Clustering Methods

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 27-Mar-2017

Copyright

Copyright © 2017 by Nathaniel E. Helwig

Outline of Notes

1) Similarity and Dissimilarity
- Defining Similarity
- Distance Measures

2) Hierarchical Clustering
- Overview
- Linkage Methods
- States Example

3) Non-Hierarchical Clustering
- Overview
- K Means Clustering
- States Example

Purpose of Clustering Methods

Clustering methods attempt to group (or cluster) objects based on some rule defining the similarity (or dissimilarity) between the objects.

Distinction between clustering and classification/discrimination:

- Clustering: the group labels are not known a priori
- Classification: the group labels are known (for a training sample)

The typical goal in clustering is to discover the "natural groupings" present in the data.

Similarity and Dissimilarity

What does it Mean for Objects to be "Similar"?

Let $\mathbf{x} = (x_1, \ldots, x_p)'$ and $\mathbf{y} = (y_1, \ldots, y_p)'$ denote two arbitrary vectors.

Problem: We want some rule that measures the "closeness" or "similarity" between **x** and **y**.

How we define closeness (or similarity) will determine how we group the objects into clusters.

- Rule 1: Pearson correlation between **x** and **y**
- Rule 2: Euclidean distance between **x** and **y**
- Rule 3: Number of matches, i.e., $\sum_{j=1}^{p} 1_{\{x_j = y_j\}}$

Card Clustering with Different Similarity Rules

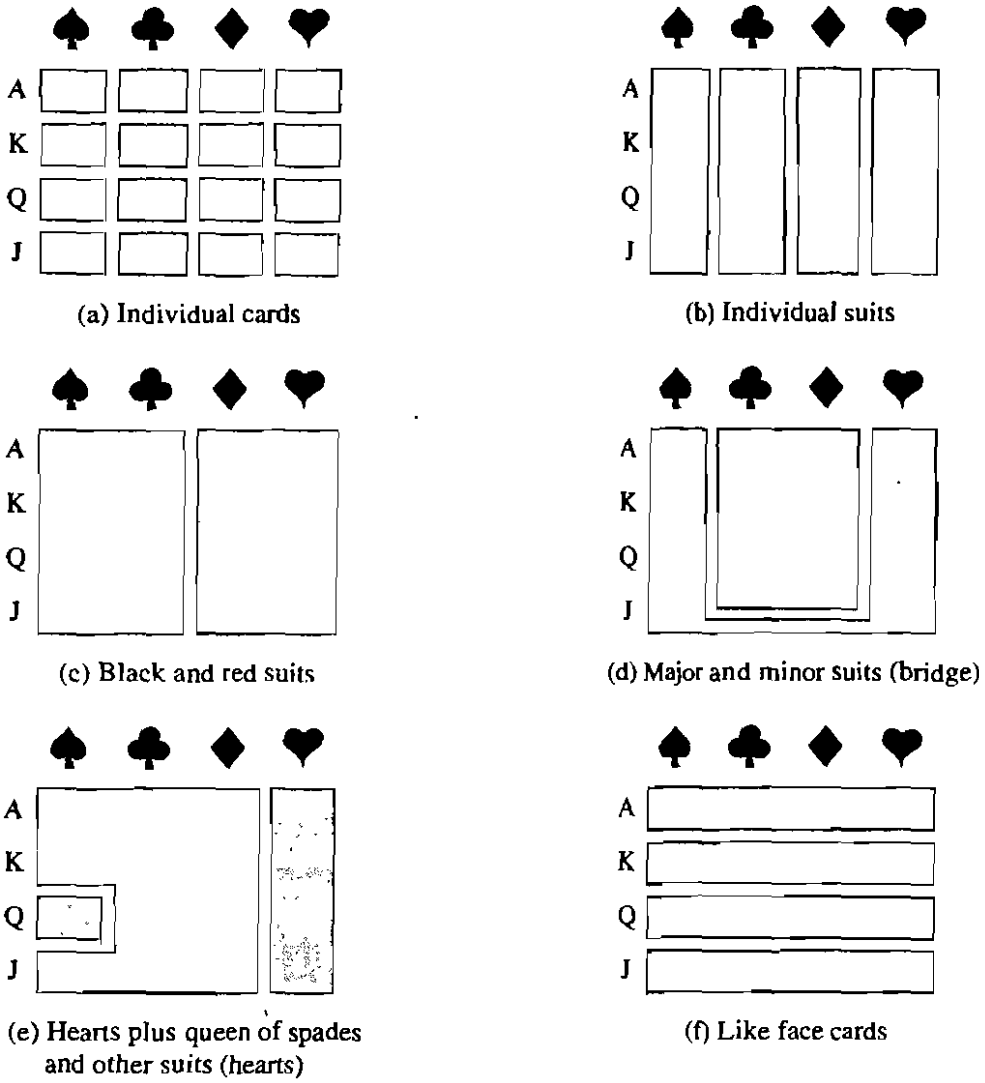


Figure 12.1 Grouping face cards.

