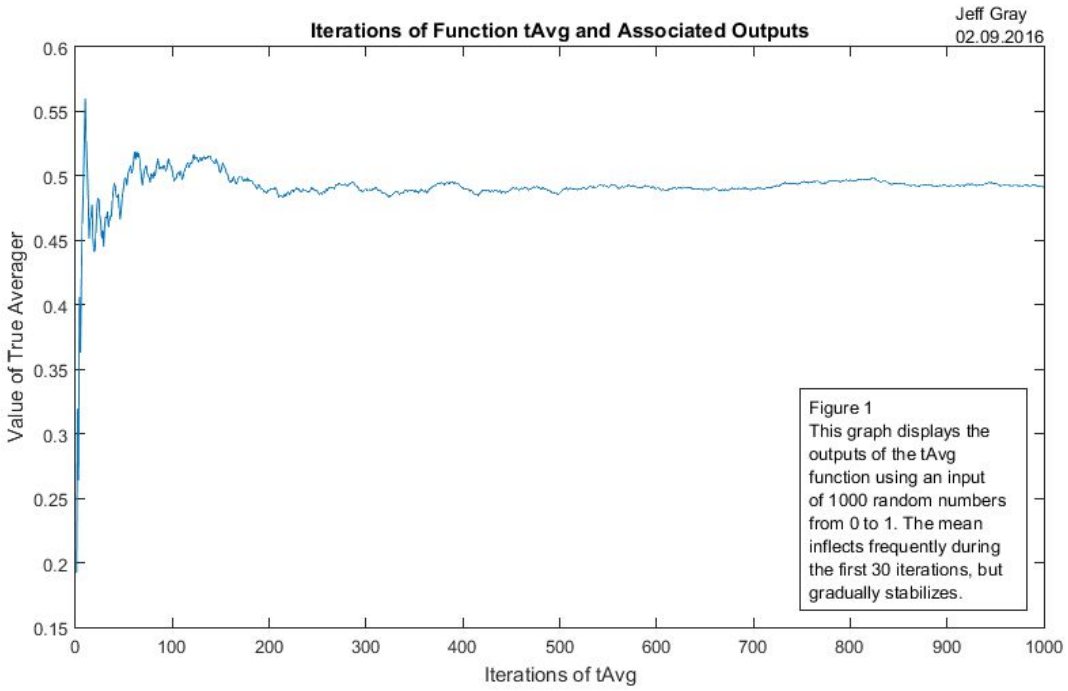
Jeff Gray
jhg7nm
02.09.2016
Workspace: https://github.com/jeffreygray/nesc5330/tree/master/lab3

Moving Averagers

Note: I conceptually discussed the lab with classmate Berk Ekmekci before attempting the problems. We, however, did not work together.

3.1: The Recursive True Averager

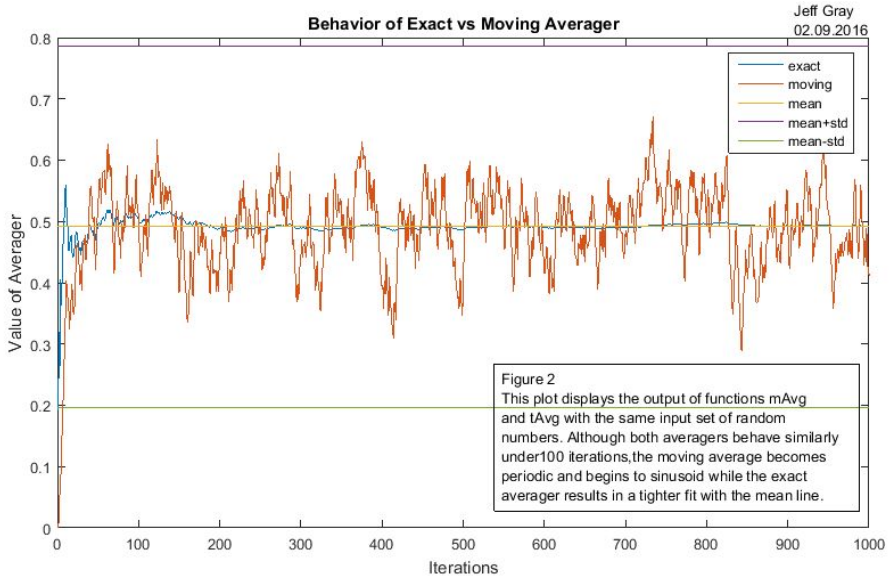



```
>>createfigure1  %     calls tAvg.m to create plot and check outputs
checking to see if output is valid...

actual =
      0.4913
calculated =
      0.4913
```

