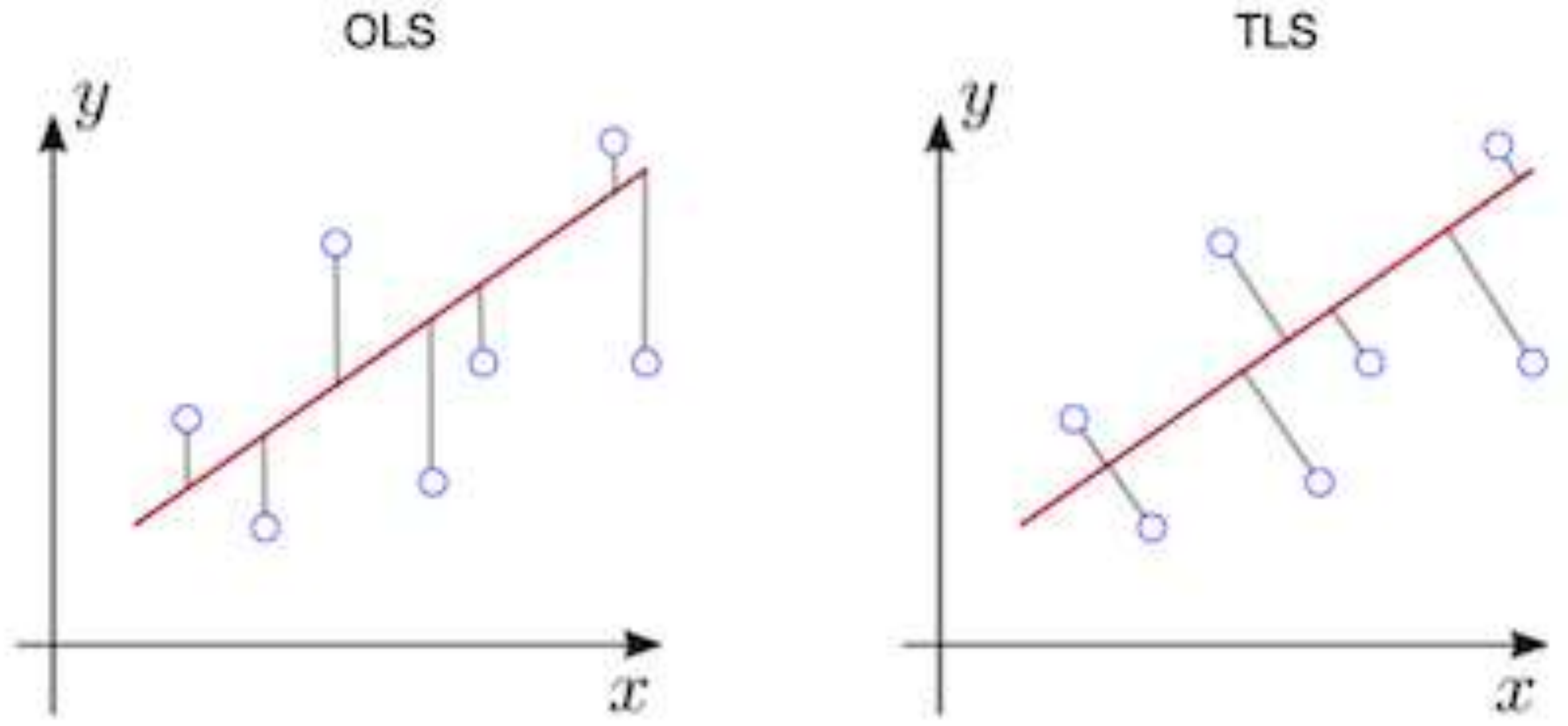
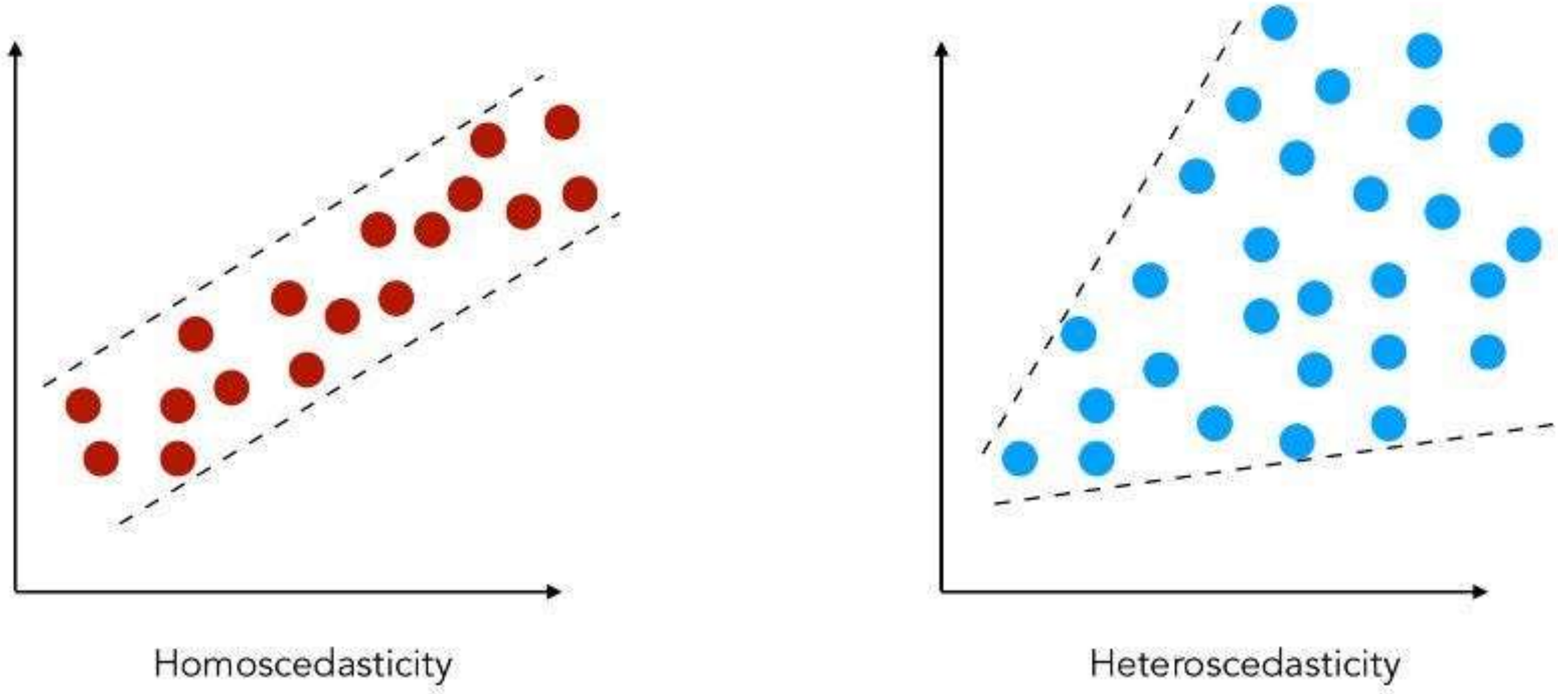
# OLS and TLS



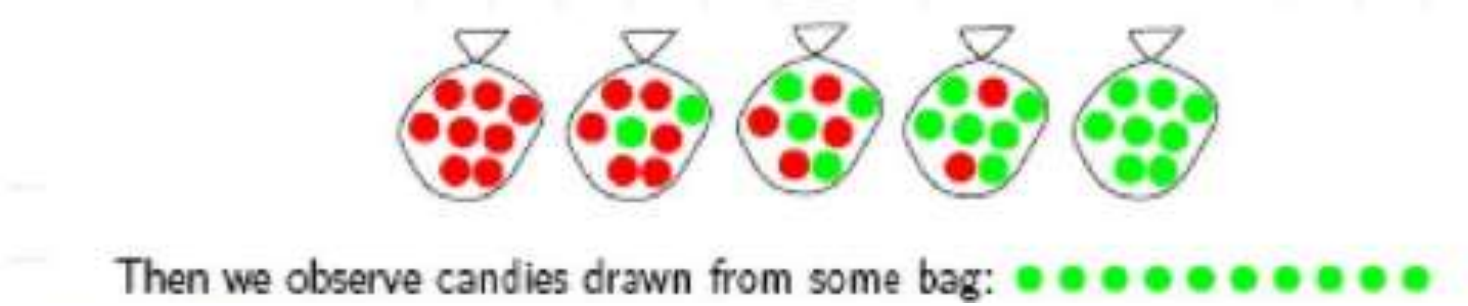
# Homo/Heteroscedasticity




PROBABILITY = EVENT / OUTCOMES
wikiHow to Calculate Probability

# Data – a probability-based perspective

- The basis for Statistical Learning Theory




  - Domain described by random variables (r.v.)
    - X = {apple, grape}
    - $b_i \in [1,5]$

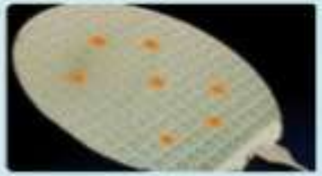
  - **Data = Instantiation of some or all r.v.'s in the domain**

# Uncertainty arises from many sources

## Process Uncertainty

Processes contain "randomness"


Uncertain travel times



Semiconductor yield

## Data Uncertainty

Data input is uncertain

Intended Spelling | Text Entry | Actual Spelling

GPS Uncertainty

Testimony


{Paris Airport}

Ambiguity

Contaminated?

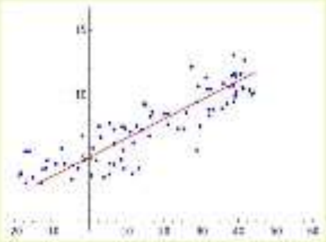
Rumors

{John Smith, Dallas}
{John Smith, Kansas}

Conflicting Data

## Model Uncertainty

All modeling is approximate



Fitting a curve to data

Forecasting a hurricane (www.noaa.gov)

© 2013 3IBM Corporation

# Data: a probabilistic perspective

| | DBAName | AKAName | Address | City | State | Zip |
|---|---|---|---|---|---|---|
| t1 | John Veliotis Sr. | Johnnyo's | 3465 S Morgan ST | ***Chicago*** | IL | *60608* |
| t2 | John Veliotis Sr. | Johnnyo's | 3465 S Morgan ST | Chicago | IL | ***60609*** |
| t3 | John Veliotis Sr. | Johnnyo's | 3465 S Morgan ST | Chicago | IL | ***60609*** |
| t4 | *Johnnyo's* | Johnnyo's | 3465 S Morgan ST | *Cicago* | IL | 60608 |

Conflicts

Does not obey data distribution

Conflict

## Output

### Proposed Cleaned Dataset

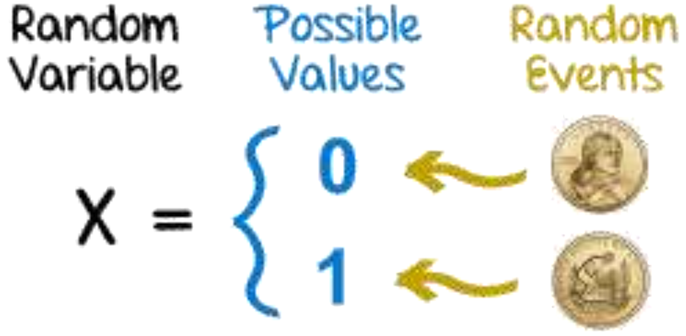
| | DBAName | Address | City | State | Zip |
|---|---|---|---|---|---|
| t1 | John Veliotis Sr. | 3465 S Morgan ST | Chicago | IL | 60608 |
| t2 | John Veliotis Sr. | 3465 S Morgan ST | Chicago | IL | **60608** |
| t3 | John Veliotis Sr. | 3465 S Morgan ST | Chicago | IL | **60608** |
| t4 | **John Veliotis Sr.** | 3465 S Morgan ST | **Chicago** | IL | 60608 |

