K Means Clustering in R Programming is an Unsupervised Non-linear algorithm that cluster data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster. In the unsupervised algorithm, high reliance on raw data is given with large expenditure on manual review for review of relevance is given. It is used in a variety of fields like Banking, healthcare, retail, Media, etc.

**Theory**
K-Means clustering groups the data on similar groups. The algorithm is as follows:

1. Choose the number **K** clusters.
2. Select at random K points, the centroids(Not necessarily from the given data).
3. Assign each data point to closest centroid that forms K clusters.
4. Compute and place the new centroid of each centroid.
5. Reassign each data point to new cluster.

After final reassignment, name the cluster as Final cluster.

**The Dataset**
**Iris** dataset consists of 50 samples from each of 3 species of Iris(Iris setosa, Iris virginica, Iris versicolor) and a multivariate dataset introduced by British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems. Four features were measured from each sample i.e length and width of the sepals and petals and based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.

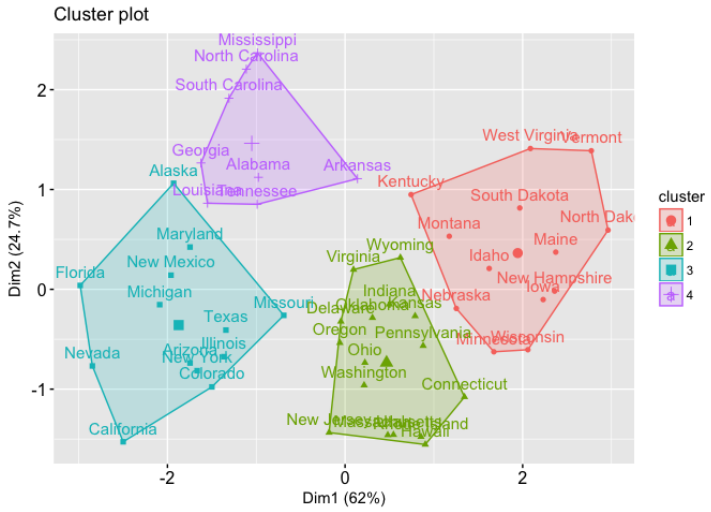
# Optimal k

One technique to choose the best k is called the **elbow method**. This method uses within-group homogeneity or within-group heterogeneity to evaluate the variability. In other words, you are interested in the percentage of the variance

explained by each cluster. You can expect the variability to increase with the number of clusters, alternatively, heterogeneity decreases. Our challenge is to find the k that is beyond the diminishing returns. Adding a new cluster does not improve the variability in the data because very few information is left to explain.
In this tutorial, we find this point using the heterogeneity measure. The Total within clusters sum of squares is the tot.withinss in the list return by kmean(). You can construct the elbow graph and find the optimal k as follow:

- Step 1: Construct a function to compute the total within clusters sum of squares
- Step 2: Run the algorithm times
- Step 3: Create a data frame with the results of the algorithm
- Step 4: Plot the results

# K-means Cluster Analysis




Clustering is a broad set of techniques for finding subgroups of observations within a data set. When we cluster observations, we want observations in the same group to be similar and observations in different groups to be dissimilar. Because there isn't a response variable, this is an unsupervised method, which implies that it seeks to find relationships between the n observations without being trained by a response variable. Clustering allows us to identify which observations are alike, and potentially categorize them therein.

K-means clustering is the simplest and the most commonly used clustering method for splitting a dataset into a set of k groups.

### tl;dr

This tutorial serves as an introduction to the k-means clustering method.

1. Replication Requirements: What you'll need to reproduce the analysis in this tutorial
2. Data Preparation: Preparing our data for cluster analysis
3. Clustering Distance Measures: Understanding how to measure differences in observations
4. K-Means Clustering: Calculations and methods for creating K subgroups of the data
5. Determining Optimal Clusters: Identifying the right number of clusters to group your data

## Replication Requirements

To replicate this tutorial's analysis you will need to load the following packages:

```
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
```

## Data Preparation

To perform a cluster analysis in R, generally, the data should be prepared as follows:

1. Rows are observations (individuals) and columns are variables
2. Any missing value in the data must be removed or estimated.the data must be standardized (i.e., scaled) to make variable

comparable. Recall that, standardization consists of transforming the variables such that they have mean zero and standard deviation one.[1]

Here, we'll use the built-in R data set `USArrests`, which contains statistics in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. It includes also the percent of the population living in urban areas

```
df <- USArrests
```

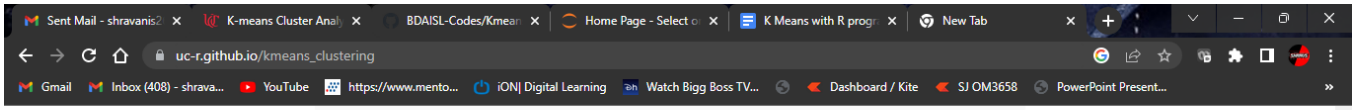
To remove any missing value that might be present in the data, type this:

```
df <- na.omit(df)
```

