

Unsupervised Clustering

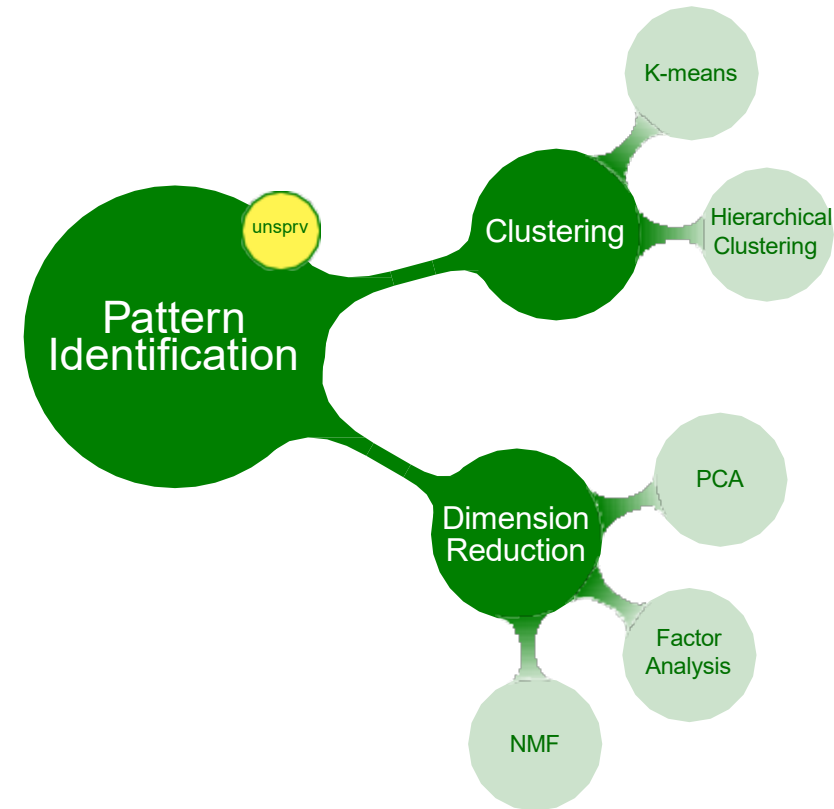
R Ladies – Boulder

Marta Jankowska

Code courtesy of Yanelli Nunez and the Environmental Mixtures
Workshop at Columbia, data courtesy of Dr. Ami Zota

Pattern Recognition

- Set of unsupervised methods (although some have supervised extensions) used to detect patterns of data that give information about a system or dataset
- Clustering
 - Better for finding 'groupings' within your data
- Dimension Reduction
 - Better for reducing the number of features (or variables) under consideration
- Today we are using traditional clustering methods, not machine learning approaches

