Abelardo Riojas
ISC 4221C Prof. Quaife
Fall 2019

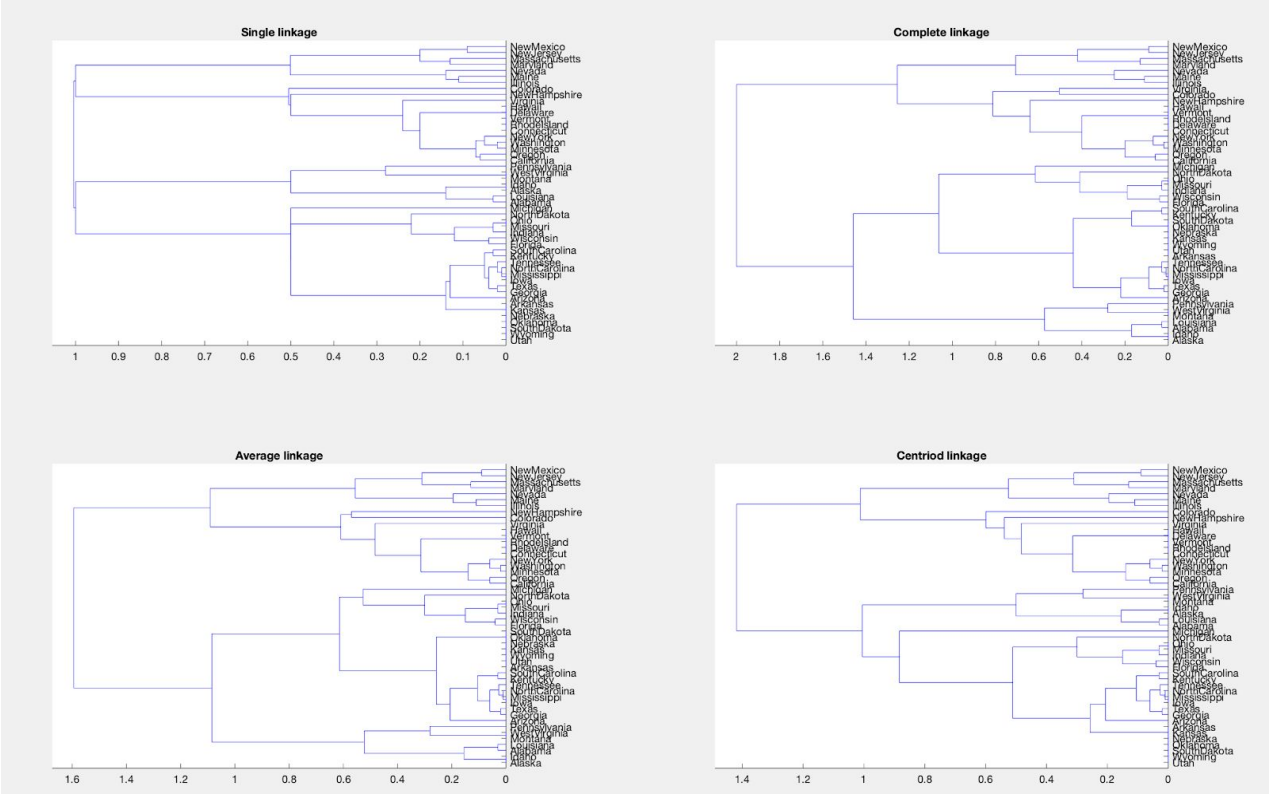
Lab 4 Report

1. Hierarchical clustering of 2016 Election Data
   i. Load the file states.txt and separate the state names from the percentages.

```
filename = fopen('states.txt');
data = textscan(filename, '%s%f%f%f%f');
states = data{1};
senators = data{2};
congressmen = data{3};
governors = data{4};
electoralCollege = data{5};
```

   ii. Using the Euclidean distance, compute the dendrogram tree using
   • single linkage, • complete linkage, • centroid linkage, • average linkage

```
A = [senators congressmen governors electoralCollege];
sv = pdist(A);
sl = linkage ( sv, 'single' );
comp = linkage ( sv, 'complete' );
average = linkage ( sv, 'average' );
centroid = linkage(sv, 'centroid' );
```



iii. Comment on any common characteristics between the states.

