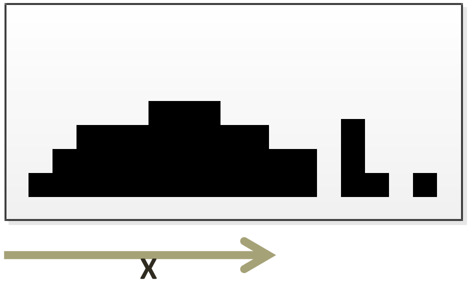
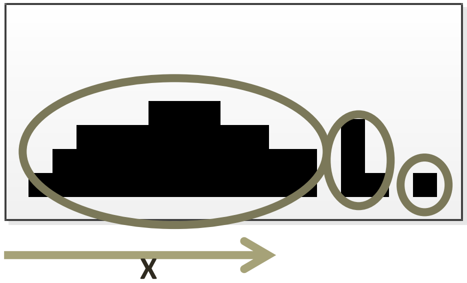
# Dimensions in Clustering

# Clustering:  Dimensions (1)



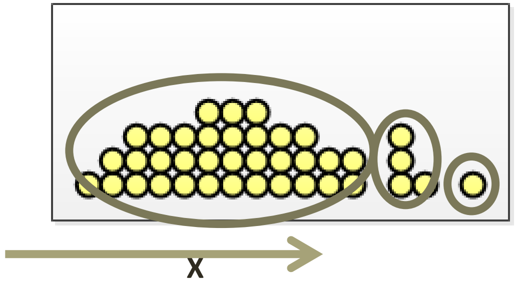
Where are the three clusters?

# Clustering:  Dimensions (2)



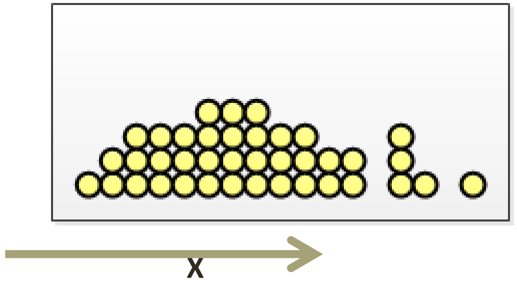
Simple assignment based on a 1D distribution

# Clustering:  Dimensions (3)



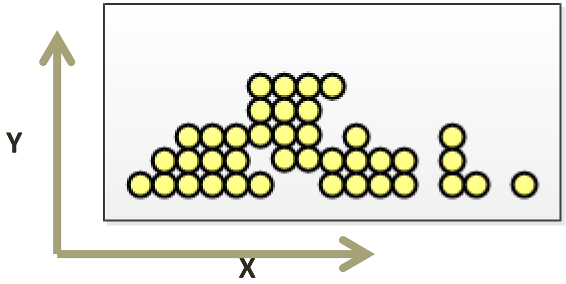
Simple assignment based on a 1D distribution

# Clustering:  Dimensions (4)



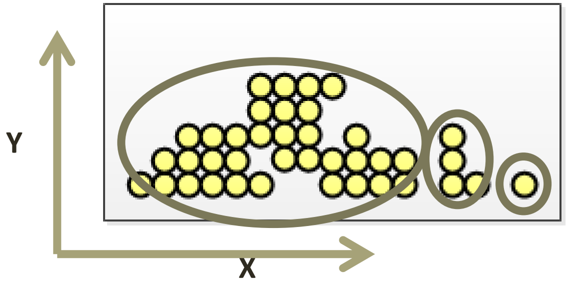
What if this was not a 1D distribution?

# Clustering:  Dimensions (5)



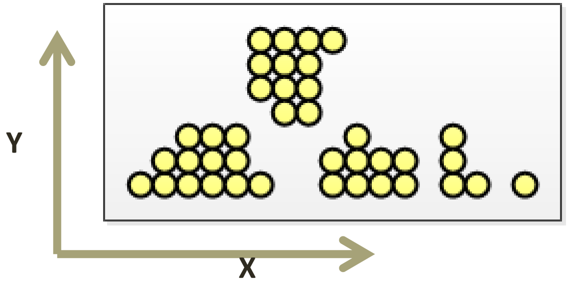
The distribution is in 2D.  Some points differ in the 2nd D

# Clustering: Dimensions (6)



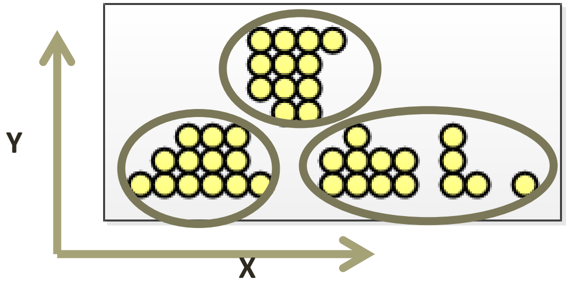
If the difference is minor, we still get the same clusters

# Clustering:  Dimensions (7)



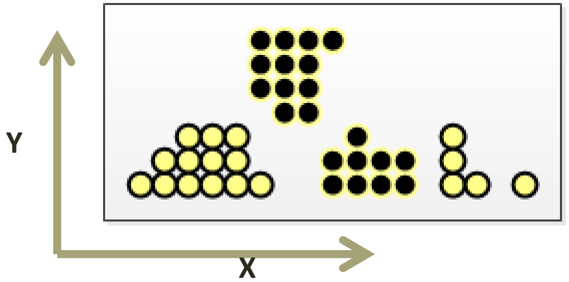
The difference could be significant

# Clustering: Dimensions (8)



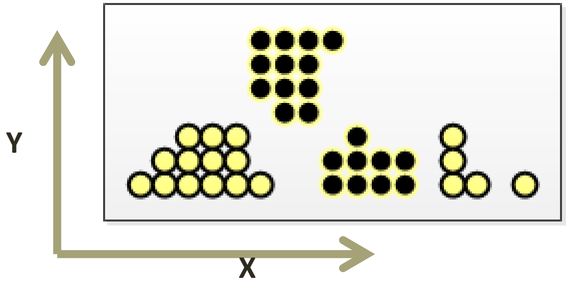
A big difference in the 2nd D can lead to different clusters

