Introduction to Data Science

Lecture 2; October 12th, 2016

Ernst Henle
ErnstHe@UW.edu
Skype: ernst-henle

Agenda

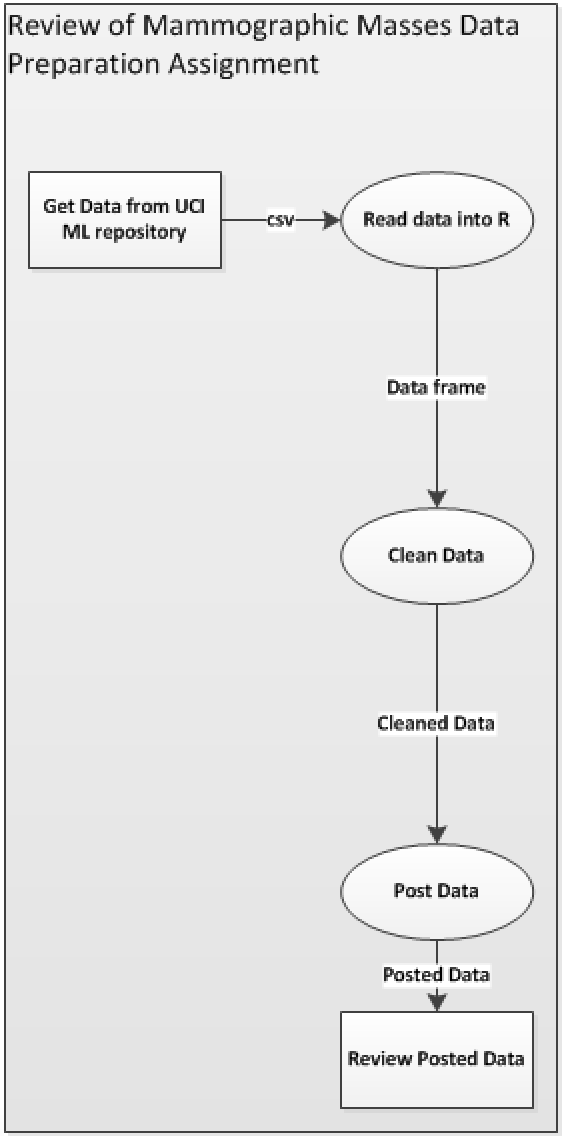
- Announcements
  - The social component is a course requirement:
    - On LinkedIn, start a discussion, make a comment on an existing discussion, or ask questions about homework.
    - Please collaborate on homework!
  - I do not use Canvas Grading or Messaging
  - Guest Lecture (45 min): Data Science Trends in the Professional World by David Porter and Emily Nichols on Oct 19th 2016
- Review
  - Optional class on programming in R
  - Homework review and Data Preparation DFD
- Quiz 02 on Data Preparation. I would use RStudio
- Introduction to K-means Clustering
- Break
- Class Exercise
- Dimensions in Clustering
- Break
- Normalization (Clustering vs Linear Regression)
- Assignment (Complete all assignments items from all assignment slides. Must be submitted by Saturday 11:57 PM)