Figure: Figure 12.1 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern).

Defining a Proper Distance

A metric (or distance) on a set $\mathcal{X}$ is a function $d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$

Let $d(\cdot, \cdot)$ denote some distance measure between objects P and Q, and let R denote some intermediate object.

A proper distance measure satisfies the following properties:

1. $d(P, Q) = d(Q, P)$ [symmetry]
2. $d(P, Q) \geq 0$ for all P, Q [non-negativity]
3. $d(P, Q) = 0$ if and only if $P = Q$ [identity of indiscernibles]
4. $d(P, Q) \leq d(P, R) + d(R, Q)$ [triangle inequality]

Distances define the similarity (or dissimilarity) between objects.

Visualization of the Triangle Inequality

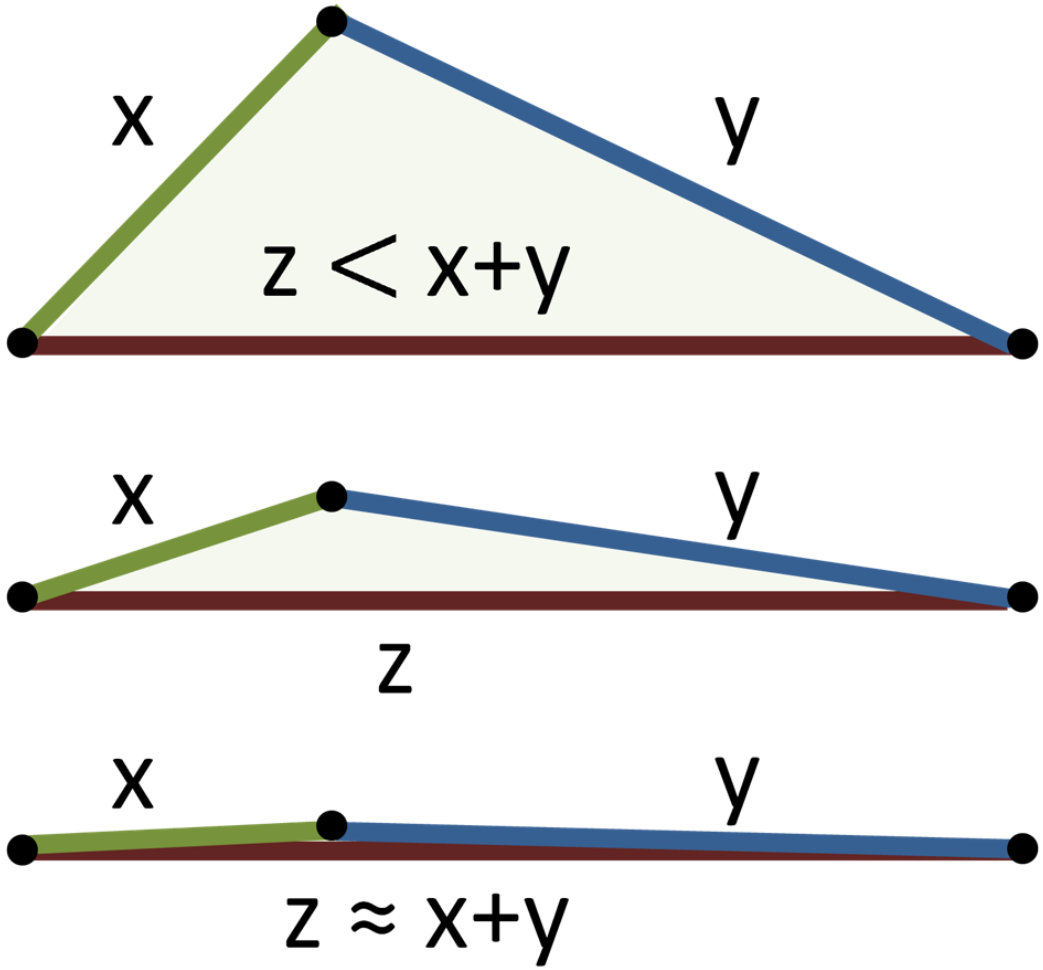


Figure: From `https://en.wikipedia.org/wiki/Triangle_inequality`

Minkowski Metric (and its Special Cases)

The Minkowski Metric is defined as

$$d_m(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{p} |x_j - y_j|^m \right)^{1/m}$$

where setting $m \geq 1$ defines a true distance metric.