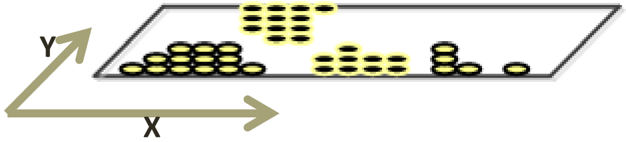
# Clustering:  Dimensions (9)



We can introduce another D by color coding.  This is a Boolean Dimension

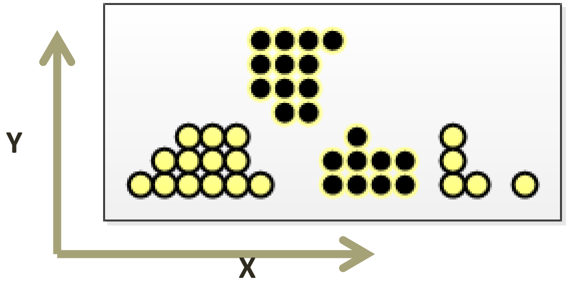
# Clustering: Dimensions (10)

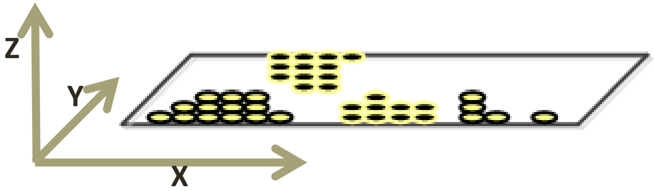




Create a 3rd Dimansion

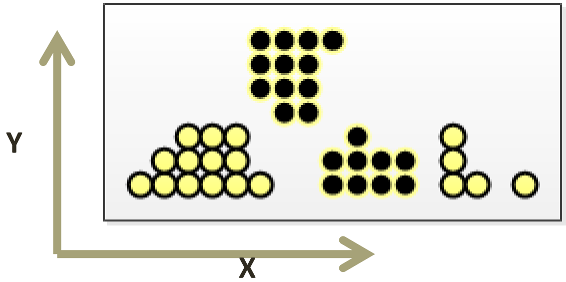
# Clustering:  Dimensions (11)

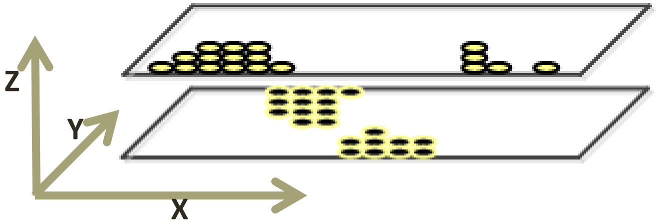




Create a 3rd Dimansion

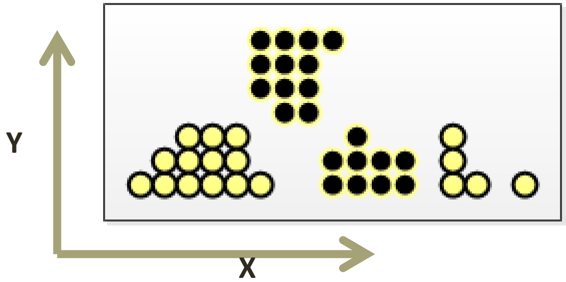
# Clustering: Dimensions (12)

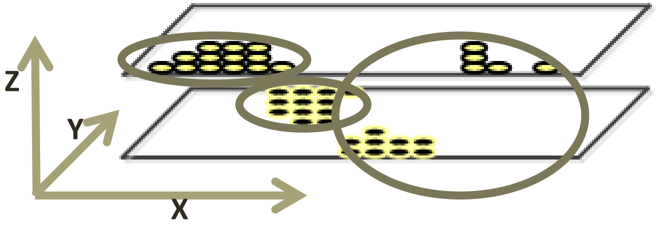


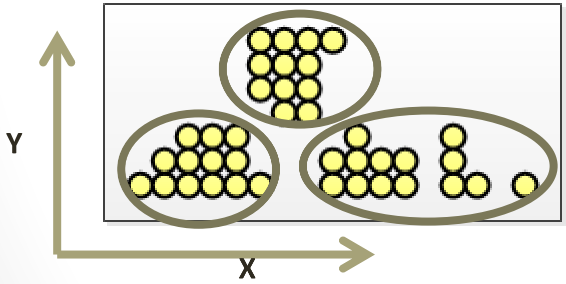


Where are the 3 clusters now?