Data Preparation Review

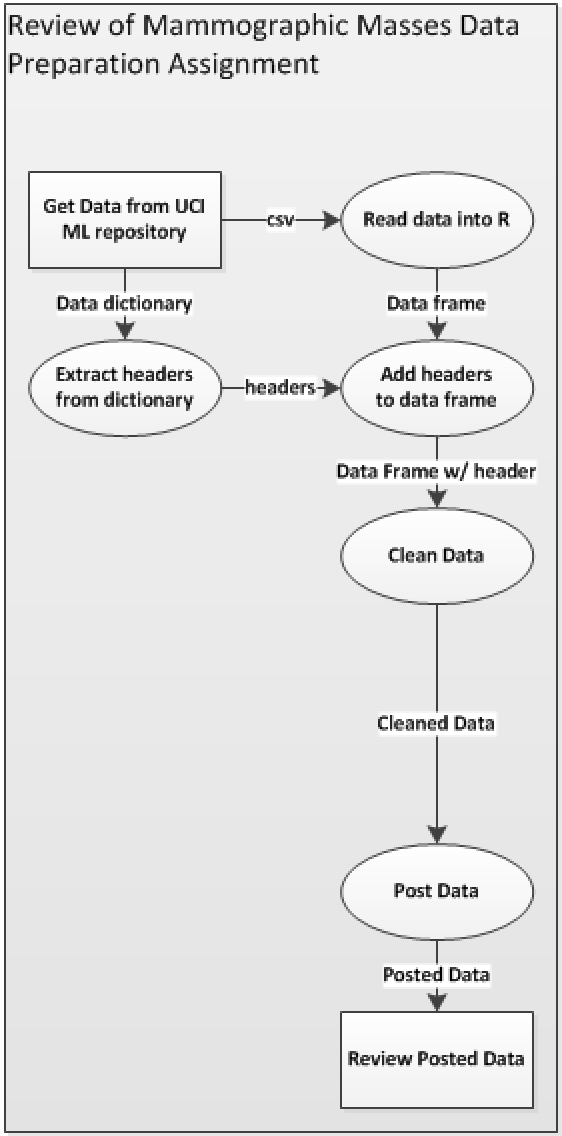
Data Preparation Review (0)

- Find in Canvas: DataScience01Homework.R

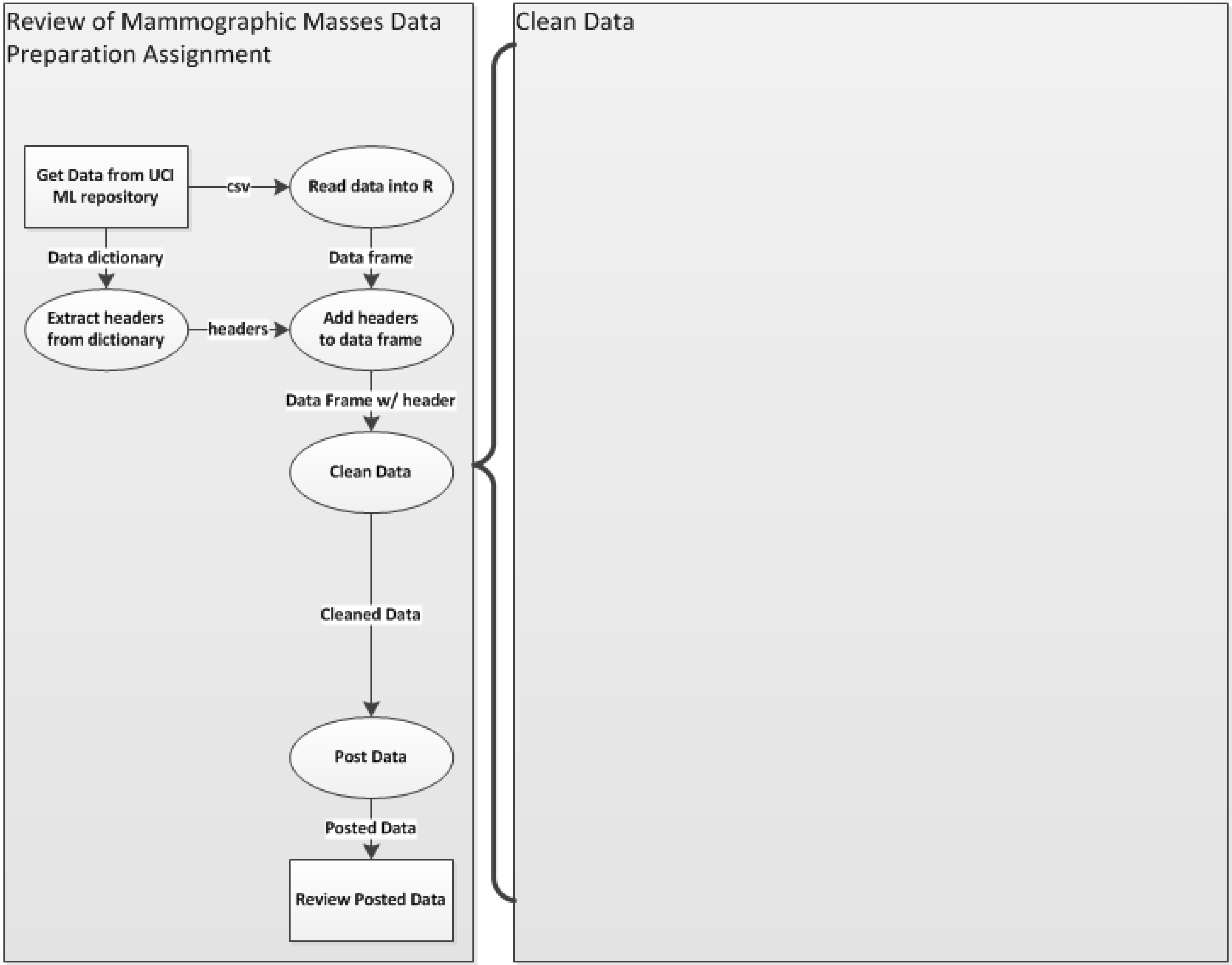
Data Preparation Review (1)