One characteristic between the states is that, because of the two party system, the clusters with the second most amount of states are two big clusters representing the Democratic and Republican controlled states. Another characteristic is that the smallest clusters tend to have states that are extremely similar in party afilliation, often down to the congressman.

2. Image Compression with k-means

i. Obtain approximations to the image using k = 2, k = 4, and k = 8 colors. For a stopping criteria, use 10 steps of Lloyd's algorithm. Compute and plot the energy versus the iteration number on a linear-linear graph. Comment on its behavior.

(I did Lloyd's algorithm until convergence, whoops!)

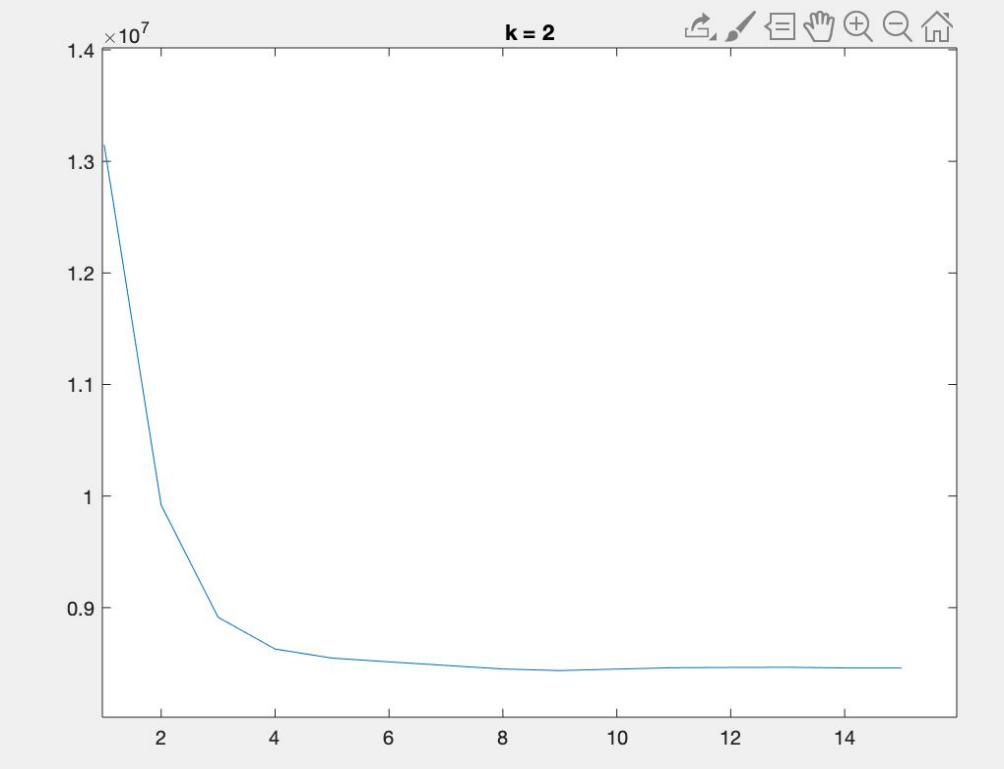
Code:

