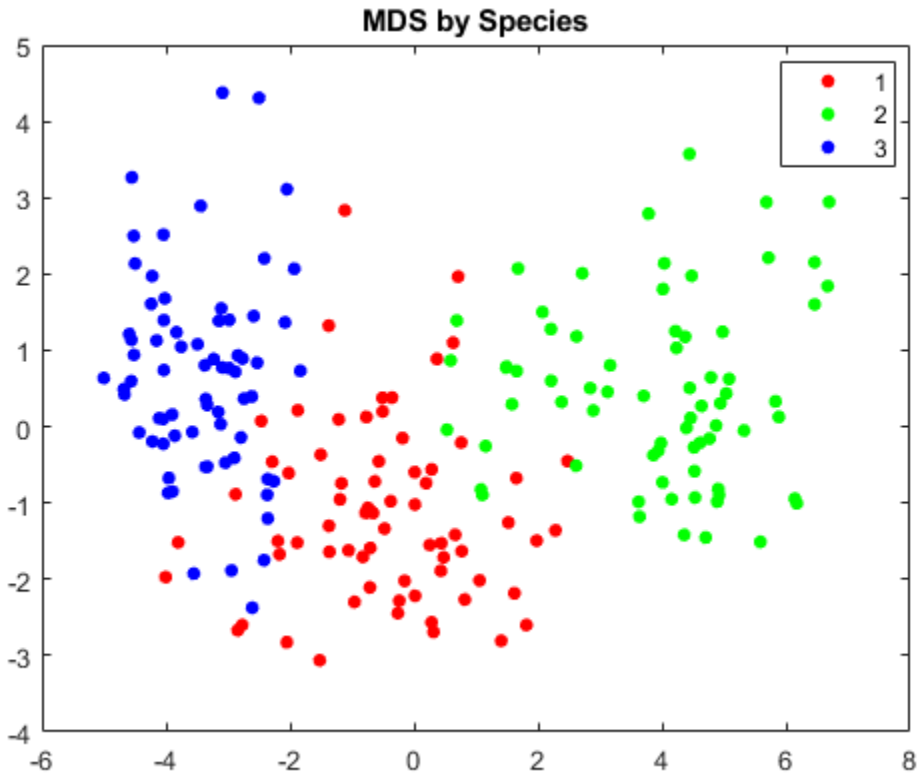
```
%Name: David George
%Student ID: 251004930

%Part A)
    T = readtable("seeds.csv");

    D = table2array(rmmissing(T(:,[1:7])));
    D = squareform(pdist(D));

    %Classical MDS
    MDS  = cmdscale(D, 2);
    gscatter(MDS(:,1), MDS(:,2),T.species);
    title("MDS by Species");
```




```
%Part B)

  methods = [ "complete" "centroid" ];

  %Performing complete and centroid clustering using a for loop
   figure
       for idx = 1:2
           rng(1234);
           subplot(2,2, idx);
           m = methods(idx);
           L = linkage(MDS,m);
```

```
            C = cluster(L,'Maxclust',3);

            C = num2str(C);

           %Plotting the clustering, using the grouping found by the
           %spefifc clustering method

            gscatter(MDS(:,1), MDS(:,2),C)
           title(methods(idx),'FontSize',12);

         GroupClassificaition = table2array(T);
         GroupClassificaition = GroupClassificaition( :, 8);
         GroupClassificaition = num2str(GroupClassificaition);

        for idx = 1:length(T.species)

            %labelling all the points of the specifc clustering with
the
            %actual group from the data set
            text(MDS(idx,1),MDS(idx, 2),  GroupClassificaition(idx));
        end


        end


        %K means clustering
        rng(1234);
        [idxCluster, centroids] = kmeans(MDS,3);

        % Color the data points wih their respective cluster:
        subplot(2,2, 3);

        idxCluster = num2str(idxCluster)

        gscatter(MDS(:,1), MDS(:,2),idxCluster)
        title("k-means Clustering",'FontSize',12)

        for idx = 1:length(idxCluster)
            %labelling all the points of the specifc clustering with
the
            %actual group from the data set