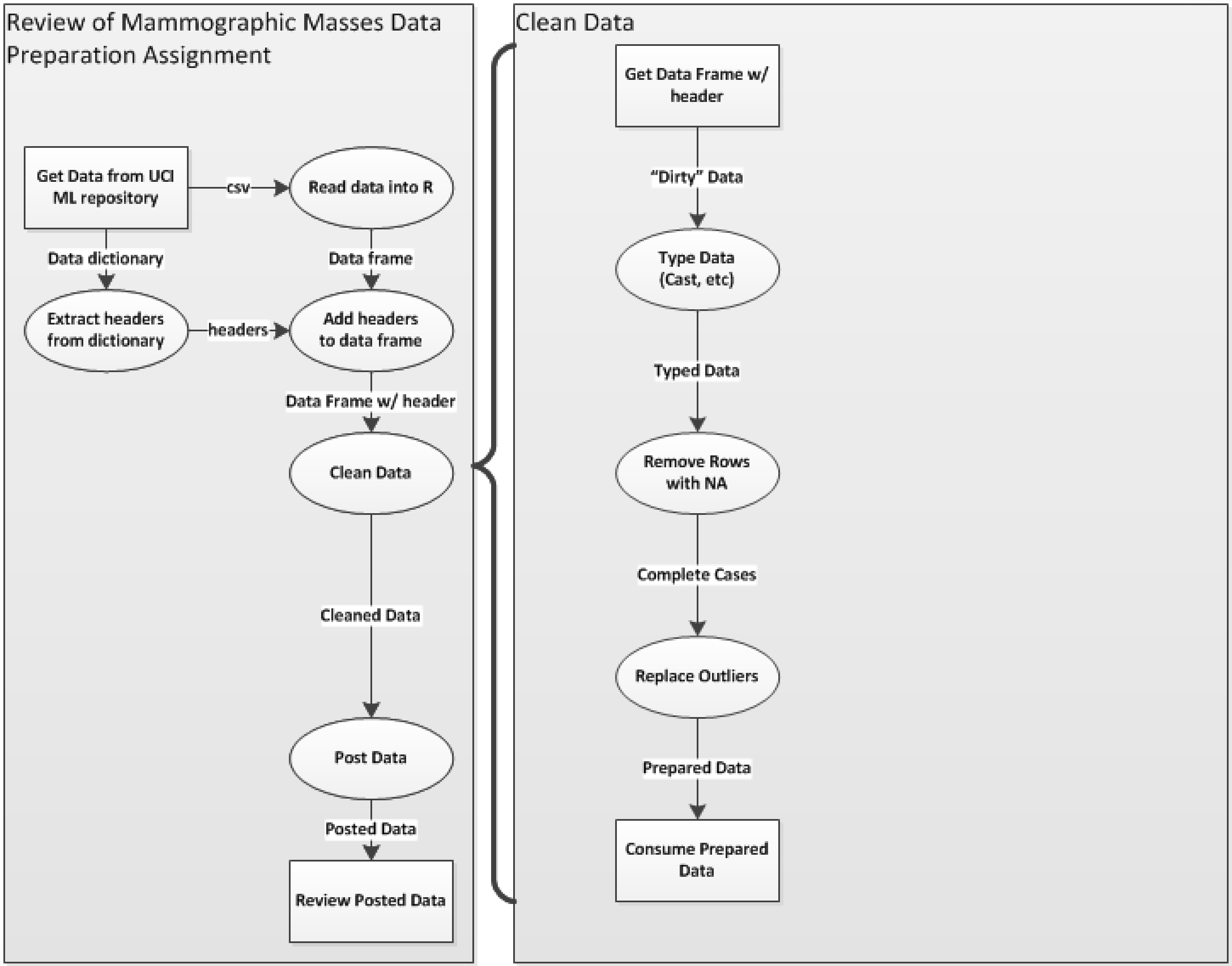
Data Preparation Review (2)