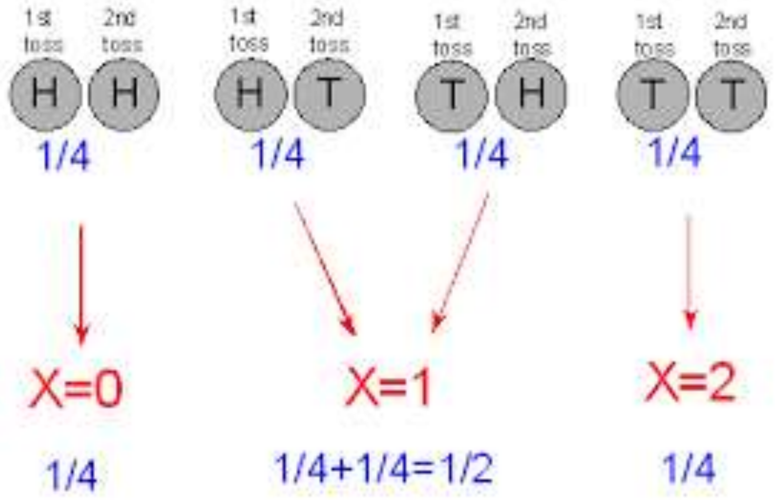
### Marginal Distribution of Cell Assignments

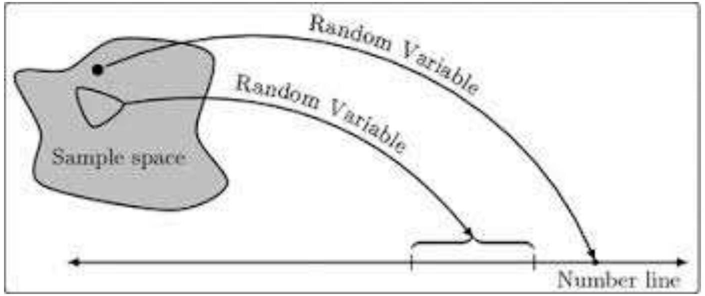
| Cell | Possible Values | Probability |
|---|---|---|
| t2.Zip | 60608 | 0.84 |
| | 60609 | 0.16 |
| t4.City | Chicago | 0.95 |
| | Cicago | 0.05 |
| t4.DBAName | John Veliotis Sr. | 0.99 |
| | Johnnyo's | 0.01 |

# Random Variables

R.V. = A numerical value from a random experiment



# Random variables

- A **discrete random variable** can assume a countable number of values.
  - Number of steps to the top of the Eiffel Tower*

# Random variables

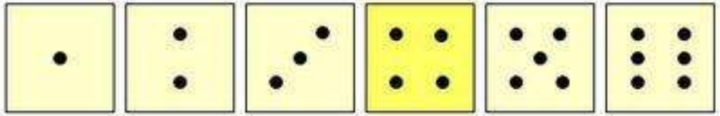
- A **discrete random variable** can assume a countable number of values.
  - Number of steps to the top of the Eiffel Tower*
- A **continuous random variable** can assume any value along a given interval of a number line.
  - The time a tourist stays at the top once s/he gets there

*Believe it or not, the answer ranges from 1,652 to 1,789. See Great Buildings

# Discrete Random Variables

- Can only take on a countable number of values

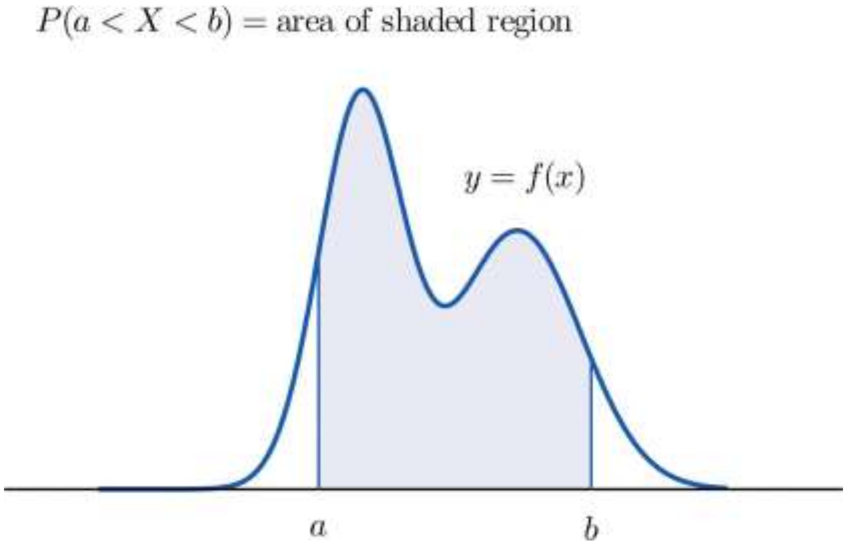
Examples:

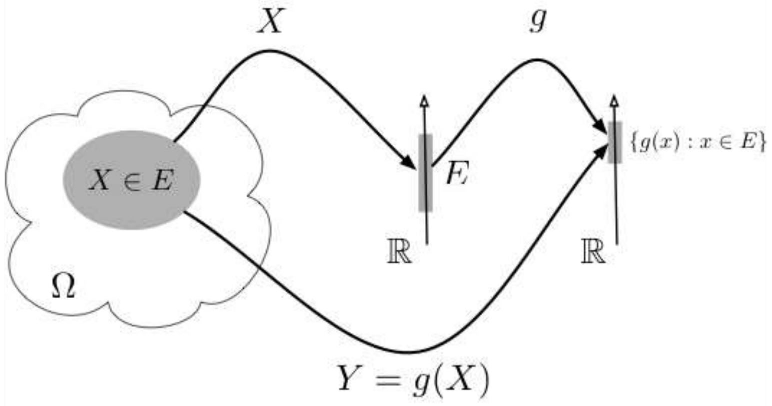


- **Roll a die twice**
  **Let X be the number of times 4 comes up**
  **(then X could be 0, 1, or 2 times)**

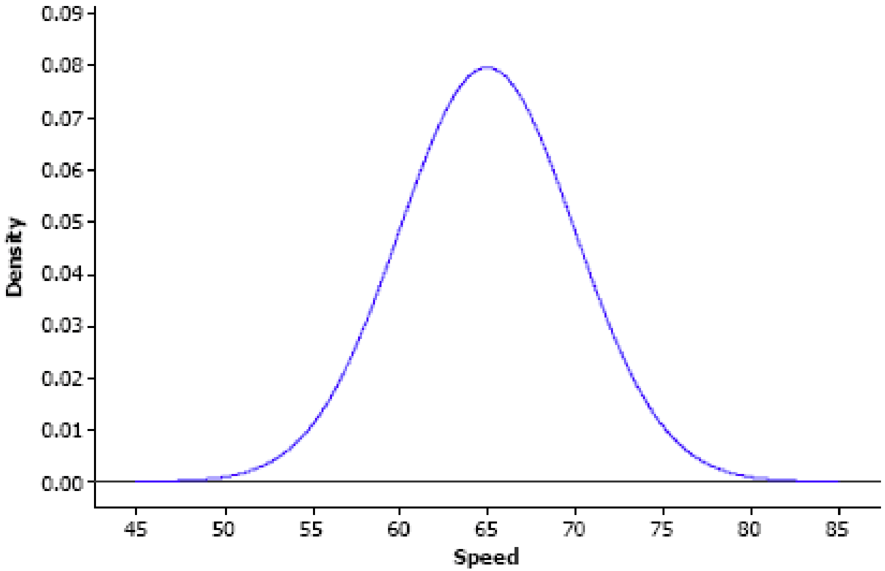
- **Toss a coin 5 times.**
  **Let X be the number of heads**
  **(then X = 0, 1, 2, 3, 4, or 5)**

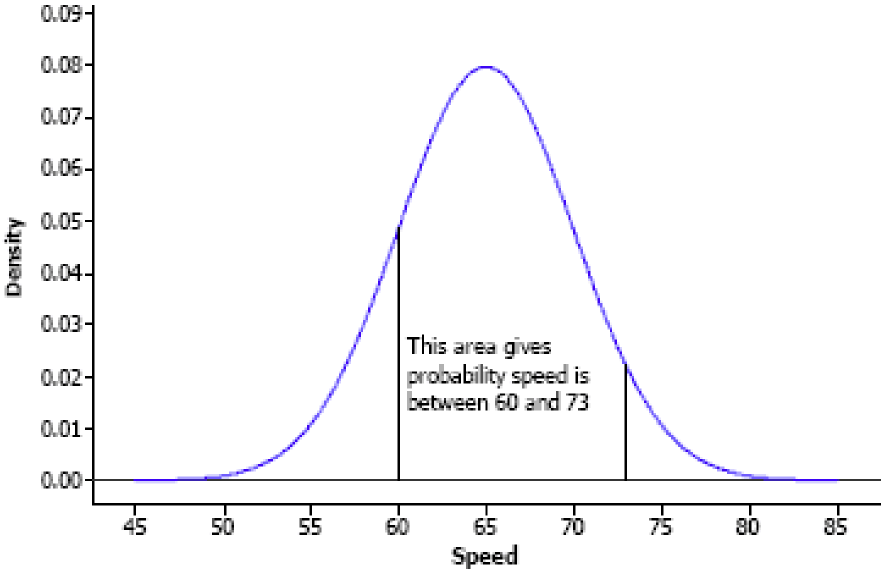
# Continuous random variable



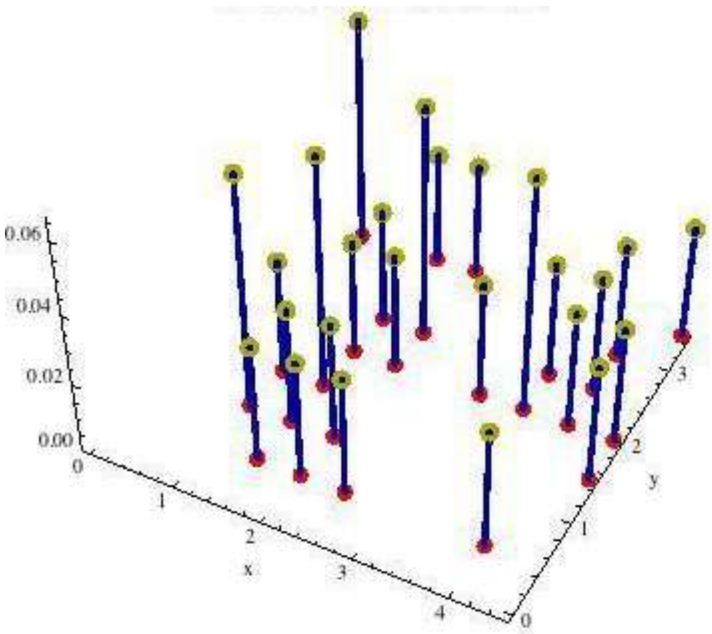



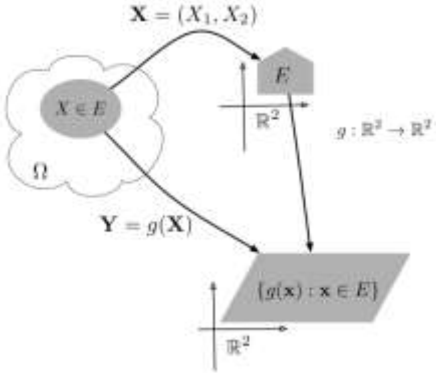
# Continuous random variable

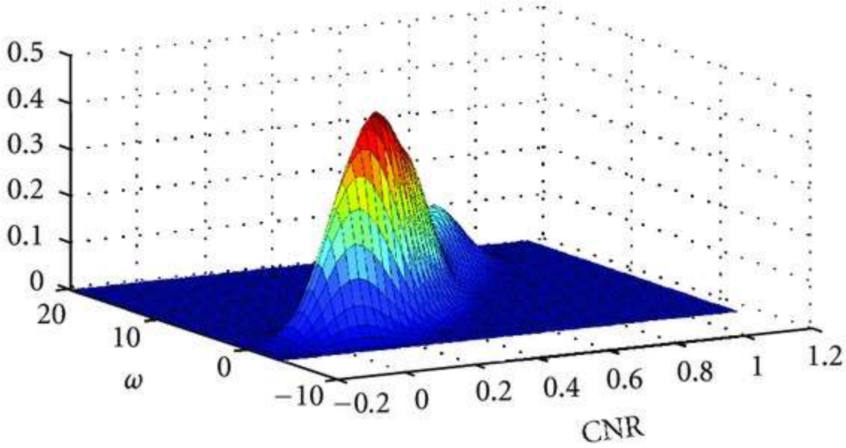


# Random vectors



# Data → r.v.

## Relative frequency

Relative frequency is the same as experimental probability. We use relative frequency to predict probabilities from experimental data.

