Submission Phase

1. Do assignment ☑ (/algorithmicthink2-002/human_grading/view/courses/974635/assessments/33/submissions)

Evaluation Phase

2. Evaluate peers ☑ (/algorithmicthink2-002/human_grading/view/courses/974635/assessments/33/peerGradingSets)

Results Phase

3. See results ☑ (/algorithmicthink2-002/human_grading/view/courses/974635/assessments/33/results/mine)

Your effective grade is 16

Your unadjusted grade is 16, which is simply the grade you received from your peers.

See below for details.

Overview

In Project 3, you implemented two methods for clustering sets of data. In this Application, we will analyze the performance of these two methods on various subsets of our county-level cancer risk data set. In particular, we will compare these two clustering methods in three areas:

- **Efficiency** - Which method computes clusterings more efficiently?
- **Automation** - Which method requires less human supervision to generate reasonable clusterings?
- **Quality** - Which method generates clusterings with less error?

Important: Please use Coursera's "Attach a file" button to attach your plots/images for this Application as required. For each question you can attach more than one image as well as including text and math (LaTeX) in the same answer box. In particular, please do not host your solution plots/images on 3rd party sites. This practice exposes your peers to extra security risks and has the potential for abuse since the contents of a link to an external site may be modified after the hard deadline. Failure to follow this policy may lead to your plots/images being counted as "not submitted".

Efficiency

The next four questions will consider the efficiency of hierarchical clustering and k-means clustering. Note that successfully computing the 3108 county images for Questions 2 and 3 in desktop Python may require some fine tuning of your code for one or both methods.

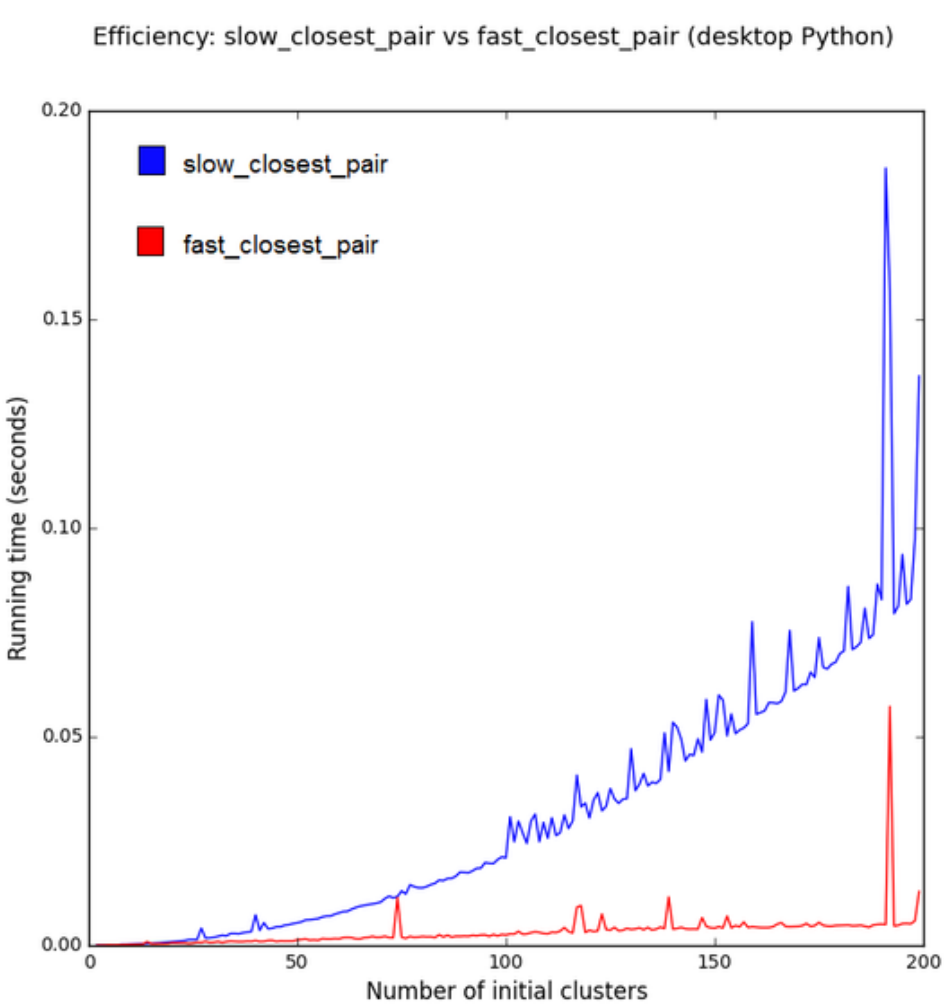
Question 1 (2 pts)

Write a function `gen_random_clusters(num_clusters)` that creates a list of clusters where each cluster in this list corresponds to one randomly generated point in the square with corners $(\pm 1, \pm 1)$. Use this function and your favorite Python timing code to compute the running times of the functions `slow_closest_pair` and `fast_closest_pair` for lists of clusters of size 2 to 200.

Once you have computed the running times for both functions, plot the result as two curves combined in a single plot. (Use a line plot for each curve.) The horizontal axis for your plot should be the the number of initial clusters while the vertical axis should be the running time of the function in seconds. Please include a legend in your plot that distinguishes the two curves.

Once you are satisfied with your plot, upload your plot in the box below using the "Attach a file" button (the button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled). Your plot will be assessed based on the answers to the following questions:

- Does the plot follow the formatting guidelines (https://class.coursera.org/algorithmicthink2-002/wiki/ides?page=plotting) for plots? Does the plot include a legend?
- Do the two curves in the plot have the correct shapes?



Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) Does the plot follow the formatting guidelines for plots? Does the plot include a legend?

The formatting guidelines include the following items:

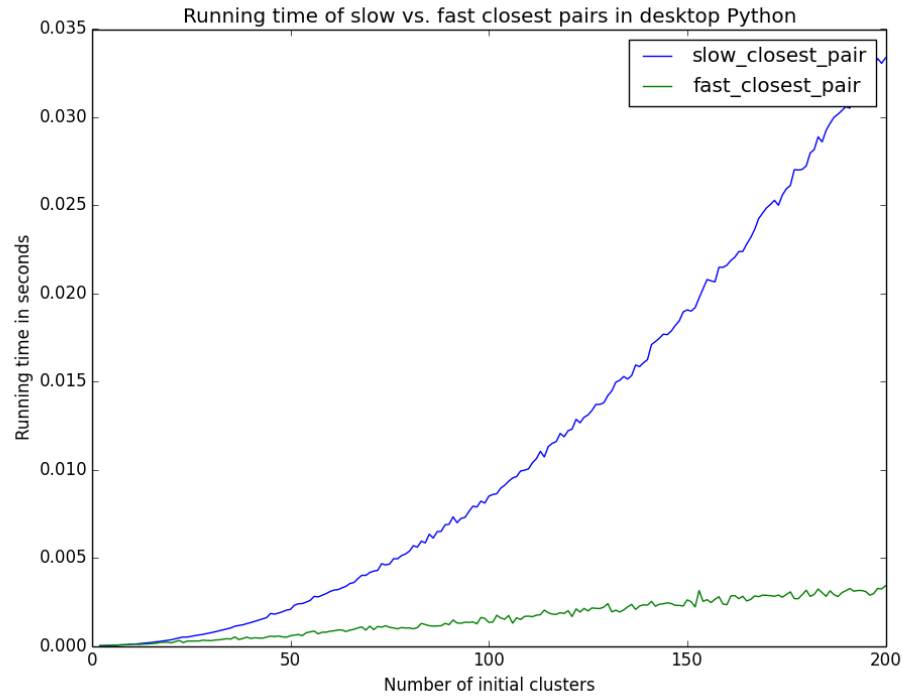
- The plot is an image and not a text file.
- The plot is appropriately trimmed. Showing the boundary of the plot's window is fine. However, the plot should not include part of the desktop.
- The elements of the plot are of the correct type. Line plots are not the same as point plots.
- Both axes should have tick marks labeled by regularly-spaced coordinate values.
- Both axes have appropriate text labels that describe the quantities being plotted.
- The plot has an appropriate title that describes the content of the plot.
- The plot has an appropriate legend (when required) that distinguishes the various components of the plot.