My true averaging algorithm (see tAvg.m in appendix) works because it traverses an input vector and updates curAvg by adding each vector element to the value of the past curAvg and dividing the result by the element's index value. It produces an output synonymous to the textbook summation from i = 1 to n of x_i/n, which in context to my code would sum the input vector and divide the result by length(vector).

3.2.1



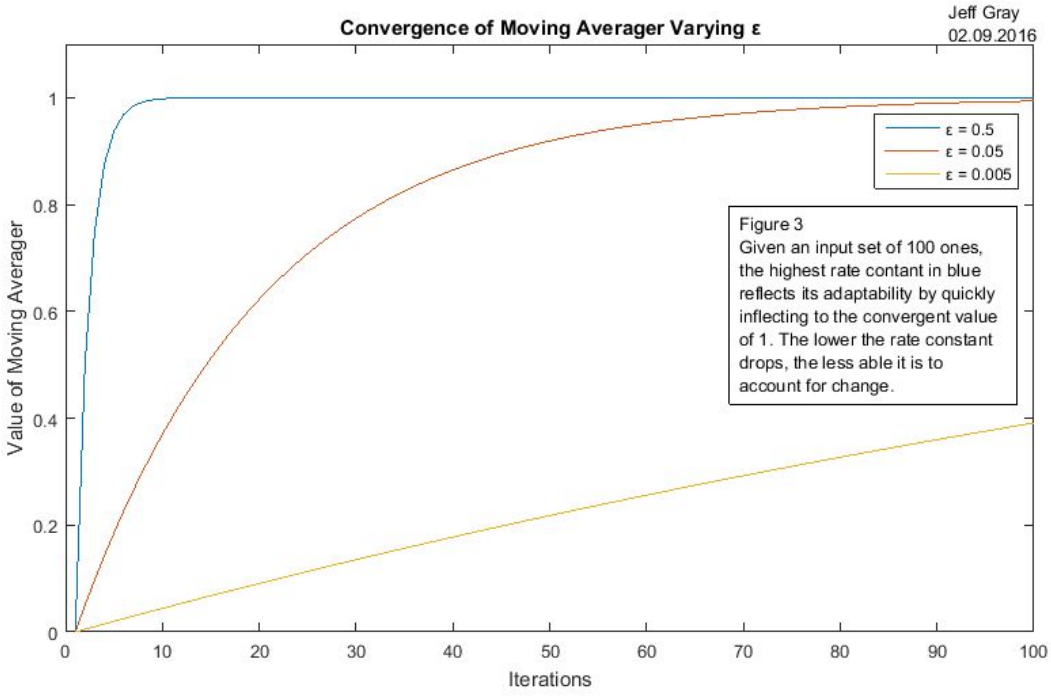
In the event that the outputs of mAvg and tAvg return the same value on the 499th index and assuming the same epsilon value of 0.08, the 500th index needs to be the following solution:

(oldAvg + newNum) / index = oldAvg + eps(newNum - oldAvg)

