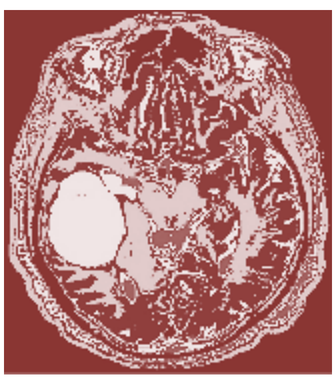


# Seeing the Big Picture

## Segmenting Images to Create Data

15.071x – The Analytics Edge

# Image Segmentation

- Divide up digital images to salient regions/clusters corresponding to individual surfaces, objects, or natural parts of objects

- Clusters should be uniform and homogenous with respect to certain characteristics (color, intensity, texture)

- Goal: Useful and analyzable image representation

# Wide Applications

- Medical Imaging
  - Locate tissue classes, organs, pathologies and tumors
  - Measure tissue/tumor volume

- Object Detection
  - Detect facial features in photos
  - Detect pedestrians in footages of surveillance videos

- Recognition tasks
  - Fingerprint/Iris recognition

# Various Methods

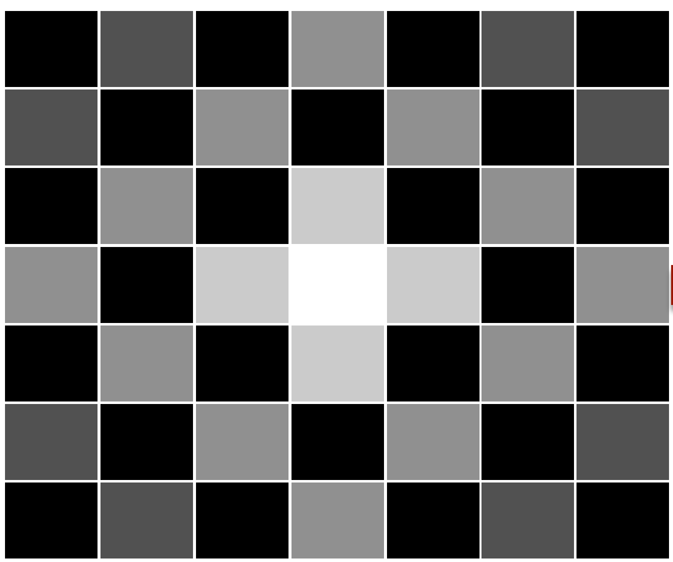
- Clustering methods
  - Partition image to clusters based on differences in pixel colors, intensity or texture
- Edge detection
  - Based on the detection of discontinuity, such as an abrupt change in the gray level in gray-scale images
- Region-growing methods
  - Divides image into regions, then sequentially merges sufficiently similar regions

# In this Recitation…

- Review hierarchical and *k*-means clustering in R
- Restrict ourselves to gray-scale images
  - Simple example of a flower image (flower.csv)
  - Medical imaging application with examples of transverse MRI images of the brain (healthy.csv and tumor.csv)
- Compare the use, pros and cons of all analytics methods we have seen so far

# Grayscale Images

- Image is represented as a matrix of pixel intensity values ranging from 0 (black) to 1 (white)
- For 8 bits/pixel (bpp), 256 color levels



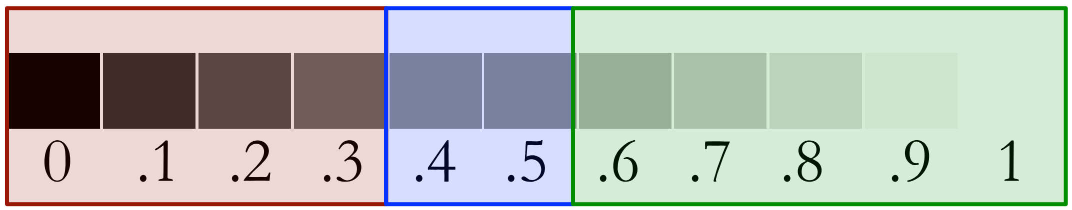
7 pixels

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | .2 | 0 | .5 | 0 | .2 | 0 |
| .2 | 0 | .5 | 0 | .5 | 0 | .2 |
| 0 | .5 | 0 | .7 | 0 | .5 | 0 |
| .5 | 0 | .7 | 1 | .7 | 0 | .5 |
| 0 | .5 | 0 | .7 | 0 | .5 | 0 |
| .2 | 0 | .5 | 0 | .5 | 0 | .2 |
| 0 | .2 | 0 | .5 | 0 | .2 | 0 |

7 pixels

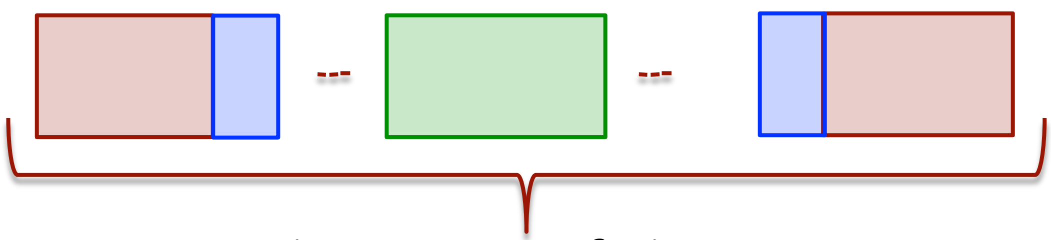
# Grayscale Image Segmentation

- Cluster pixels according to their intensity values




| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | .2 | 0 | .5 | 0 | .2 | 0 |
| .2 | 0 | .5 | 0 | .5 | 0 | .2 |
| 0 | .5 | 0 | .7 | 0 | .5 | 0 |
| .5 | 0 | .7 | 1 | .7 | 0 | .5 |
| 0 | .5 | 0 | .7 | 0 | .5 | 0 |
| .2 | 0 | .5 | 0 | .5 | 0 | .2 |
| 0 | .2 | 0 | .5 | 0 | .2 | 0 |