**The experiment**

This spinner was spun 40 times and the results recorded in this table:

| Colour | Frequency |
|---|---|
| Blue | 20 |
| Yellow | 10 |
| Red | 5 |
| Green | 5 |

**Relative frequency**

$$\frac{\text{frequency of event}}{\text{total number of trials}}$$

Event means one possible outcome; here, one colour on the spinner.

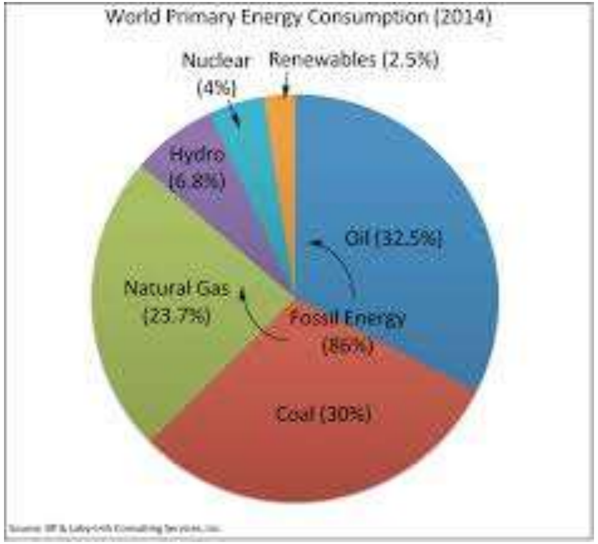
There were 20 blues recorded...

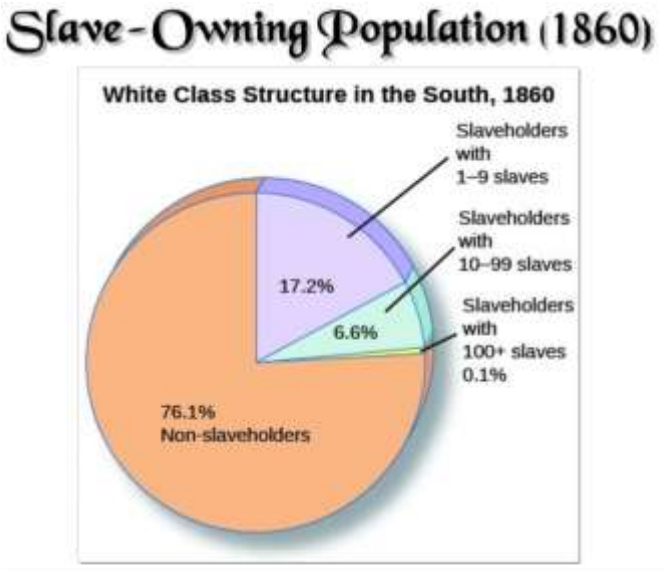
...out of 40 spins.

$$P(\text{blue}) = \frac{20}{40}$$

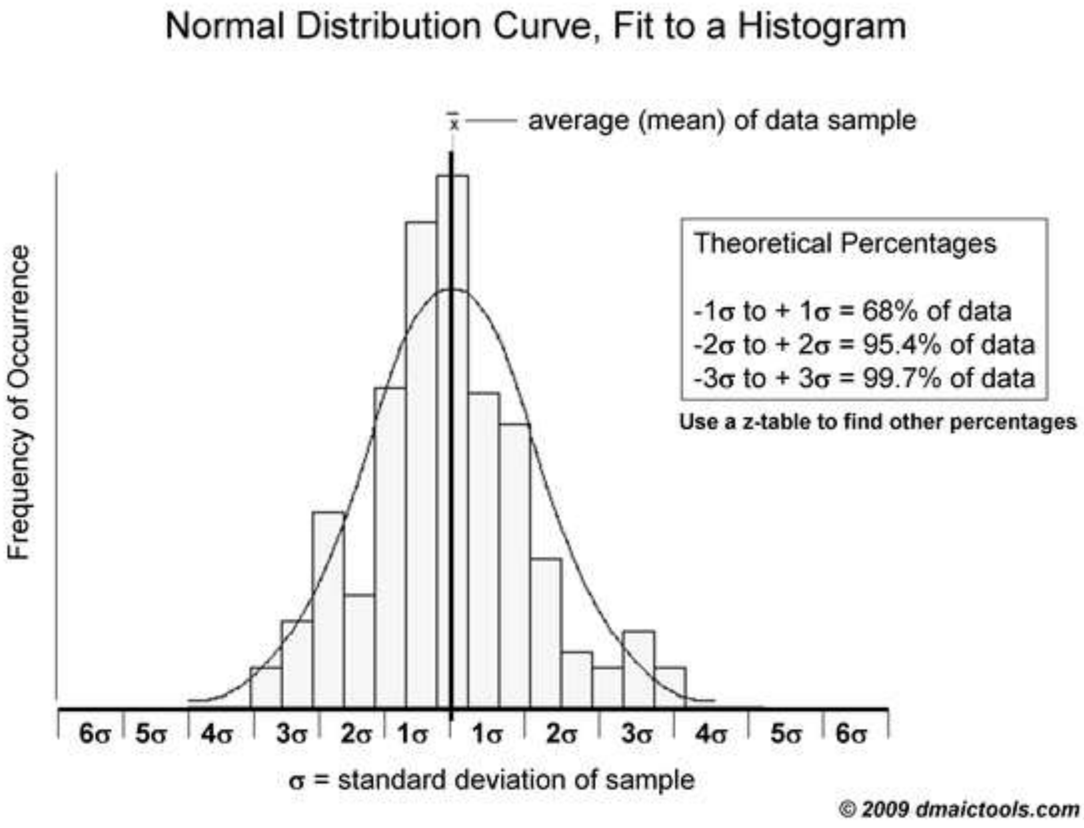
Simplify: $P(\text{blue}) = \frac{20}{40} = \frac{2}{4} = \frac{1}{2}$

# Discrete Prior distributions



# Data → r.v.

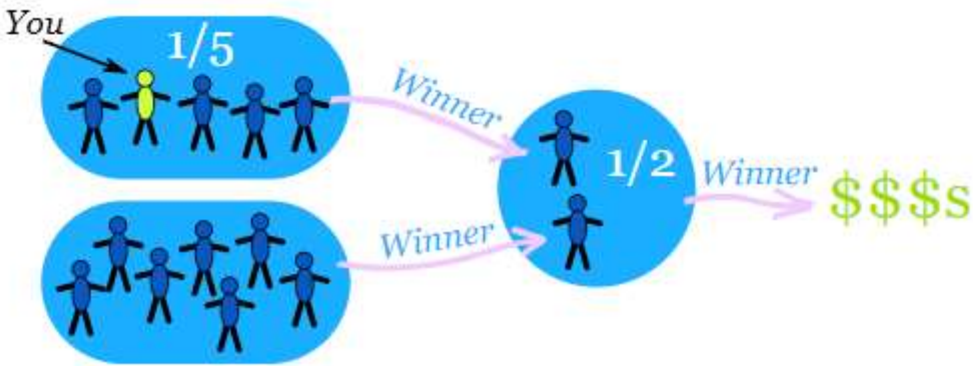


Normal Distribution Curve, Fit to a Histogram
x̄ average (mean) of data sample
Frequency of Occurrence
Theoretical Percentages
-1σ to + 1σ = 68% of data
-2σ to + 2σ = 95.4% of data
-3σ to + 3σ = 99.7% of data
Use a z-table to find other percentages
6σ 5σ 4σ 3σ 2σ 1σ 1σ 2σ 3σ 4σ 5σ 6σ
σ = standard deviation of sample
© 2009 dmaictools.com

# Independent Events

Imagine there are two groups:

- A member of each group gets randomly chosen for the winners circle,
- **then** one of those gets randomly chosen to get the big money prize:




What is your chance of winnning the big prize?

# Independent vs. Dependent Events

Using the bag of marbles on the left, what is the probability of pulling a black marble two times in a row? $P(black, black)$

When you put 1st marble back in (*Independent Events*)

$$\frac{2}{10} * \frac{2}{10}$$

$$\frac{1}{5} * \frac{1}{5} \boxed{= \frac{1}{25}}$$

When you KEEP 1st marble (*Dependent Events*)

$$\frac{2}{10} * \frac{1}{9}$$

$$\frac{1}{5} * \frac{1}{9}$$

## Independent Events

The outcome of one event does not affect the outcome of the other.

If A and B are independent events then the probability of both occurring is

$$P(A \text{ and } B) = P(A) \times P(B)$$

## Dependent Events

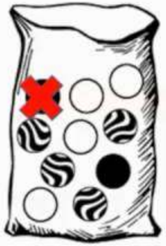
The outcome of one event affects the outcome of the other.

If A and B are dependent events then the probability of both occurring is

$$P(A \text{ and } B) = P(A) \times P(B|A)$$

Probability of B given A

# *Independent vs. Dependent Events*



Using the bag of marbles on the left, what is the probability of pulling a black marble two times in a row? $P(black, black)$

When you put 1st marble back in (*Independent Events*)

$$\frac{2}{10} * \frac{2}{10}$$

$$\frac{1}{5} * \frac{1}{5} = \frac{1}{25}$$

When you KEEP 1st marble (*Dependent Events*)

$$\frac{2}{10} * \frac{1}{9}$$

$$\frac{1}{5} * \frac{1}{9}$$

$$P(A \text{ and } B) = P(A) \times P(B)$$

$$P(A \text{ and } B) = P(A) \times P(B|A)$$

Probability of B given A

# Marginal Probabilities

$$\begin{cases} x = 1 & \text{(Rains)} \\ x = 0 & \text{(Doesn't rain)} \end{cases}$$

$Pr(x = 1) = 0.6$

$Pr(x = 0) = 0.4$

$$\begin{cases} y = 1 & \text{(Have umbrella)} \\ y = 0 & \text{(Don't have umbrella)} \end{cases}$$

$Pr(y = 1) = 0.3$

$Pr(y = 0) = 0.7$

# Joint Probability

$$\begin{cases} x = 1 & \text{(Rains)} \\ x = 0 & \text{(Doesn't rain)} \end{cases}$$

$$\begin{cases} y = 1 & \text{(Have umbrella)} \\ y = 0 & \text{(Don't have umbrella)} \end{cases}$$

$Pr(x = 1) = 0.6$

$Pr(x = 0) = 0.4$

$Pr(y = 1) = 0.3$

$Pr(y = 0) = 0.7$

$$\begin{aligned} Pr(x=0) &= \sum_{y=0}^{1} Pr(x=0, y) \\ &= Pr(x=0, y=0) + Pr(x=0, y=1) \\ &= 0.28 + 0.12 = 0.4 \end{aligned}$$