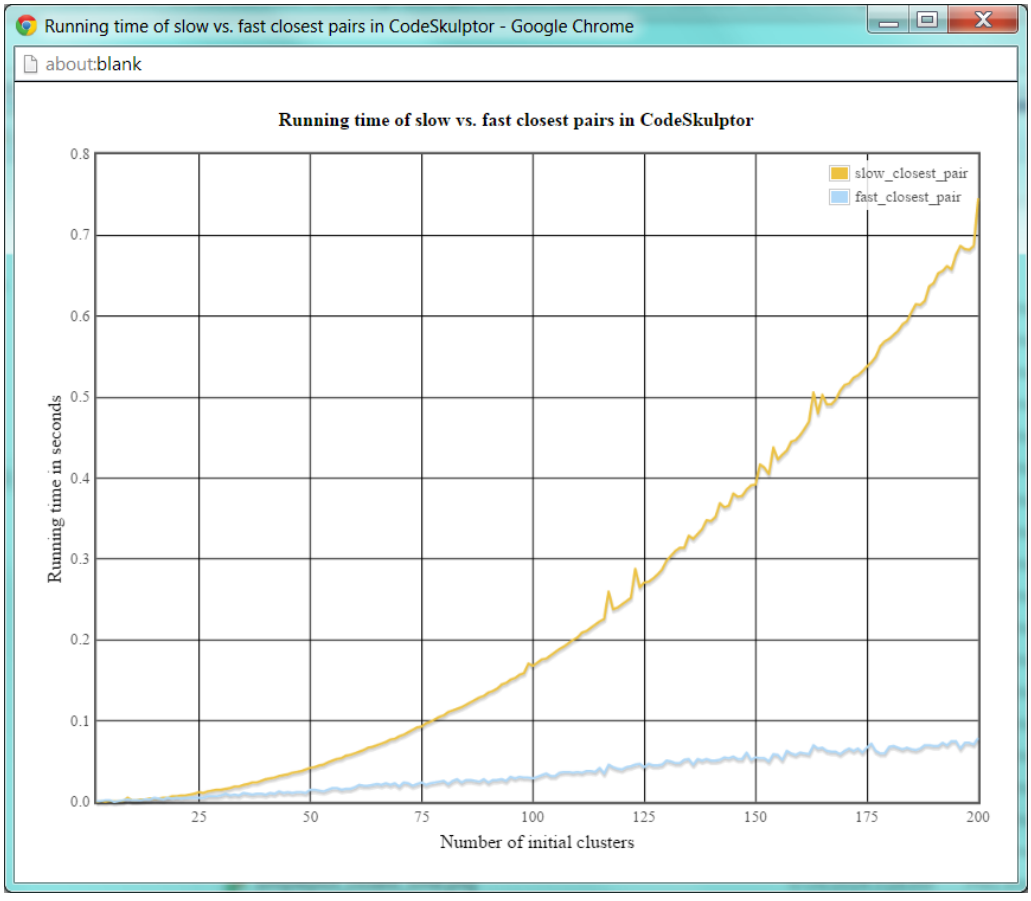
Assess the submitted plot based on these guidelines. For this question, do not deduct if the title of the plot does not distinguish between desktop Python and CodeSkulptor.

Score from your peers: **1**

Item b (1 pt) Do the two curves in the plot have the correct shapes?

The running times of the functions `slow_closest_pair` and `fast_closest_pair` should be $O(n^2)$ and $O(n \log^2 n)$, respectively. Here are correct plots in both `matplotlib` and `simpleplot`.





For `fast_closest_pair`, the timing curve should look almost linear with a very slight upward bend, indicating the function is growing at an $O(n \log^2 n)$ rate. If the timing curve appears to be growing quadratically (i.e. it is a constant fraction of the timing curve for `slow_closest_pair`), count this item as incorrect. If you are unsure, score your peer's answer as correct.

Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

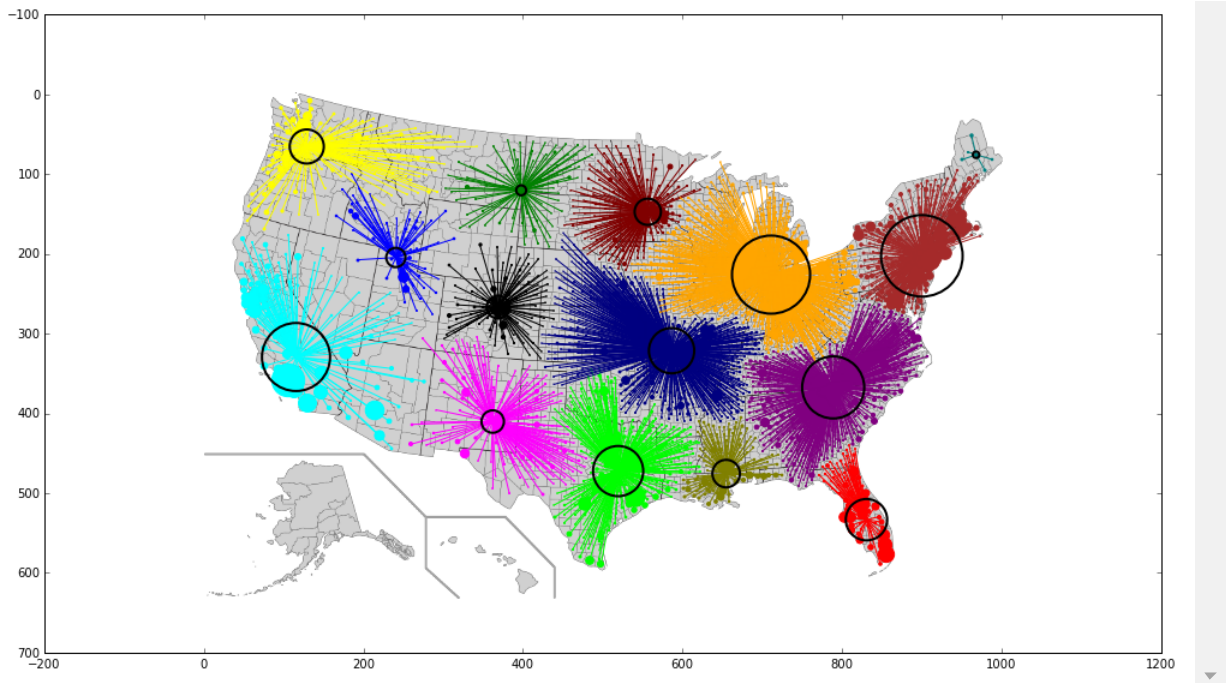
peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → Spikes may be garbage collector...I'm not sure but I gave you the point/

Question 2 (1 pt)

Use `alg_project3_viz` to create an image of the 15 clusters generated by applying hierarchical clustering to the 3108 county cancer risk data set. You may submit an image with the 3108 counties colored by clusters or an enhanced visualization with the original counties colored by cluster and linked to the center of their corresponding clusters by lines. You can generate such an enhanced plot using our `alg_clusters_matplotlib` code by modifying the last parameter of `plot_clusters` to be `True`. Note that plotting only the resulting cluster centers is not acceptable.

Once you are satisfied with your image, upload your image in the box below using the "Attach a file" button. (The button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled.) Your submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axis labels, or a title for this image.



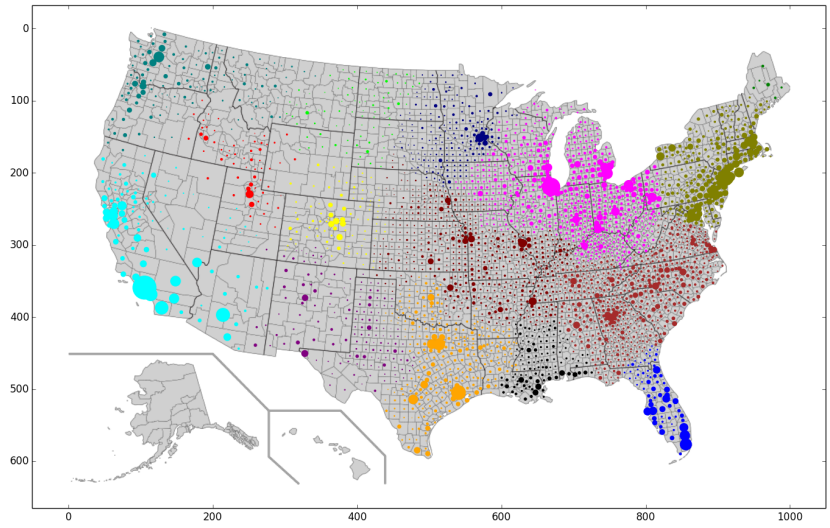
Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

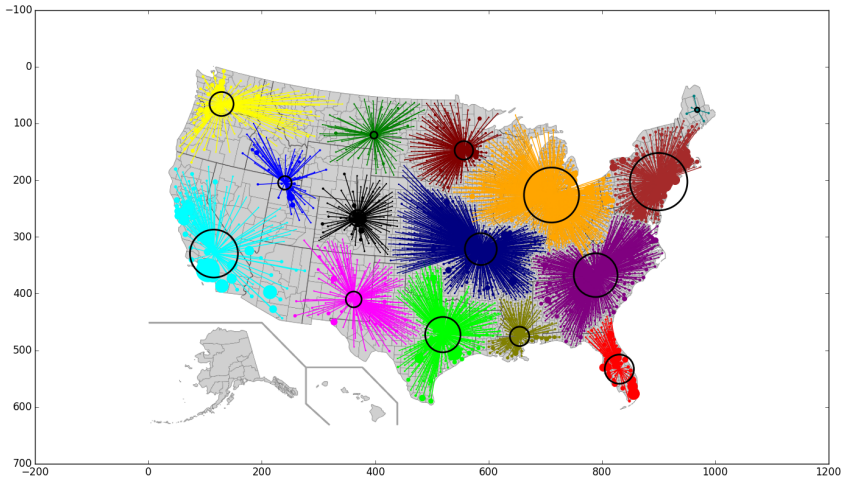
The submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axes labels, or a title for this image.

Our code for computing this image takes approximately 10 minutes to run in IDLE so we will only provide the `matplotlib` solution image below. Observe that the solution image has a small green cluster in the far upper right (to the right of the olive cluster). Images created with other tools such as SimpleGUI or GNUPlot are fine as long as they include the map of the USA in the background and the shape of the clusters match those in the the image below fairly accurately.