- Setting $m = 1$ gives the Manhattan distance (city block)
  $d_1(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{p} |x_j - y_j|$
- Setting $m = 2$ gives the Euclidean distance
  $d_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{p} [x_j - y_j]^2 \right)^{1/2}$
- Setting $m = \infty$ gives the Chebyshev distance
  $d_\infty(\mathbf{x}, \mathbf{y}) = \max_j |x_j - y_j|$

Hierarchical Clustering

Two Approaches to Hierarchical Clustering

Hierarchical clustering uses a series of successive mergers or divisions to group *N* objects based on some distance.

Agglomerative Hierarchical Clustering (bottom up)

1. Begin with *N* clusters (each object is own cluster)
2. Merge the most similar objects
3. Repeat 2 until all objects are in the same cluster

Divisive Hierarchical Clustering (top down)

1. Begin with 1 cluster (all objects together)
2. Split the most dissimilar objects
3. Repeat 2 until all objects are in their own cluster

Dissimilarity between Objects (and Clusters?)

Our input for hierarchical clustering is an $N \times N$ dissimilarity matrix

$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{pmatrix}$$

where $d_{uv} = d(X_u, X_v)$ is the distance between objects X_u and X_v.

We know how to define dissimilarity between objects (i.e., d_{uv}), but how do we define dissimilarity between clusters of objects?

Measuring Inter-Cluster Distance (Dissimilarity)

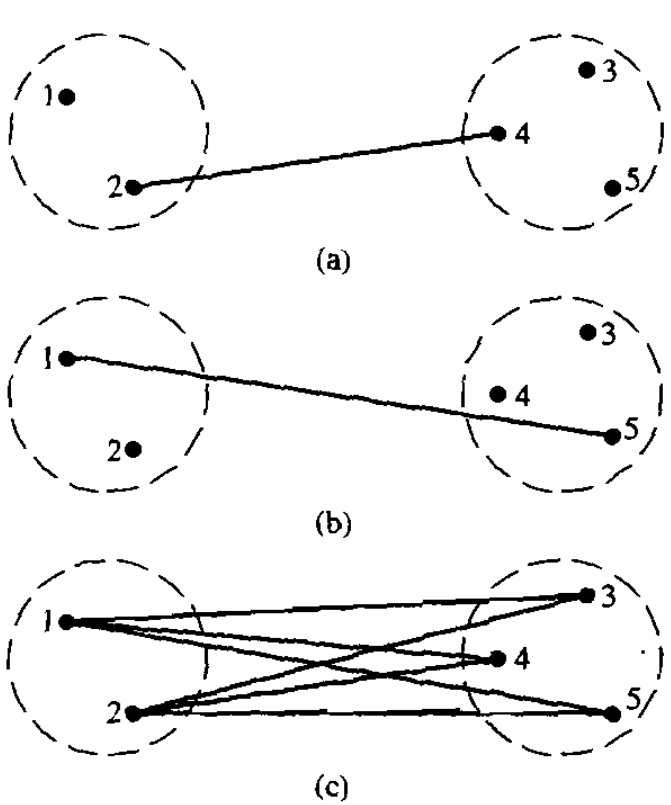
Let $C_X = \{X_1, \ldots, X_m\}$ and $C_Y = \{Y_1, \ldots, Y_n\}$ denote two clusters.

- X_j is the j-th object in cluster C_X for $j = 1, \ldots, m$
- Y_k is the k-th object in cluster C_Y for $k = 1, \ldots, n$

To quantify the distance between two clusters, we could use:

- Single Linkage: minimum (or nearest neighbor) distance
  $d(C_X, C_Y) = \min_{j,k} d(X_j, Y_k)$
- Complete Linkage: maximum (or furthest neighbor) distance
  $d(C_X, C_Y) = \max_{j,k} d(X_j, Y_k)$
- Average Linkage: average (across all pairs) distance
  $d(C_X, C_Y) = \frac{1}{mn} \sum_{j=1}^{m} \sum_{k=1}^{n} d(X_j, Y_k)$

Visualizing the Different Linkage Methods



Cluster distance

d_{24}

d_{15}

$$\frac{d_{13} + d_{14} + d_{15} + d_{23} + d_{24} + d_{25}}{6}$$

Figure 12.2 Intercluster distance (dissimilarity) for (a) single linkage, (b) complete linkage, and (c) average linkage.