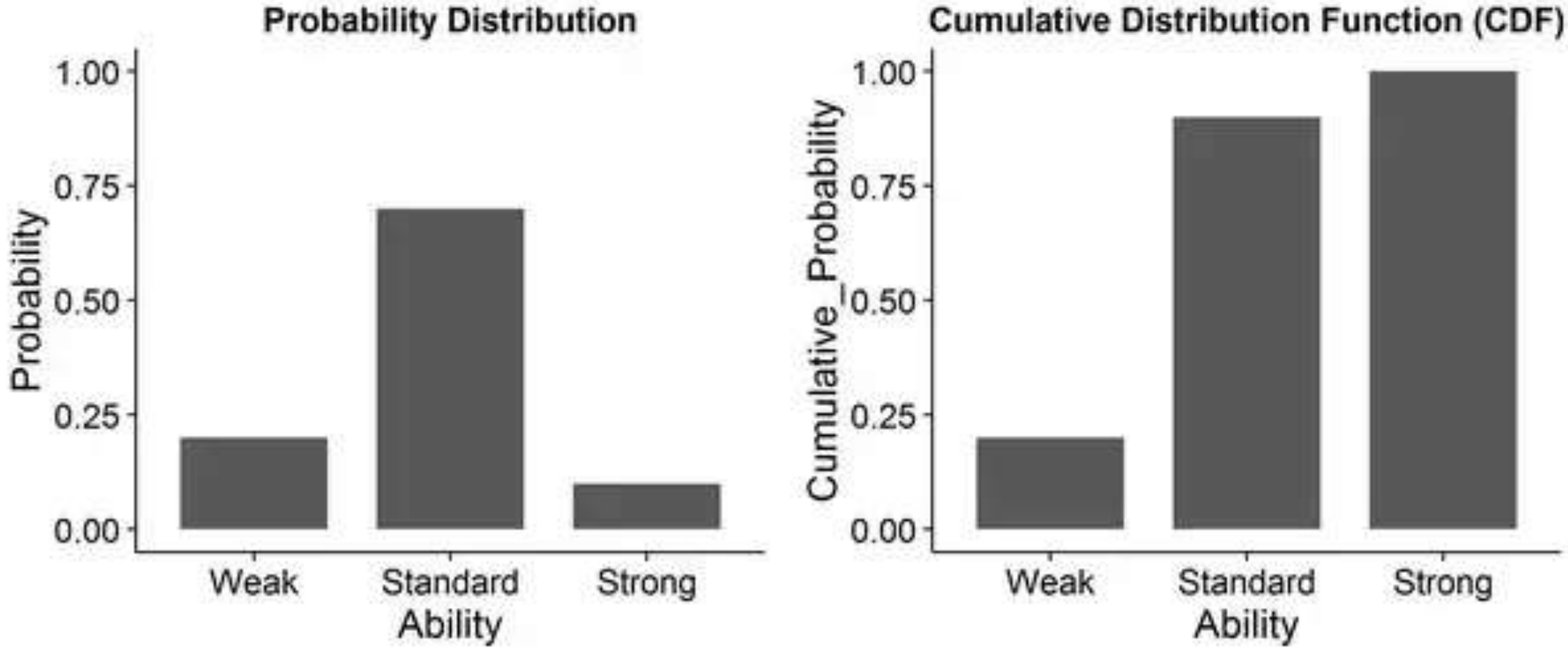
<u>Case 1 : Rains but you have an umbrella</u>

$$\begin{aligned} Pr(x=1, y=1) &= Pr(x=1) \times Pr(y=1) \\ &= 0.6 \times 0.3 \\ &= 0.18 \end{aligned}$$

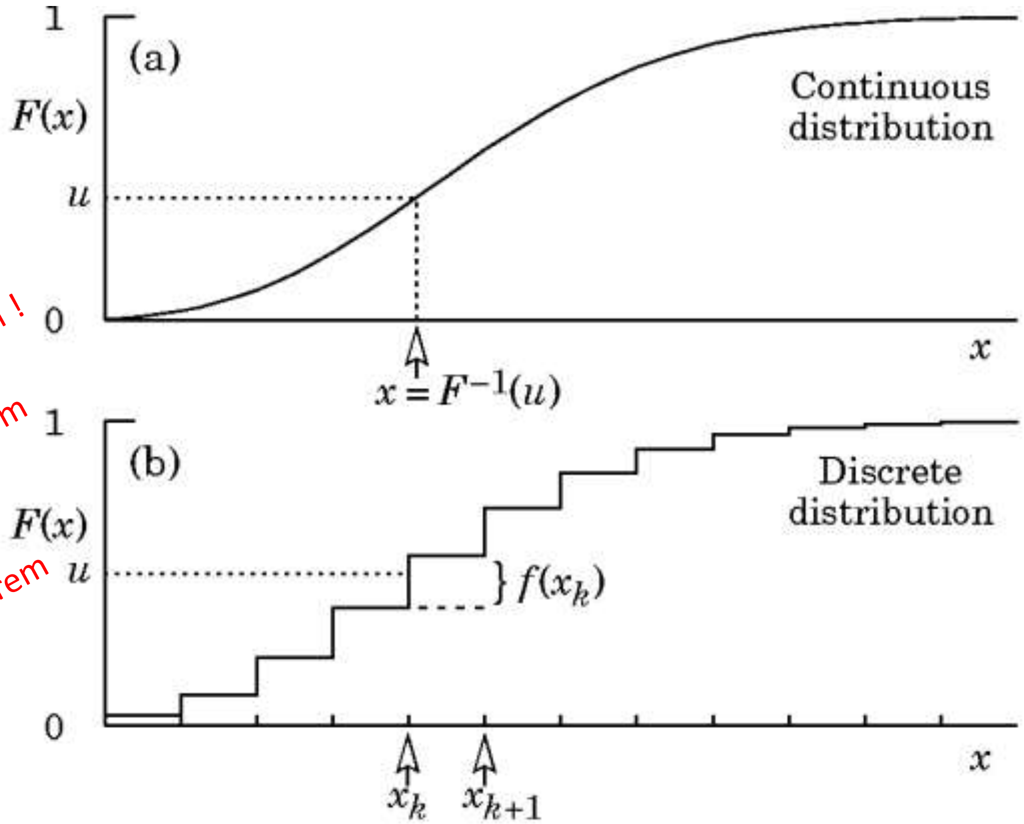
<u>Case 2 : Rains but you DON'T have an umbrella</u>

$$\begin{aligned} Pr(x=1, y=0) &= Pr(x=1) \times Pr(y=0) \\ &= 0.6 \times 0.7 \\ &= 0.42 \end{aligned}$$

# Inverse Transform Sampling



# Inverse Transform Sampling




Does not work with Gaussian !
CDF not in closed form.
Need Box-Muller Transform

Sample multiple times and
Invoke Central Limit Theorem

23.08.2024

# Statistical Methods in AI (CS7.403)

## Lecture-7: Clustering (k-means)

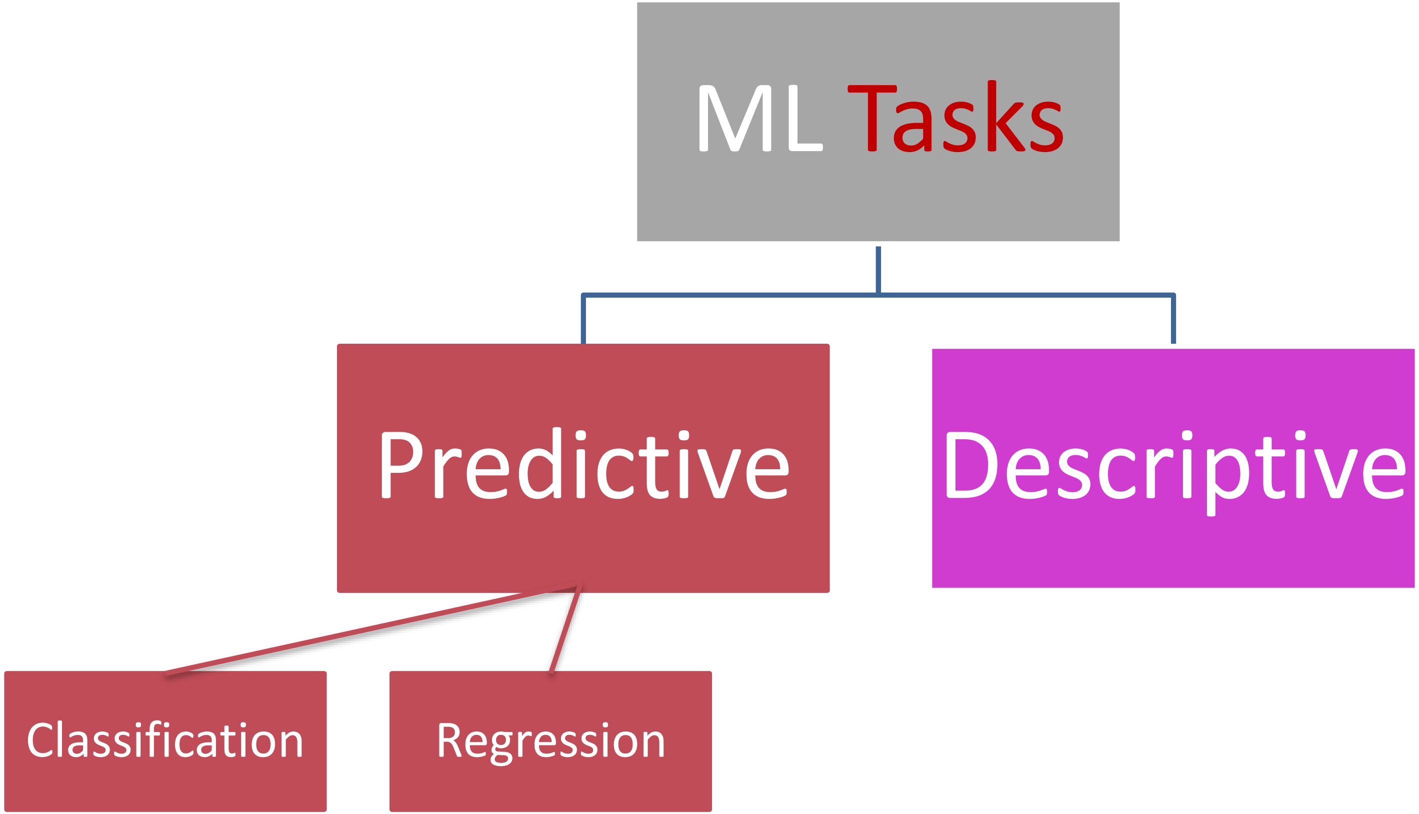

Ravi Kiran (ravi.kiran@iiit.ac.in)

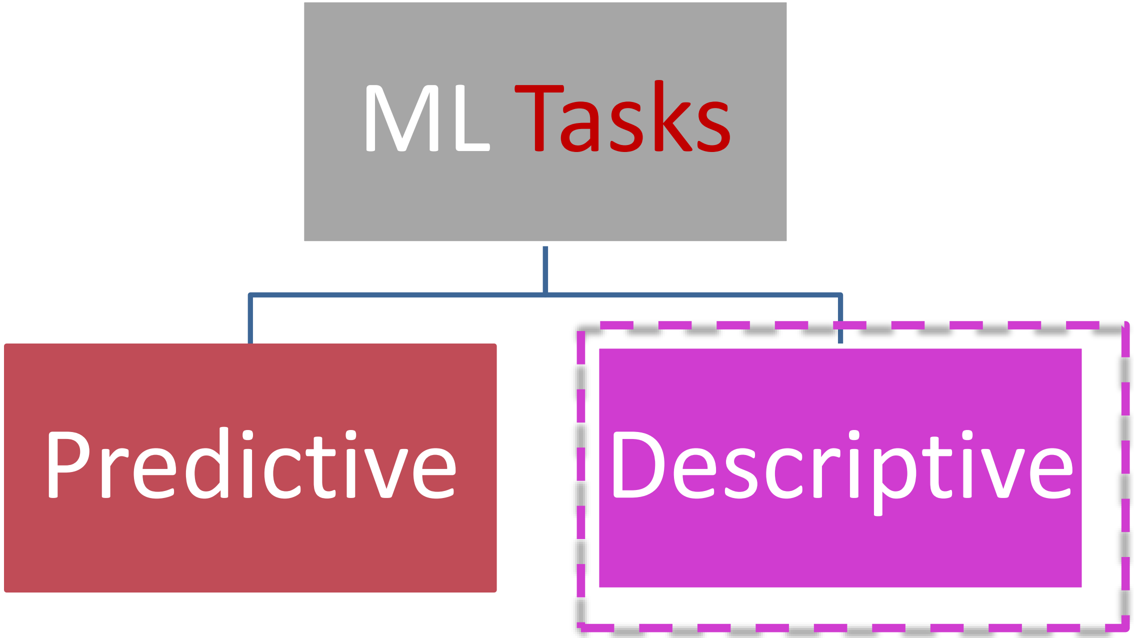
https://ravika.github.io

@vikataravi

Center for Visual Information Technology (CVIT)

