Chem/Stat3240: Homework 10a
Matlab

November 2, 2015

1. Suppose you have a sensor array that generates measurements of ambient temperature, wind velocity, and relative humidity every hour and saves it to a 3-column matrix where each column represents a particular set of measurements in the order given above. Suppose the matrix is generated every 20 hours.

   The problem is that when the temperature gets below freezing and the relative humidity is high, the wind speed sensor might freeze. But you don't want to delete all the data values where the temperature is below freezing. Instead, look for data where the temperature was sufficiently low and the relative humidity sufficiently high, and see if the wind speed was very low which could potentially mean the sensor was frozen. That way, you don't have to discard all data at low temperatures.

   Write a function so that it performs as specified:

```
function data_out= frozenSensor(sensor_data,thresholds)
% The function frozenSensor determines if a given set of
% wind, temperature, and humidity measurements are unreliable
% based on a set of conditions and removes them from the original
% data set.
% sensor_data = matrix of sensor measurements
% thresholds = vector of thresholds for sensor measurements
% data_out = Cell array containing the new data set matrix and
% and the discarded data matrix.
```

   Your implementation should not use any loops by making effective use of the `find` command. Remember that an element of a vector can be deleted by setting the element equal to [] (empty brackets). Also

remember that a subset of a vector x can be specified by using a vector of indices (either integer or logical vectors) as the argument of the vector x.

The following are examples of running the function frozenSensor in a script 3 times and displaying the output. You are only responsible for having your frozenSensor function pass the Cody tests.

```
>> hwk7_1
cleaned data
    13     0    52
   -15     6    68
    16    10    50
    -6     8    68
     4     6    61
     3     8    52
     7     8    53
     6     0    64
   -15     7    50
    10     5    65
   -11    10    70
     6     8    57
    15     6    63
   -17     3    57
    19     2    54
   -19     9    66
    14     4    65
    14     8    55
discarded data
   -17     3    68
    -3     3    67
```

```
>> hwk7_1
cleaned data
    -5     9    61
    20     2    64
     9     6    57
    -4     5    51
   -16     4    59
    10     7    51
     6    10    65
   -17     6    50
    20     2    65
   -18     9    55
   -15     6    55
    16     2    65
     2     0    57
     0     9    65
   -18     4    52
discarded data
    -7     3    61
   -16     3    70
     0     4    69
   -20     0    60
    -1     5    70
```

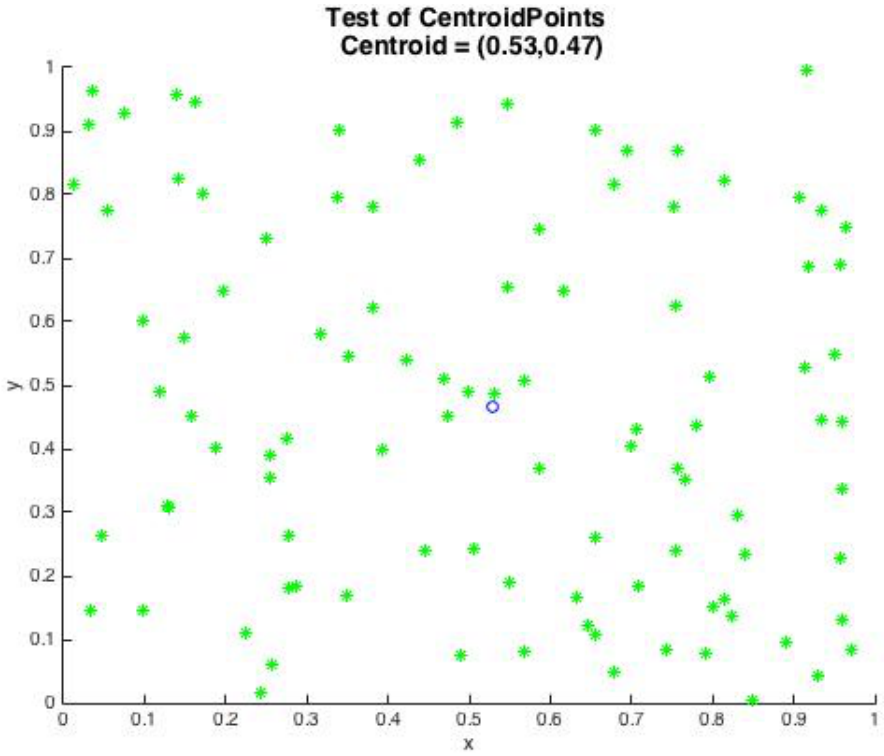
```
>> hwk7_1
cleaned data
     9     9    53
   -12     1    55
   -12     3    51
    20     2    62
     4     6    66
    -8     6    60
    19     3    50
    16     8    70
   -13     8    60
   -20     8    56
     9     3    53
    15     0    63
   -16     7    62
   -19    10    65
     4     1    53
     4     1    57
     1    10    53
    10     5    67
     8     9    57
    18     8    55
no data was discarded
```