Figure: Figure 12.2 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern).

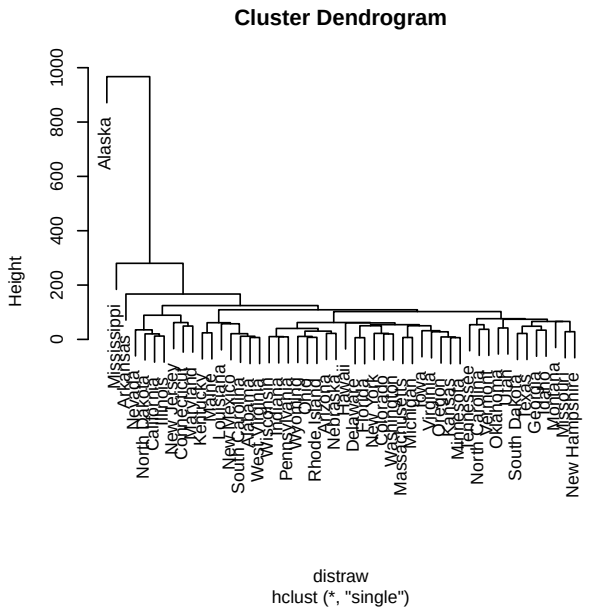
States Example: Dissimilarity Matrix

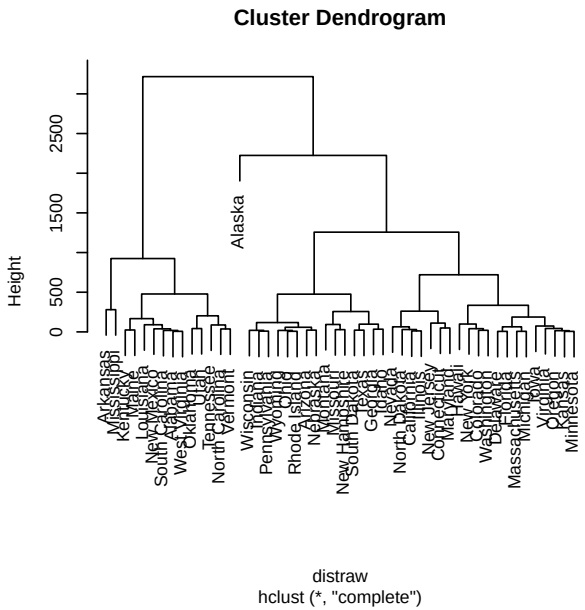
```
# look at states data
> ?state.x77
> vars <- c("Income","Illiteracy","Life Exp","HS Grad")
> head(state.x77[,vars])
           Income Illiteracy Life Exp HS Grad
Alabama      3624        2.1    69.05    41.3
Alaska       6315        1.5    69.31    66.7
Arizona      4530        1.8    70.55    58.1
Arkansas     3378        1.9    70.66    39.9
California   5114        1.1    71.71    62.6
Colorado     4884        0.7    72.06    63.9
> apply(state.x77[,vars], 2, mean)
    Income Illiteracy   Life Exp    HS Grad
 4435.8000     1.1700    70.8786    53.1080
> apply(state.x77[,vars], 2, sd)
     Income  Illiteracy    Life Exp     HS Grad
614.4699392   0.6095331   1.3423936   8.0769978

# create distance (raw and standarized)
> distraw <- dist(state.x77[,vars])
> diststd <- dist(scale(state.x77[,vars]))
```