# Clustering: Dimensions (13)

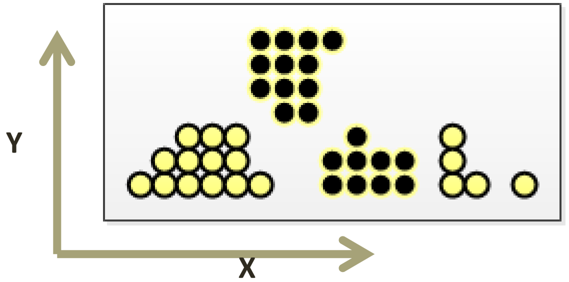


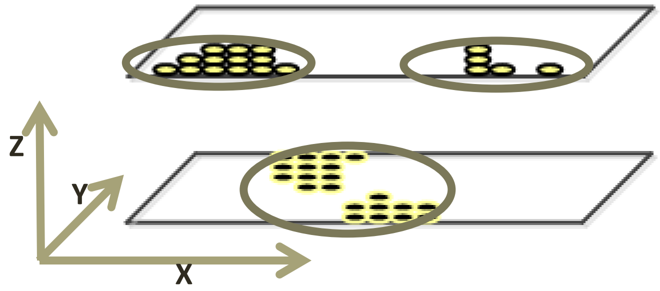


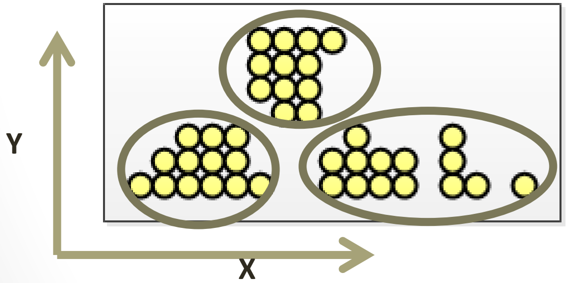


If the 3rd is small, then the clustering is the same as in 2D

# Clustering: Dimensions (14)



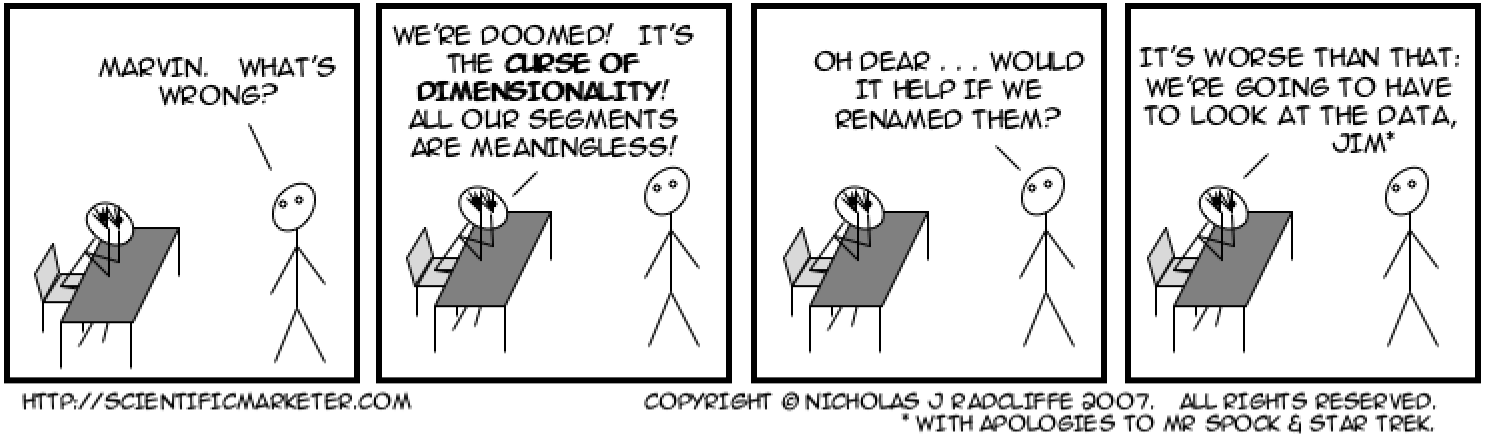




If the 3rd is big, then the clustering differs from 2D

# Dimensions in Clustering

# Break

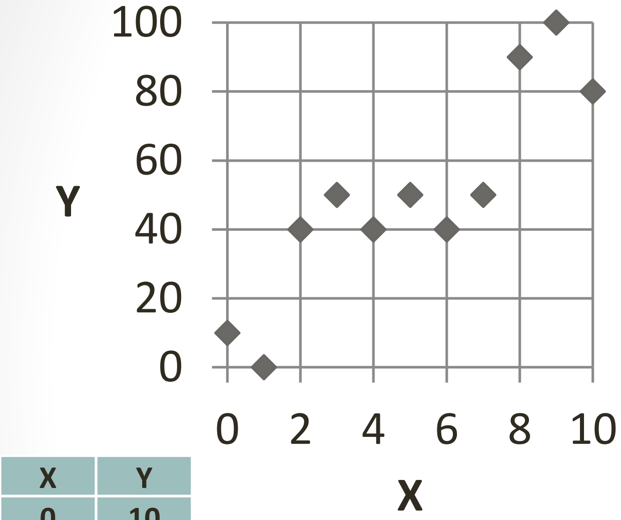

MARVIN. WHAT'S WRONG?
WE'RE DOOMED! IT'S THE **CURSE OF DIMENSIONALITY**! ALL OUR SEGMENTS ARE MEANINGLESS!
OH DEAR . . . WOULD IT HELP IF WE RENAMED THEM?
IT'S WORSE THAN THAT: WE'RE GOING TO HAVE TO LOOK AT THE DATA, JIM*
HTTP://SCIENTIFICMARKETER.COM
COPYRIGHT © NICHOLAS J RADCLIFFE 2007. ALL RIGHTS RESERVED.
* WITH APOLOGIES TO MR SPOCK & STAR TREK.