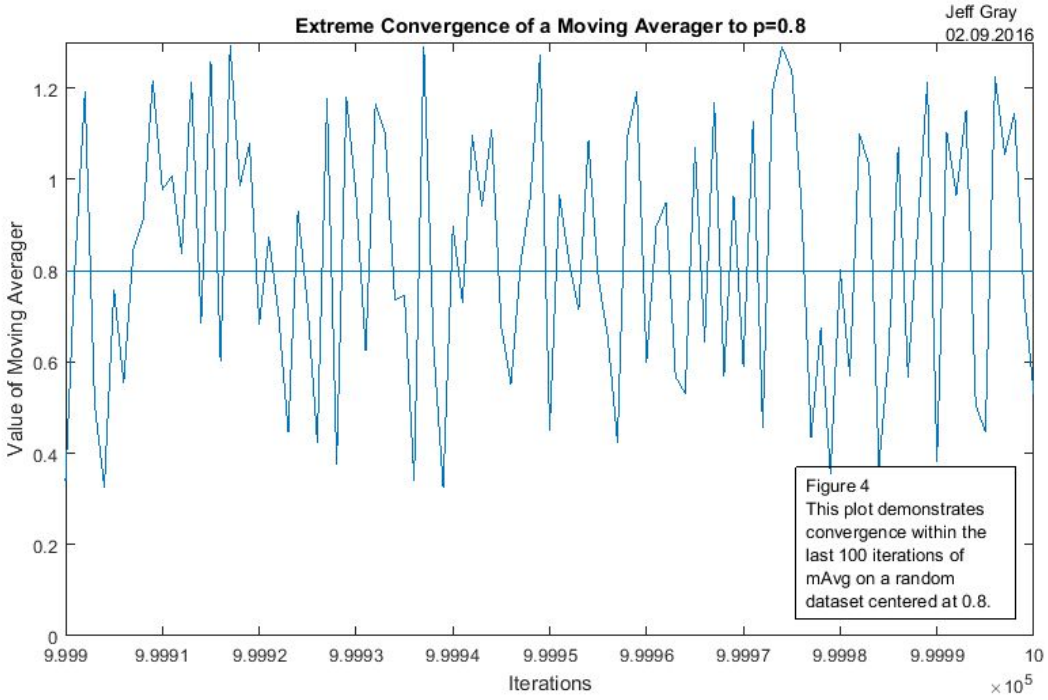
(0.5 + x) / 500 = 0.5 + 0.8(x - 0.5)