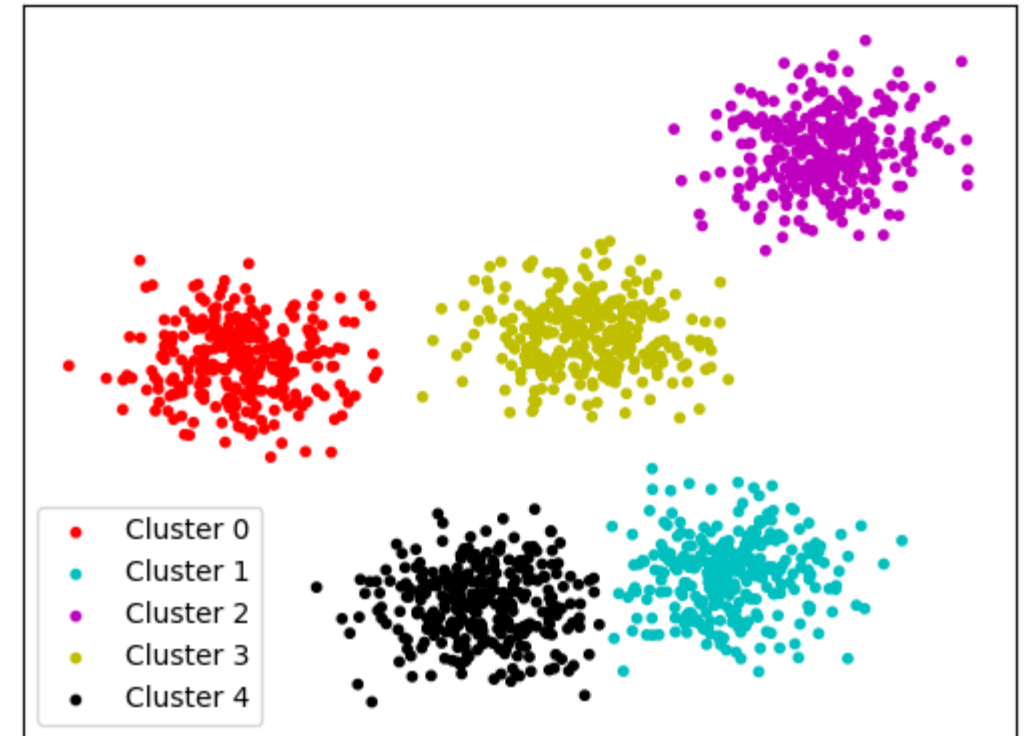


What is clustering?

- Clustering is a set of unsupervised techniques to identify subgroups
- The goal is to partition observations in a dataset into distinct groups so that
 - Observations in a group are like each other
 - Observations in different groups are different from each other

K-means Clustering

- Partitions data into a pre-specified (k) number of non-overlapping clusters
 - Iterative
 - Goal is to minimize within-cluster variation and maximize between-cluster distance
- Definition of 'distance'
 - Usually using Euclidean distance

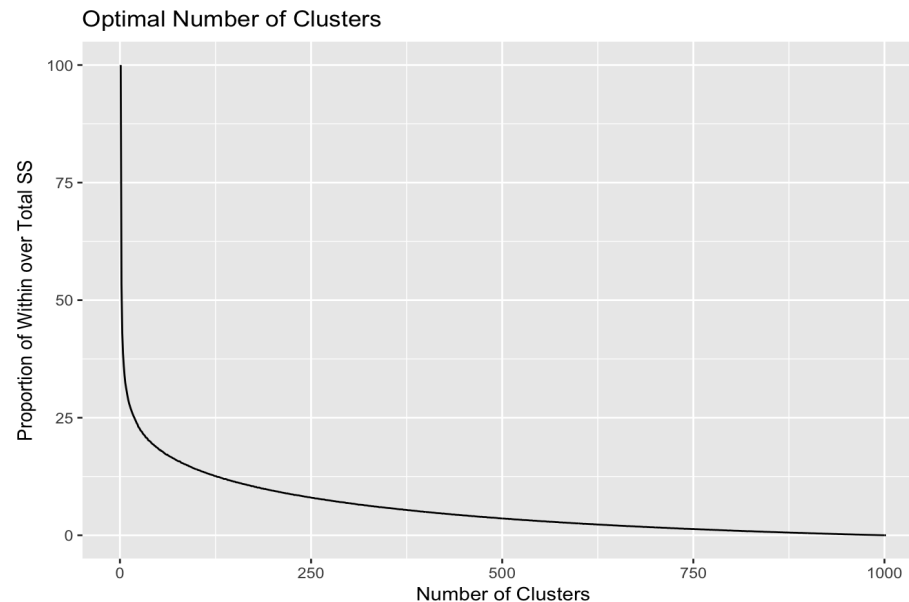


K-means in R (part of the stats package)