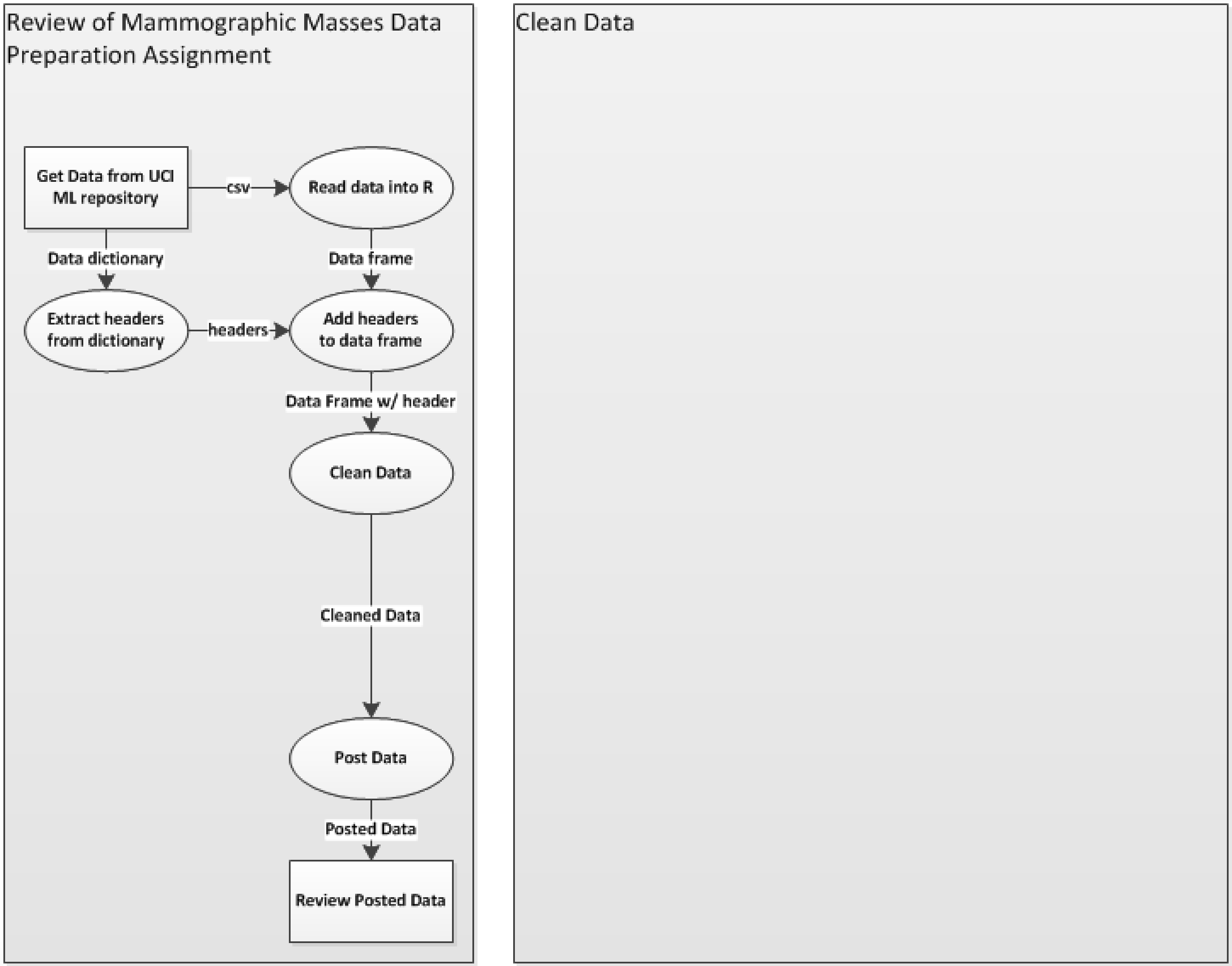
Data Preparation Review (3)