# Normalization in Clustering

# Normalization of a linear relationship (1)

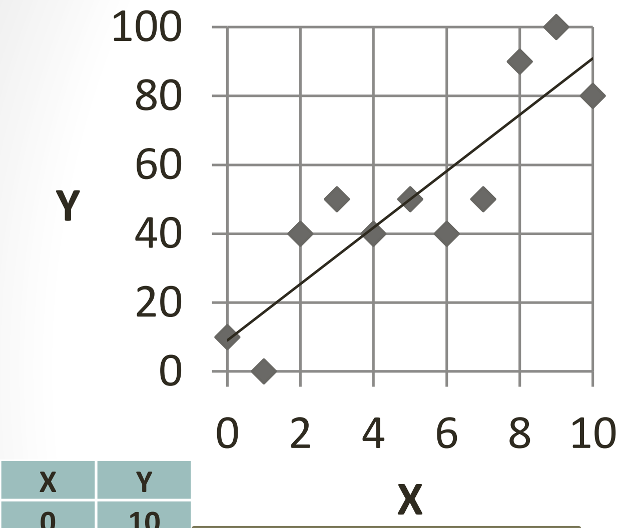
| X | Y |
|---|---|
| 0 | 10 |
| 1 | 0 |
| 2 | 40 |
| 3 | 50 |
| 4 | 40 |
| 5 | 50 |
| 6 | 40 |
| 7 | 50 |
| 8 | 90 |
| 9 | 100 |
| 10 | 80 |

# Normalization of a linear relationship (2)



| X | Y |
|---|---|
| 0 | 10 |
| 1 | 0 |
| 2 | 40 |
| 3 | 50 |
| 4 | 40 |
| 5 | 50 |
| 6 | 40 |
| 7 | 50 |
| 8 | 90 |
| 9 | 100 |
| 10 | 80 |

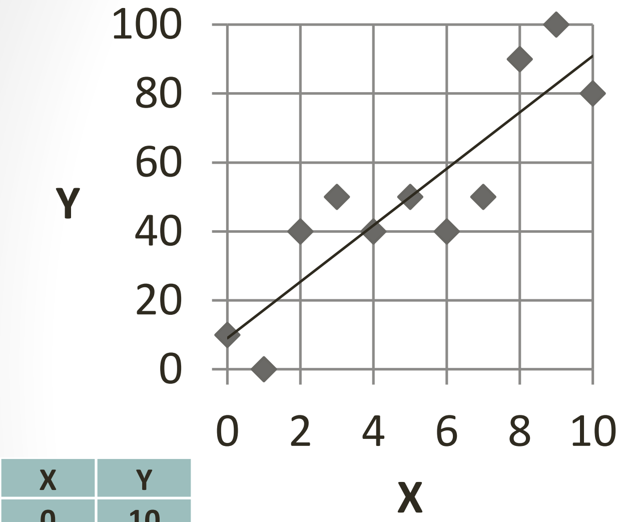
# Normalization of a linear relationship (3)



| X | Y |
|---|---|
| 0 | 10 |
| 1 | 0 |
| 2 | 40 |
| 3 | 50 |
| 4 | 40 |
| 5 | 50 |
| 6 | 40 |
| 7 | 50 |
| 8 | 90 |
| 9 | 100 |
| 10 | 80 |

Y = 10 + 8*X

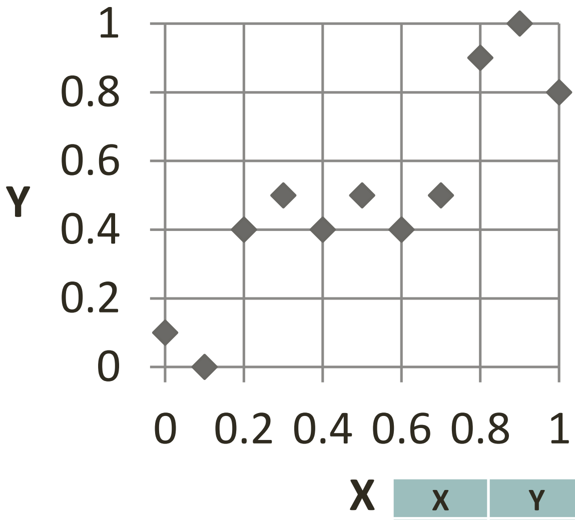
# Normalization of a linear relationship (4)



| X | Y |
|---|---|
| 0 | 10 |
| 1 | 0 |
| 2 | 40 |
| 3 | 50 |
| 4 | 40 |
| 5 | 50 |
| 6 | 40 |
| 7 | 50 |
| 8 | 90 |
| 9 | 100 |
| 10 | 80 |