Compare the shape of the clusters in the submitted image to the shape of the clusters in the solution image. Since hierarchical clustering is deterministic, the shape of the clusters (where each cluster is visualized as a collection of counties) should match exactly. **Do not compare the colors of the clusters.** Reordering the clusters (which is allowed) causes the visualization code to assign different colors to each cluster. In some regions, the coloring of adjacent clusters may make it difficult to determine the boundary of the clusters in that region. In these regions, assume that the shape is correct.



Images that show the centers of the resulting clusters **as well as the original counties** are acceptable. Again, the position of each cluster center should exactly match the solution below. The absolute size of each cluster center need not match the solution image below since submitted images may have been rescaled. However, the size of each cluster center in relation to its neighbors should agree.



Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

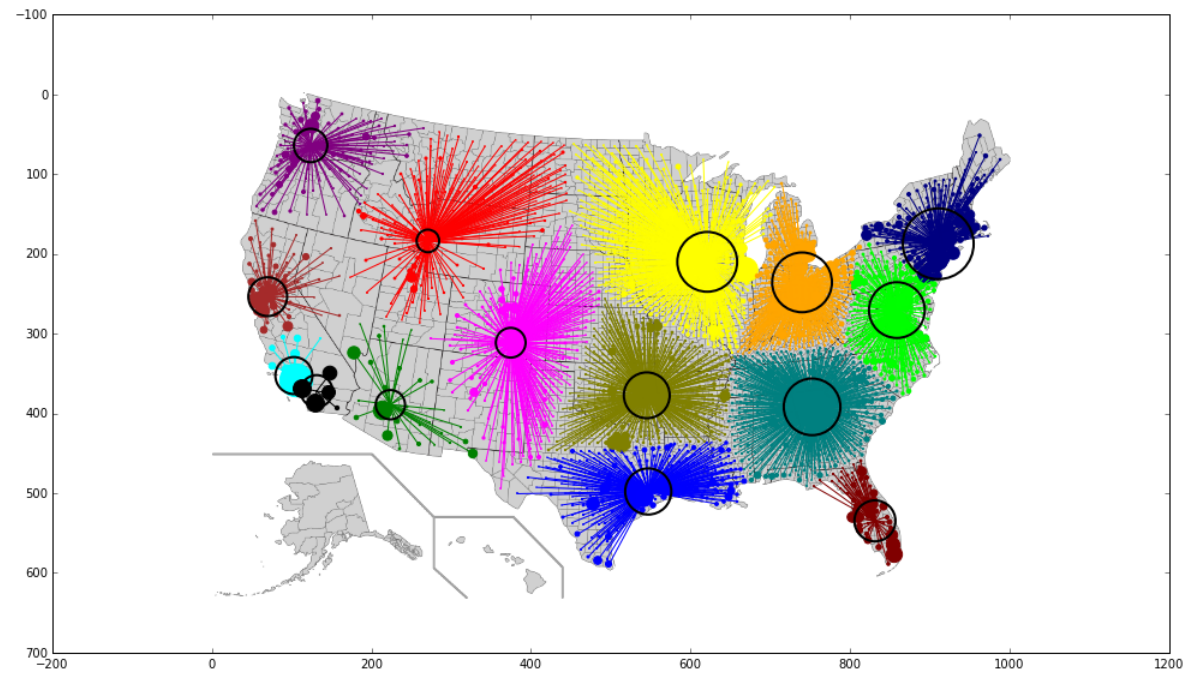
peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Question 3 (1 pt)

Use `alg_project3_viz` to create an image of the 15 clusters generated by applying 5 iterations of k-means clustering to the 3108 county cancer risk data set. You may submit an image with the 3108 counties colored by clusters or an enhanced visualization with the original counties colored by cluster and linked to the center of their corresponding clusters by lines. As in Project 3, the initial clusters should correspond to the 15 counties with the largest populations.

Once you are satisfied with your image, upload your image in the box below using the "Attach a file" button (the button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled). Your submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axis labels, or a title for this image.



Evaluation/feedback on the above work

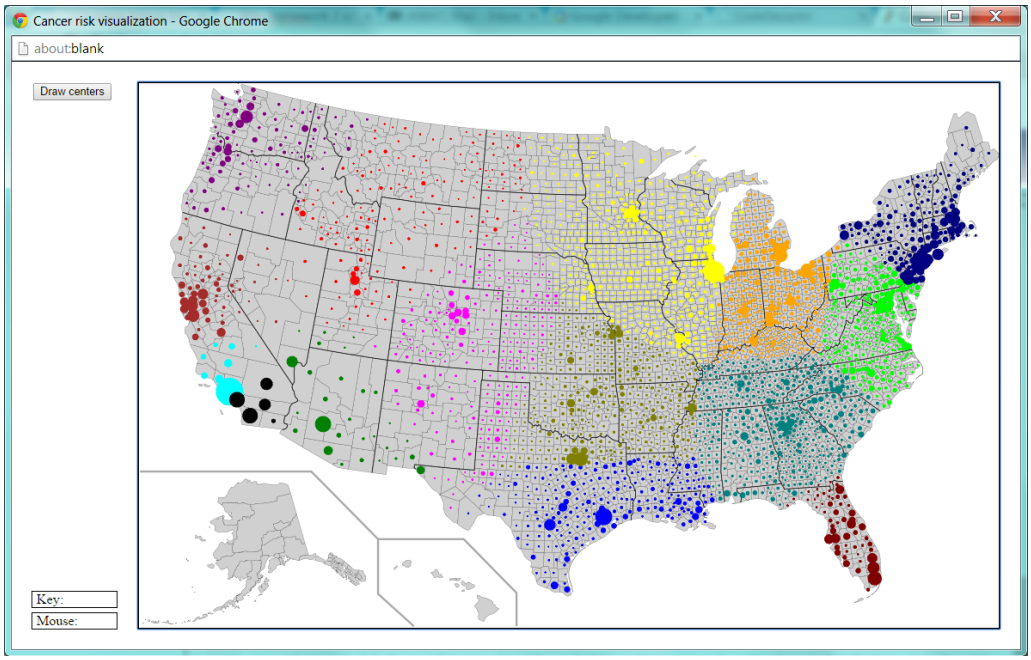
Note: this section can only be filled out during the evaluation phase.

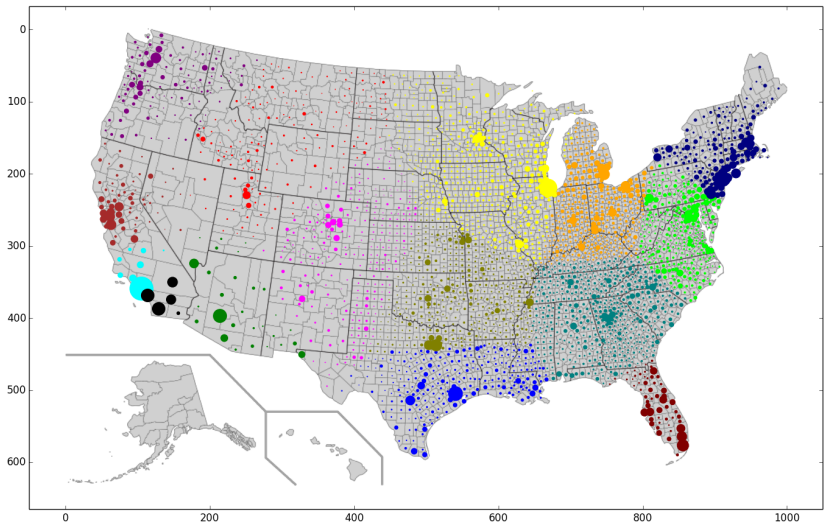
Item a (1 pt) The submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axes labels, or a title for this image.

Our code for computing this image takes only a few seconds to compute in CodeSkulptor so we will provide both `simplegui` and `matplotlib` solution images below. (Note that including the entire window in the `simplegui` image is fine.)

Compare the shape of the clusters in the submitted image to the shape of the clusters in the solution image. Since the computation of this clustering is deterministic, the shape of the clusters (where each cluster is visualized as a collection of counties) should agree exactly. For example, the correct image should have 4 clusters that touch the east coast of the USA and 4 clusters that touch the west coast of the USA. Even small differences in the shape of the clusters should result in the submitted image being counted as incorrect.

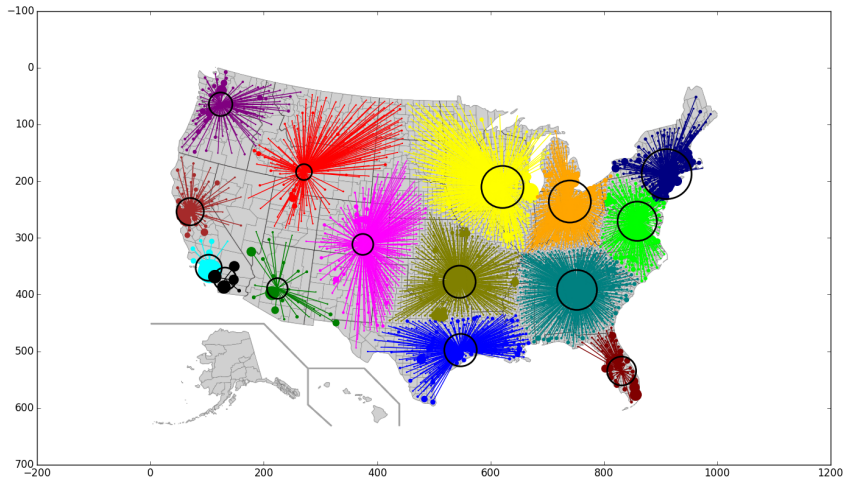
Important note: Compare only the shape of the clusters, not their colors. Reordering the clusters (which is allowed) causes the visualization code to assign different colors to each cluster.