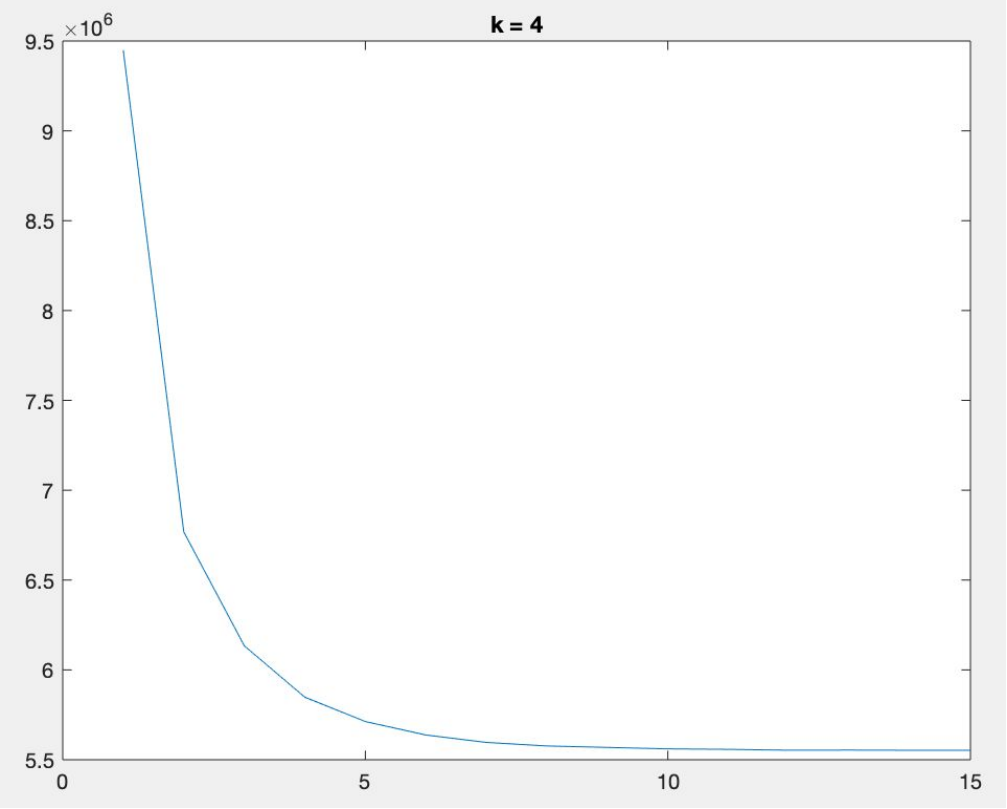
```
pixel = imread('mandrill.tiff');
%pixel = imread('gg.jpg');

k = 8;
centriods = randi(256, k, 3) - 1;

x = size(pixel,1);
y = size(pixel,2);

distances = zeros(x,y,k);


dispcolor(centriods);
sgtitle('Original guesses');

% For every pixel, calculate the distance from the centroid color
to the
% pixel color, then find the closest centroid.
count = 0;
conv = 0;
energies = [];
while (conv == 0)
closestcentriod = zeros(x,y);
mindists = zeros(1,k);
dist = 0;
for i = 1:x
    for j = 1:y
        p1 = double(pixel(i,j,1));
```