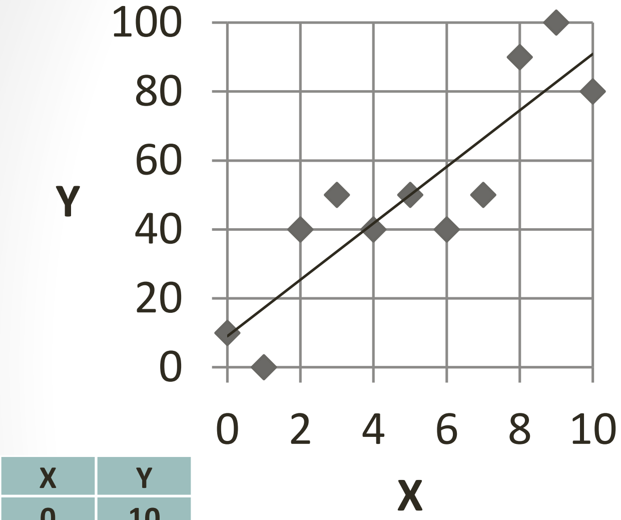
Y = 10 + 8*X

Normalize

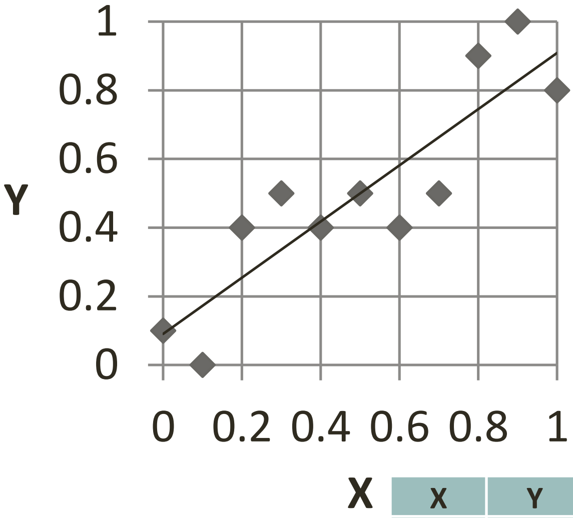


| X | Y |
|---|---|
| 0 | 0.1 |
| 0.1 | 0 |
| 0.2 | 0.4 |
| 0.3 | 0.5 |
| 0.4 | 0.4 |
| 0.5 | 0.5 |
| 0.6 | 0.4 |
| 0.7 | 0.5 |
| 0.8 | 0.9 |
| 0.9 | 1 |
| 1 | 0.8 |

# Normalization of a linear relationship (5)



Normalize



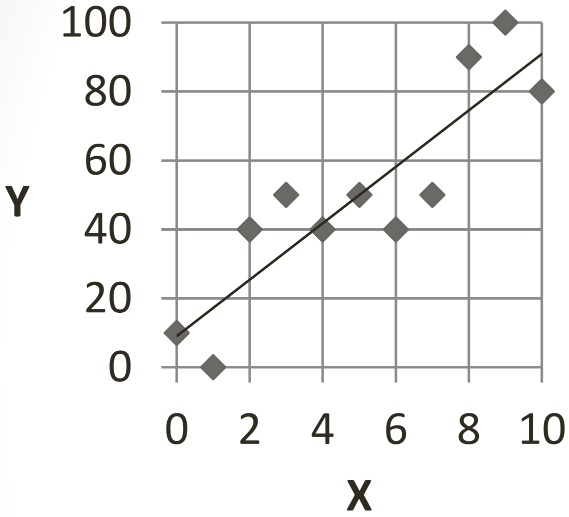
| X | Y |
|---|---|
| 0 | 10 |
| 1 | 0 |
| 2 | 40 |
| 3 | 50 |
| 4 | 40 |
| 5 | 50 |
| 6 | 40 |
| 7 | 50 |
| 8 | 90 |
| 9 | 100 |
| 10 | 80 |

Y = 10 + 8*X

Y = 0.1 + 0.8*X

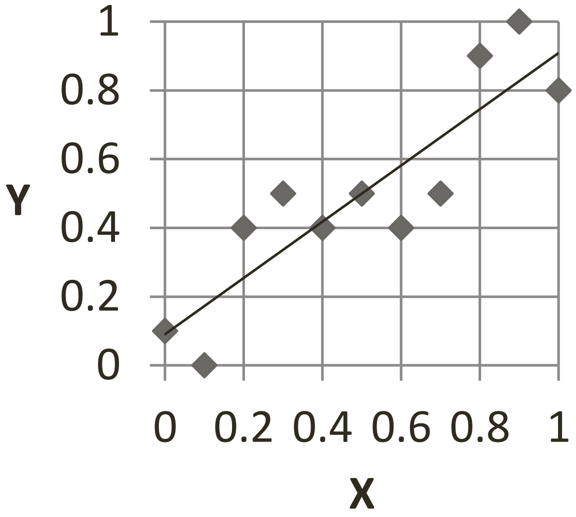
| X | Y |
|---|---|
| 0 | 0.1 |
| 0.1 | 0 |
| 0.2 | 0.4 |
| 0.3 | 0.5 |
| 0.4 | 0.4 |
| 0.5 | 0.5 |
| 0.6 | 0.4 |
| 0.7 | 0.5 |
| 0.8 | 0.9 |
| 0.9 | 1 |
| 1 | 0.8 |

