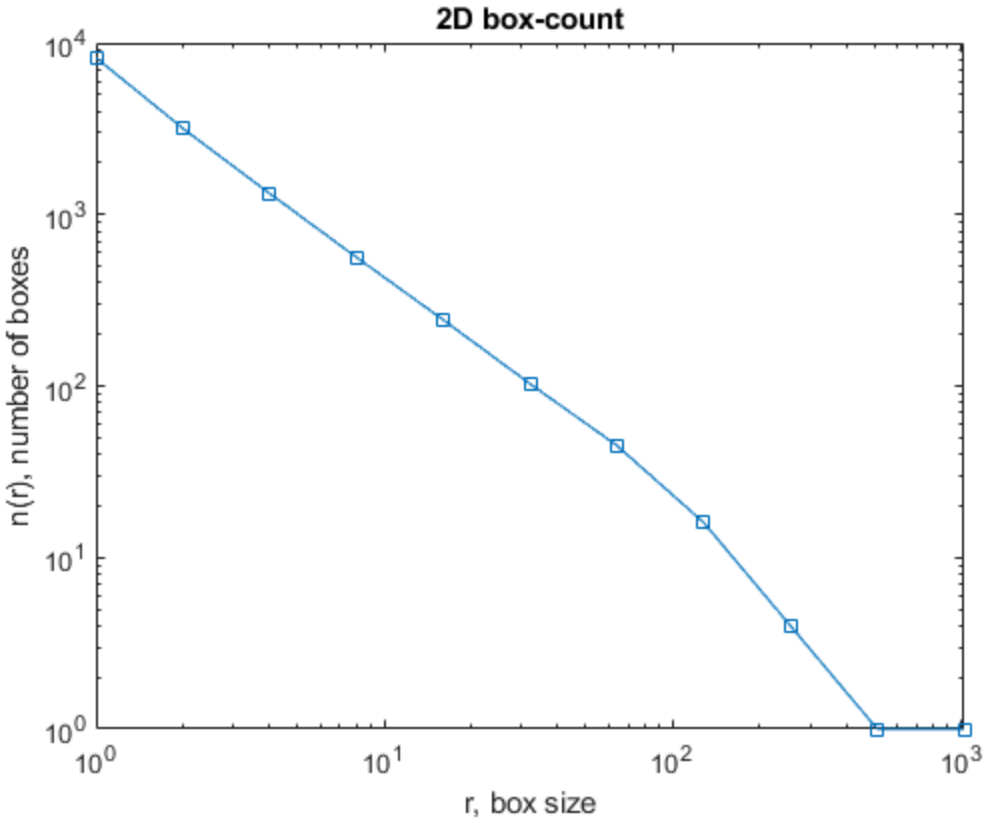
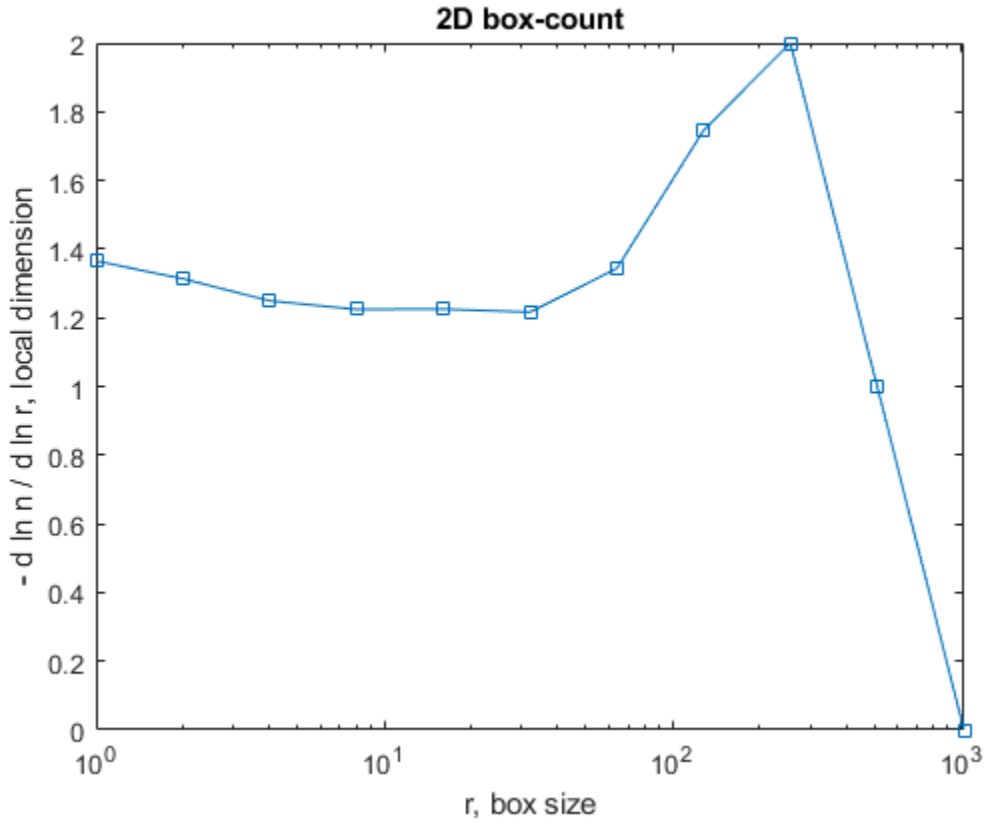
```
%Part 4
%Fractal dimension: how complex or complicated a self-similar object
 is.
%This is achieved by considering the ratio between how complicated a
 fractal pattern is
%and how the scale it's measured at changes.

c = imread('COUNTMYBOxes.JPG');%Takes image from Part 3
c = (c<198);%if c is a fractal set, with fractal dimension DF<D then N
 scales as R^(-DF)
imagesc(~c)
colormap gray
axis square
figure
boxcount(c)%Box counting method is useful for determining fractal
 properties of 1D segments, 2D images or a 3D array.
figure
boxcount(c,'slope')%shows the semi-log plot of N as a function of R
function [n,r] = boxcount(c,varargin)% c is a 2D array, this method
 counts the number N of 2D boxes of size R
%needed to cover the nonzero elements of C.
narginchk(1,2);


if ndims(c)==3
    if size(c,3)==3 && size(c,1)>=8 && size(c,2)>=8
        c = sum(c,3);
    end
end

warning off
c = logical(squeeze(c));
warning on

dim = ndims(c);
if dim>3
    error('Maximum dimension is 3.');
end


if length(c)==numel(c)
    dim=1;
    if size(c,1)~=1
        c = c';
    end
end

width = max(size(c));
p = log(width)/log(2);


if p~=round(p) || any(size(c)~=width)
```

```
    p = ceil(p);
    width = 2^p;
    switch dim
        case 1
            mz = zeros(1,width);
            mz(1:length(c)) = c;
            c = mz;
        case 2