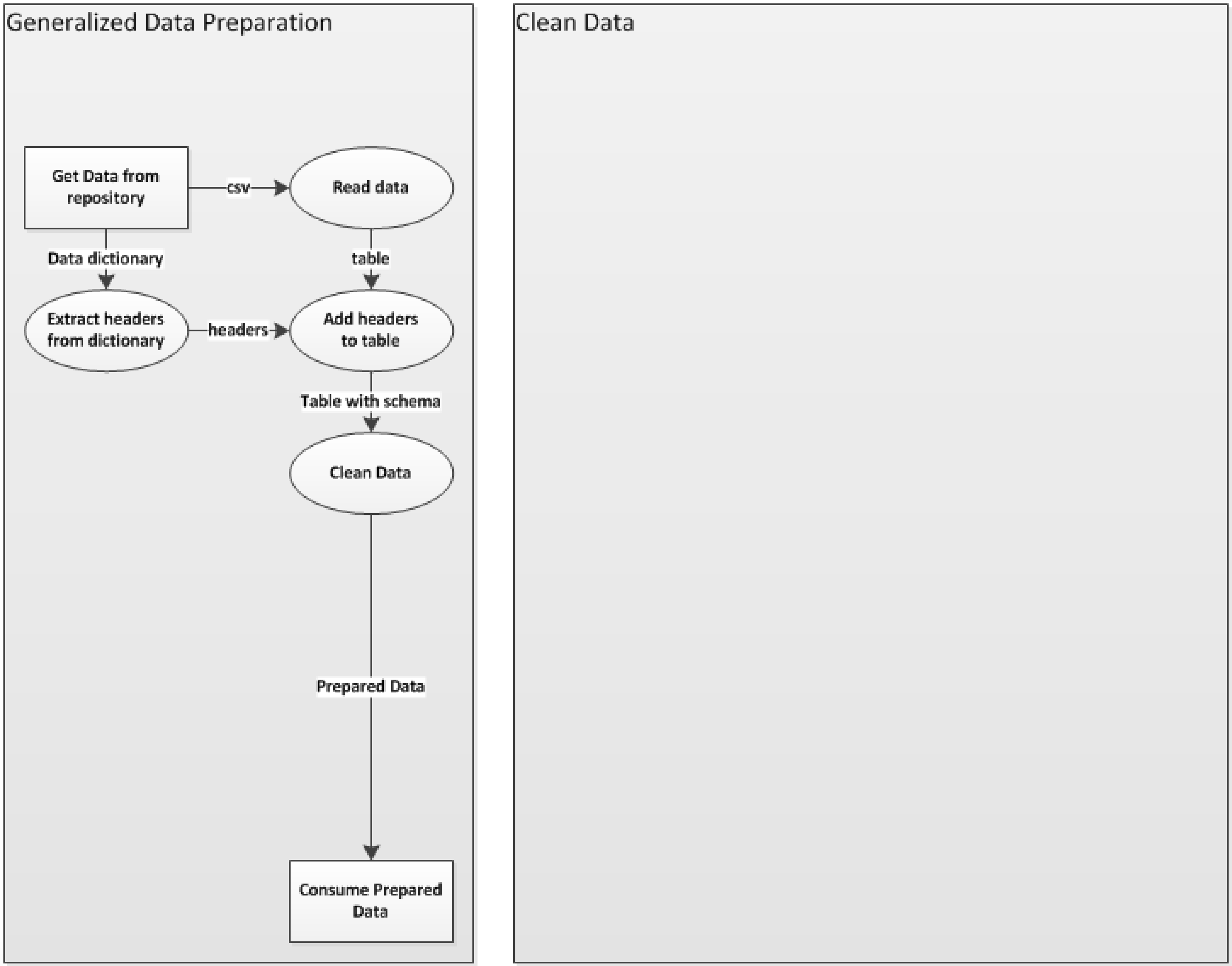
Data Preparation Review (4)