# Normalization of a linear relationship (6)



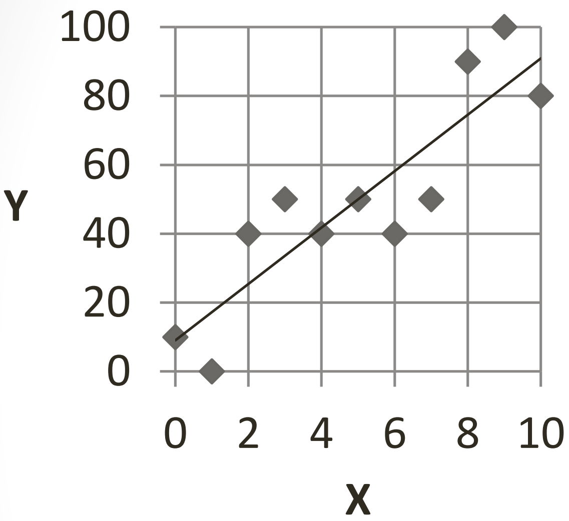
Y = 10 + 8*X

Normalize

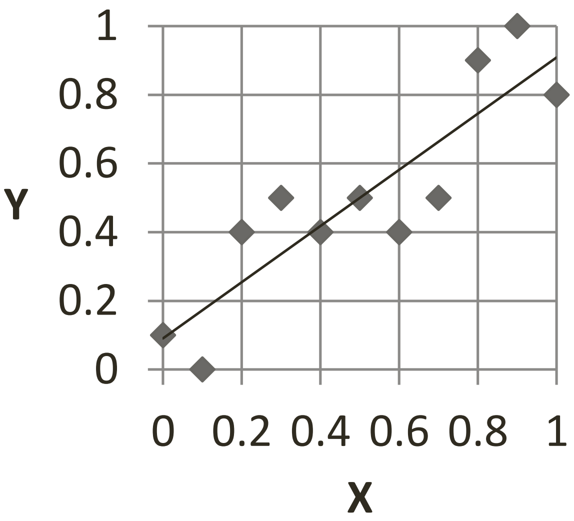


Y = 0.1 + 0.8*X

# Normalization of a linear relationship (7)



Normalize



Y = 10 + 8*X

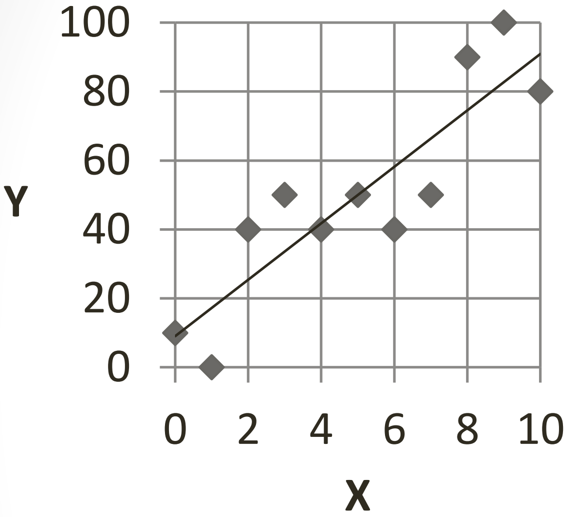
Normalize Input
X = 2 -> X’ = 0.2

Predict Output
X’ = 0.2 -> Y’= 0.26

Denormalize Output
Y’= 0.26 -> Y = 26

Y = 0.1 + 0.8*X

# Normalization of a linear relationship (8)



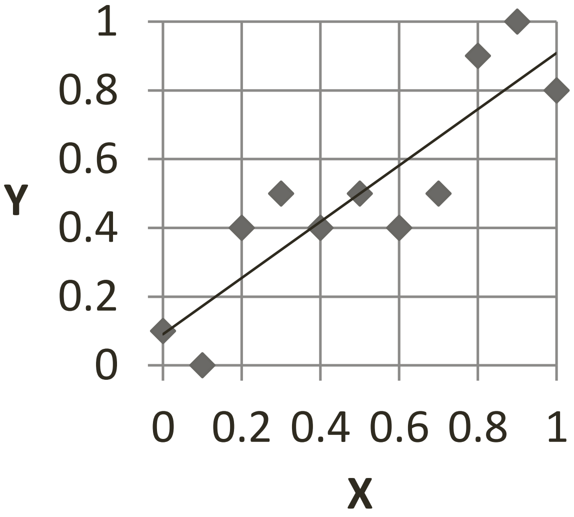
Y = 10 + 8*X

Normalize

Normalize Input
X = 2 -> X’ = 0.2

Predict Output
X’ = 0.2 -> Y’= 0.26

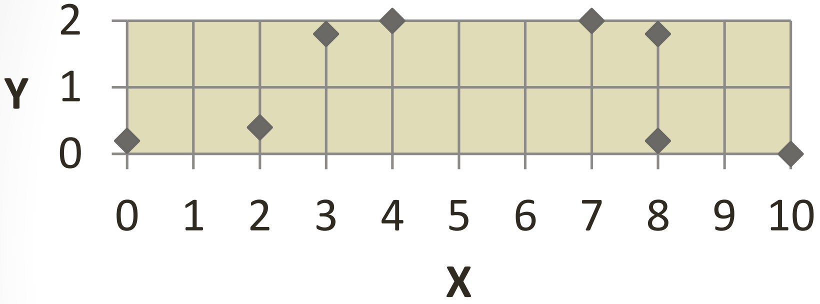
Denormalize Output
Y’= 0.26 -> Y = 26



Y = 0.1 + 0.8*X

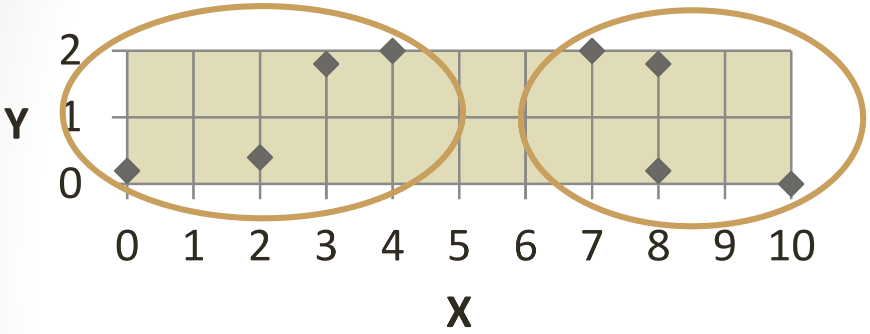
Prediction in Original Space:
X = 2 -> Y = 26