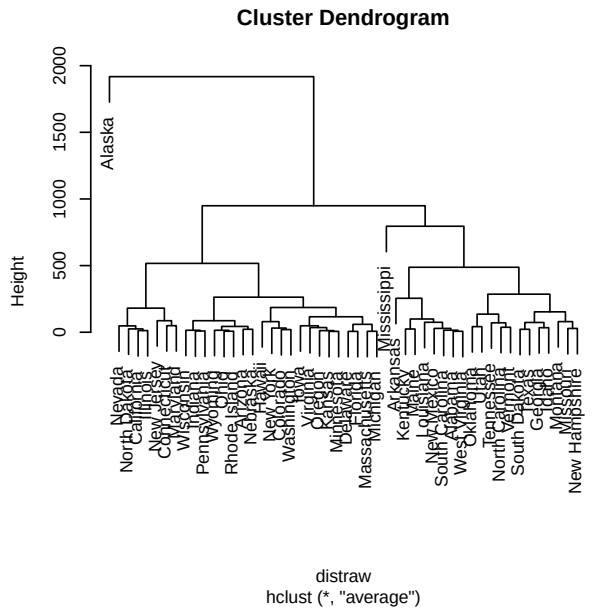
States Example: HCA via Three Linkage Methods

```
# hierarchical clustering (raw data)
> hcrawSL <- hclust(distraw, method="single")
> hcrawCL <- hclust(distraw, method="complete")
> hcrawAL <- hclust(distraw, method="average")


# hierarchical clustering (standardized data)
> hcstdSL <- hclust(diststd, method="single")
> hcstdCL <- hclust(diststd, method="complete")
> hcstdAL <- hclust(diststd, method="average")
```

States Example: Results for Raw Data

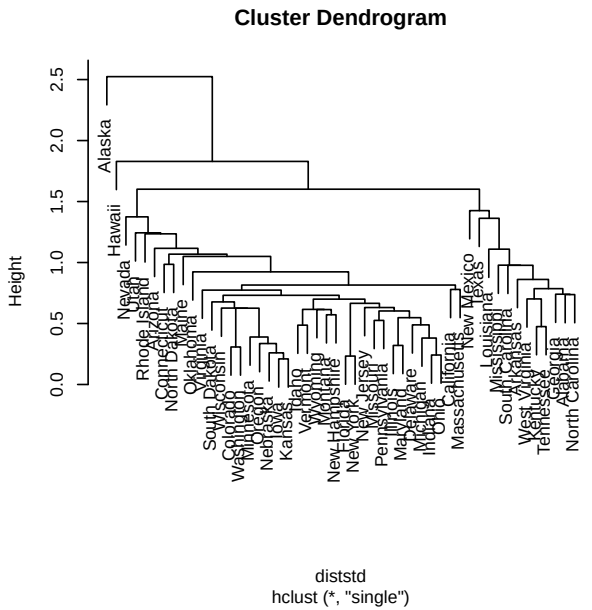


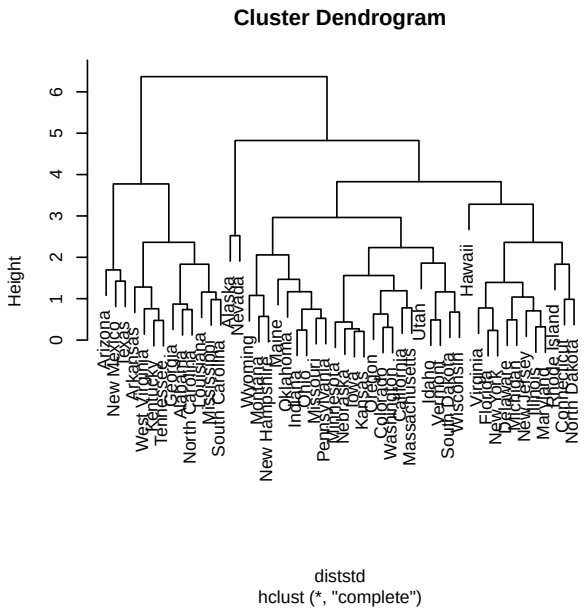


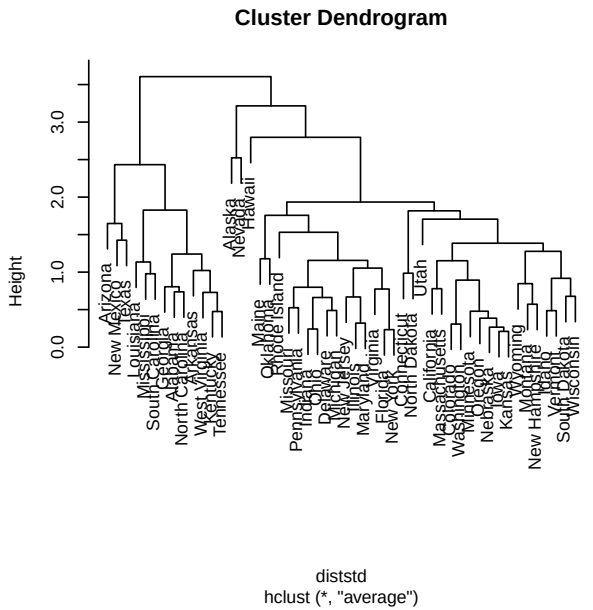



```
plot(hcrawSL)
plot(hcrawCL)
plot(hcrawAL)
```