            text(MDS(idx,1),MDS(idx, 2), GroupClassificaition(idx));

         end

        rng(1234);
        spclust = spectralcluster(MDS,3);

        subplot(2,2, 4);
         spclust  = num2str(spclust);
         gscatter(MDS(:,1), MDS(:,2),spclust);
        title("Spectral Clustering",'FontSize',12)
```

```
        for idx = 1:length(idxCluster)
            %labelling all the points of the specifc clustering with
 the
            %actual group from the data set
            text(MDS(idx,1),MDS(idx, 2), GroupClassificaition(idx));
        end
```

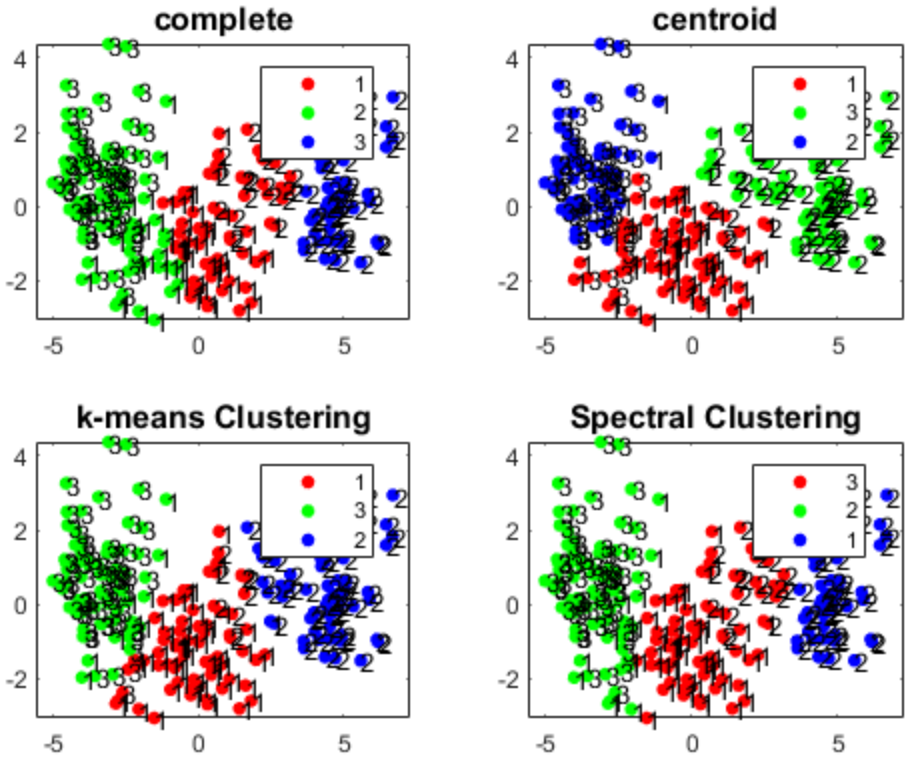
```
Warning: Non-monotonic cluster tree -- the centroid linkage is
 probably not
appropriate.

idxCluster =

  210×1 char array

    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '3'
    '1'
    '1'
    '3'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '3'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '1'
    '2'
```

```
'1'
'3'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'1'
'3'
'3'
'3'
'3'
'1'
'1'
'1'
'1'
'1'
'3'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
```

```
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'1'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'1'
'2'
'1'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'2'
'1'
'1'
'1'
'1'
'2'
'1'
'1'
'1'
'3'
'3'
'3'
'3'
'3'
'3'
```

```
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'1'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
```

```
'3'
'1'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
'3'
```



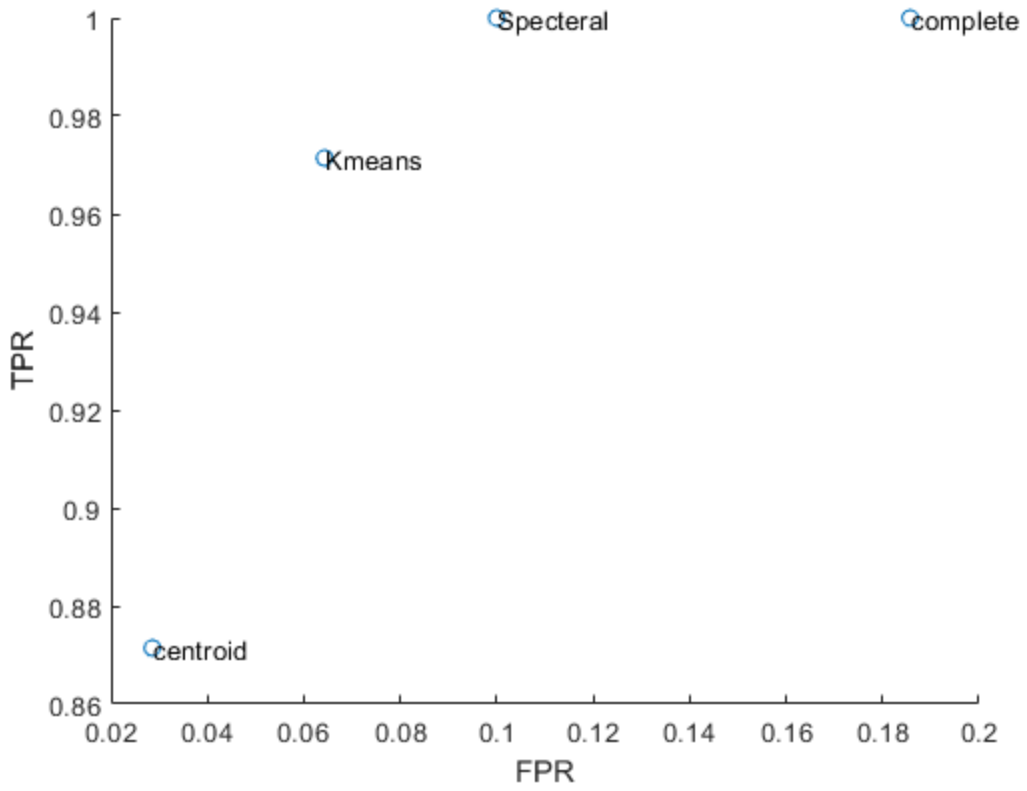

```
  %Part C

        %Conduct each clustering method, use the cluster grouping
array
        % Compare it to the original data's results to determine
results.