IIIT Hyderabad


ML Tasks
Predictive
Descriptive
Classification
Regression

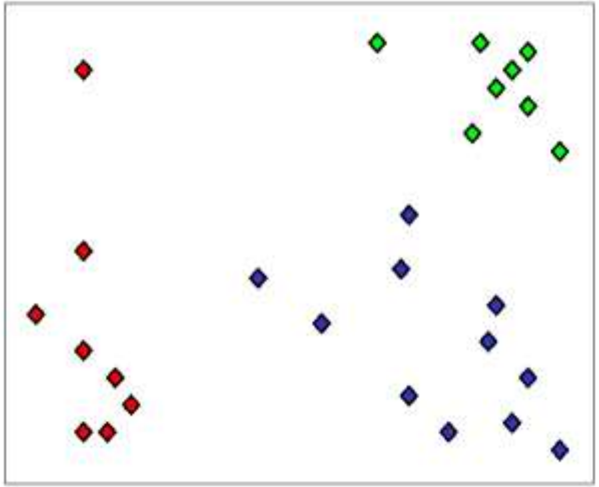

ML Tasks
Predictive
Descriptive

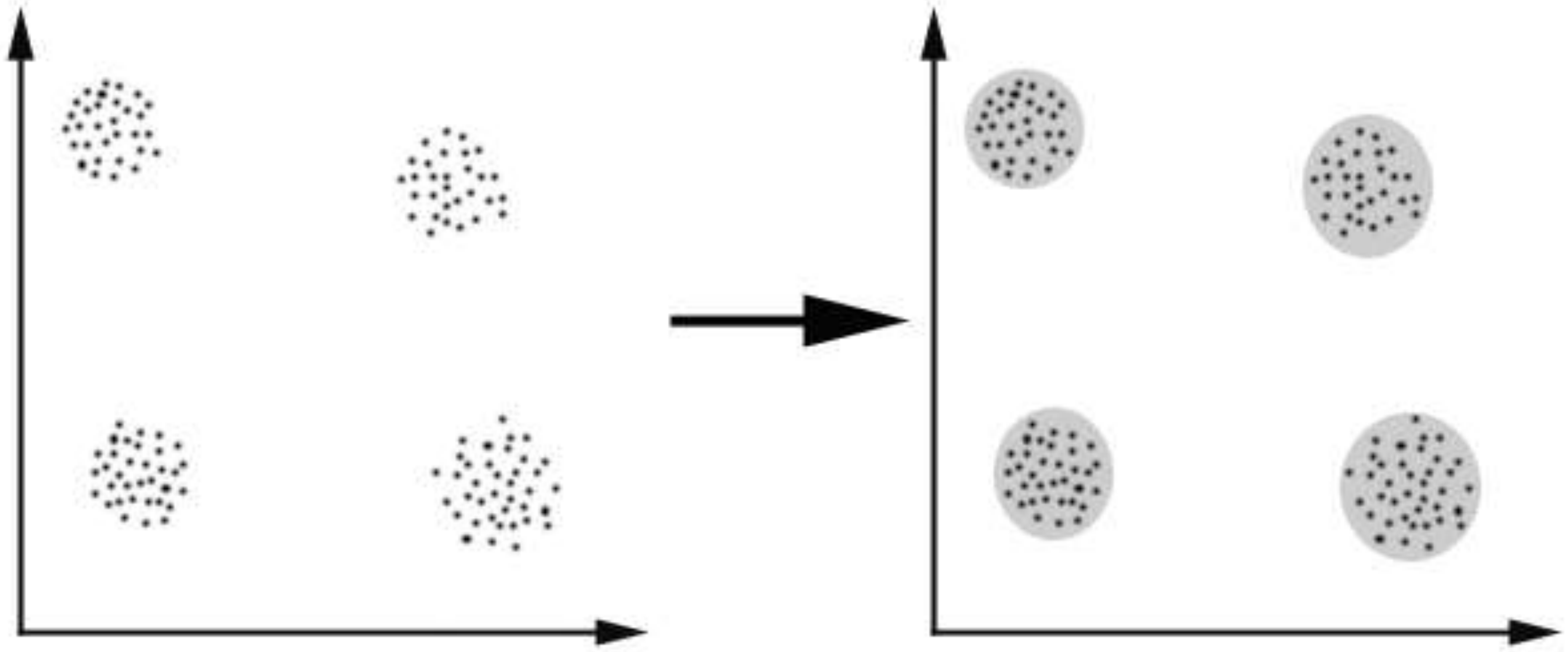
# Unsupervised Learning → Clustering

- Determine groups of people in image above
  - based on clothing styles
  - gender, age, etc

# What is Clustering?

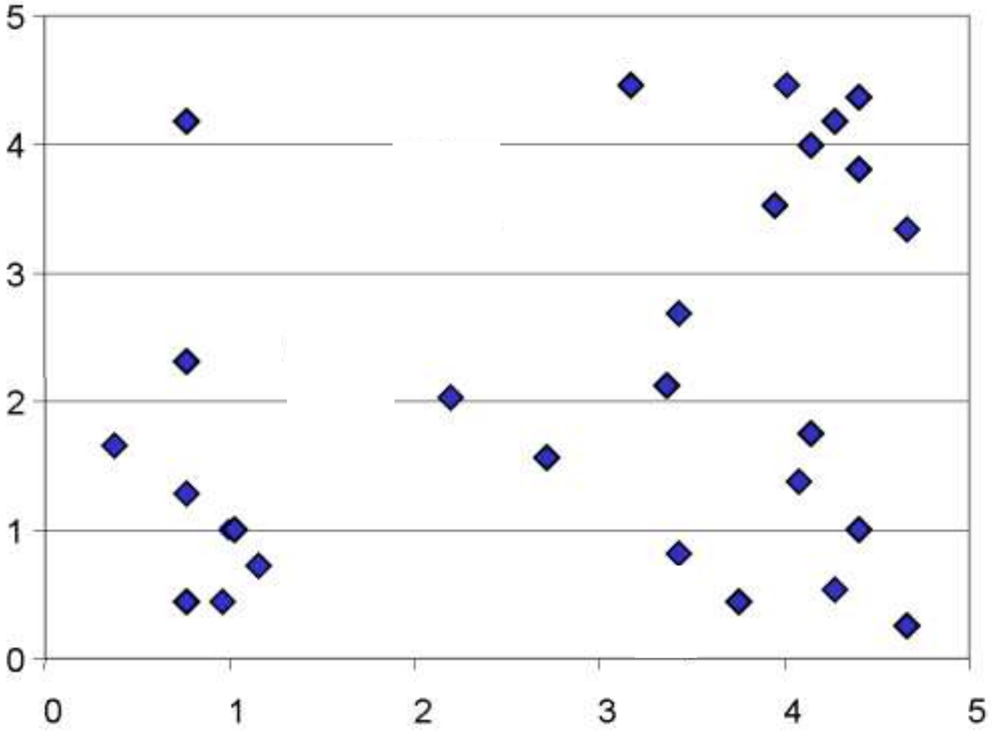
- Organizing data into *clusters* such that there is
    - high intra-cluster similarity
    - low inter-cluster similarity
- Informally, finding natural groupings among objects.





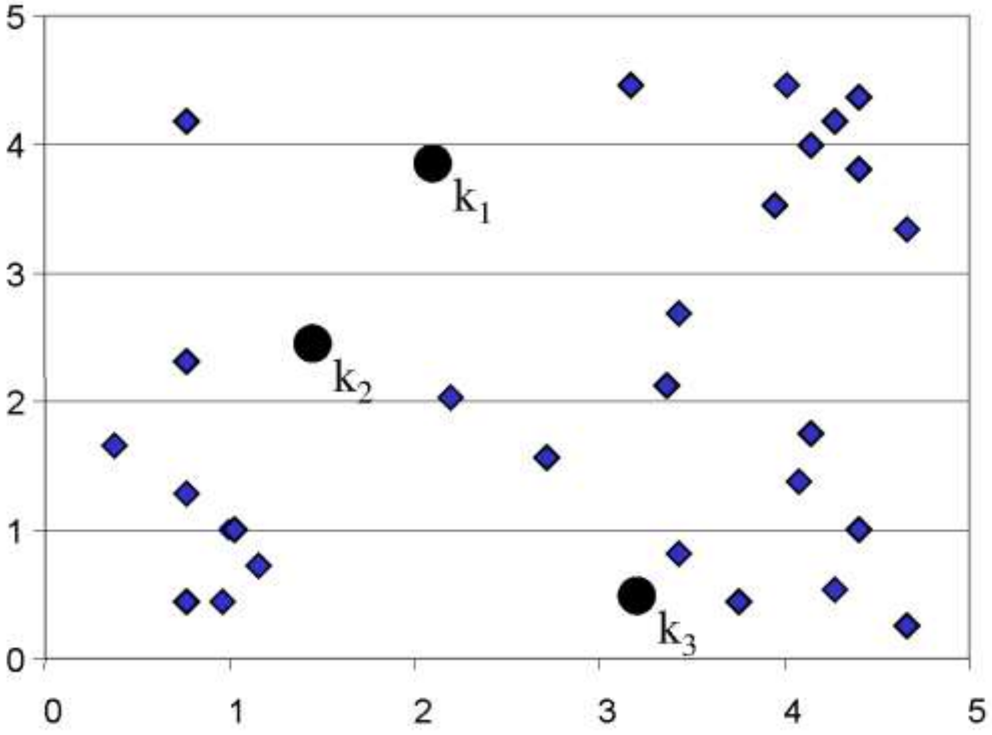
# K-means Clustering: Initialization

Decide $K$, and initialize $K$ centers (randomly)



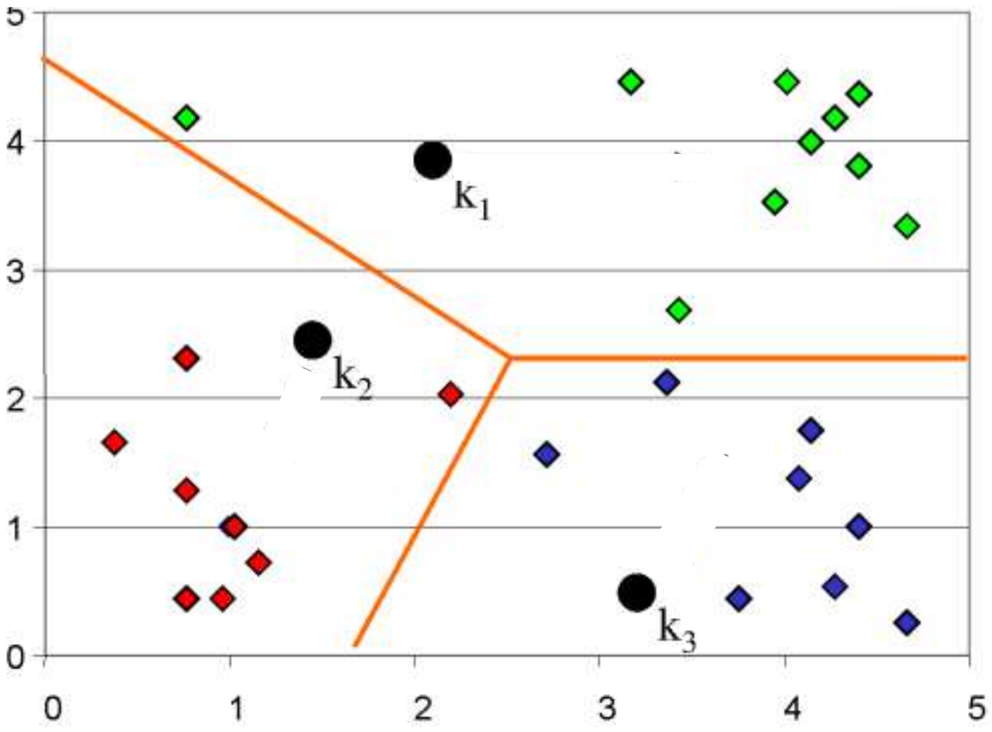
# K-means Clustering: Initialization

Decide *K*, and initialize *K* centers (randomly)