Images that show the centers of the resulting clusters **as well as the original counties** are acceptable. Again, the position of each cluster center should exactly match the solution. The absolute size of each cluster center need not match the solution image below

since submitted images may have been rescaled. However, the size of each cluster center in relation to its neighbors should agree. Finally, the colors of the cluster centers do not need to match.



Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Question 4 (1 pt)

Which clustering method is faster when the number of output clusters is either a small fixed number or a small fraction of the number of input clusters? Provide a short explanation in terms of the asymptotic running times of both methods. You should assume that `hierarchical_clustering` uses `fast_closest_pair` and that k-means clustering always uses a small fixed number of iterations.

Analyzing the asymptotic running times for **hierarchical_clustering** and **kmeans_clustering**, it is obtained $O(n^2 + nh(n))$ for the first and $O(qnk)$ for the latter (where $h(n)$ is the running time

for **fast_closest_pair**), therefore, if q (number of iterations) and k (number of clusters) are a small fixed number or a small fraction of the input clusters, for practical purposes, $O(qnk)$ can be considered $O(n)$ linear running time, that is why kmeans_clustering is faster than hierarchical_clustering which runs at least on the order of $O(n^2)$, regardless of the number of clusters (running times are verifiable in Homework 3).

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) Which clustering method is faster when the number of output clusters is either a small fixed number or a small fraction of the number of input clusters? Provide a short explanation in terms of the asymptotic running times of both methods. You should assume that `hierarchical_clustering` uses `fast_closest_pair` and that k-means clustering always uses a small fixed number of iterations.

As your computations in Questions 2 and 3 should have illustrated, k-means clustering is substantially faster than hierarchical clustering when the number of output clusters is fixed or a small fraction of the number of input clusters. The asymptotic analysis of both methods supports this observation.

If there are n input clusters and k output clusters, hierarchical clustering makes $n - k$ calls to `fast_closest_pair`. If k is fixed or a small fraction of n, each call to `fast_closest_pair` is $O(n \log^2 n)$ and, therefore, the running time for hierarchical clustering using `fast_closest_pair` is $O(n^2 \log^2 n)$. For k-means, the running time is either $O(n)$ or $O(n^2)$ depending on whether the size of output cluster is fixed or varies as a function n. Even in the this second case, k being a small fraction of n will reduce the running time in practice.

Note that the submitted explanation does not need to go into this level of detail. However, it should state that k-means clustering is more efficient and include a short explanation that plausibly (possibly correct) matches the explanation above.

Score from your peers: **1**

Conclusion: One important reason that hierarchical clustering is slower than k-means clustering is that each call to `fast_closest_pair` throws away most of the information used in computing the closest pair. Smarter implementations of hierarchical clustering can substantially improve their performance by using an implementation of dynamic closest pairs (http://en.wikipedia.org/wiki/Closest_pair_of_points_problem#Dynamic_closest-pair_problem). Dynamic methods build an initial data structure that can be used to accelerate a sequence of insertions and deletions from the set of points.

Using dynamic closest pairs leads to implementations of hierarchical clustering that are more competitive with k-means clustering. This paper (http://www.ics.uci.edu/~eppstein/projects/pairs/Talks/ClusterGroup.pdf) provides a nice introduction to approaches of this type.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

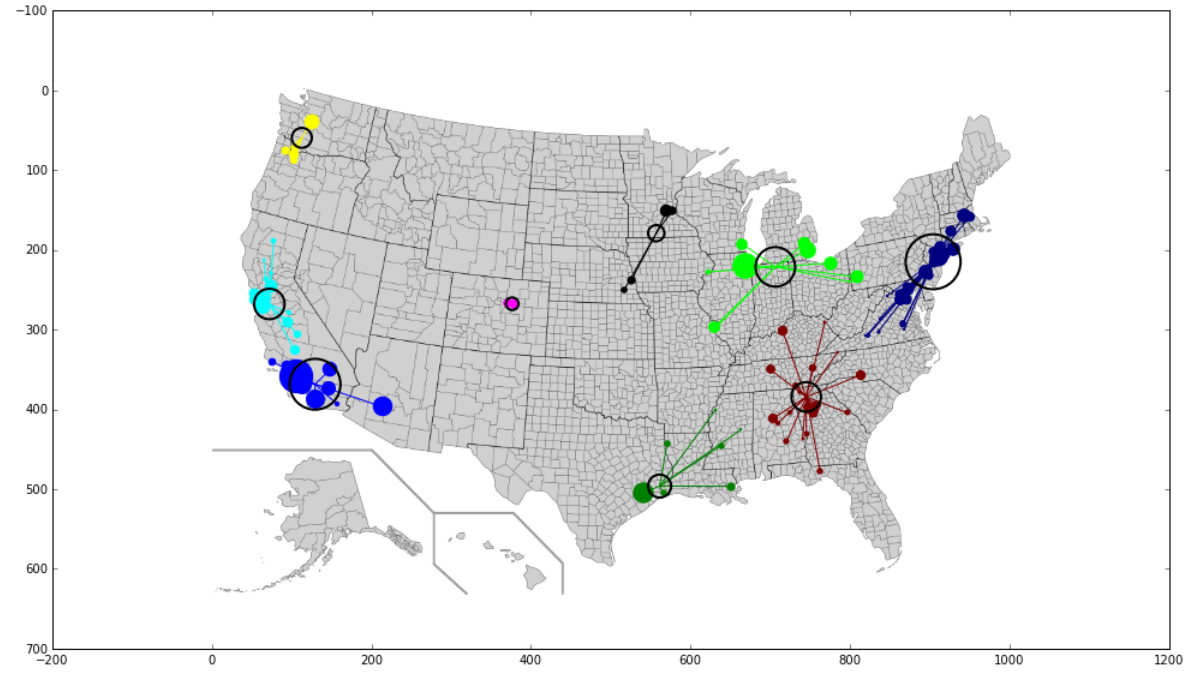
Automation

In the next five questions, we will compare the level of human supervision required for each method.

Question 5 (1 pt)

Use `alg_project3_viz` to create an image of the 9 clusters generated by applying hierarchical clustering to the 111 county cancer risk data set. You may submit an image with the 111 counties colored by clusters or an enhanced visualization with the original counties colored by cluster and linked to the center of their corresponding clusters by lines.

Once you are satisfied with your image, upload your image in the box below using the "Attach a file" button (the button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled). Your submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axes labels, or a title for this image.



Evaluation/feedback on the above work

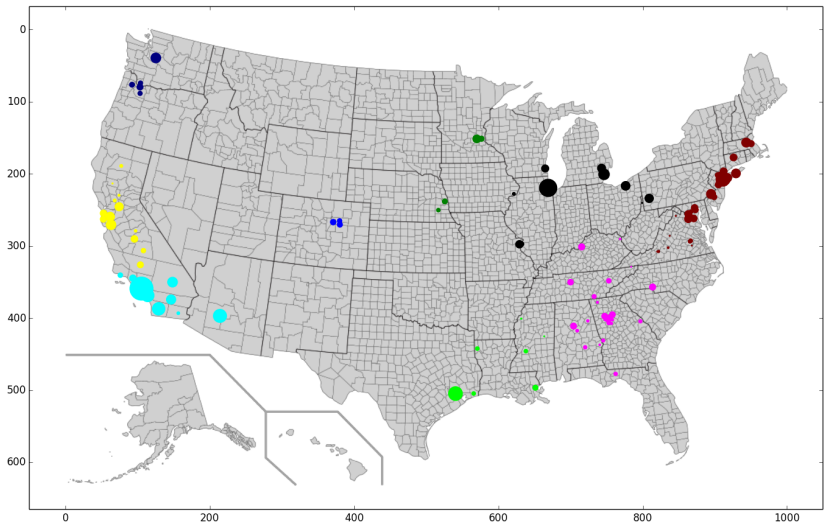
Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) The submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axes labels, or a title for this image.

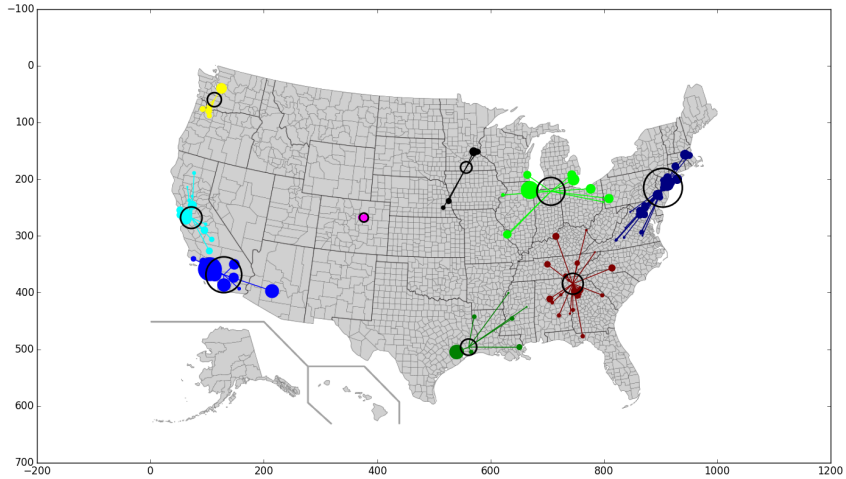
The `matplotlib` solution image is provided below. Creating the image in `simplegui` is also fine. (Note that including the entire window in the `simplegui` image is fine.)