As we don't want the clustering algorithm to depend to an arbitrary variable unit, we start by scaling/standardizing the data using the R function `scale`:

```
df <- scale(df)
head(df)
```

```
##                Murder   Assault   UrbanPop         Rape
## Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
## Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
## Arizona    0.07163341 1.4788032  0.9989801  1.042878388
## Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
## California 0.27826823 1.2628144  1.7589234  2.067820292
## Colorado   0.02571456 0.3988593  0.8608085  1.864967207
```




## Clustering Distance Measures

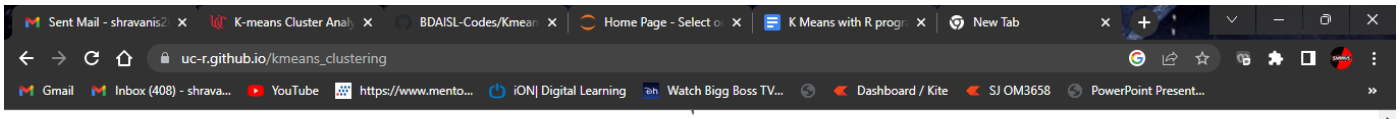
The classification of observations into groups requires some methods for computing the distance or the (dis)similarity between each pair of observations. The result of this computation is known as a dissimilarity or distance matrix. There are many methods to calculate this distance information; the choice of distance measures is a critical step in clustering. It defines how the similarity of two elements (x, y) is calculated and it will influence the shape of the clusters.

The choice of distance measures is a critical step in clustering. It defines how the similarity of two elements (x, y) is calculated and it will influence the shape of the clusters. The classical methods for distance measures are *Euclidean* and *Manhattan distances*, which are defined as follow:

**Euclidean distance:**

$$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

**Manhattan distance:**




**Manhattan distance:**

$$d_{man}(x,y) = \sum_{i=1}^{n}|(x_i - y_i)| \qquad (2)$$

Where, $x$ and $y$ are two vectors of length $n$.

Other dissimilarity measures exist such as correlation-based distances, which is widely used for gene expression data analyses. Correlation-based distance is defined by subtracting the correlation coefficient from 1. Different types of correlation methods can be used such as:

**Pearson correlation distance:**

$$d_{cor}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad (3)$$

**Spearman correlation distance:**

The spearman correlation method computes the correlation between the rank of $x$ and the rank of $y$ variables.

**Spearman correlation distance:**

The spearman correlation method computes the correlation between the rank of *x* and the rank of *y* variables.

$$d_{spear}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x'_i - \bar{x}')(y'_i - \bar{y}')}{\sqrt{\sum_{i=1}^{n}(x'_i - \bar{x}')^2 \sum_{i=1}^{n}(y'_i - \bar{y}')^2}} \quad (4)$$

Where $x'_i = rank(x_i)$ and $y'_i = rank(y_i)$.

**Kendall correlation distance:**

Kendall correlation method measures the correspondence between the ranking of *x* and *y* variables. The total number of possible pairings of *x* with *y* observations is *n(n − 1)/2*, where *n* is the size of *x* and *y*. Begin by ordering the pairs by the *x* values. If *x* and *y* are correlated, then they would have the same relative rank orders. Now, for each $y_i$, count the number of $y_j > y_i$ (concordant pairs (c)) and the number of $y_j < y_i$ (discordant pairs (d)).

Kendall correlation distance is defined as follow:

$$n_c - n_d$$

Sent Mail - shravanis2 | K-means Cluster Anal | BDAISL-Codes/Kmean | Home Page - Select o | K Means with R progr | New Tab
uc-r.github.io/kmeans_clustering
Gmail | Inbox (408) - shrava... | YouTube | https://www.mento... | iON| Digital Learning | Watch Bigg Boss TV... | Dashboard / Kite | SJ OM3658 | PowerPoint Present...

Kendall correlation distance is defined as follow:

$$d_{kend}(x,y) = 1 - \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \quad (5)$$

The choice of distance measures is very important, as it has a strong influence on the clustering results. For most common clustering software, the default distance measure is the Euclidean distance. However, depending on the type of the data and the research questions, other dissimilarity measures might be preferred and you should be aware of the options.

Within R it is simple to compute and visualize the distance matrix using the functions `get_dist` and `fviz_dist` from the `factoextra` R package. This starts to illustrate which states have large dissimilarities (red) versus those that appear to be fairly similar (teal).

- `get_dist` : for computing a distance matrix between the rows of a data matrix. The default distance computed is the Euclidean; however, `get_dist` also supports distanced described in equations 2-5 above plus others.
- `fviz_dist` : for visualizing a distance matrix