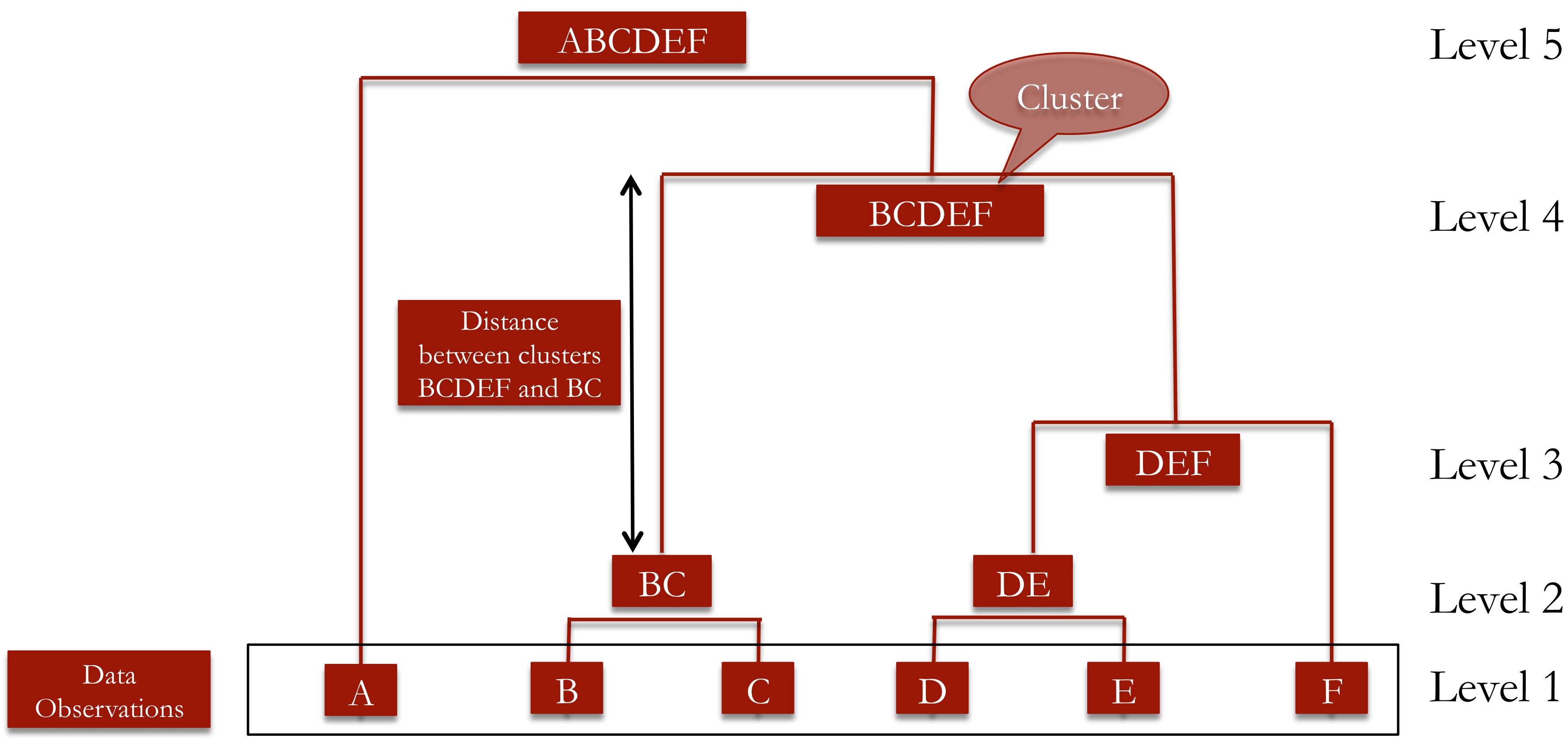
**7 pixels**



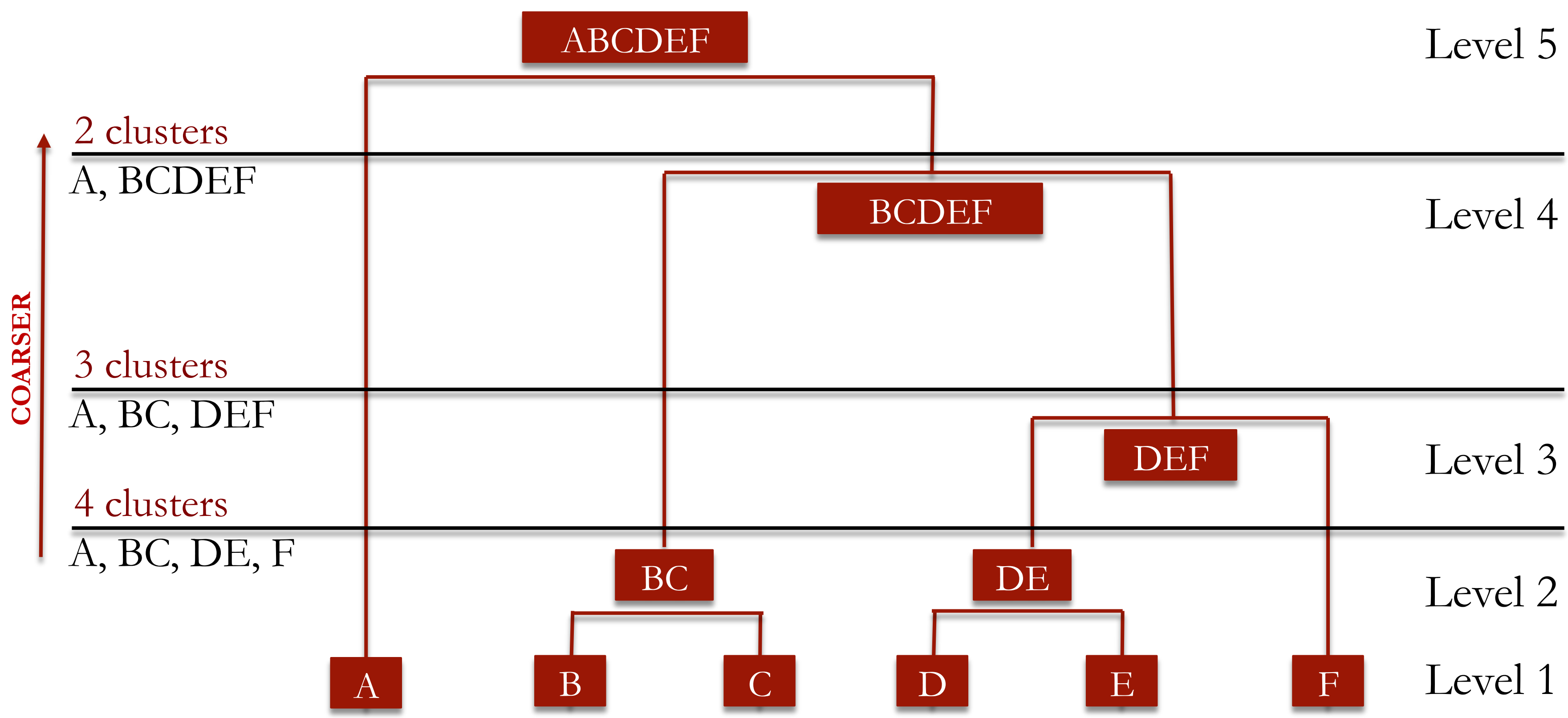
Intensity Vector of size n = 7x7

Pairwise distances n(n-1)/2

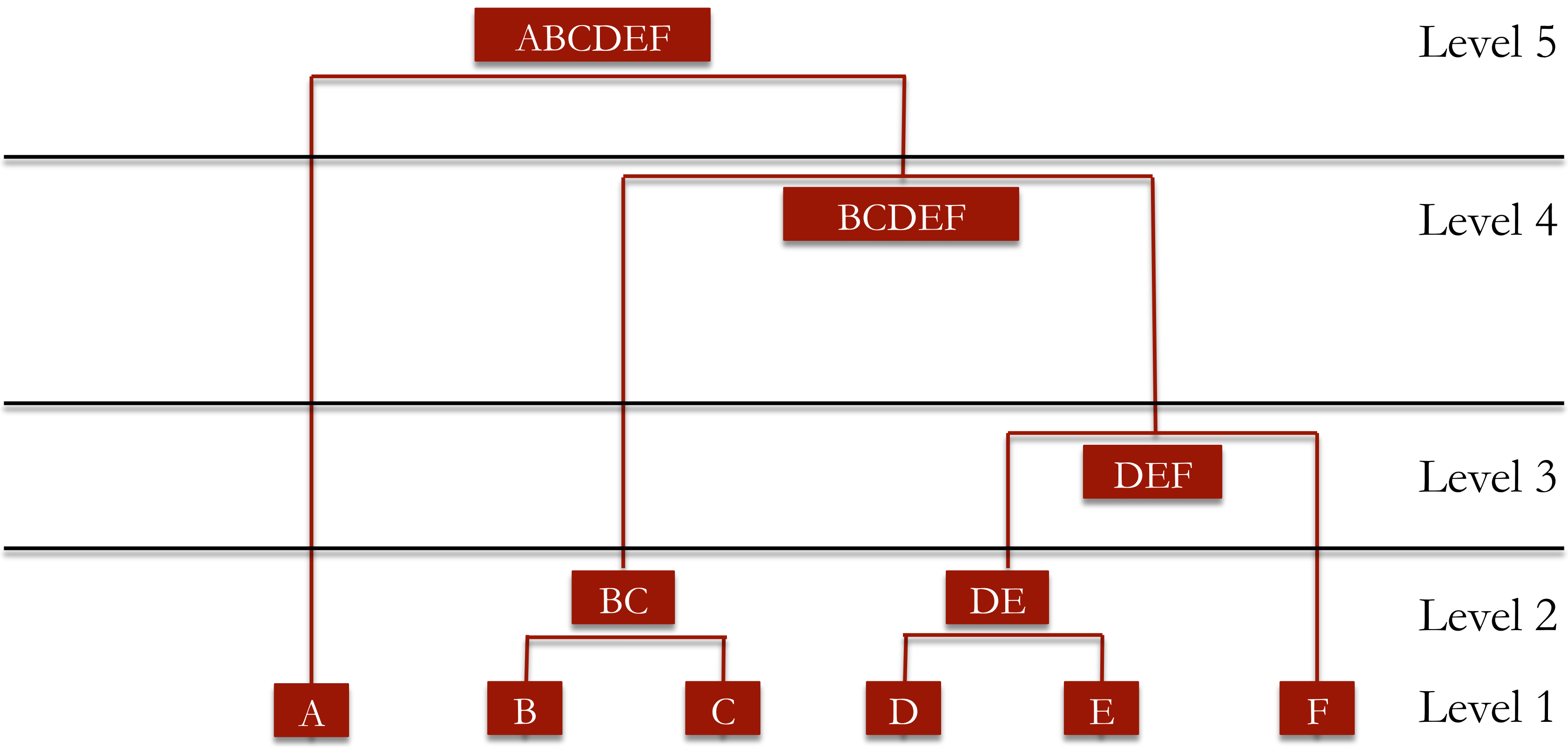
# Dendrogram Example



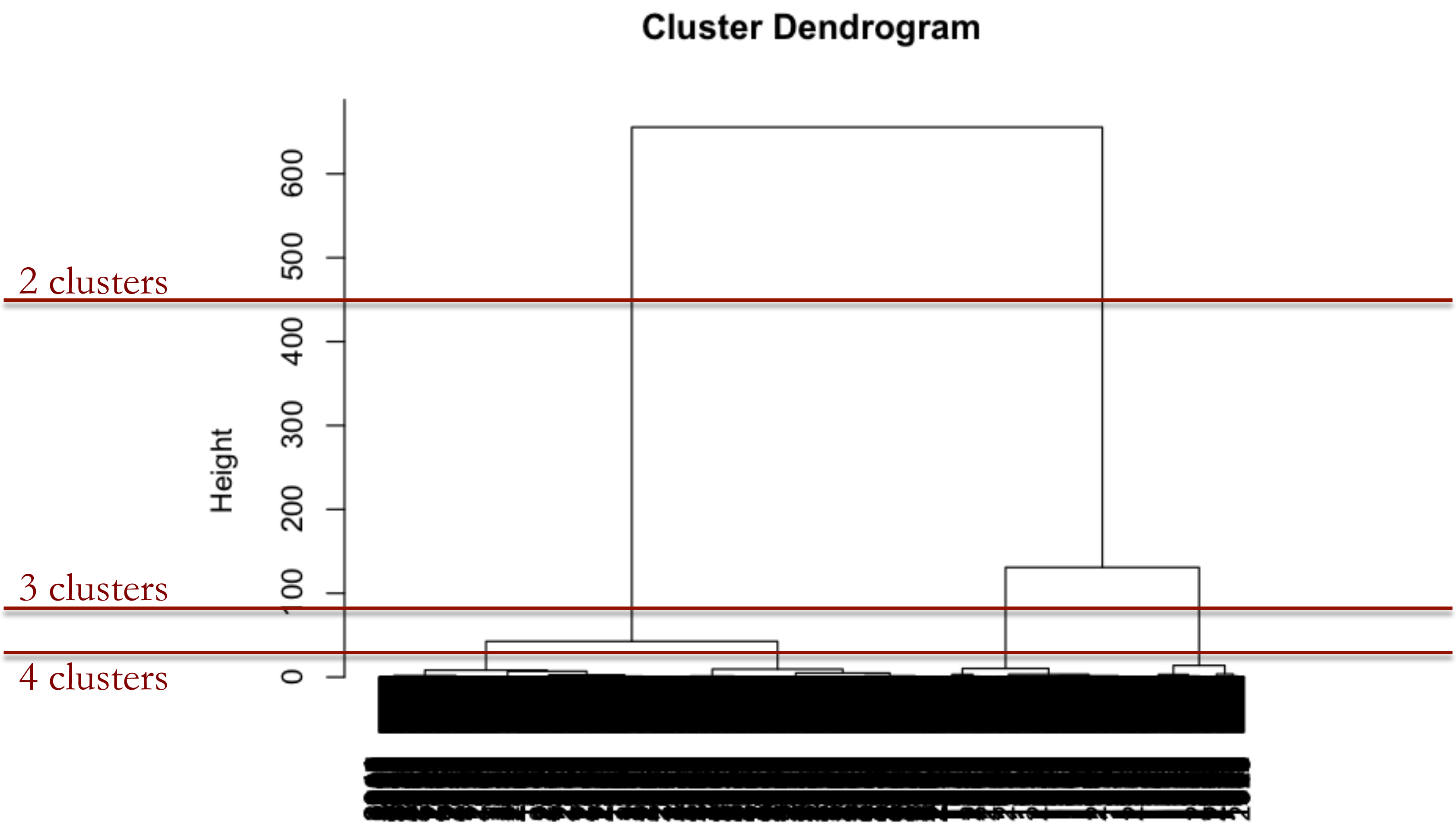
# Dendrogram Example



# Dendrogram Example



# Flower Dendrogram



# $k$-Means Clustering

- The $k$-means clustering aims at partitioning the data into $k$ clusters in which each data point belongs to the cluster whose mean is the nearest

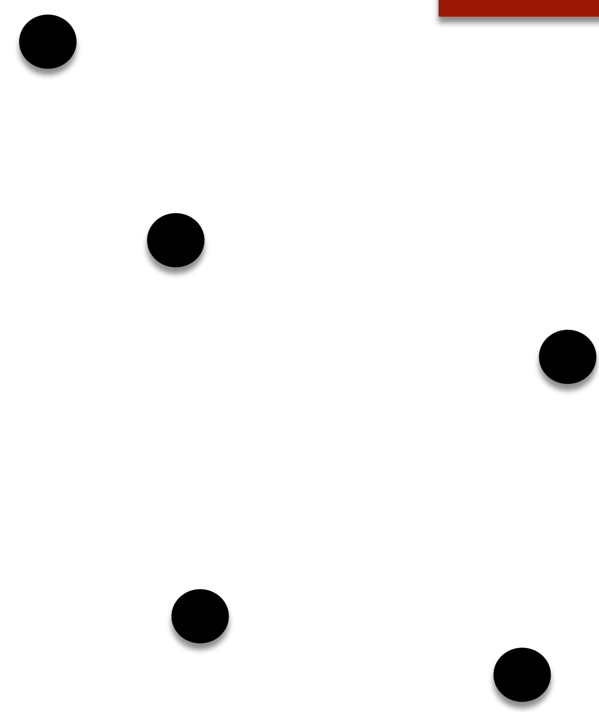
| **$k$-Means Clustering Algorithm** |
|---|
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |
| 3. Compute cluster centroids |
| 4. Re-assign each point to the closest cluster centroid |