- `kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"), trace=FALSE)`
- `x` = numeric matrix of data
- `centers` = number of clusters
- `iter.max` = maximum number of iterations (check for convergence errors and increase if needed)
- `nstart` = how many random sets should be chosen
- `algorithm` = k-means clustering algorithm used, default is Hartigan-Wong

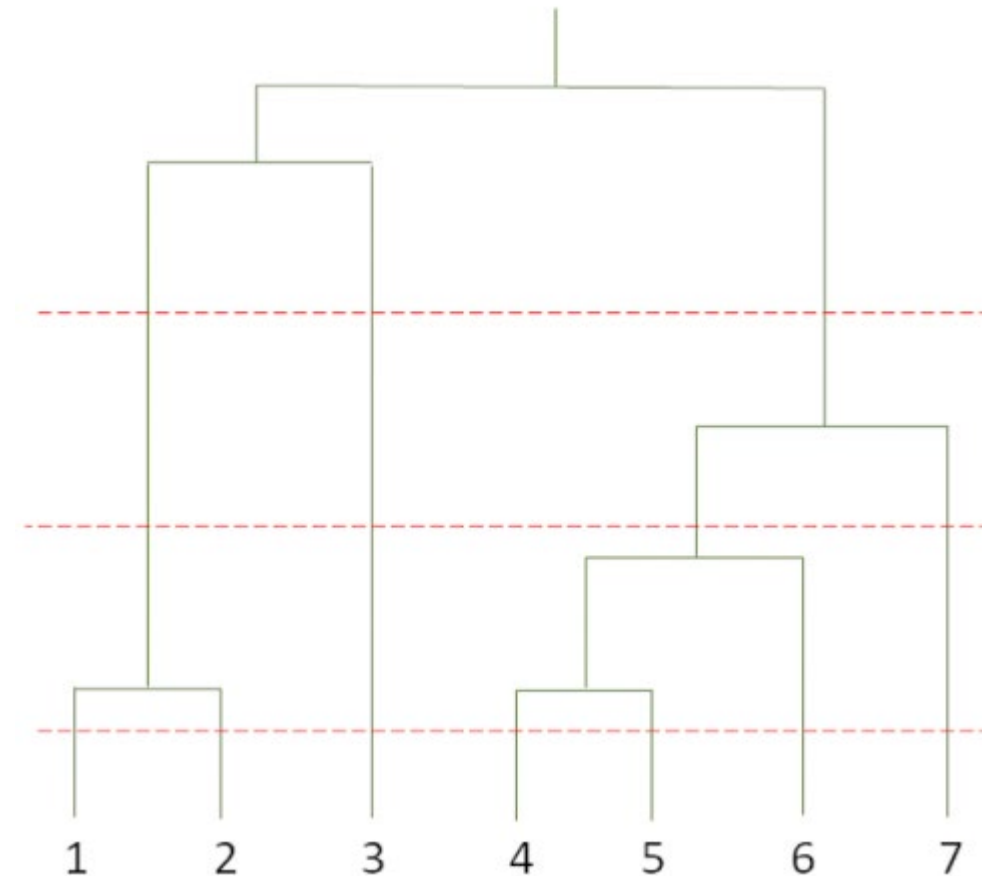
How to choose K?

- Ultimately you know your data best, and you are the expert.
- But...
 - You can also try several outputs and plot number of clusters against within cluster variation (want that number to be smallest possible!) and look for where you are getting minimizing returns