Compare the shape of the clusters in the submitted image to the shape of the clusters in the solution image. Since the computation of this clustering is deterministic, the shape of the clusters (where each cluster is visualized as a collection of counties) should agree exactly. Even small differences in the shape of the clusters should result in the submitted image being counted as incorrect.

Important note: Compare only the shape of the clusters, not their colors. Reordering the clusters (which is allowed) causes the visualization code to assign different colors to each cluster.



Images that show the centers of the resulting clusters **as well as the original counties** are acceptable. Again, the position of each cluster center should exactly match the solution. The absolute size of each cluster center need not match the solution image below since submitted images may have been rescaled. However, the size of each cluster center in relation to its neighbors should agree. Finally, the colors of the cluster centers do not need to match.



Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

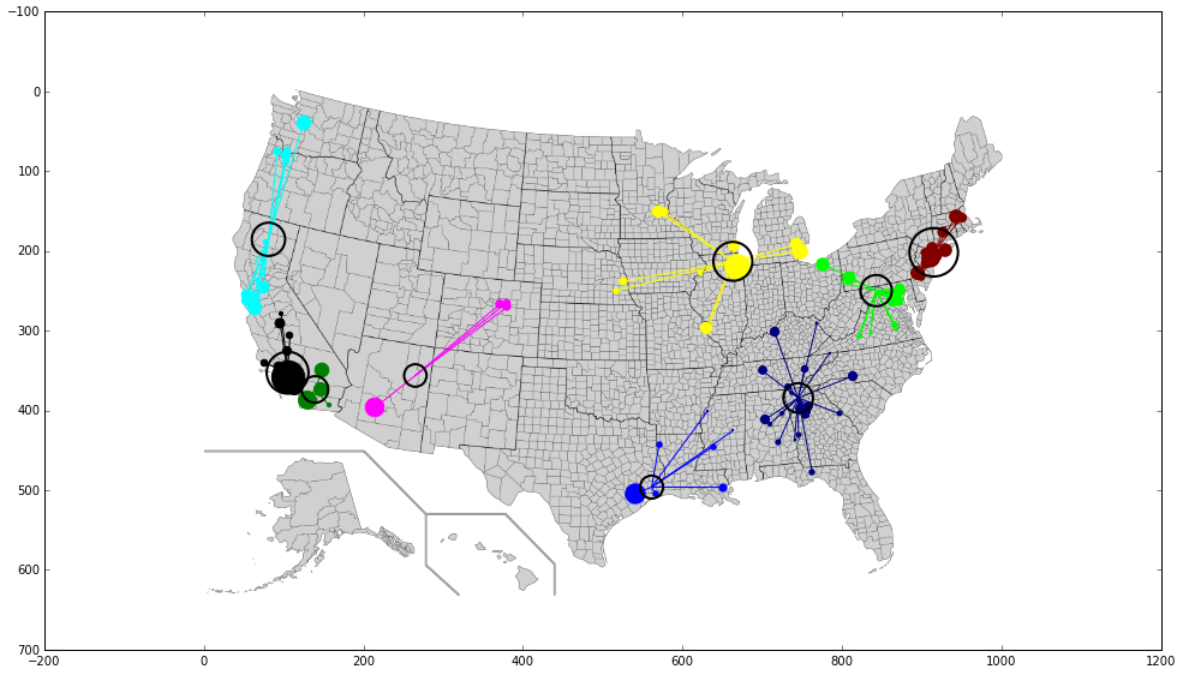
peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Question 6 (1 pt)

Use `alg_project3_viz` to create an image of the 9 clusters generated by applying 5 iterations of k-means clustering to the 111 county cancer risk data set. You may submit an image with the 111 counties colored by clusters or an enhanced visualization with the original counties colored by cluster and linked to the center of their corresponding clusters by lines. As in Project 3, the initial clusters should correspond to the 9 counties with the largest populations.

Once you are satisfied with your image, upload your image in the box below using the "Attach a file" button (the button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled). Your submitted image will be assessed based on whether it matches our solution image. You do not need to include axes, axes labels, or a title for this image.



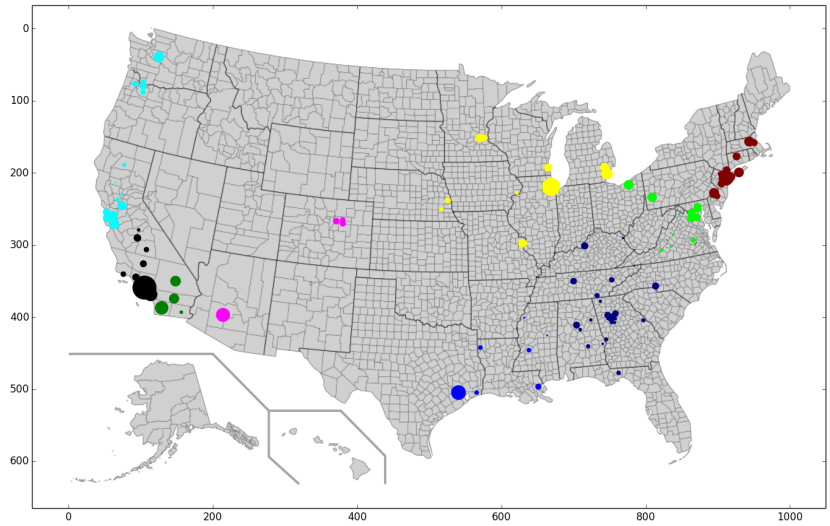
Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

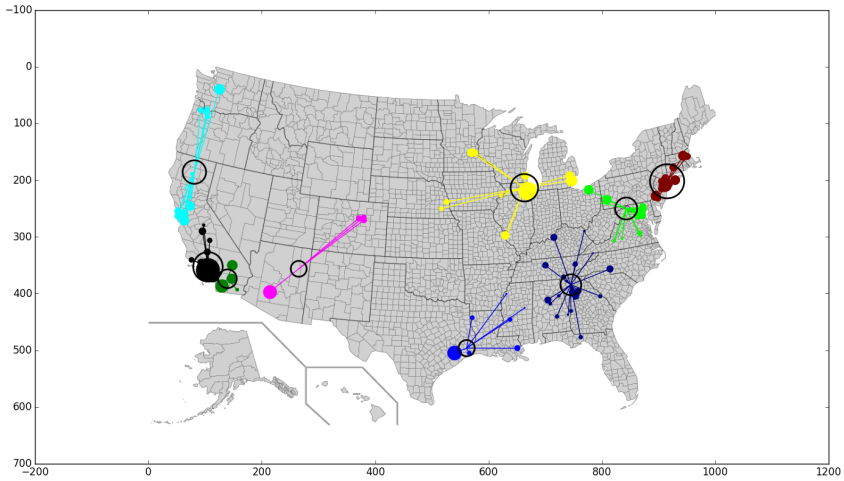
Item a (1 pt) The `matplotlib` solution image is provided below. Creating the image in `simplegui` is also fine. (Note that including the entire window in the `simplegui` image is fine.)

Compare the shape of the clusters in the submitted image to the shape of the clusters in the solution image. Since the computation of this clustering is deterministic, the shape of the clusters (where each cluster is visualized as a collection of counties) should agree exactly. Even small differences in the shape of the clusters should result in the submitted image being counted as incorrect.

Important note: Compare only the shape of the clusters, not their colors. Reordering the clusters (which is allowed) causes the visualization code to assign different colors to each cluster.



Images that show the centers of the resulting clusters **as well as the original counties** are acceptable. Again, the position of each cluster center should exactly match the solution. The absolute size of each cluster center need not match the solution image below since submitted images may have been rescaled. However, the size of each cluster center in relation to its neighbors should agree. Finally, the colors of the cluster centers do not need to match.



Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Question 7 (1 pt)

The clusterings that you computed in Questions 5 and 6 illustrate that not all clusterings are equal. In particular, some clusterings are better than others. One way to make this concept more precise is to formulate a mathematical measure of the error associated with a cluster. Given a cluster C, its *error* is the sum of the squares of the distances from each county in the cluster to the cluster's center, weighted by each county's population. If p_i is the position of the county and w_i is its population, the cluster's error is:

$$error(C) = \sum_{p_i \in C} w_i (d_{p_i c})^2$$

where c is the center of the cluster C. The `Cluster` class includes a method `cluster_error(data_table)` that takes a `Cluster` object and the original data table associated with the counties in the cluster and computes the error associated with a given cluster.

Given a list of clusters L, the *distortion* of the clustering is the sum of the errors associated with its clusters.

$$distortion(L) = \sum_{C \in L} error(C).$$

Write a function `compute_distortion(cluster_list)` that takes a list of clusters and uses `cluster_error` to compute its distortion. Now, use `compute_distortion` to compute the distortions of the two clusterings in questions 5 and 6. Enter the values for the distortions (with at least four significant digits) for these two clusterings in the box below. Clearly indicate the clusterings to which each value corresponds.

As a check on the correctness of your code, the distortions associated with the 16 output clusters produced by hierarchical clustering and k-means clustering (with 5 iterations) on the 290 county data set are approximately 2.575×10^{11} and 2.323×10^{11}, respectively.

hierarchical_clustering: 1.7516×10^{11} (9 clusters on the 111 county data set)
kmeans_clustering: 2.7125×10^{11} (9 clusters, 5 iterations on the 111 county data set)

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) Enter the values for the distortions for these two clusterings in the box below. Clearly indicate which values correspond to which clustering.