| 5. Re-compute cluster centroids |
| 6. Repeat 4 and 5 until no improvement is made |

# $k$-Means Clustering

**$k$-Means Clustering Algorithm**

1. Specify desired number of clusters $k$

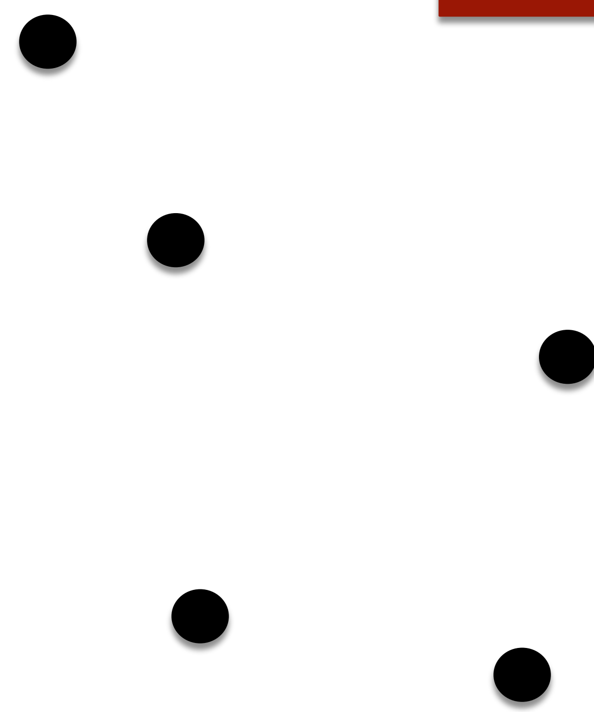
$k = 2$



# $k$-Means Clustering

| **$k$-Means Clustering Algorithm** |
| --- |
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |

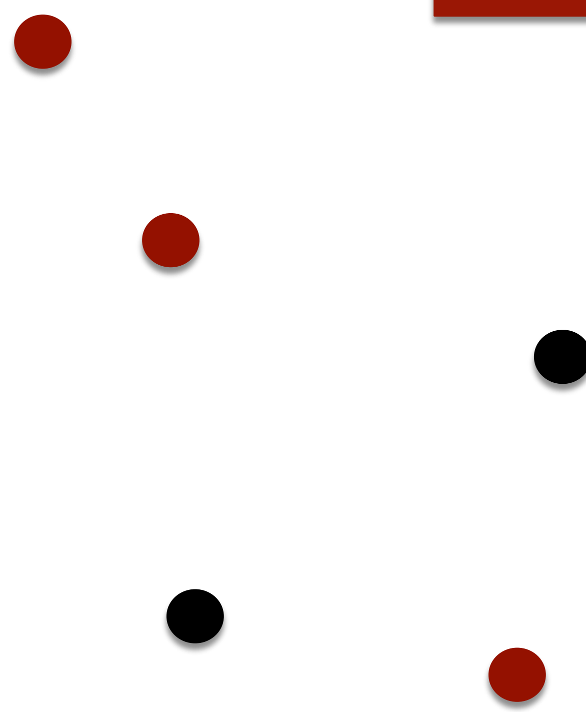
$k = 2$



# $k$-Means Clustering

| **$k$-Means Clustering Algorithm** |
| --- |
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |

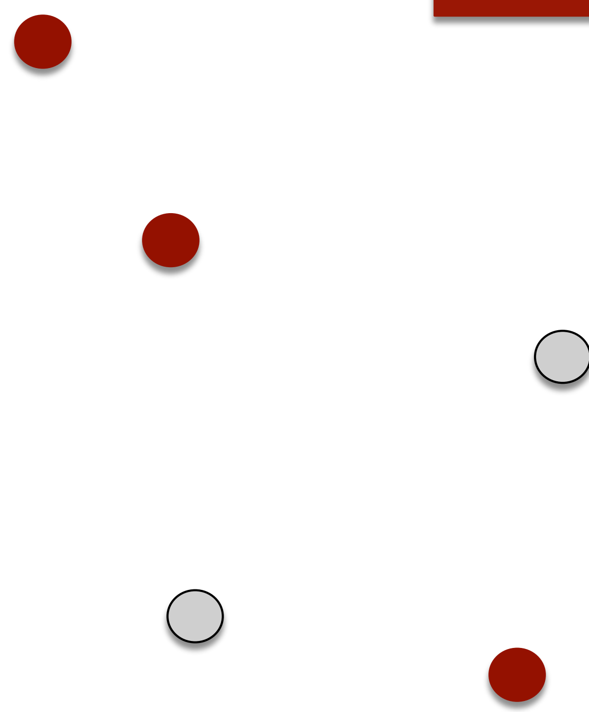
$k = 2$



# $k$-Means Clustering

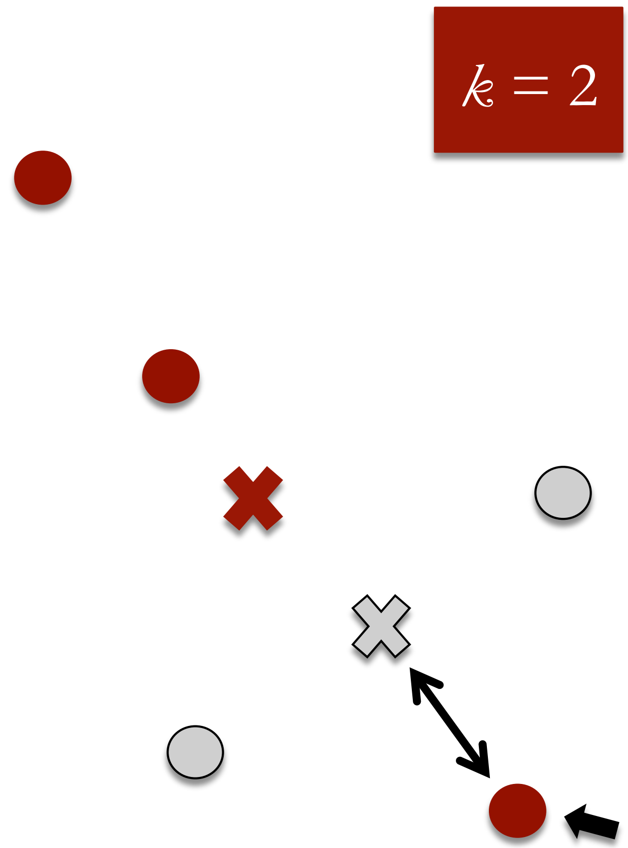
| **$k$-Means Clustering Algorithm** |
| --- |
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |

$k = 2$



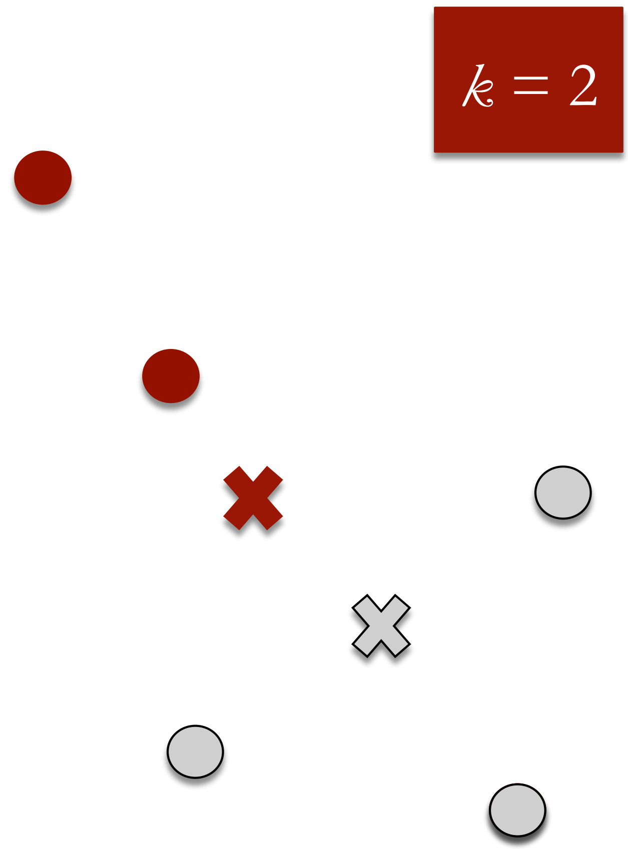
# $k$-Means Clustering

| $k$-Means Clustering Algorithm |
|---|
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |
| 3. Compute cluster centroids |
| 4. Re-assign each point to the closest cluster centroid |



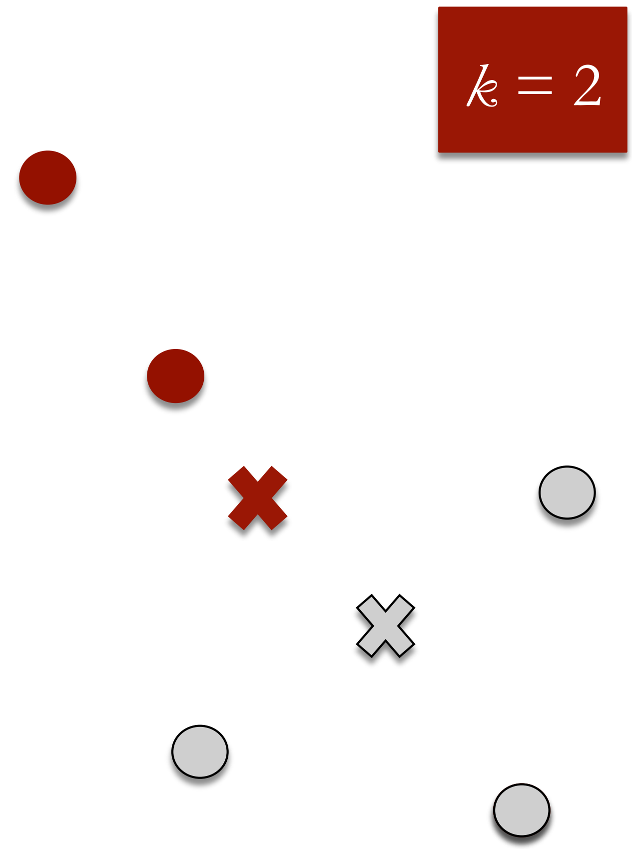
# $k$-Means Clustering

| $k$-Means Clustering Algorithm |
|---|
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |
| 3. Compute cluster centroids |
| 4. Re-assign each point to the closest cluster centroid |

