The **Euclidean distance** is appropriate when I have continuous numerical variables and I want to reflect absolute distances. This distance takes into account every variable and doesn’t remove redundancies.

# Non-Hierarchical methods:

1. **K-means**

k-means can only handle numerical data and the results might be skewed if we do not normalize it

k-means algorithm, the resulting set of clusters strongly depends on the selection of initial centroids which is random.

K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular (spherical cluster).

 With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).

Different initial partitions can result in different final clusters.

 Despite the fact that k-means is easy to use we showed that is also comes with some disadvantages. These disadvantages don’t make it impossible to use k-means but they might affect the results of the analysis. This should be considered while interpreting the results.

* *Sensitive to scale:* Rescaling your datasets will completely change results. While this itself is not bad, not realizing that *you have to spend extra attention to scaling your data* is bad. Scaling factors are extra dd hidden parameters in k-means that "default" to 1 and thus are easily overlooked, yet have a major impact (but of course this applies to many other algorithms, too).
* k-means cannot find non-convex clusters, it can find only circular shaped clusters.
* Does not work well with [non-globular](http://www.improvedoutcomes.com/docs/WebSiteDocs/Glossary/Glossary_of_Terms_Acronym_List.htm#N_Index) clusters.
* K-Means is also dependent upon initialization; give it multiple different random starts and you can get multiple different clusterings. This does not engender much confidence in any individual clustering that may result.
* It does not work well with clusters (in the original data) of Different size and Different density

For our dataset, Kmeans works better since data is normalized, Clustering is mostly globular. Kmeans did give good number of clustering groups. Feature importance is consistence with other algorithms used.

1. **Spectral clustering**

Spectral Clustering: Builds models based on non-convex cluster structure, or when a measure of the center is not a suitable description of the complete cluster.

Suitable for non-globular clusters

Although, Spectral clustering has done a decent job of our dataset, it is more time consuming.

# Hierarchical methods

The only problem with the technique is that it is able to only handle small number of data-points and is very time consuming. This is because it tries to calculate the distance between all possible combination and then takes one decision to combine two groups/individual data-point.

1. Wards:

* Biased towards globular clusters
* Chaining effect
* **Ward’s** method, or minimal **increase of sum-of-squares** (MISSQ), sometimes incorrectly called "minimum variance" method. Proximity between two clusters is the magnitude by which the summed square in their joint cluster will be greater than the combined summed square in these two clusters , Results in too few clusters if distances are too close.

1. Complete Linkage / **farthest neighbour clustering**

In complete-linkage clustering, the link between two clusters contains all element pairs, and the distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other

1. Average Linkage

Average-link clustering merges in each iteration the pair of clusters with the highest cohesion.

Time complexity

 In average-link clustering, every subset of vectors can have a different cohesion, so we cannot precompute all possible cluster-cluster similarities.

Hierarchical clustering is certainly not suitable for our dataset. Ward returned too few clusters since data is very close to each other. This might be due to the fact we used T-SNE approach which lost the distance information to an extent. Average and complete did give more than 2 clusters but not give good silhouette score and overall processing time I a little high.

# T-sne

t-SNE reveals approximate contiguity in an underlying high-dimensional manifold, so clusters on the low-dimensional representation of the high-dimensional space maximize the "likelihood" that contiguous individuals will not be in the same cluster

The problem with t-SNE is that it does not preserve distances nor density. It only to some extend preserves nearest-neighbors. The difference is subtle, but affects any density- or distance based algorithm.