States Example: Results for Standardized Data

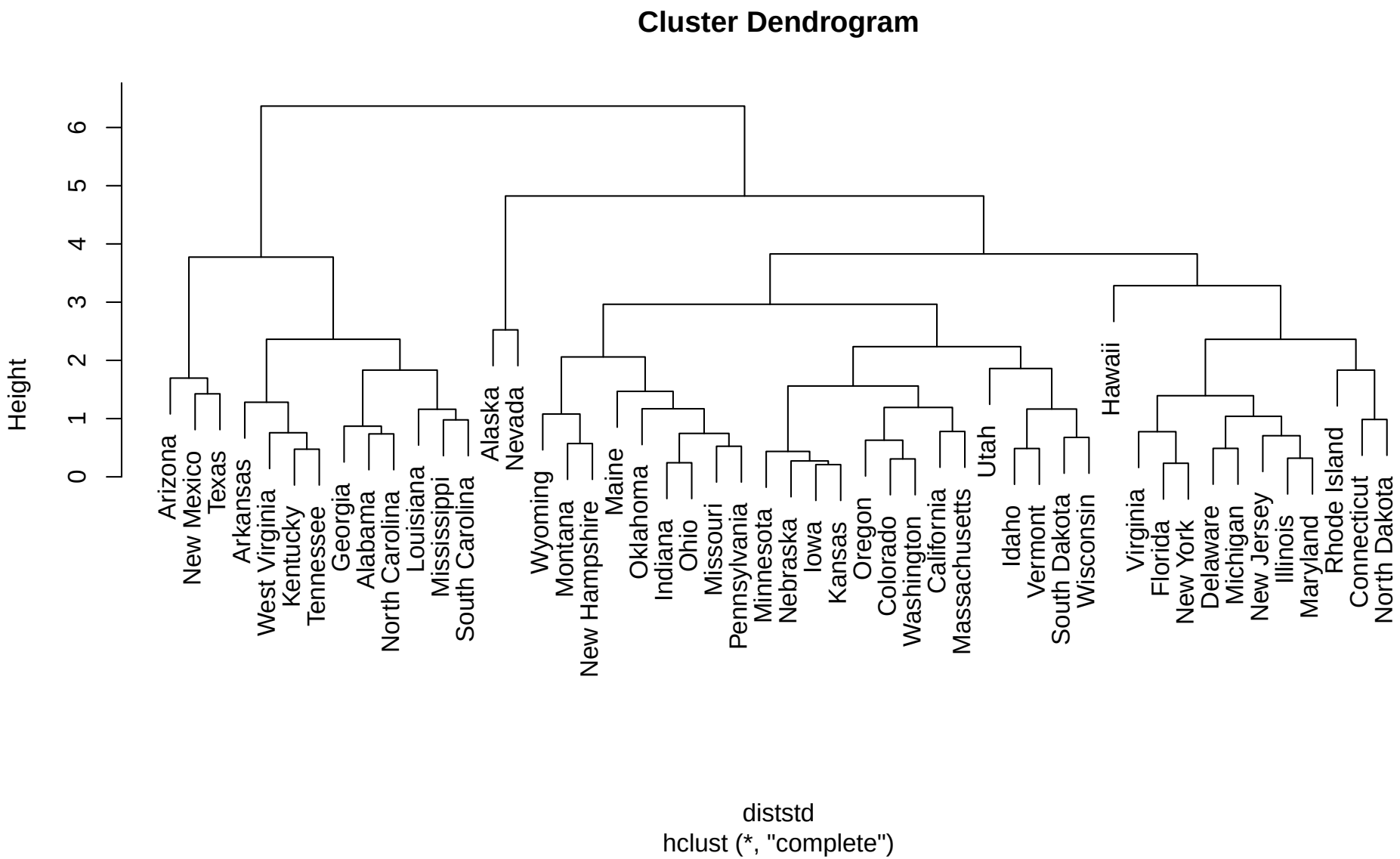







```
plot(hcstdSL)
plot(hcstdCL)
plot(hcstdAL)
```

States Example: Standardized Data w/ Complete Link



Non-Hierarchical Clustering

Non-Hierarchical Clustering: Definition

Non-hierarchical clustering partitions a set of N objects into K distinct groups based on some distance (or dissimilarity).