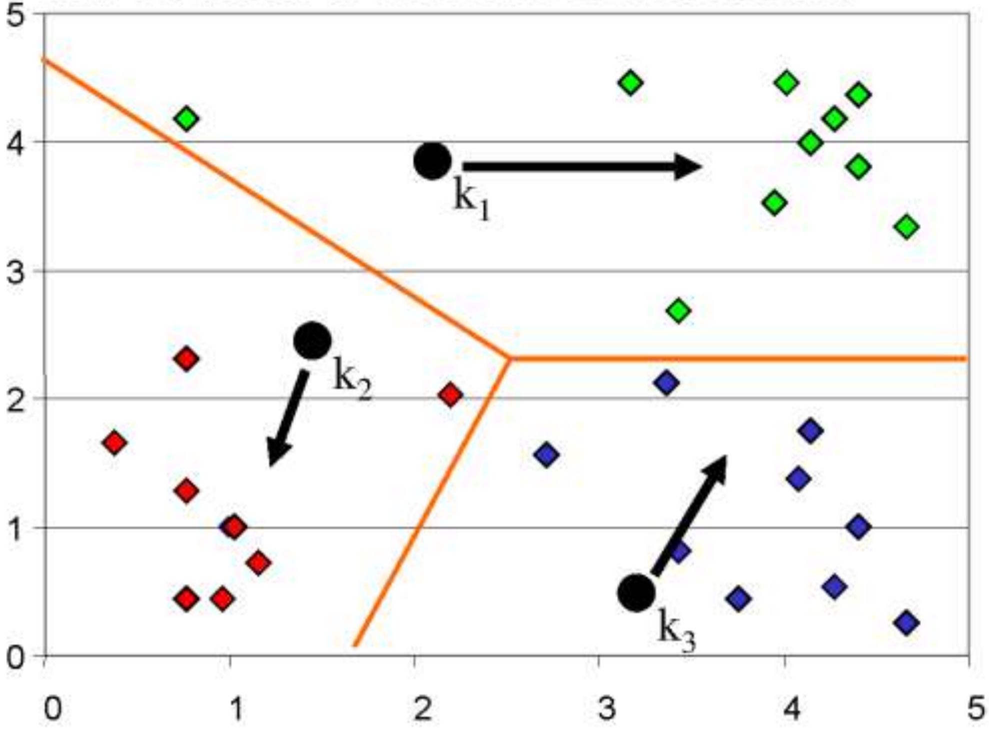
# K-means Clustering: Iteration 1

Assign all objects to the nearest center.

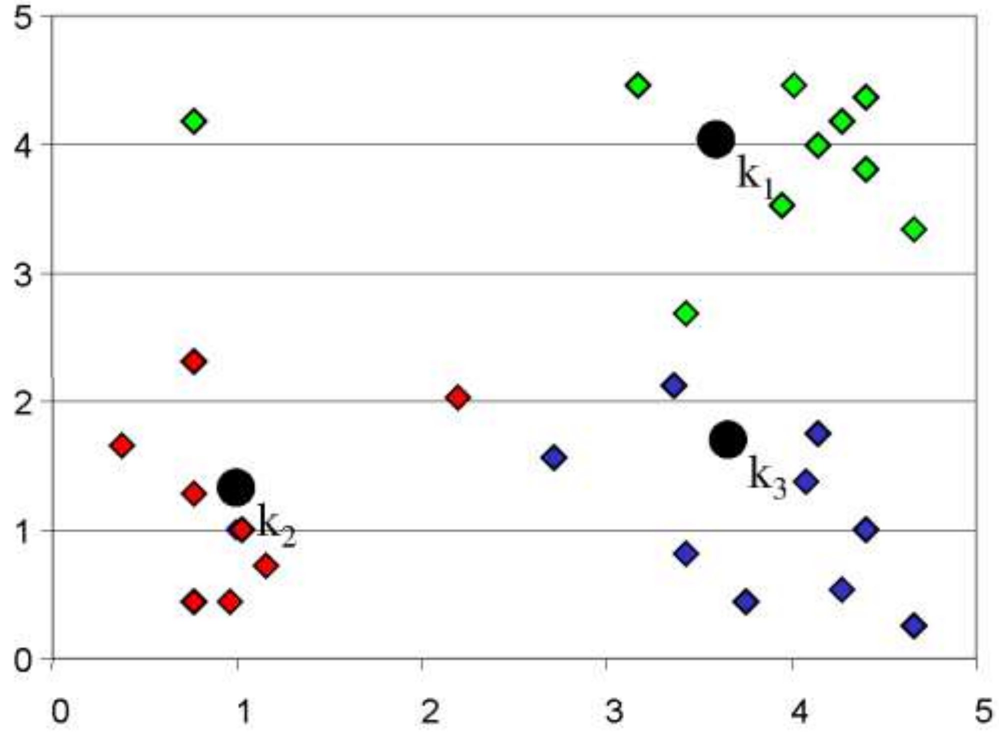


# K-means Clustering: Iteration 1

Assign all objects to the nearest center.
Move a center to the mean of its members.

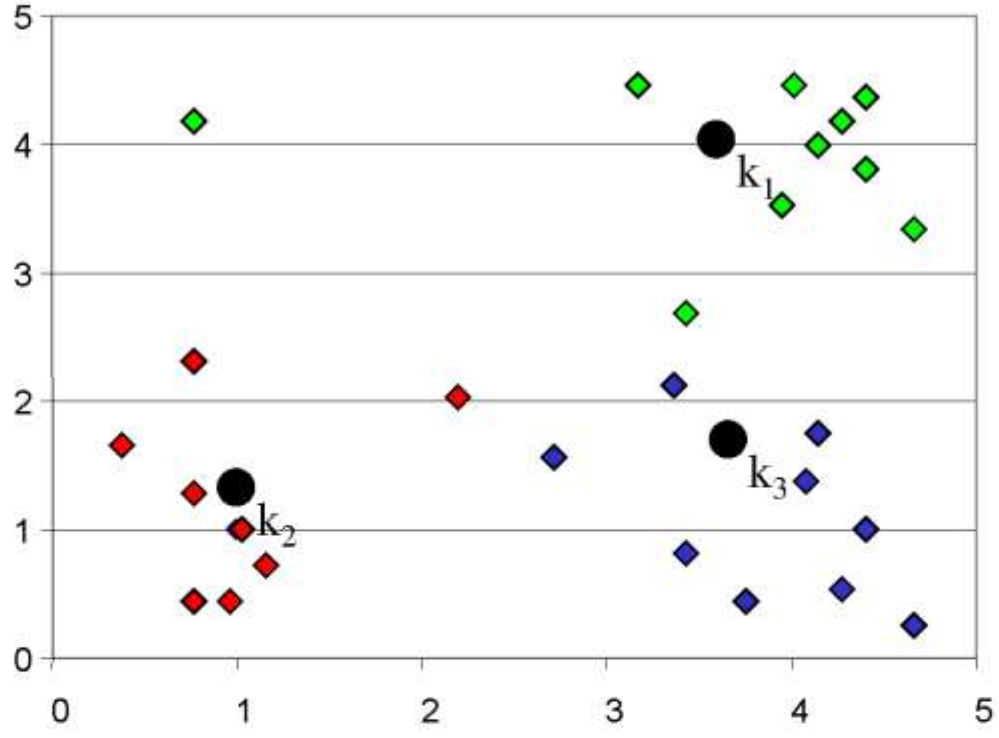


# K-means Clustering: Iteration 2



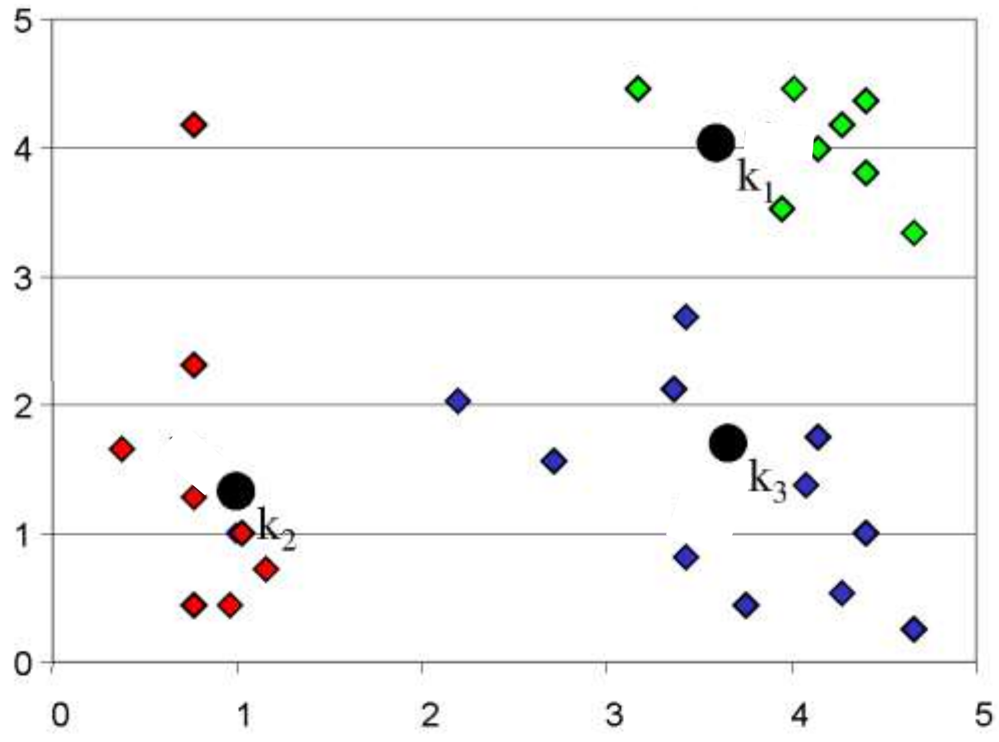
# K-means Clustering: Iteration 2

After moving centers, re-assign the objects…



# K-means Clustering: Iteration 2

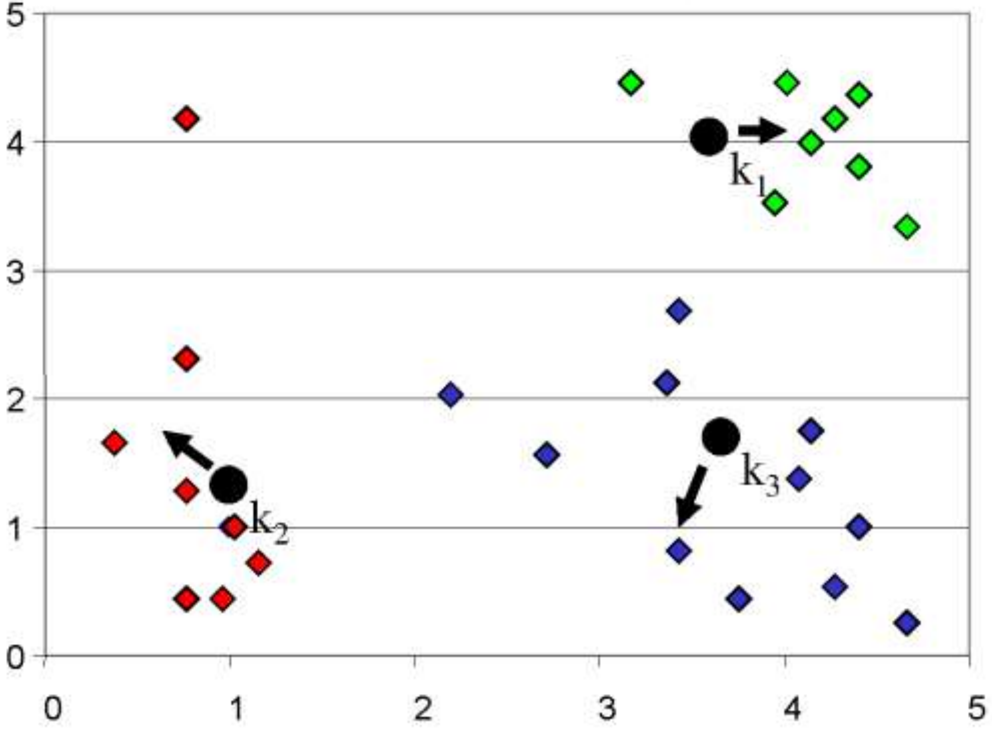
After moving centers, re-assign the objects to nearest centers.