# Normalization of a non-linear relationship (1)



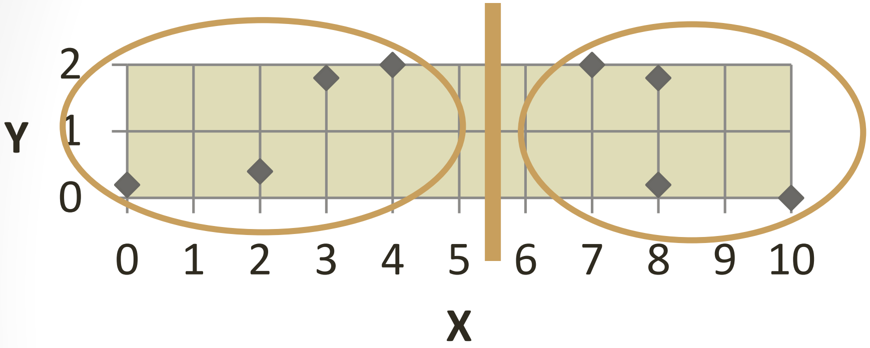
Original data in 2D:
Find 2 clusters

# Normalization of a non-linear relationship (2)



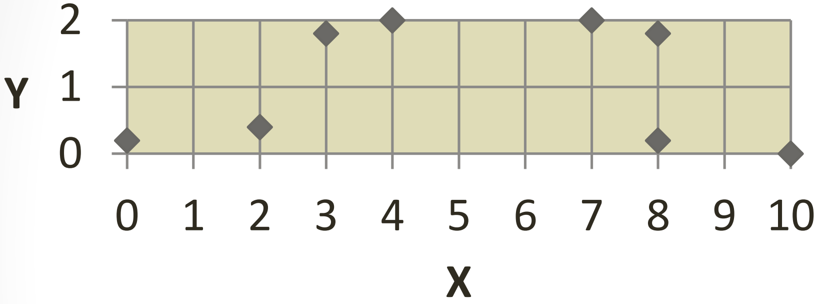
Found 2 Clusters

# Normalization of a non-linear relationship (3)



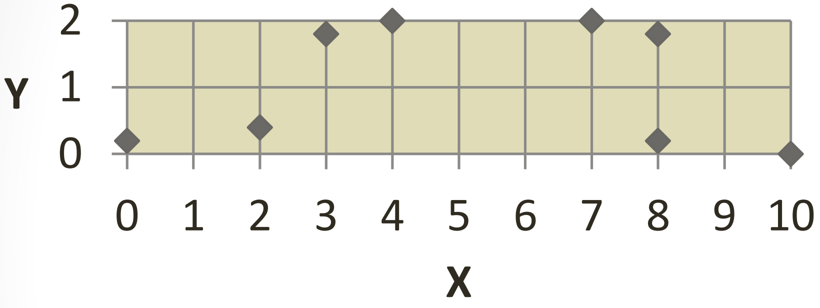
Clusters  segment the image

# Normalization of a non-linear relationship (4)

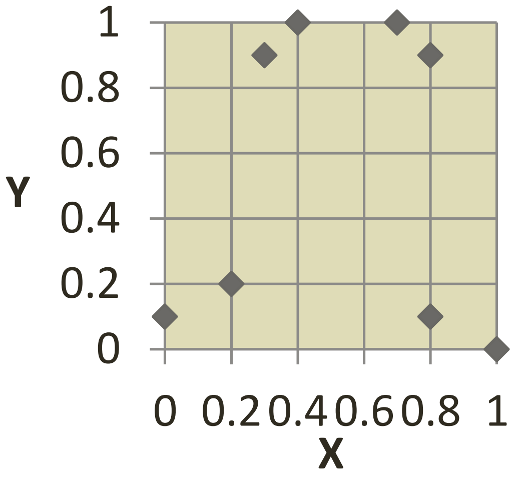


Non-normalized 2D data

# Normalization of a non-linear relationship (5)



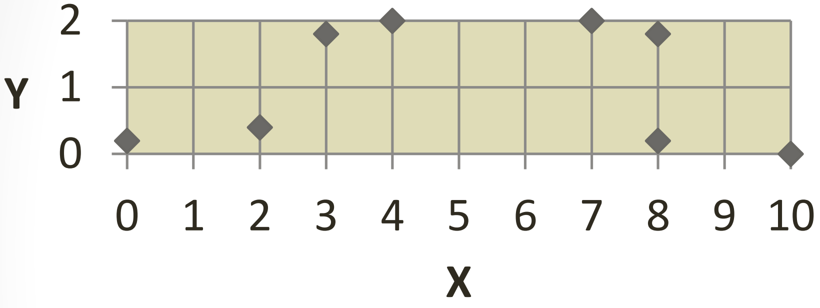
Normalize



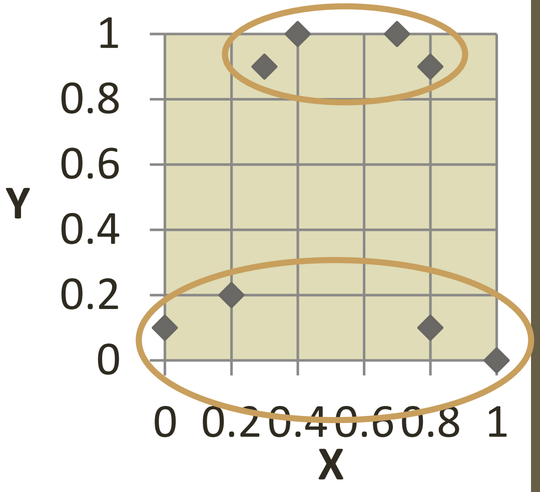
Non-normalized 2D data

Normalize the data:
Search for 2 Clusters

# Normalization of a non-linear relationship (6)



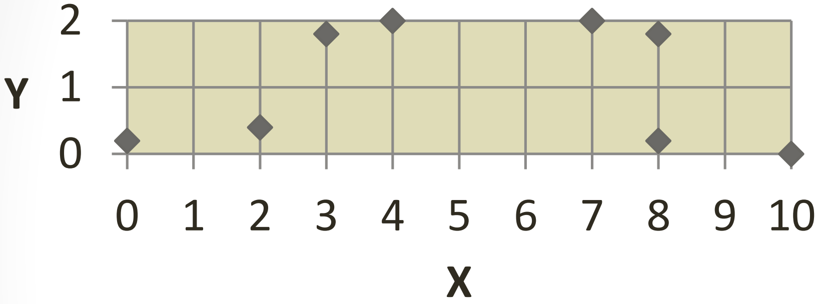
Normalize



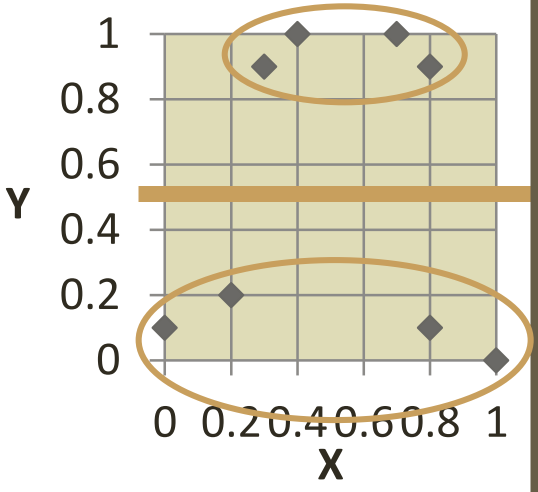
Non-normalized 2D data

Found 2 Clusters in the normalized data

# Normalization of a non-linear relationship (6)



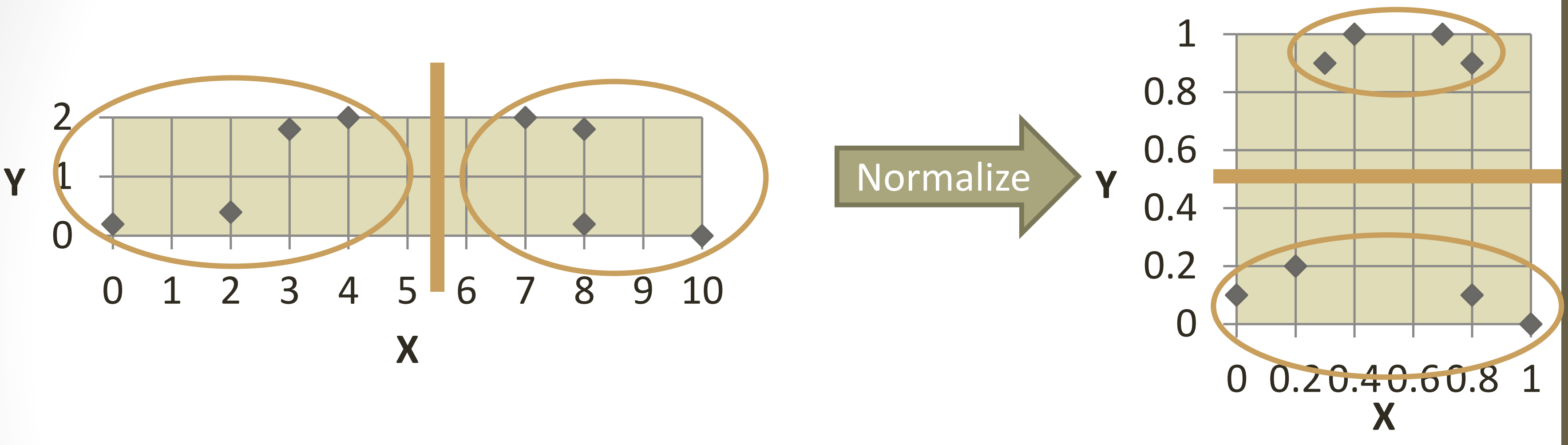
Normalize



Non-normalized 2D data

Clusters Segment the Image

# Normalization of a non-linear relationship (7)



Clustering before normalization

Clustering after normalization

# Normalization of Linear and Non-Linear Outcomes

- Non-linear (Normalization can change outcome):
  - K-Means
  - Neural Net
- Linear (Normalization should not change outcome):
  - Logistic Regression
  - Linear Regression
  - Mixture of Gaussians
- https://en.wikipedia.org/wiki/Linearity
- https://en.wikipedia.org/wiki/Linear_function