The number of clusters K can be known a priori or can be estimated as a part of the procedure.

Regardless, we need to start with some initial partition or "seed points" which define cluster centers.

- Try many different randomly generated seed points

K Means: Clustering via Distance to Centroids

K means clustering refers to the algorithm:

1. Partition the N objects into K distinct clusters $C_1, \ldots, C_K$
2. For each $i = 1, \ldots, N$:
   - 2a Assign object X_i to cluster C_k that has closest centroid (mean)
   - 2b Update cluster centroids if X_i is reassigned to new cluster
3. Repeat 2 until all objects remain in the same cluster

Note: we could replace step 1 with "Define K seed points giving the centroids of clusters $C_1, \ldots, C_K$".

It is good to use MANY random starts of the above algorithm.

States Example: K Means on Raw Data

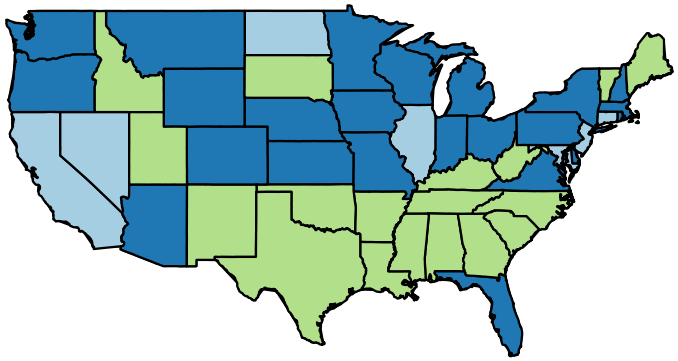
```
# look at states data
> ?state.x77
> vars <- c("Income","Illiteracy","Life Exp","HS Grad")
> apply(state.x77[,vars], 2, mean)
    Income Illiteracy   Life Exp    HS Grad
 4435.8000     1.1700    70.8786    53.1080


# fit k means for k = 2, ..., 10 (raw data)
> kmlist <- vector("list", 9)
> for(k in 2:10){
+     set.seed(1)
+     kmlist[[k-1]] <- kmeans(state.x77[,vars], k, nstart=5000)
+ }
```

States Example: Scree Plot for Raw Data

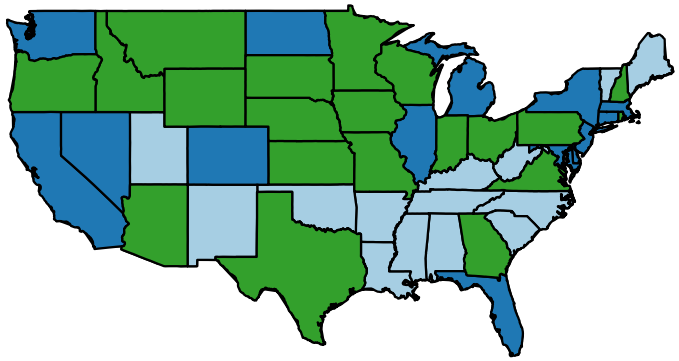

```
tot.withinss <- sapply(kmlist, function(x) x$tot.withinss)
plot(2:10, tot.withinss / kmlist[[1]]$totss, type="b", xlab="# Clusters",
     ylab="SSW / SST", main="Scree Plot: Raw Data")
```