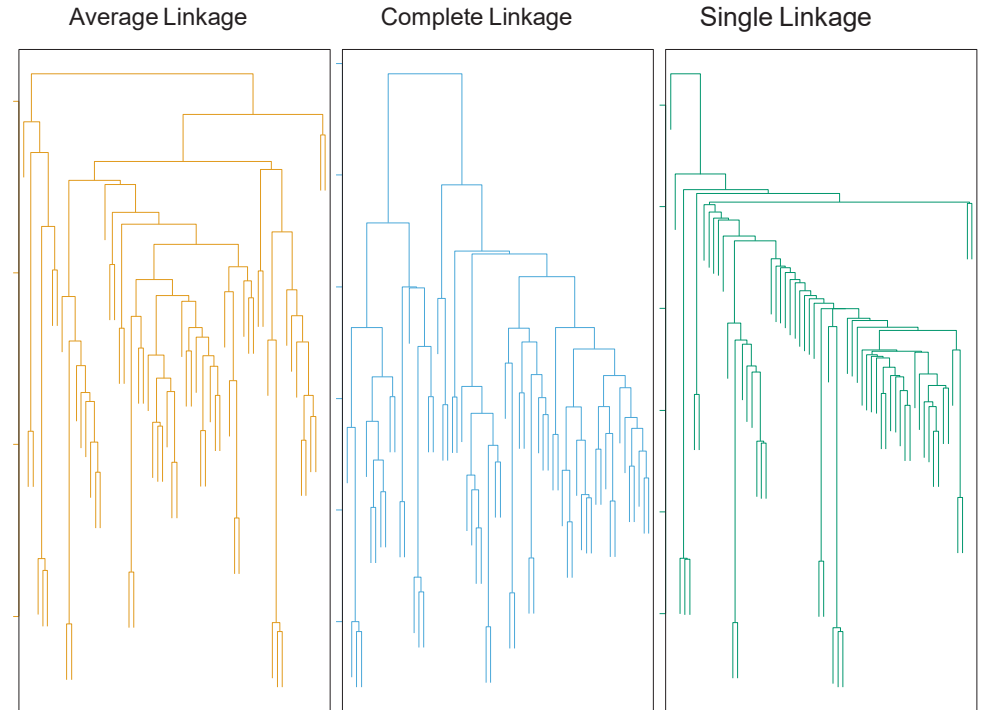
Hierarchical clustering

- Tree-like visual representation of all possible clusters
- Don't need to pre-specify how many clusters you want, but do need to decide when to 'chop' your tree
- Agglomerative clustering
 - Starts from the bottom (each individual data point) and builds clusters as it moves up the tree and fuses more similar data points together
 - Data in earlier fusion groups will be more similar to each other than later fusion groups



Linkage

- Dissimilarity between two groups of observations
- Average
 - Mean inter-cluster dissimilarity
 - robust against noise
- Complete
 - Max inter-cluster dissimilarity
 - compact clusters
- Single
 - Min inter-cluster dissimilarity
 - Extended, trailing clusters



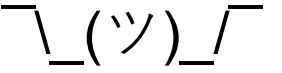
Hierarchical Clustering in R (part of the stats package)

- `hclust(d, method = "complete", members = NULL)`
 - `d` = dissimilarity structure as produced by `dist`
 - `method` = agglomeration method to be used
-
- There is also an `hclust` function in the `fastcluster` package

Some notes...

- KNOW. YOUR. DATA.
 - Aim for interpretable results and solutions
- Might be a good idea to scale data so that a variable with values that is very different from other values doesn't skew your clusters
- Use clusters for:
 - Data exploration
 - Categorical variables in a statistical model
 - Effect modification between one of the cluster member variables and the outcome of interest
 - Identify data subgroups and assess as modifiers in a different relationship you are looking to quantify

Libraries we are using...

- tidyverse – collection of packages for data science
- janitor – data cleaning (formatting, crosstabs, duplicates)
- ggcorrplot – correlation matrix visualization
- ggfortify – stats analysis visualization
- ggdendro – tree diagram and dendrogram visualization
- ggplotify – conversion function to make things compatible with grid and ggplot
- gridExtra – set of add on functions to grid graphics
- knitr – dynamic report generation in R using Literate Programming techniques
- dendextend – extends dendrogram functionality in R (visual and statistical comparisons)
- Pryr – “provides tools to pry back the surface of R and dig into the details” 
- reshape2 – tools for reshaping data (reshape version 2)