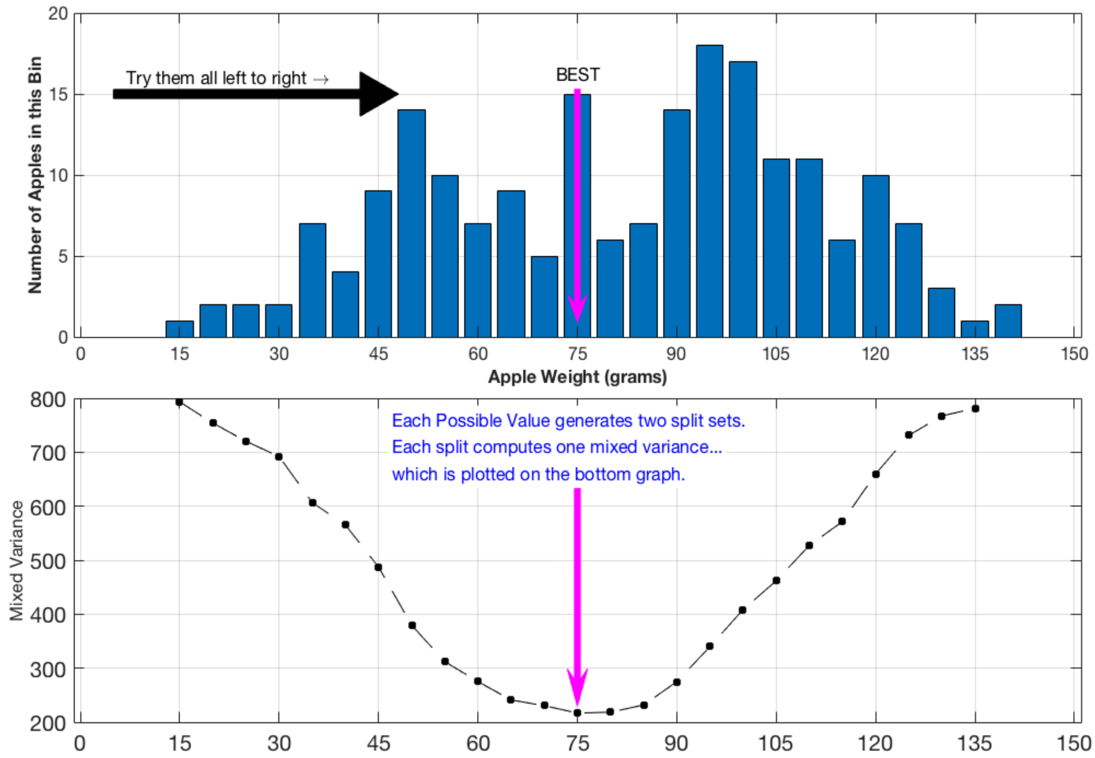
```
function HW_01_Classify_Apples()
%
%  A working example of finding the best threshold for
%  some apple data.
%
%  I am doing this in MATLAB.  You try it in
%  another language. :-)
%
%  Thomas B. Kinsman,
%  '04-Sep-2020'
%
MYFS = 16;

    %
    % We cluster to learn the structure of the data.
    %
    % Here we will load the data, and form a histogram of the data.
    %
    data_table =
 readtable( 'HW_01_CS420_Apple_Weights_Unclustered.csv' );

    % The first column is just a record id.
    %
    % So, ignore column ONE.
    %
    % Get the variable values out of column two:
    %
    % Your data will have the column you need to pay attention
    % to in another place.   Look at your data by pulling it
    % into Excel, or printing the top five lines of the file.
    the_raw_weights = data_table(:,2).Variables;

    %  Now we quantize this data, by putting it into bins.
    %  Why?
    %
    %  BECAUSE binning the data is a form of NOISE REDUCTION.
    %  It means that we do not have to worry about small
    %  changes in the data.
    %
    %  Let us imagine that we are using bins of size 5 grams:
    BIN_SIZE = 5;
    quantized_data = floor(the_raw_weights/BIN_SIZE) * BIN_SIZE;
```

```
   % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %
   %  CREATE AND FORM A HISTOGRAM
   %
   %  Let's form a histogram to see what it looks like:
   edges_for_the_histograms    = (BIN_SIZE/2):BIN_SIZE:150;
   centers_of_each_range       = BIN_SIZE:BIN_SIZE:145;
   [hist_counts,hist_bins]     = histcounts( quantized_data,
edges_for_the_histograms );

   %  Show a histogram:
   figure( 'Position', [10 10 1024 768]);
   hTop = subplot(2,1,1);
   bar( centers_of_each_range, hist_counts );
   set( gca, 'Position', [0.075 0.55 0.90 0.42] );
   % Make the axis numbers bigger for reading.
   set( gca, 'FontSize', MYFS );
   xlabel('\bfApple Weight (grams)', 'FontSize', MYFS );
   ylabel('\bfNumber of Apples in this Bin', 'FontSize', MYFS );
   grid on;
   set(gca,'XTick', 0:15:150 );


   arrow( [5,15], [48,15], 'Width', 10, 'Length', 35, 'TipAngle',
35 );
   text( 7, 16, 'Try them all left to right \rightarrow', 'FontSize',
MYFS );







   % Continued on next page....