solution = -0.588462

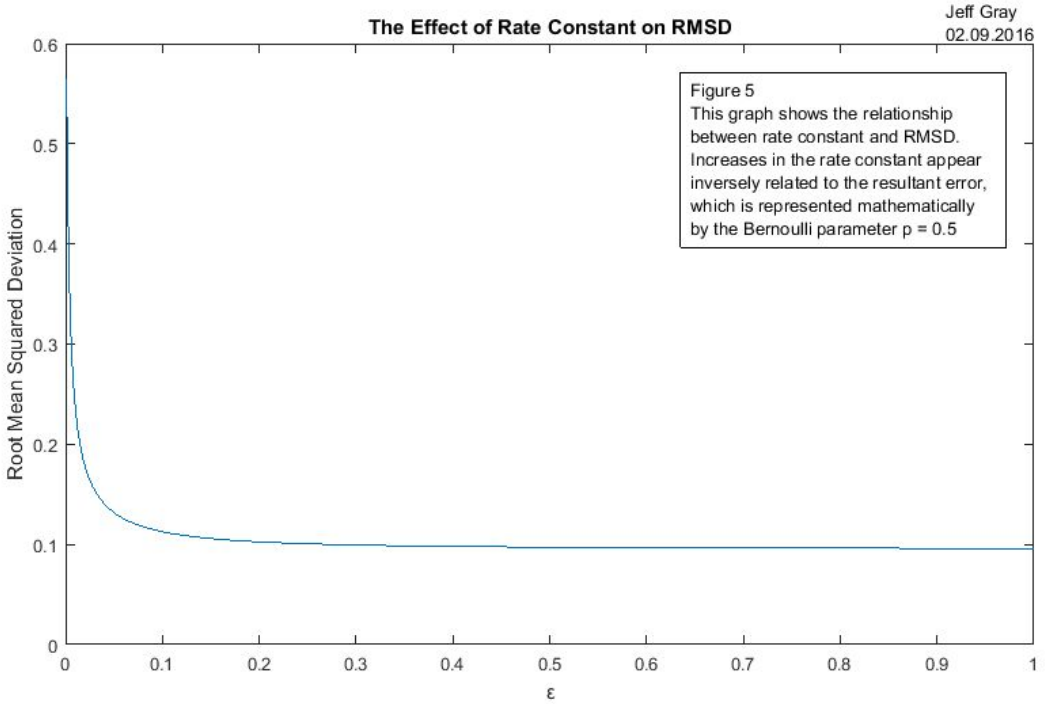
3.2.2



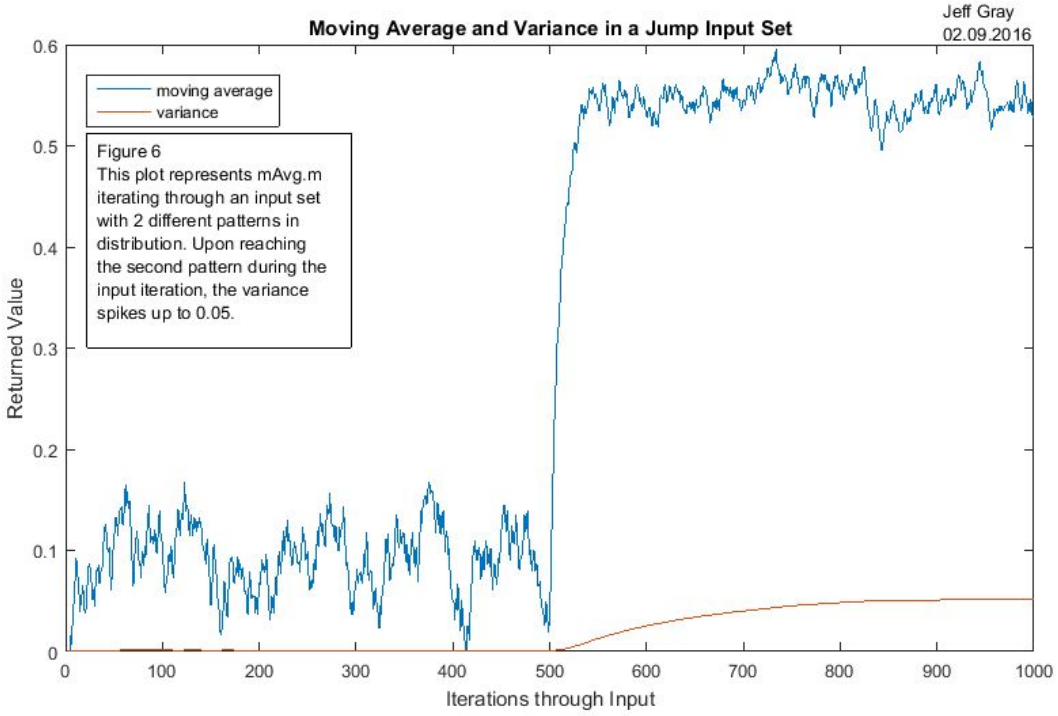
3.3B.1


Extreme Convergence of a Moving Averager to p=0.8
Jeff Gray
02.09.2016
Value of Moving Averager
Iterations
×10^5
Figure 4
This plot demonstrates convergence within the last 100 iterations of mAvg on a random dataset centered at 0.8.

3.3B.2


The Effect of Rate Constant on RMSD
Jeff Gray
02.09.2016
Root Mean Squared Deviation
ε
Figure 5
This graph shows the relationship between rate constant and RMSD. Increases in the rate constant appear inversely related to the resultant error, which is represented mathematically by the Bernoulli parameter p = 0.5

3.6B


Moving Average and Variance in a Jump Input Set
Jeff Gray
02.09.2016
moving average
variance
Figure 6
This plot represents mAvg.m iterating through an input set with 2 different patterns in distribution. Upon reaching the second pattern during the input iteration, the variance spikes up to 0.05.
Returned Value
Iterations through Input
0
0.1
0.2
0.3
0.4
0.5
0.6
0
100
200
300
400
500
600
700
800
900
1000

Workspace: https://github.com/jeffreygray/nesc5330/tree/master/lab3

CODE

3.1: tAvg.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: calculates true average

function [trueAvg] = tAvg(inputVec)
%    clear;
    clc;

    %   error checking
    if (size(inputVec, 2) > 1)
        disp('ERROR: input is not a column vector!')
        disp('continuing anyway...')
    end