            mz = zeros(width, width);
            mz(1:size(c,1), 1:size(c,2)) = c;
            c = mz;
        case 3
            mz = zeros(width, width, width);
            mz(1:size(c,1), 1:size(c,2), 1:size(c,3)) = c;
            c = mz;
    end
end

n=zeros(1,p+1);

switch dim %
    case 2

        n(p+1) = sum(c(:));
        for g=(p-1):-1:0
            siz = 2^(p-g);
            siz2 = round(siz/2);
            for i=1:siz:(width-siz+1)
                for j=1:siz:(width-siz+1)
                    c(i,j) = ( c(i,j) || c(i+siz2,j) || c(i,j+siz2) ||
 c(i+siz2,j+siz2) );
                end
            end
            n(g+1) = sum(sum(c(1:siz:(width-siz+1),1:siz:(width-siz
+1))));
        end



end
n = n(end:-1:1);
r = 2.^(0:p);

if any(strncmpi(varargin,'slope',1))
    s=-gradient(log(n))./gradient(log(r));
    semilogx(r, s, 's-');
    ylim([0 dim]);
    xlabel('r, box size'); ylabel('- d ln n / d ln r, local
 dimension');
    title([num2str(dim) 'D box-count']);
elseif nargout==0 || any(strncmpi(varargin,'plot',1))
    loglog(r,n,'s-');
    xlabel('r, box size'); ylabel('n(r), number of boxes');
    title([num2str(dim) 'D box-count']);
```

```
end
if nargout==0
    clear r n
end
end
```




2D box-count
n(r), number of boxes
r, box size
10^0
10^1
10^2
10^3
10^4


2D box-count
- d ln n / d ln r, local dimension
r, box size
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
10^0
10^1
10^2
10^3

Published with MATLAB® R2019b