The distortion for the 9 clusters generated by hierarchical clustering on the 111 county data set is between 1.751×10^{11} and 1.752×10^{11}. Count any answer in this range as being correct.

The distortion for the 9 clusters generated by k-means clustering on the 111 county data set is between 2.712×10^{11} and 2.713×10^{11}. Count any answer in this range as being correct.

Note that the exponential part of the answer (10^{11}) may also be expressed in Python as the string `e+11`.

Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Question 8 (1 pt)

Examine the clusterings generated in Questions 5 and 6. In particular, focus your attention on the number and shape of the clusters located on the west coast of the USA.

Describe the difference between the shapes of the clusters produced by these two methods on the west coast of the USA. What caused one method to produce a clustering with a much higher distortion? To help you answer this question, you should consider how k-means clustering generates its initial clustering in this case.

In explaining your answer, you may need to review the geography of the west coast of the USA (http://en.wikipedia.org/wiki/West_Coast_of_the_United_States).

The three clusters created by hierarchical clustering on the west coast are better balanced and better distributed than those three created by kmeans clustering, that is, while they seem to be more reasonable spaced and fairly compact, in the first case, there are two small, compact and close clusters and one large and vertically spread cluster in the latter case.

kmeans clustering initializes the centers to be the coordinates of the 9 most populated cities in the U.S., It happens that the state of California has 2 of them (#2 Los Angeles, #8 San Diego), which are very close to each other, therefore kmeans computes two clusters around this cities despite they are very close (when it would be more natural to put this two cities in the same cluster, as hierarchical clustering

does). To the north of the west coast remain two other large concentrations of people (#10 San Jose, #20 Seattle), but if only one cluster is dedicated to this cities the outcome is clear, it will be a cluster whose center is middle way between this two cities yielding a vertically spread and large cluster, and increasing in this way the cluster_error.

The initialization of kmeans does not take into account the actual distances between cities, the "largest population criteria" does not obey to a spatially well distributed characteristic, therefore it is reasonable to expect higher distortion than the output produced by hierarchical clustering, because the initial clusters will be placed around large cities, despite if they are close or not, while hierarchical clustering will connect counties greedily by closest pairs, minimizing in this way the error associated.

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) Describe the difference between the shapes of the clusters produced by these two methods on the west coast of the USA. What caused one method to produce a clustering with a much higher distortion?

Each method generates 3 clusters on the west coast of the USA. Hierarchical clustering generates one cluster in Washington state, one in northern California and one in southern California. K-means clustering generates one cluster that includes Washington state and parts of northern California, one cluster that includes the Los Angeles area, and one cluster that includes San Diego. The k-means clustering has substantially higher distortion due in part to the fact that southern California is split into two clusters while northern California is clustered with Washington state.

This difference in cluster shape is due to the fact that the initial clustering used in k-means clustering includes the 3 counties in southern California with high population and no counties in northern California or Washington state. Due to a poor choice of the initial clustering based on large population counties, k-means clustering produces a clustering with relatively high distortion.

When scoring the answer/explanation for this question, you are welcome to use your judgment about the correctness of the submitted explanation. If the explanation is plausible (possibly correct), score it as correct. If not, check whether the submitted explanation mentions that three of the nine initial clusters in k-means clustering are located in southern California. If so, also score these explanations as being correct.. Otherwise, score the explanation as being incorrect.

Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → Good.

Question 9 (1 pt)

Based on your answer to Question 8, which method (hierarchical clustering or k-means clustering) requires less human supervision to produce clusterings with relatively low distortion?

hierarchical clustering

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) Based on your answer to Question 8, which method (hierarchical clustering or k-means clustering) requires less human supervision to produce clustering with relatively low distortion?

Hierarchical clustering requires less human supervision than k-means clustering to produce clustering of relatively low distortion as it requires no human interaction beyond the choice of the number of output clusters. On the other hand, k-means clustering requires a good strategy for choosing the initial cluster centers. As we saw in Question 8, strategies that appear reasonable at first glance may lead to clusterings with relatively high distortion.

Score from your peers: **1**

Conclusion: In general, k-means clustering requires a higher level of human supervision since the quality of the output clustering is sensitive to the initial choice of cluster centers. k-means clustering also requires that the size of the output clustering be known beforehand. For hierarchical clustering, the distortion of the clustering can be monitored as closest pairs of clusters are merged with almost no extra cost.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Quality

In the last two questions, you will analyze the quality of the clusterings produced by each method as measured by their distortion.

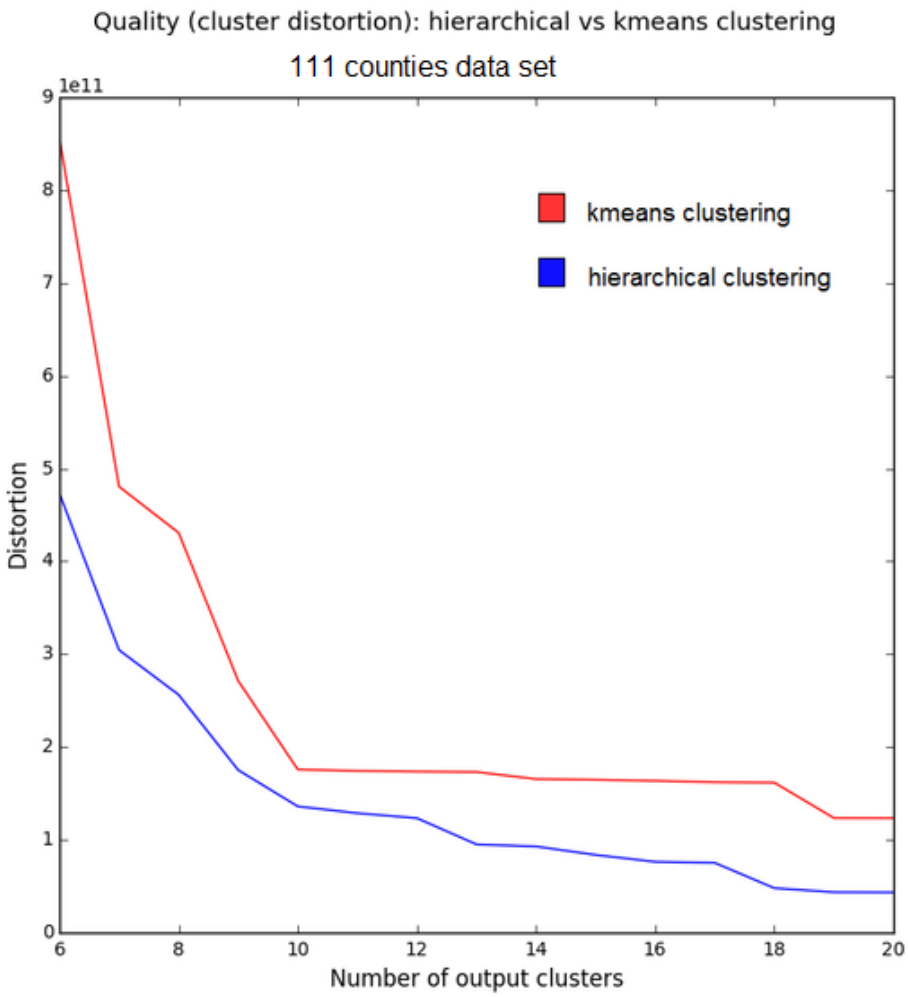
Question 10 (4 pts)

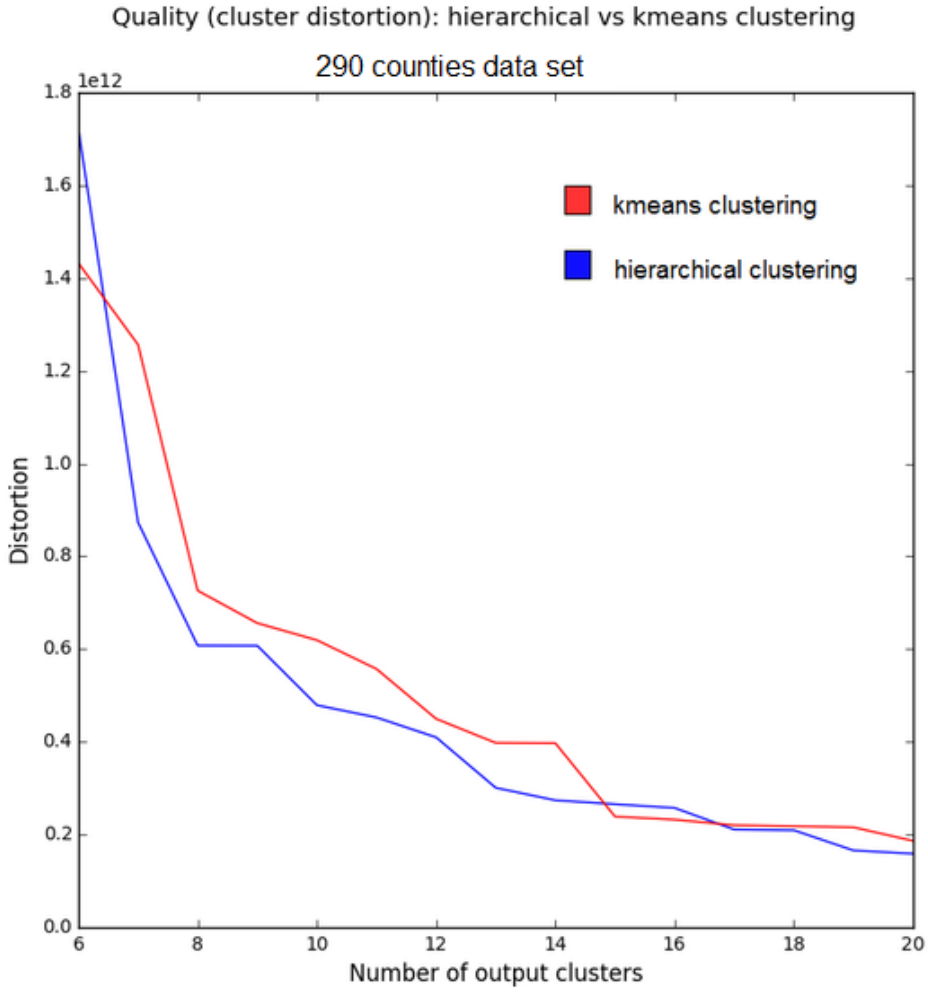
Compute the distortion of the list of clusters produced by hierarchical clustering and k-means clustering (using 5 iterations) on the 111, 290, and 896 county data sets, respectively, where the number of output clusters ranges from 6 to 20 (inclusive). **Important note:** To compute the distortion for all 15 output clusterings produced by `hierarchical_clustering`, you should remember that you can use the hierarchical cluster of size 20 to compute the hierarchical clustering of size 19 and so on. Otherwise, you will introduce an unnecessary factor of 15 into the computation of the 15 hierarchical clusterings.