        %Isolating the eigth column of the data
        GroupClassificaition = table2array(T);
        GroupClassificaition = GroupClassificaition( :, 8);
        rng(1234);

        %Performing specteral and kmeans clustering once more
        spclust = spectralcluster(MDS,3);
```

```
        rng(1234);
        [idxCluster, centroids] = kmeans(MDS,3);

        methods = [ "complete" "centroid" ];
        L = linkage(MDS,methods(1));
        rng(1234);
        C = cluster(L,'Maxclust',3);

        % Compare it to the original data's results to determine
rates.
        [CompleteTPR, CompleteFPR] =
calculateRate(GroupClassificaition, C, "co");

        L = linkage(MDS,methods(2));
        C = cluster(L,'Maxclust',3);
        rng(1234);
        % Compare it to the original data's results to determine
rates.
        [CentroidTPR, CentroidFPR] =
calculateRate(GroupClassificaition, C, "ce");
        % Compare it to the original data's results to determine
rates.
        [KmeansTPR, KmeansFPR] = calculateRate(GroupClassificaition,
idxCluster, "k");
        % Compare it to the original data's results to determine
rates.
        [specterTPR, specterFPR] =
calculateRate(GroupClassificaition,spclust, "s");

        %Plotting the data
        TPR = [CompleteTPR, CentroidTPR, KmeansTPR, specterTPR];
        FPR = [CompleteFPR, CentroidFPR, KmeansFPR, specterFPR];
        figure
        methods = [ "complete" ,"centroid", "Kmeans", "Specteral" ];

        scatter(FPR, TPR);
        text(FPR(1), TPR(1),"complete");

        text(FPR(2), TPR(2),"centroid");
        text(FPR(3), TPR(3),"Kmeans");
        text(FPR(4), TPR(4),"Specteral")

        xlabel("FPR");
        ylabel("TPR");

    %Based on the criteria of having a TPR of at least 95%, and
minnimzing
    %the FPR, the best method is therefore K means clustering as it
meets
    %the criteria of at least 0.95 TPR and its FPR is the smallest
amongst all
    %the possible options

  function [TPR, FPR] = calculateRate(T, C, type)
```

```
%Basic idea of this funciton is as follows
%compare specfic clustering method results, to the actual data
% results. INcrement tp, fn, fp, and tn accordingly
%Use these results to calcualte TPR and FPR

tp = 0;
fn = 0;
fp = 0;
tn = 0;

if type == "co"
   num =2;
end

if type == "ce"
   num =2;
end

if type == "k"
   num =3;
end
if type == "s"
   num =2;
end


for idx = 1:length(T)

    if C(idx) == num & T(idx) == 3

        tp = tp +1;

    end

    if C(idx) ~= num & T(idx) == 3

        fn = fn +1;

    end

     if C(idx) == num & T(idx) ~= 3

        fp = fp +1;
     end

       if C(idx) ~= num & T(idx) ~= 3

       tn = tn +1;

    end
end
```

```
        %This if statments are used to ensure NaN is not returned if
 the
        %numerator is 0.

        if tp == 0 & fp~=0
            TPR =0;
            FPR = fp / (fp + tn);
            return;
        end

        if fp == 0 & tp~=0
           FPR =0;
           TPR = tp /(tp +fn);
           return;
        end


        if fp ~= 0 & tp~=0
          FPR =fp / (fp + tn);
          TPR = tp /(tp +fn);
          return;
        end


        if fp == 0 & tp ==0
          FPR = 0;
          TPR = 0;
          return;
        end



    end
```

```
Warning: Non-monotonic cluster tree -- the centroid linkage is
 probably not
appropriate.
```



Published with MATLAB® R2019b