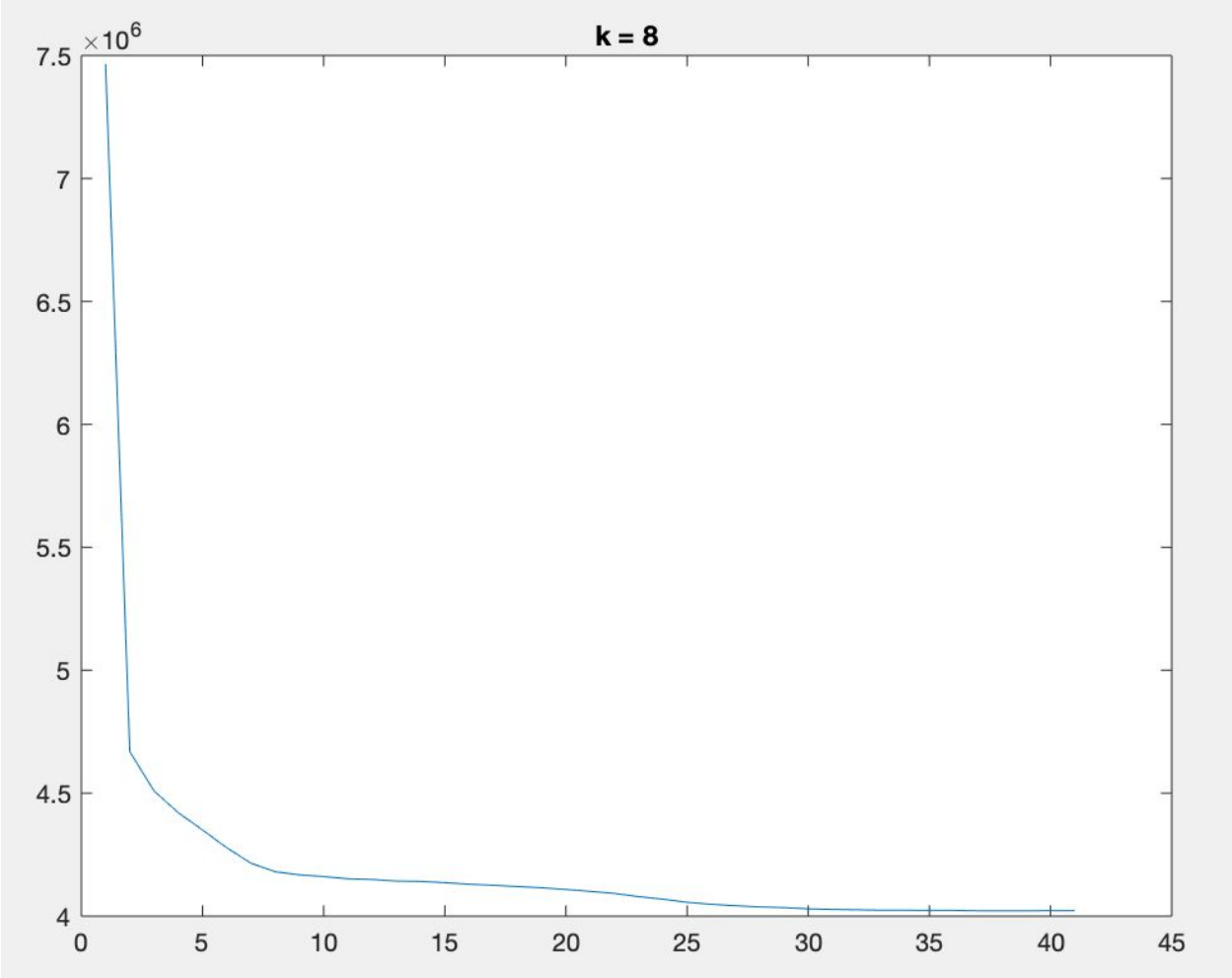
```
        p2 = double(pixel(i,j,2));
        p3 = double(pixel(i,j,3));
        for c = 1:k
            c1 = centriods(c,1);
            c2 = centriods(c,2);
            c3 = centriods(c,3);
            distance = (p1-c1)^2 + (p2-c2)^2 + (p3-c3)^2;
            distances(i,j,c) = sqrt(distance);
        end
        [mindist, centriod] = min(distances(i,j,:));
        closestcentriod(i,j) = centriod;
        mindists(1,centriod) = mindists(1,centriod)+mindist;
    end
end

energies(end+1) = sum(mindists)/2;

% Change the centriod to the average value of each
% centriod's closest pixel.
for i = 1:k
    [row, col] = find(closestcentriod==i);
    pixelnum = numel(row);
    cluster = zeros(pixelnum,3);
    for xi = 1:pixelnum
        cluster(xi,1) = pixel(row(xi), col(xi),1);
        cluster(xi,2) = pixel(row(xi), col(xi),2);
        cluster(xi,3) = pixel(row(xi), col(xi),3);
    end
    centriods(i,1) = ceil(mean(cluster(:,1)));
    centriods(i,2) = ceil(mean(cluster(:,2)));
    centriods(i,3) = ceil(mean(cluster(:,3)));

end
centriods(isnan(centriods)) = 0; %random bug where 0's would
appear as NaN.
count = count + 1;
%check for convergence
if (numel(energies) >= 2)
    if (abs(energies(end-1) - energies(end))) < .001
        conv = 1;
    end
end
end


dispcolor(centriods);
%make a new picture with the pixels as the centriod colors
newpicture = zeros(x,y,3);
```

```
for i = 1:x
    for j = 1:y
        color = closestcentriod(i,j);
        r = uint8(centriods(color,1));
        g = uint8(centriods(color,2));
        b = uint8(centriods(color,3));
        newpicture(i,j,1) = r;
        newpicture(i,j,2) = g;
        newpicture(i,j,3) = b;
    end
end


figure
plot(1:count, energies);
str = strcat("k = ", num2str(k));
title(str);
str = strcat('new', num2str(k), '.tiff');
imwrite(uint8(newpicture), str)

function dispcolor(colors)
n = size(colors,1);
figure
sgtitle('Final colors');
for i = 1:n
    dim = ceil(sqrt(n));
    x = [0 1 1 0];
    y = [0 0 1 1];
    subplot(dim,dim,i)
    fill(x,y,colors(i,:)/255);
    title(num2str(i));
end
End
```

Plots below.


k = 2
×10^7
1.4
1.3
1.2
1.1
1
0.9
2
4
6
8
10
12
14


k = 4
×10^6
9.5
9
8.5
8
7.5
7
6.5
6
5.5
0
5
10
15



ii. For each k, display your compressed image along with the original image.

k = 2 

k = 4

k = 8