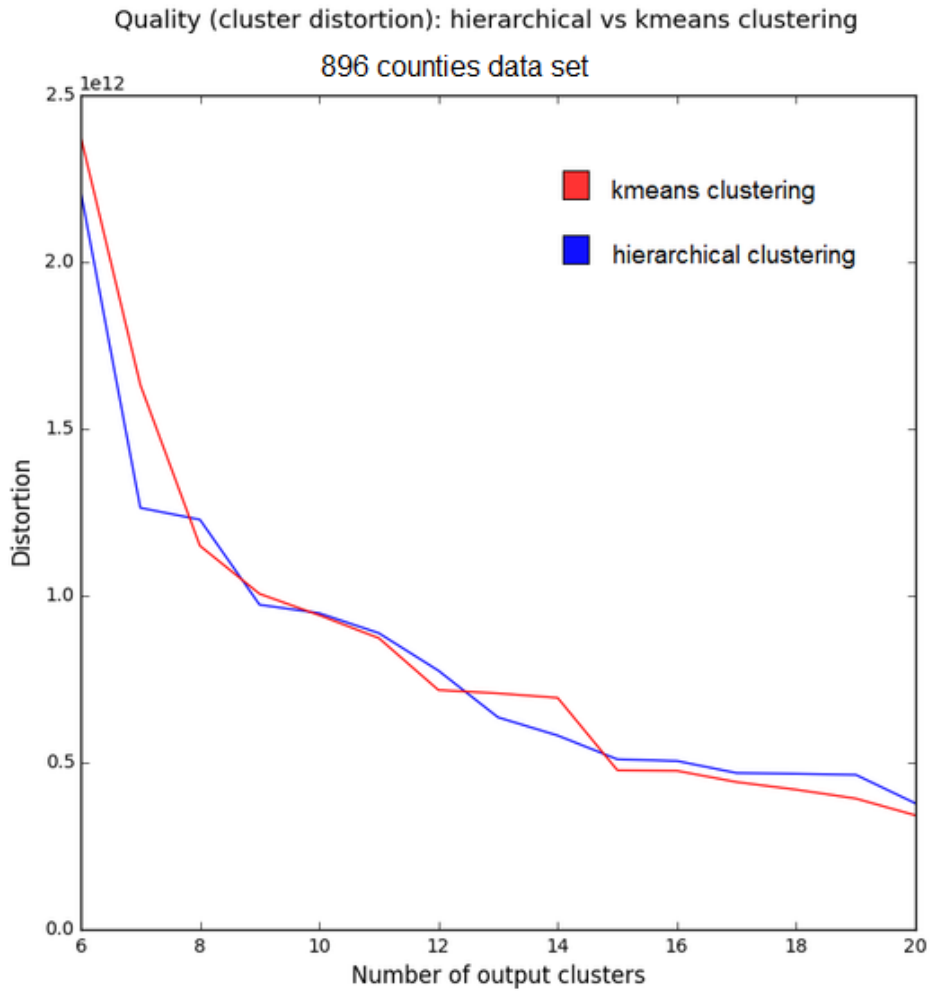
Once you have computed these distortions for both clustering methods, create three separate plots (one for each data set) that compare the distortion of the clusterings produced by both methods. Each plot should include two curves drawn as line plots. The horizontal axis for each plot should indicate the number of output clusters while the vertical axis should indicate the distortion associated with each output clustering. For each plot, include a title that indicates the data set used in creating the plots and a legend that distinguishes the two curves.

Once you are satisfied with your plots, upload them in the box below using the "Attach a file" button (the button is disabled under the 'html' edit mode; you must be under the 'Rich' edit mode for the button to be enabled). Your plots will be assessed based on the answers to the following questions:

- Do the plots follow the formatting guidelines (https://class.coursera.org/algorithmicthink2-002/wiki/ides?page=plotting) for plots? Does the title of each plot indicate which data was used to create the plot? Do the plots include a legend?
- Do the two curves in each plot have the correct shapes?


Quality (cluster distortion): hierarchical vs kmeans clustering
111 counties data set
1e11
kmeans clustering
hierarchical clustering
Distortion
Number of output clusters


Quality (cluster distortion): hierarchical vs kmeans clustering
290 counties data set
1e12
kmeans clustering
hierarchical clustering
Distortion
Number of output clusters



Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

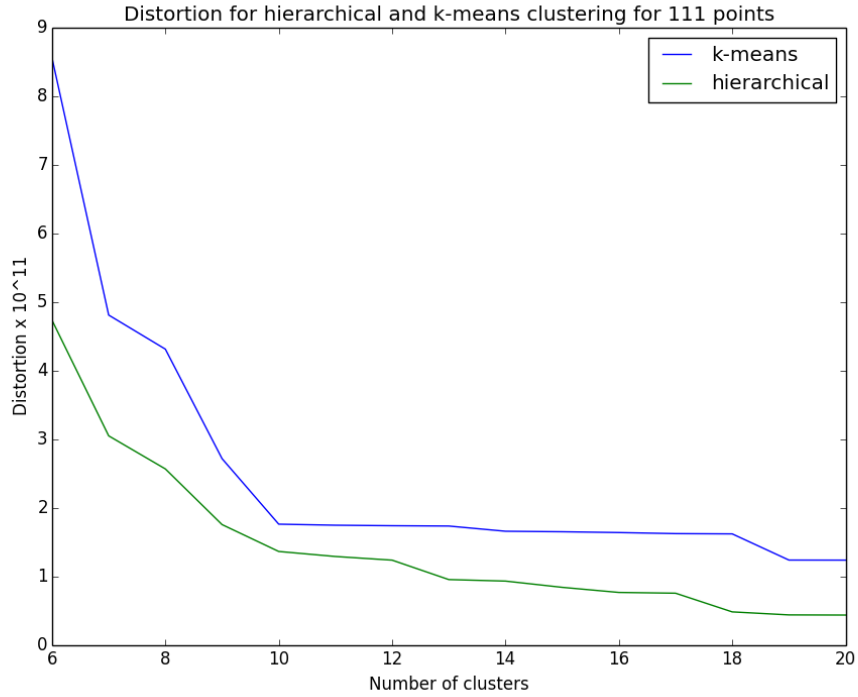
Item a (1 pt) Do the plots follow the formatting guidelines for plots? Does the title of each plot indicate which data was used to create the plot? Do the plots include a legend?

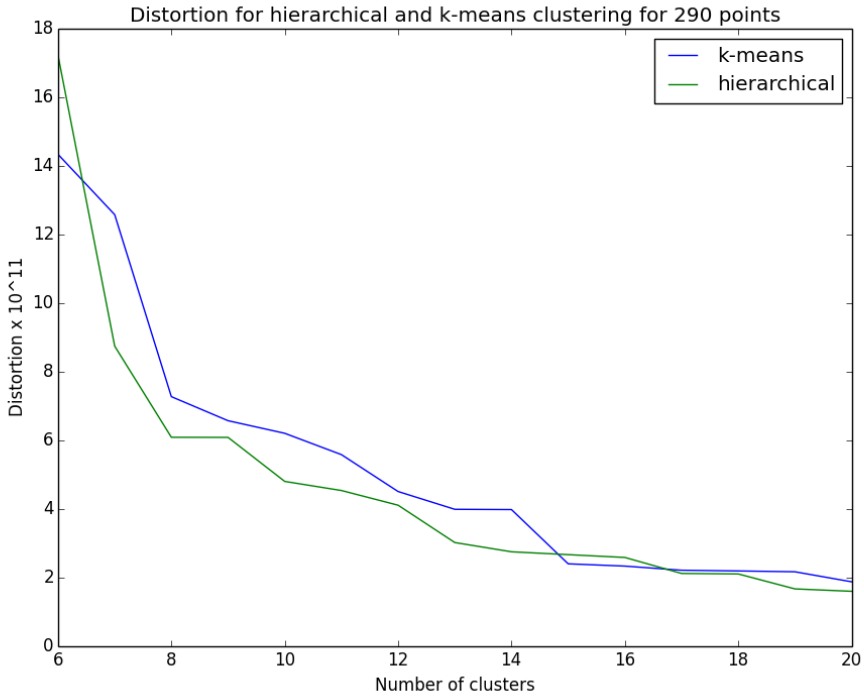
Review the plotting guidelines from Question 1a. Be generous on the labeling of axes. However, for these plots, it is critical that the title identifies which plot goes with which data set and the legend identifies the clustering method corresponding to each curve. Therefore, score this item as incorrect if either is not present.