# K-means Clustering: Iteration 2

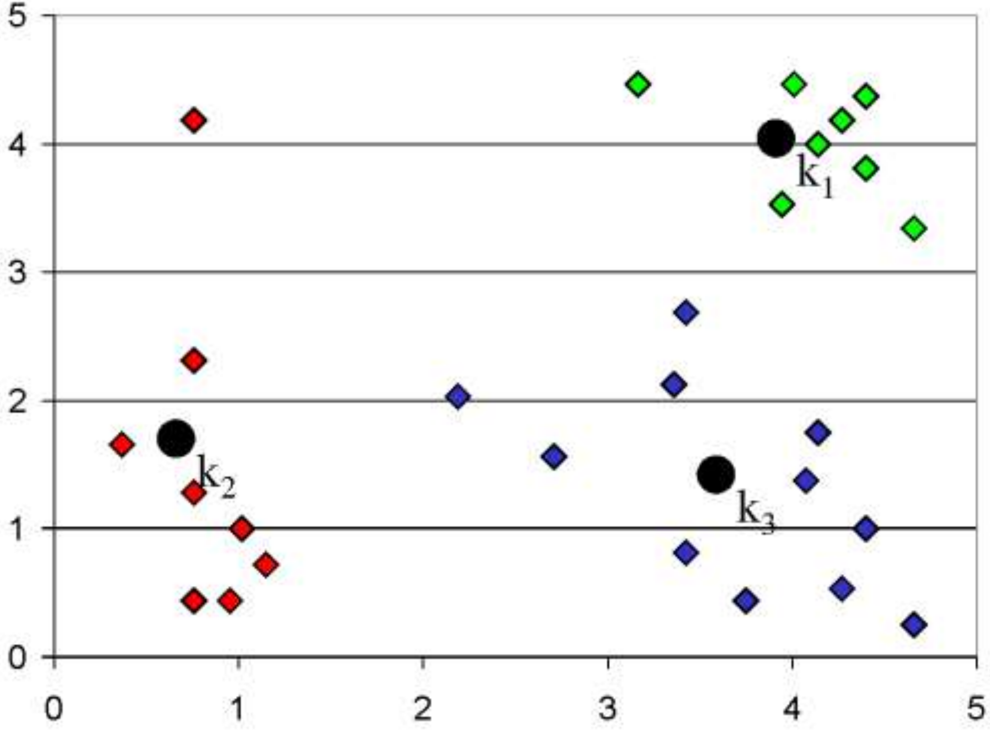
After moving centers, re-assign the objects to nearest centers.

Move a center to the mean of its new members.



# K-means Clustering: Finished!

Re-assign and move centers, until …
no objects changed membership.



$$\{x^{(1)}, \ldots, x^{(m)}\} \qquad x^{(i)} \in \mathbb{R}^n$$

The $k$-means clustering algorithm is as follows:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

   For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

Assignment step: Assign each data point to the closest cluster

   For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

Refitting step: Move each cluster center to the center of the data assigned to it

}

## EXPECTATION MAXIMIZATION

$$\{x^{(1)}, \ldots, x^{(m)}\} \qquad x^{(i)} \in \mathbb{R}^n$$

The $k$-means clustering algorithm is as follows:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

   For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

E

Assignment step: Assign each data point to the closest cluster

   For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

M

Refitting step: Move each cluster center to the center of the data assigned to it

}

$$\{x^{(1)}, \ldots, x^{(m)}\} \qquad x^{(i)} \in \mathbb{R}^n$$

The $k$-means clustering algorithm is as follows:

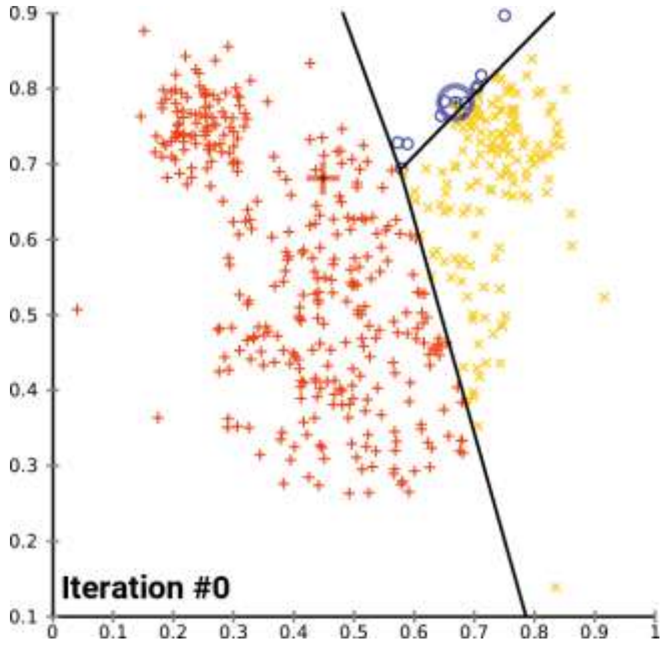
1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

   For every $i$, set
$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each $j$, set
$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }



## **Algorithm** *k-means*

1. Decide on a value for $K$, the number of clusters.

2. Initialize the $K$ cluster centers (randomly, if necessary).

3. Decide the class memberships of the $N$ objects by assigning them to the nearest cluster center.

4. Re-estimate the $K$ cluster centers, by assuming the memberships found above are correct.

5. Repeat 3 and 4 until none of the $N$ objects changed membership in the last iteration.

# Algorithm *k-means*

1. Decide on a value for $K$, th

2. Initialize the $K$ cluster cent
necessary).

3. Decide the class memberships of the $N$ objects by assigning them to the nearest cluster center.

4. Re-estimate the $K$ cluster centers, by assuming the memberships found above are correct.

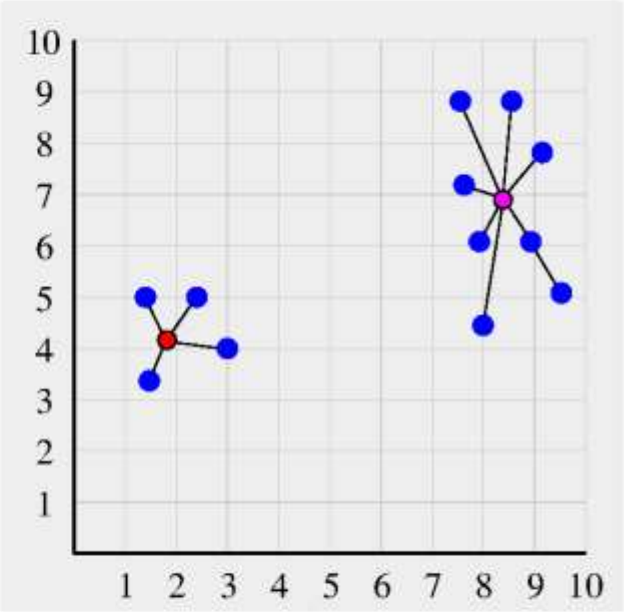
5. Repeat 3 and 4 until none of the $N$ objects changed membership in the last iteration

Use one of the distance / similarity functions we discussed earlier

Average / median of class members

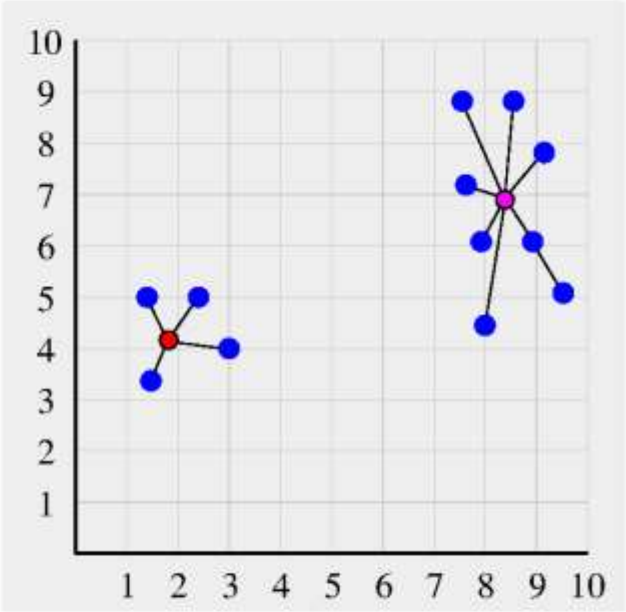
# Why K-means Works

- What is a good partition?
- High intra-cluster similarity



# Why K-means Works

- What is a good partition?
- High intra-cluster similarity
- K-means optimizes




$$se = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$

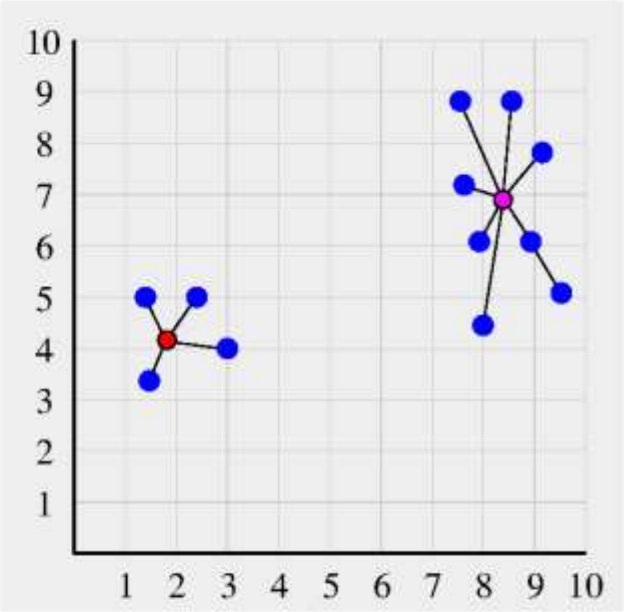
# Why K-means Works

- What is a good partition?
- High intra-cluster similarity
- K-means optimizes
  - the average distance to members of the same cluster

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \left\| x_{ki} - x_{kj} \right\|^2$$

  - which is twice the total distance to centers, also called squared error

$$se = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



Repeat until convergence: {

For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

$$\sum_{k=1}^{K} \sum_{i=1}^{n_k} \|x_{ki} - \mu_k\|^2$$

- Whenever an assignment is changed, the sum squared distances $J$ of data points from their assigned cluster centers is reduced.

- Whenever an assignment is changed, the sum squared distances $J$ of data points from their assigned cluster centers is reduced.