States Example: Cluster Plot for Raw Data

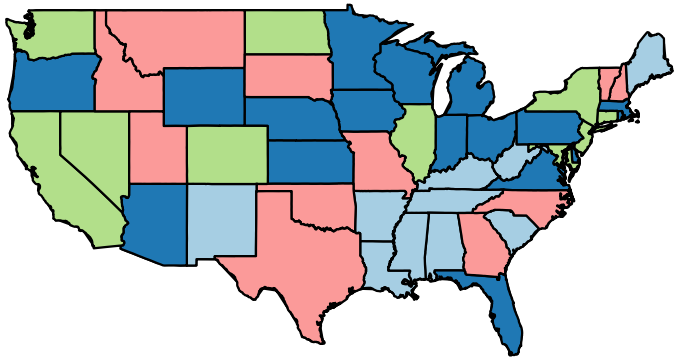
K=3 Clusters: Raw Data



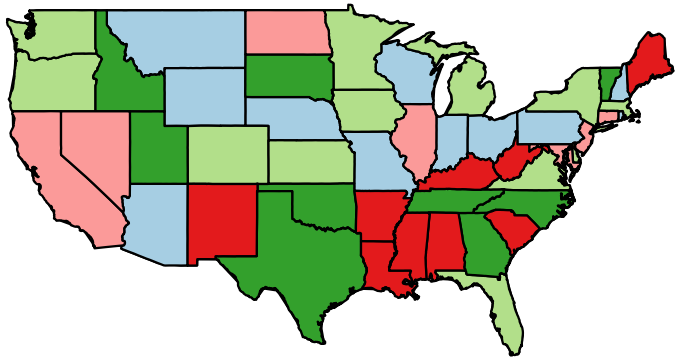
K=4 Clusters: Raw Data



K=5 Clusters: Raw Data



K=6 Clusters: Raw Data



States Example: K Means on Standardized Data

```
# look at states data
> ?state.x77
> vars <- c("Income","Illiteracy","Life Exp","HS Grad")
> apply(state.x77[,vars], 2, mean)
    Income Illiteracy   Life Exp    HS Grad
 4435.8000     1.1700    70.8786    53.1080
```

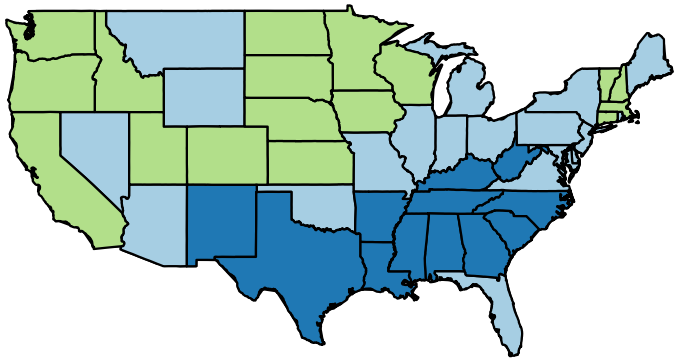
```
# fit k means for k = 2, ..., 10 (standardized data)
> Xs <- scale(state.x77[,vars])
> kmlist.std <- vector("list", 9)
> for(k in 2:10){
+     set.seed(1)
+     kmlist.std[[k-1]] <- kmeans(Xs, k, nstart=5000)
+ }
```

States Example: Scree Plot for Standardized Data

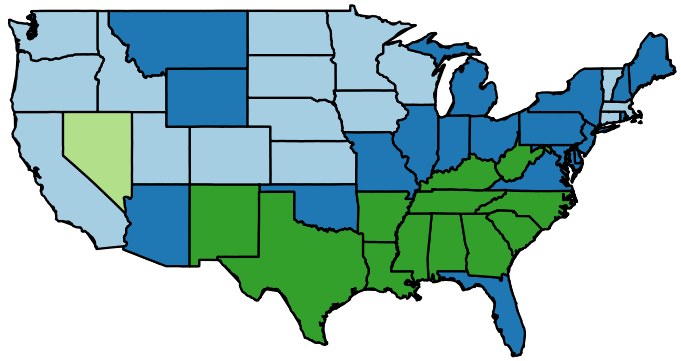

```
tot.withinss.std <- sapply(kmlist.std, function(x) x$tot.withinss)
plot(2:10, tot.withinss.std / kmlist.std[[1]]$totss, type="b",
     xlab="# Clusters", ylab="SSW / SST", main="Scree Plot: Std. Data")
```

States Example: Cluster Plot for Standardized Data

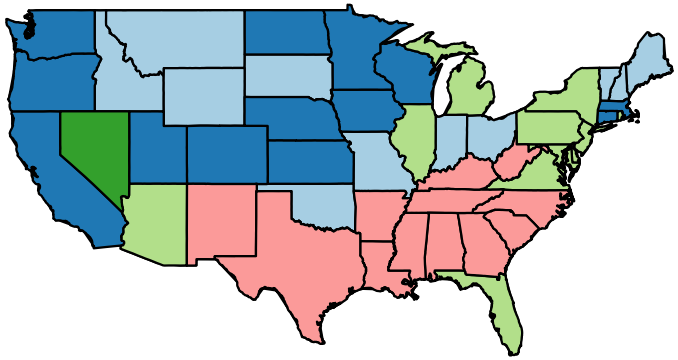
K=3 Clusters: Std. Data



K=4 Clusters: Std. Data



K=5 Clusters: Std. Data



K=6 Clusters: Std. Data