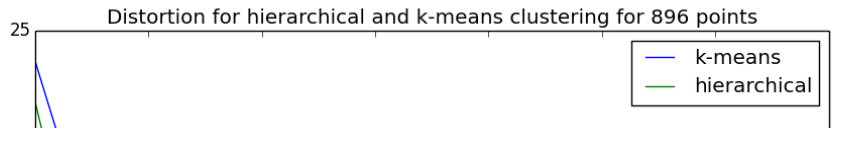
Score from your peers: **1**

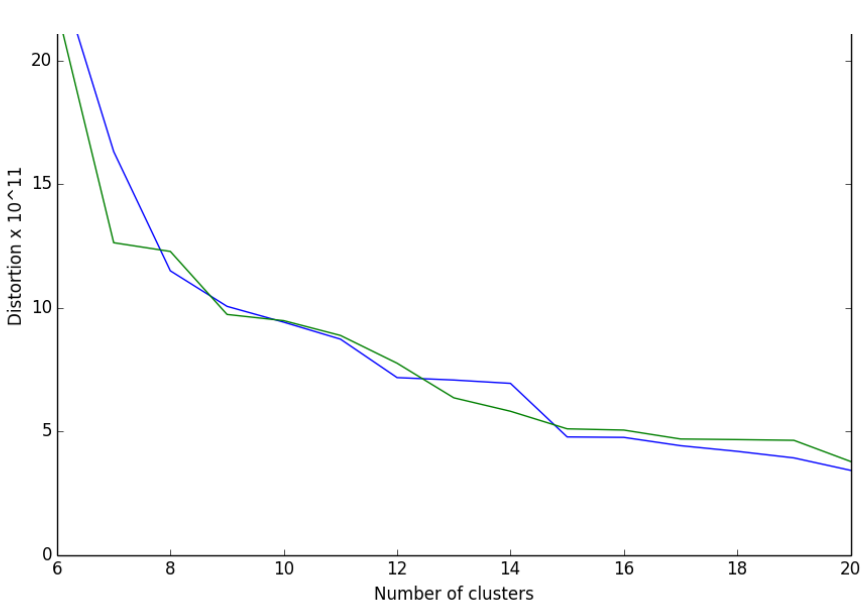
Item b (3 pts) Do the two curves in each plot have the correct shapes?

Below are the three plots for the distortion curves for hierarchical clustering and k-means clustering. When assessing each submitted plot, the shape of both curves should match the answer plots closely. Award one point for each correct plot. If one of the methods is producing curves with correct distortion, but the other is incorrect, award one point.


Distortion for hierarchical and k-means clustering for 111 points
k-means
hierarchical
Distortion x 10^11
Number of clusters
0
1
2
3
4
5
6
7
8
9
6
8
10
12
14
16
18
20


Distortion for hierarchical and k-means clustering for 290 points
k-means
hierarchical
Distortion x 10^11
Number of clusters
0
2
4
6
8
10
12
14
16
18
6
8
10
12
14
16
18
20


Distortion for hierarchical and k-means clustering for 896 points
k-means
hierarchical
25



Score from your peers: **3**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Question 11 (1 pt)

For each data set (111, 290, and 896 counties), does one clustering method consistently produce lower distortion clusterings when the number of output clusters is in the range 6 to 20? Is so, indicate on which data set(s) one method is superior to the other.

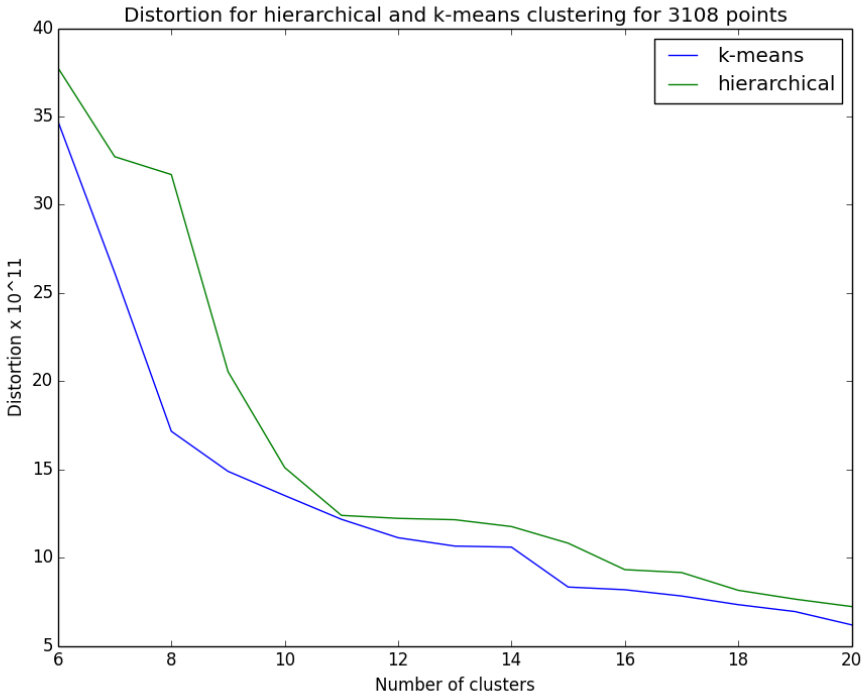
- On the 111 data set, hierarchical clustering is clearly superior to kmeans clustering, producing lower distortion clusterings consistently.
- On the 290 data set, hierarchical clustering produces lower distortion clusterings for most of the values *number of output clusters*, nevertheless the differences between distortion values are smaller in comparison to the 111 data set.
- On the 896 data set, it is not clear whether one method is superior to the other.

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) For each data set (111, 290, and 896 counties), does one clustering method consistently produce lower distortion clusterings when the number of output clusters is in the range 6 to 20? Is so, indicate on which data set(s) one method is superior to the other.

For the 111 county data set, hierarchical clustering consistently produces clusterings with less distortion. For the other two data sets, neither clustering method consistently dominates. Interestingly, k-means clustering produces lower distortion clusterings for the 3108 county data set. Here is a plot of distortion for the clusterings produced by both methods.



For this item, award full credit if the submitted answer states that hierarchical clustering produces lower distortion clusterings for the 111 county data set. Award no credit if the submitted answer states k-means produces lower distortion (or that the results are mixed) for the 111 country data set. Since the results are mixed for the other two data sets, do not consider the answers for these two data sets in scoring this problem.

Score from your peers: **1**

Conclusions: Beyond the smallest data set, neither method consistently produces lower distortion clusterings.

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

Overall evaluation (1 pt Extra Credit)

Which clustering method would you prefer when analyzing these data sets? Provide a summary of each method's strengths and weaknesses on these data sets in the three areas considered in this application. Your summary should be at least a paragraph in length (4 sentences minimum).

- In terms of efficiency, it is a no brainer! kmeans clustering is the winning method hands down (linear vs quadratic running time). If the application requires a fast algorithm, at the cost of quality perhaps, then kmeans is the way to go.
- In terms of automation, hierarchical clustering has a more natural way to create clusters, after all, the measure function to define proximity is the Euclidean distance, which is the core idea by using closest pair on each iteration. On the other hand, kmeans criteria to initialize (largest city), does not take into account the relative proximity of the counties, thus yielding some strange behavior: some clusters may be too close from each other, some clusters may be too spread.
- In terms of quality, it seems that if the data set is small, hierarchical clustering yields lower distortion values, as opposed to kmeans clustering. But as the size of the data set increases, the difference between methods seems immaterial.

Overall, if time is no concern, I think the best way to go is with hierarchical clustering. But for fast applications, kmeans should be the natural choice.

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Item a (1 pt) Which clustering method would you prefer when analyzing these data sets? Provide a summary of each method's strengths and weaknesses on these data sets in the three areas considered in this application.

On these data sets, neither method dominates in all three areas: efficiency, automation, and quality. In terms of efficiency, k-means clustering is preferable to hierarchical clustering as long as the desired number of output clusters is known beforehand. However, in terms of automation, k-means clustering suffers from the drawback that a reliable method for determining the initial cluster centers needs to be available. Finally, in terms of quality, neither method produces clusterings with consistently lower distortion on larger data sets.

As you may suspect, this answer is basically impossible to reliable peer assess for content since the answer to this question is subject to much interpretation. Therefore, award full credit if the submitted answer contains at least 4 sentences. Feel free to add comments critiquing your peer's answer if you so desire. Enjoy your 1 point of extra credit!

Score from your peers: **1**

Comments: Please enter an explanation for your scoring, especially if you deducted any points for one of the rubric items for this question.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → *[This area was left blank by the evaluator.]*

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*