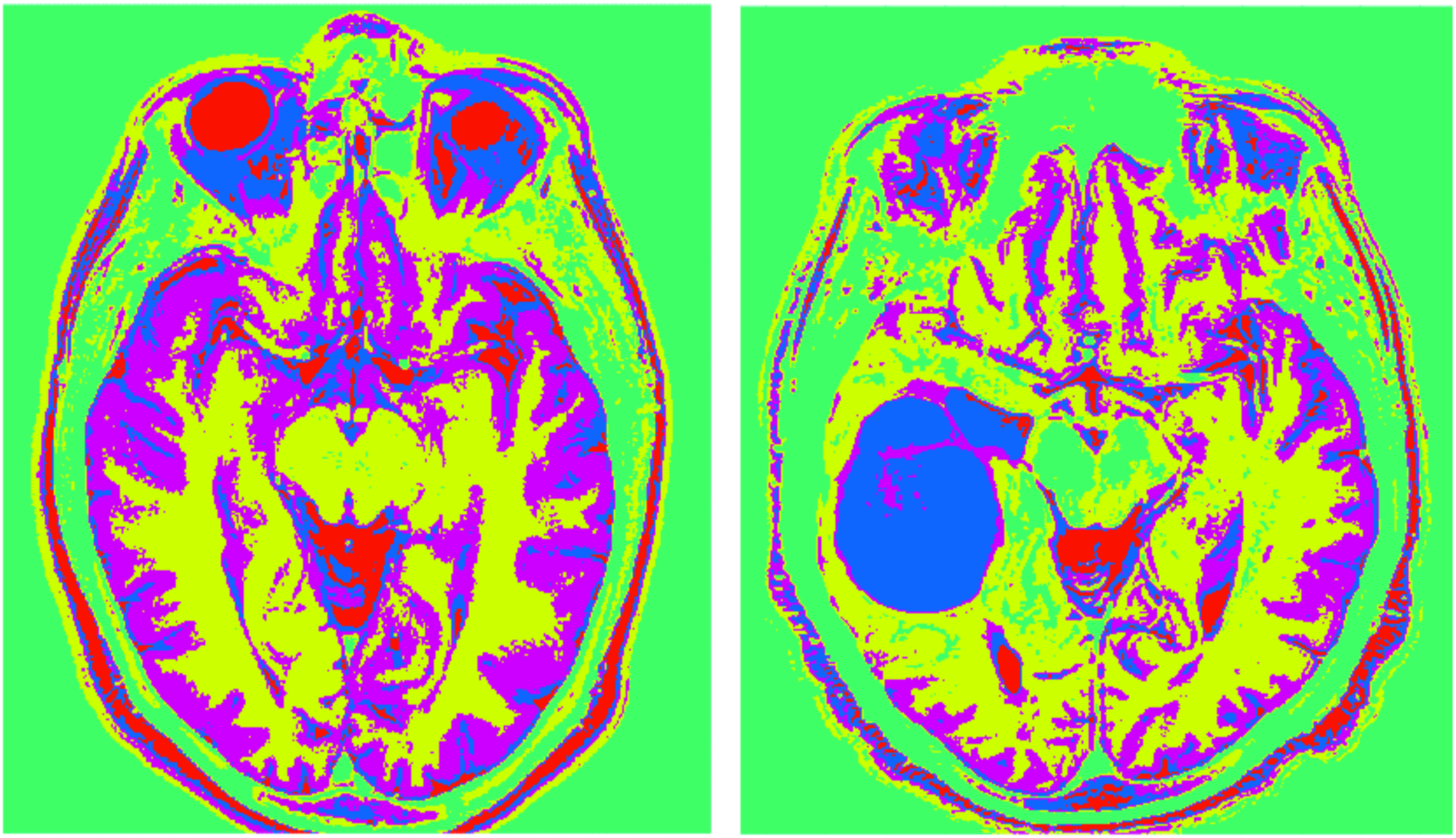


# $k$-Means Clustering

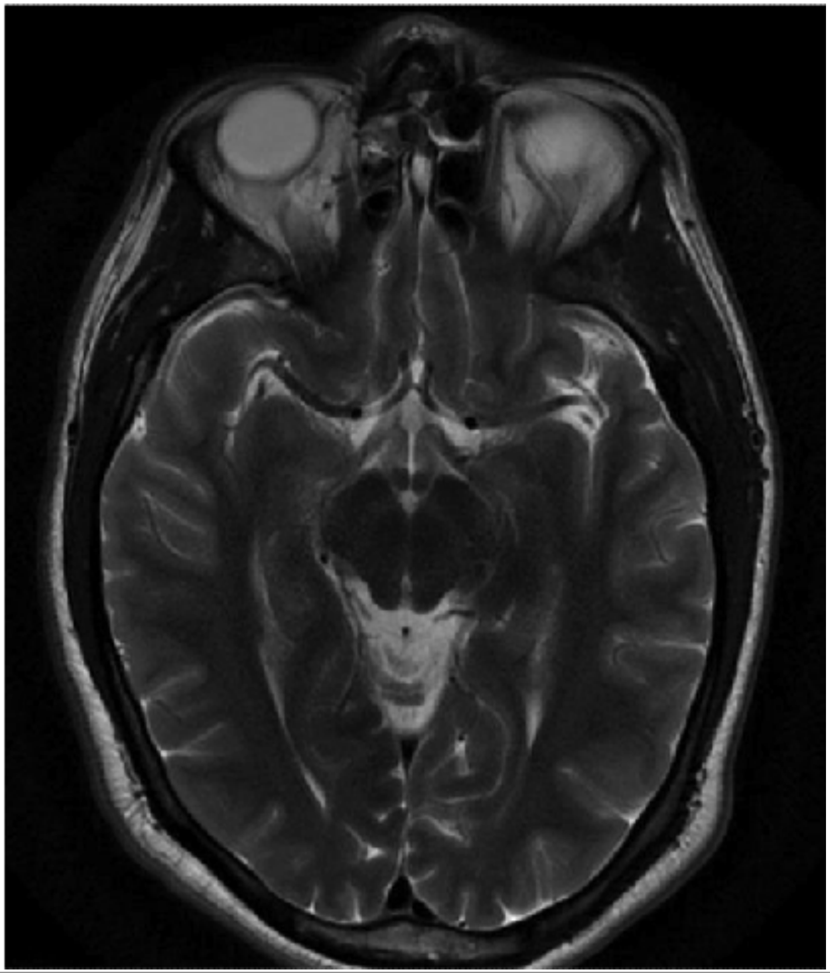
| $k$-Means Clustering Algorithm |
|---|
| 1. Specify desired number of clusters $k$ |
| 2. Randomly assign each data point to a cluster |
| 3. Compute cluster centroids |
| 4. Re-assign each point to the closest cluster centroid |
| 5. Re-compute cluster centroids |
| 6. Repeat 4 and 5 until no improvement is made |