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  COMPUTE THE MIXED VARIANCE FOR EACH POSSIBLE TO SPLIT THE DATA
%  INTO SPLIT INTO TWO GROUPS. THEN SEE WHAT THE AVG VARIANCE OF
%  THE TWO GROUPS IS.
%
%  WE WANT TO FIND THE MINIMUM AVERAGE VARIANCE,
%  so find that using a linear search ... try them all.
%
%
%
%  Now we know what we are working with.
%
%  We are going to split the apples into two groups.
%  The question, is, what is the best way to split them up?
%  The answer is -- we do not know.
%
%  The only way to find out is to TRY THEM ALL.
%  This uses a linear search to find the answer.
%

% For every possible weight to split the apples into:
%
% But first initialize the variables to sentinel values:
% This is a bogus, sentinel value.
best_minimum_mixed_variance_ever = Inf;
best_value_yet                   = Inf;

for counter = 1 : numel( centers_of_each_range )

    % Which value will we split on this time through?
    splitting_weight = centers_of_each_range(counter);

    % Find the indices of the values on the left side:
    %
    % This is a boolean variable that is 1 (or True) if this
    % data item (i.e. this particular apple) has a weight <= the
splitting_weight.
    %
    % We set this to '1' to indicate that this item would go in
the left hand group.
    %
    % Similarly for the right hand values.
    %
    boolean_indices_of_the_left__hand_values = quantized_data <=
splitting_weight;
    boolean_indices_of_the_right_hand_values = quantized_data >
splitting_weight;

    % Using those boolean variables,
    % Get the actual values for the data on the left side:
    set_of_left__hand_values     =
quantized_data( boolean_indices_of_the_left__hand_values );
```

```
        set_of_right_hand_values     =
quantized_data( boolean_indices_of_the_right_hand_values );

        % What is the fraction of the data in the left hand set?
        % It is the number of elements in the left set, divided by the
number of all elements:
        Wleft       = numel( set_of_left__hand_values ) /
numel( quantized_data );
        Wright      = numel( set_of_right_hand_values ) /
numel( quantized_data );

        % How mixed up are each set of data?
        % We use the Variance of the sets as a measure of how mixed up
they are.
        VarianceLeft    = var( set_of_left__hand_values );
        VarianceRight   = var( set_of_right_hand_values );

        % Evaluate this splitting point by computing the mixed
variance:
        MixedVariance(counter) = Wleft * VarianceLeft + Wright *
VarianceRight;
        if ( MixedVariance(counter) <
best_minimum_mixed_variance_ever )
            best_minimum_mixed_variance_ever    =
MixedVariance(counter);
            best_value_yet                      = splitting_weight;
        end

    end


    %  THAT's IT.
    %
    %  THAT's HOW TO IMPLEMENT OTSU's METHOD.
    %
    %  Everything else is for decoration.
    %

    % Add an arrow where the best splitting point is:
    haxTop = axis();
    arrow( [ best_value_yet haxTop(4)*0.9 ], ...
           [ best_value_yet haxTop(3)+(haxTop(4)-
haxTop(3))*0.05 ], ...
            'Color', 'm', 'Width', 5, 'BaseAngle', 35, 'Length', 30 );
    text( best_value_yet, haxTop(4)*0.9, 'BEST', ...
        'HorizontalAlign', 'Center', ...
        'BackGroundColor', 'w', ...
        'FontSize', MYFS, 'Color', 'k' );
    set(gca,'XTick', 0:15:150 );

    fprintf('Best Value to split at is : %4.2f grams\n',
best_value_yet );
    fprintf('Best Mixed Variance is    : %6.2f\n',
best_minimum_mixed_variance_ever );
```

```
    % Now Plot the reult:
    hBot = subplot(2,1,2);
    plot( centers_of_each_range, MixedVariance, 'ks--', ...
        'MarkerFaceColor', 'k', ...
        'LineWidth', 1 );

    % Add an arrow where the best splitting point is:
    haxBot = axis();
    arrow( [ best_value_yet haxBot(4)*0.9 ], [ best_value_yet
 haxBot(3)+(haxBot(4)-haxBot(3))*0.05 ], ...
            'Color', 'm', 'Width', 5, 'BaseAngle', 35, 'Length', 30 );

    % Line the axes up with each other.
    haxBot( 1:2 ) = haxTop( 1:2 );
    axis( haxBot );

    set( gca, 'Position', [0.075 0.05 0.90 0.42] );
    set( gca, 'FontSize', 20 );
    xlabel('Apple Weight (grams)', 'FontSize', MYFS );
    ylabel('Mixed Variance', 'FontSize', MYFS );
    set(gca,'XTick', 0:15:150 );
    grid on;

    text( 47, 760, 'Each Possible Value generates two split
 sets.', ...
                   'FontSize', MYFS, ...
                   'Color', 'b', 'BackgroundColor', 'w' );
    text( 47, 710, 'Each split computes one mixed variance...',     ...
                   'FontSize', MYFS, ...
                   'Color', 'b', 'BackgroundColor', 'w' );
    text( 47, 660, 'which is plotted on the bottom graph.',       ...
                   'FontSize', MYFS, ...
                   'Color', 'b', 'BackgroundColor', 'w' );


    capture_graph( 'Fig_Answer_For_Apples.png', 'PNG', 32, 'w', 0,
 0 );
end

Best Value to split at is : 75.00 grams
Best Mixed Variance is    : 216.93
```



Published with MATLAB® R2018a