Repeat until convergence: {

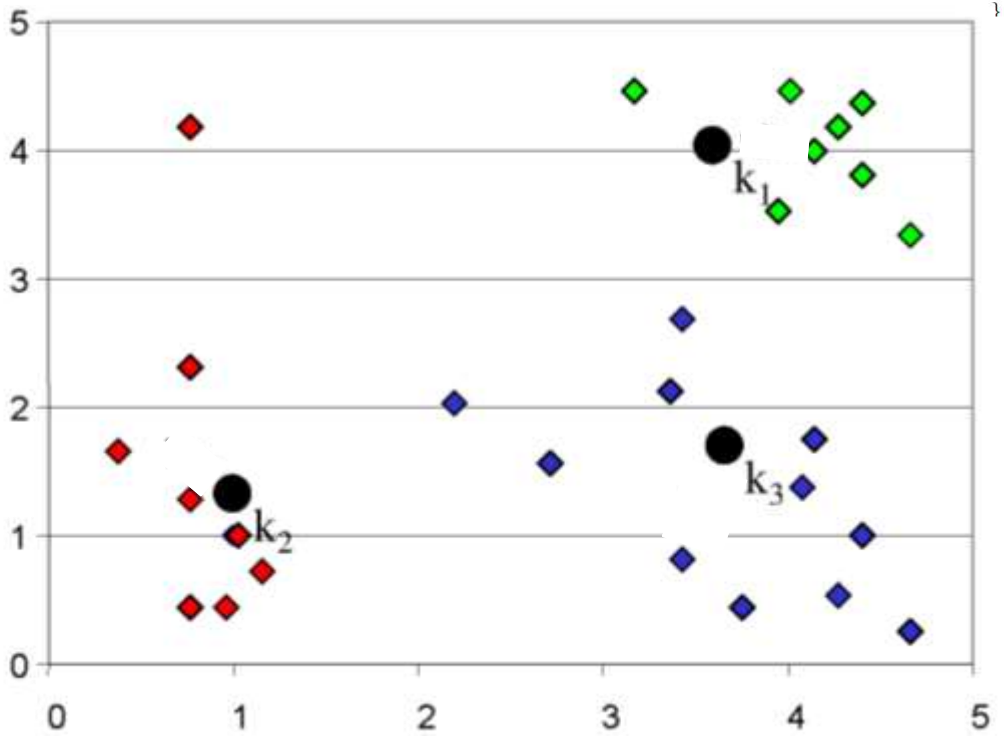
For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}



- Whenever an assignment is changed, the sum squared distances $J$ of data points from their assigned cluster centers is reduced.

Repeat until convergence: {

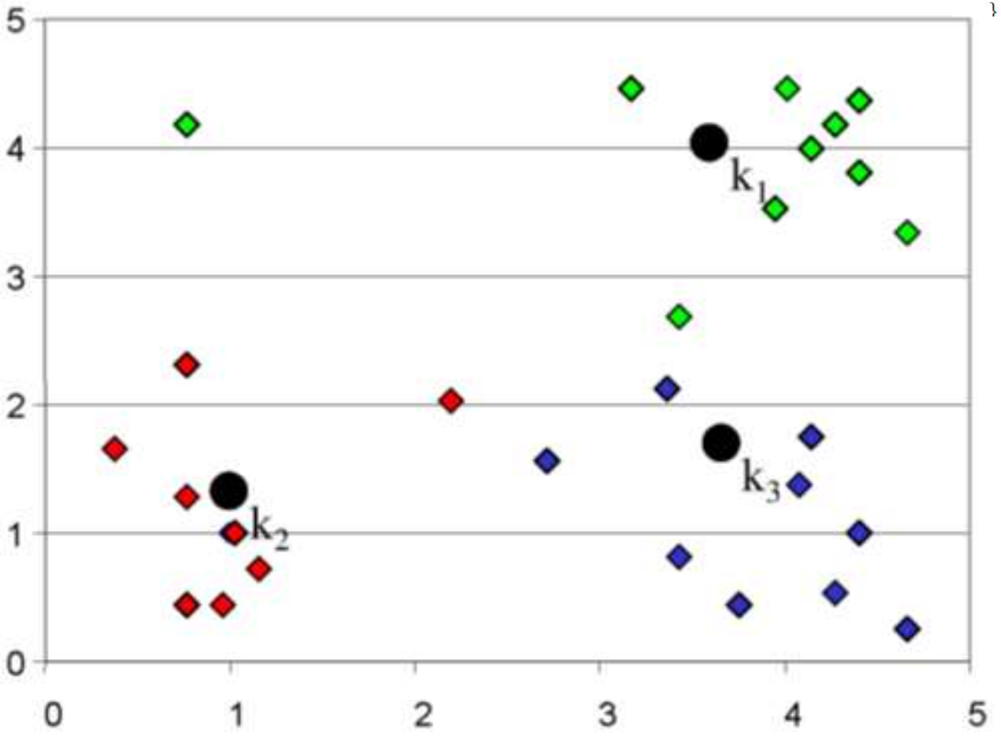
For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}



Repeat until convergence: {

For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

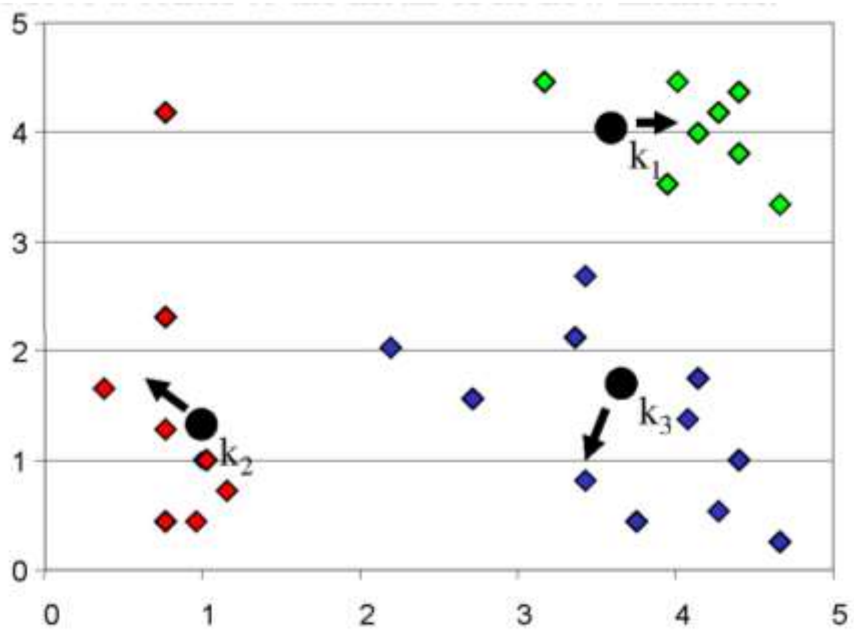
For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

$$\sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$

- Whenever an assignment is changed, the sum squared distances $J$ of data points from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, $J$ is reduced.



Repeat until convergence: {

For every $i$, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

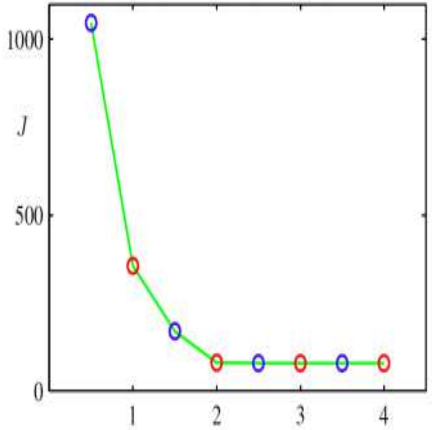
For each $j$, set

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\}x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

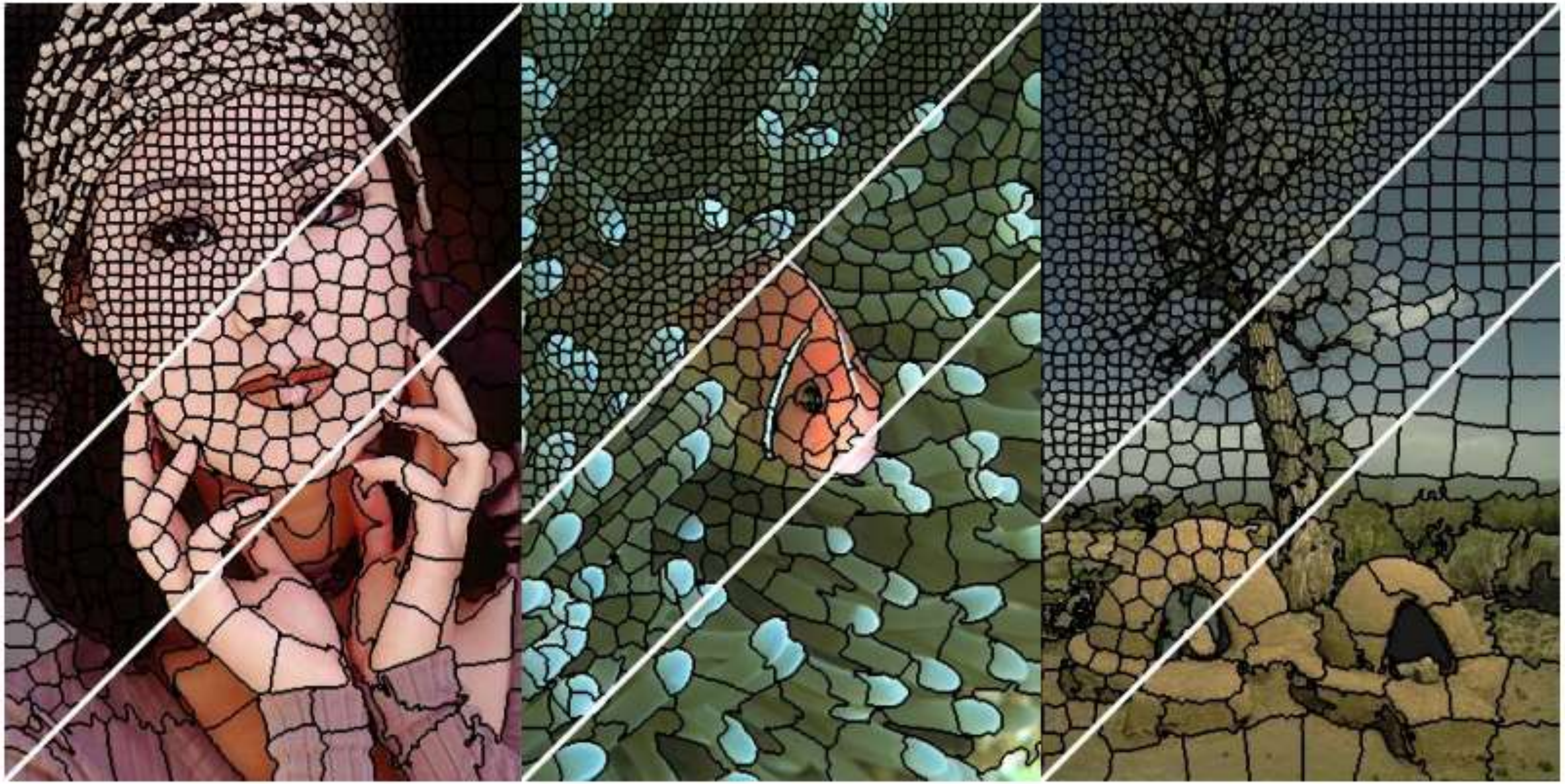
$$\sum_{k=1}^{K}\sum_{i=1}^{n_k}\left\|x_{ki} - \mu_k\right\|^2$$

- Whenever an assignment is changed, the sum squared distances $J$ of data points from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, $J$ is reduced.
- Test for convergence: If the assignments do not change in the assignment step, we have converged (to at least a local minimum).




- K-means cost function after each E step (blue) and M step (red). The algorithm has converged after the third M step


$K = 2$
$K = 3$
$K = 10$
Original image



- How would you modify k-means to get super pixels?