    curAvg = 0; %   base case

    %   populate vector trueAvg by iterating through inputVec
    for i = 1 : length(inputVec)
        curAvg = (curAvg*(i-1) + inputVec(i,1))/i;
        trueAvg(i,1) = curAvg;
    end
```

3.1: createfigure1.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: creates figure 1

function createfigure1()
clf;
clc;
clear;

rng(9711963);
randVec = rand(1000,1);
input = tAvg(randVec);

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'on');

% Create plot
plot(input);
```

```
% Create xlabel
xlabel('Iterations of tAvg','FontSize',11);

% Create ylabel
ylabel('Value of True Averager','FontSize',11);

% Create title
title('Iterations of Function tAvg and Associated Outputs','FontSize',11);

% Create textbox
annotation(figure1,'textbox',...
    [0.826985854189337 0.914122139338318 0.0946681150220036 0.0839694637151165],...
    'String',{'Jeff Gray','02.09.2016'},...
    'FitBoxToText','off',...
    'EdgeColor','none');

% Create textbox
annotation(figure1,'textbox',...
    [0.710089849214986 0.134187576858853 0.181719254973923 0.311068693588253],...
    'String',{'Figure 1','This graph displays the','outputs of the tAvg','function using an
input','of 1000 random numbers','from 0 to 1. The mean','inflects frequently during','the
first 30 iterations, but','gradually stabilizes.'});

disp('checking to see if output is valid...')
actual = mean(randVec(1:1000),1)
calculated = input(1000,1)
```

3.2.1: mAvg.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: calculates moving average

function [movingAvg] = mAvg(inputVec)
%    clear;
    clc;

    %   error checking
    if (size(inputVec, 2) > 1)
        disp('ERROR: input is not a column vector!')
        disp('continuing anyway...')
    end

    movingAvg(1,1) = 0;      %   base case
    epsilon = .08;            %   setting rate constant, could be specified as parameter

    %    populate vector movingAvg while iterating through inputVec
    for i = 1 : length(inputVec) - 1
```

```
        movingAvg(1, i+1) = movingAvg(1, i) + epsilon * (inputVec(i,1) - movingAvg(1,i));
    end
```

3.2.1: createfigure2.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: creates figure 2

function createfigure2()
clf;
clc;
clear;

rng(9711963);
randInput = rand(1000,1);
input = tAvg(randInput);

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
%% Uncomment the following line to preserve the X-limits of the axes
% xlim(axes1,[0 1000]);
%% Uncomment the following line to preserve the Y-limits of the axes
% ylim(axes1,[0 0.8]);
box(axes1,'on');
hold(axes1,'on');

% Create plot
plot(tAvg(randInput));
hold on
plot(mAvg(randInput));
hold on
plot([0 1000], [mean(randInput) mean(randInput)])
hold on
plot([0 1000], [mean(randInput)+std(randInput) mean(randInput)+std(randInput)])
hold on
plot([0 1000], [mean(randInput)-std(randInput) mean(randInput)-std(randInput)])
axis([0 1000 0 .8])

% Create xlabel
xlabel('Iterations','FontSize',11);

% Create ylabel
ylabel('Value of Averager','FontSize',11);

% Create title
title('Behavior of Exact vs Moving Averager','FontSize',11);
```

```
% Create textbox
annotation(figure1,'textbox',...
    [0.539535675423598 0.169859314076119 0.349292699381369 0.244318175101371],...
    'String',{'Figure 2','This plot displays the output of functions mAvg','and tAvg with the
same input set of random','numbers. Although both averagers behave similarly','under100
iterations,the moving average becomes','periodic and begins to sinusoid while the
exact','averager results in a tighter fit with the mean line.'});

legend(axes1,'show');

% Create textbox
annotation(figure1,'textbox',...
    [0.825897714907509 0.910984850403937 0.0946681150220036 0.0833333314142445],...
    'String',{'Jeff Gray','02.09.2016'},...
    'EdgeColor','none');
```

3.2.2: mAvgEps.m

simply change input function to take epsilon value as parameter, modify mAvg.m

3.2.2: converge.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: runs problem 3.2.2

function converge(input, eps1, eps2, eps3)
clf;
clc;
rng(9711963);
figure1 = figure;

plot(mAvgEps(input, eps1))
hold on
plot(mAvgEps(input, eps2))
hold on
plot(mAvgEps(input, eps3))

axis([0 100 0 1.1])
xlabel('Iterations')
ylabel('Value of Moving Averager')
title('Convergence of Moving Averager')
annotation(figure1,'textbox',...
    [0.661500544069641 0.426691736941946 0.230685521196878 0.274436082606925],...
    'String',['Figure 3',sprintf('\n'),'Given an input set of 100 ones,',sprintf('\n'),'the
highest rate contant in blue',sprintf('\n'),'reflects its adaptability by
quickly',sprintf('\n'),'inflecting to the convergent value',sprintf('\n'),'of 1. The lower the
```