# Normalization in Clustering

# In-Class Exercise

Normalization in K-Means

- Download L07-3-KMeansNorm_Incomplete.py from Canvas and load into Spyder.
- Run the script:  Some results will be wrong
- Add code to normalize each input dimension
- Add code to de-normalize the output
- Specifically, replace all lines that say:  "**Replace this line with code**".
- Run the script:  Results should be correct

# In-Class Exercise

1. KMeansNorm_Incomplete.py
    a. Get mean and standard deviation  of point dimensions. Use the np.mean and np.std functions
    b. Z-Normalize points and centroid guesses based on distribution of points
    c. Let the KMeans function determine the labels and the centroids in normalized space
    d. De-normalize the centroids
    e. Return the labels and the de-normalized centroids
2. Answer the following questions
    a) What is the single most obvious difference between the distributions of the first and second dimensions?
    b) Does separation of clusters in Test 1 occur along the x, y, or both dimensions? Why?
    c) Does separation of clusters in Test 2 occur along the x, y, or both dimensions? Why?
    d) Does separation of clusters in Test 3 occur along the x, y, or both dimensions? Why?
    e) Does separation of clusters in Test 4  occur along the x, y, or both dimensions? Why?
3. Why is normalization important in K-means clustering?
4. How do you encode categorical data in a K-means clustering?
5. Why is clustering un-supervised learning as opposed to supervised learning?