2. Download the script `Eg10_1` and functions `MakePoint`, `GetDist`, `DiameterPoints`, and `ShowPointSet` from the Matlab Example Code/Structures folder of the course Collab site. You can use the functions (or code from them) in the assignment.

   Write a function `CentroidPoints` that takes a structure (list) array of points P and returns a point Q that is the centroid of the set of points contained in P. Here is the Wikipedia definition of the centroid of a finite set of points:
   `http://en.wikipedia.org/wiki/Centroid#Of_a_finite_set_of_points`

   Remember you can't add the point structures directly. Your function should also use a circle marker to plot the centroid on that same plot with the point set and the diameter of the point set. Include the coordinates of the computed centroid in the plot title as shown below.

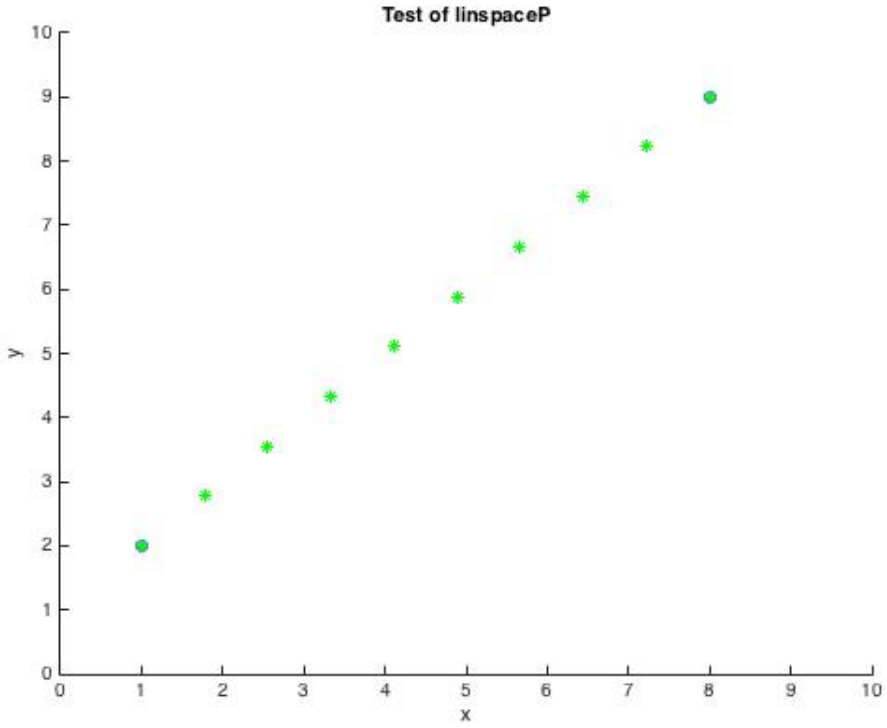
Submit the function `CentroidPoints` to Cody and to the collab site and a pdf of the plot to the collab site



3. Complete the following function so that it performs as specified:

```
function P = linspaceP(Q1, Q2, n)
%  Q1 and Q2 are points.
%  n is an integer >= 2
%  P is a length n structure array of points with the property that
%   P(1),...,P(n) are equally spaced along the line segment
%   that connects Q1 and Q2. Note: P(1) = Q1 and P(n) = Q2.
```

The function should also plot Q1 and Q2 as blue circle markers, and the points in P as green asterisks (*) as shown below and save the plot as a pdf. Submit the function `linspaceP` to Cody and to the collab site and a pdf of the plot to the collab site.


Test of linspaceP
y
x
0
1
2
3
4
5
6
7
8
9
10