```
rate constant',sprintf('\n'),'drops, the less able it is to',sprintf('\n'),'account for
change.']);
annotation(figure1,'textbox',...
    [0.829142002176279 0.900877192982455 0.140071428571429 0.0928571428571456],...
    'String',{'Jeff Gray','02.09.2016'},...
    'FitBoxToText','off',...
    'EdgeColor','none');
```

3.3B.1: averagingSim.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: runs problem 3.3B.1

function averagingSim()
    clc;
    clf;
    figure1 = figure;

    randNums = rand(1000000, 1) + 0.3;    %   generating 1m random numbers centered at 0.8

    %   plotting
    plot(randNums)
    hold on
    line([0 1000000], [.8, .8])
    hold on

    %   formatting plot and adding text
    axis([1000000-100 1000000 0 1.3])
    annotation(figure1,'textbox',...
    [0.826985854189336 0.938432835820895 0.0891392818280741 0.0541044776119401],...
    'String',{'Jeff Gray','02.09.2016'},...
    'FitBoxToText','off',...
    'EdgeColor','none');
annotation(figure1,'textbox',...
    [0.714866003419867 0.132924668073611 0.176278558726928 0.20895521820926],...
    'String',{'Figure 4','This plot demonstrates','convergence within the','last 100
iterations of','mAvg on a random','dataset centered at 0.8.'});
    title('Extreme Convergence of a Moving Averager to p=0.8')
    xlabel('Iterations')
    ylabel('Value of Moving Averager')
```

3.3B.1: bernoulli.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: runs problem 3.3B.2

function bernoulli()
```

```
    clc;
    clf;
    rng(9711963);
    figure1 = figure;

    randNums = rand(1000,1);
    epsRanges = [.001:.001:1]';

    for i = 1 : 1000
        output(i, 1) = sqrt(mean((.5 - mAvgEps(randNums, epsRanges(i, 1)))));
    end

    plot(epsRanges, output)
    xlabel('?')
    ylabel('Root Mean Squared Deviation')
    title('The Effect of Rate Constant on RMSD')
    annotation(figure1,'textbox',...
    [0.826985854189336 0.933823529411765 0.105461371055495 0.0588235294117647],...
    'String',{'Jeff Gray','02.09.2016'},...
    'FitBoxToText','off',...
    'EdgeColor','none');

% Create textbox
annotation(figure1,'textbox',...
    [0.622327529923831 0.648897065342787 0.26115342018005 0.237132346421919],...
    'String',{'Figure 5','This graph shows the relationship','between rate constant and
RMSD.','Increases in the rate constant appear','inversely related to the resultant
error,','which is represented mathematically','by the Bernoulli parameter p = 0.5'});
```

3.6B: jumpVar.m

```
%   jeff gray
%   jhg7nm
%   lab3
%   description: runs problem 3.6B

function jumpVar
    %   standard clear routine
    clc;
    clf;
    rng(9711963);
    figure1 = figure;

    %   defining variables
    set1 = rand(500, 1) * .6 - .2;
    set2 = rand(500, 1) * .3 + .4;
    set = [set1; set2];
    data = mAvg(set)';

    %   plotting
```

```
    plot(mAvg(set))
    hold on

    for i = 1 : 1000
        varData(i, 1) = var(data(1:i));
    end
    plot(varData)
    xlabel('Iterations through Input')
    ylabel('Returned Value')
    title('Moving Average and Variance in a Jump Input Set')

    % Create legend
    legend1 = legend('show');

    annotation(figure1,'textbox',...
    [0.146810663764963 0.51824817518248 0.211187159956474 0.288321167883212],...
    'String',{'Figure 6','This plot represents mAvg.m','iterating through an input set','with
2 different patterns in','distribution. Upon reaching','the second pattern during the','input
iteration, the variance','spikes up to 0.05.'},...
    'FitBoxToText','off');
    annotation(figure1,'textbox',...
    [0.824721436343853 0.917883211678831 0.0871392818280742 0.0766423357664232],...
    'String',{'Jeff Gray','02.09.2016'},...
    'FitBoxToText','off',...
    'EdgeColor','none');
    axis([0 1000 0 .6])
```