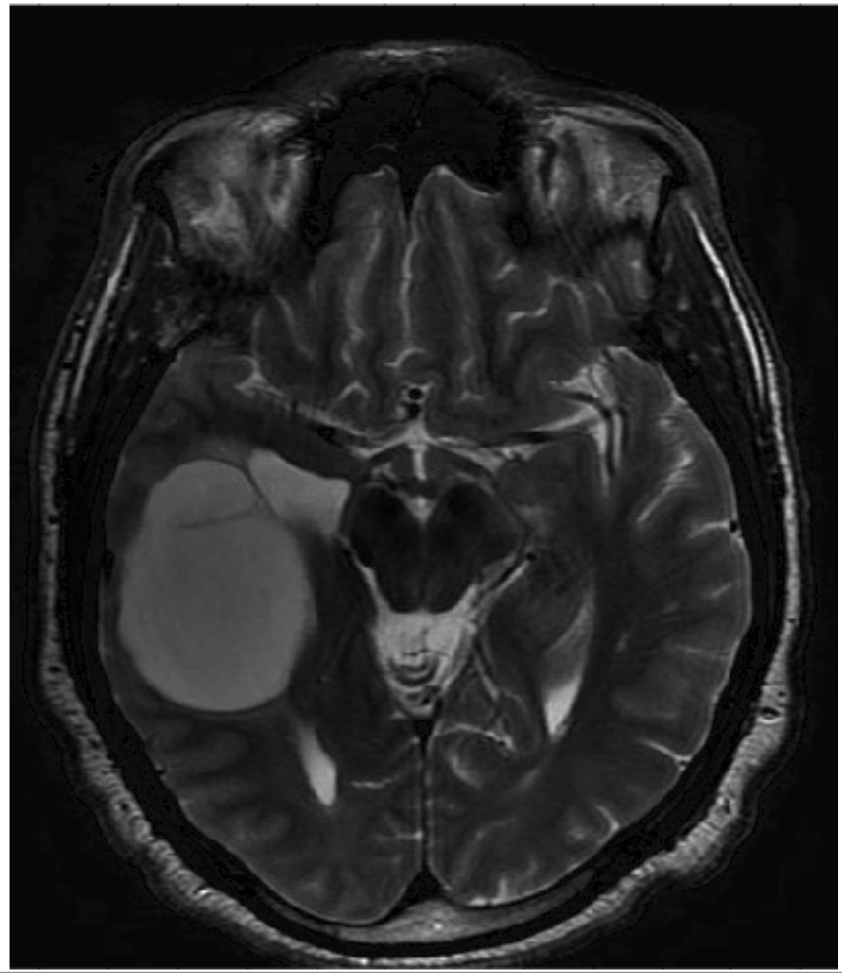


# Segmented MRI Images



# T2 Weighted MRI Images





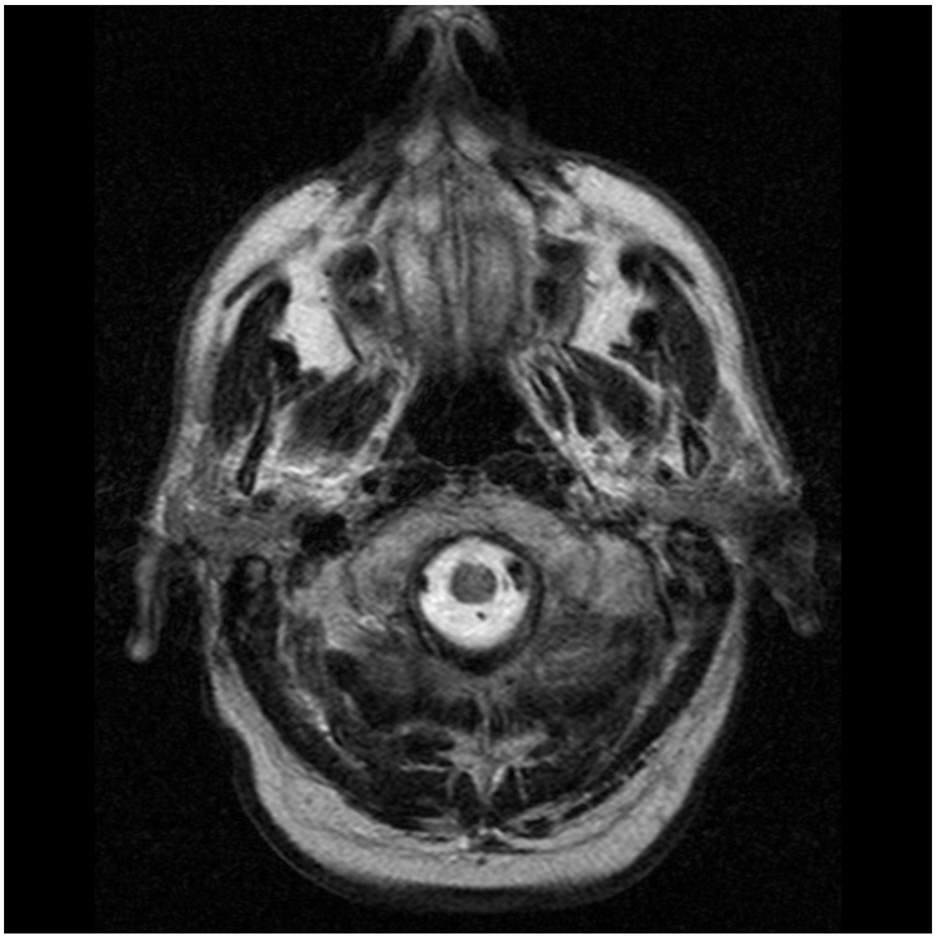
# First Taste of a Fascinating Field

- MRI image segmentation is subject of ongoing research

- *k*-means is a good starting point, but not enough
  - Advanced clustering techniques such as the modified fuzzy k-means (MFCM) clustering technique
  - Packages in R specialized for medical image analysis

    http://cran.r-project.org/web/views/MedicalImaging.html

# 3D Reconstruction



- 3D reconstruction from 2D cross sectional MRI images
- Important in the medical field for diagnosis, surgical planning and biological research

# Comparison of Methods

| | Used For | Pros | Cons |
|---|---|---|---|
| **Linear Regression** | Predicting a continuous outcome (salary, price, number of votes, etc.) | • Simple, well recognized<br>• Works on small and large datasets | • Assumes a linear relationship<br>$Y = a\underbrace{\log(X)}+b$ |
| **Logistic Regression** | Predicting a categorical outcome (Yes/No, Sell/Buy, Accept/Reject, etc.) | • Computes probabilities that can be used to assess confidence of the prediction | • Assumes a linear relationship |

# Comparison of Methods

| | Used For | Pros | Cons |
|---|---|---|---|
| **CART** | Predicting a categorical outcome (quality rating 1--5, Buy/Sell/Hold) or a continuous outcome (salary, price, etc.) | • Can handle datasets without a linear relationship<br>• Easy to explain and interpret | • May not work well with small datasets |
| **Random Forests** | Same as CART | • Can improve accuracy over CART | • Many parameters to adjust<br>• Not as easy to explain as CART |

# Comparison of Methods

| | Used For | Pros | Cons |
|---|---|---|---|
| Hierarchical Clustering | • Finding similar groups<br>• Clustering into smaller groups and applying predictive methods on groups | • No need to select number of clusters a priori<br>• Visualize with a dendrogram | • Hard to use with large datasets |
| *k*-means Clustering | Same as Hierarchical Clustering | • Works with any dataset size | • Need to select number of clusters before algorithm |