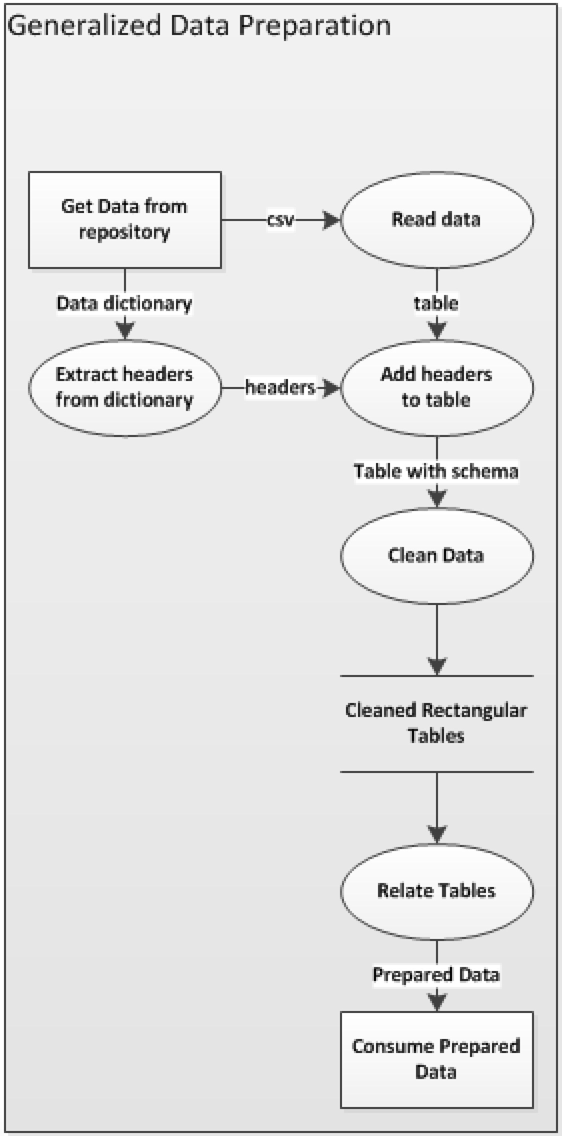
Data Preparation Review (5)

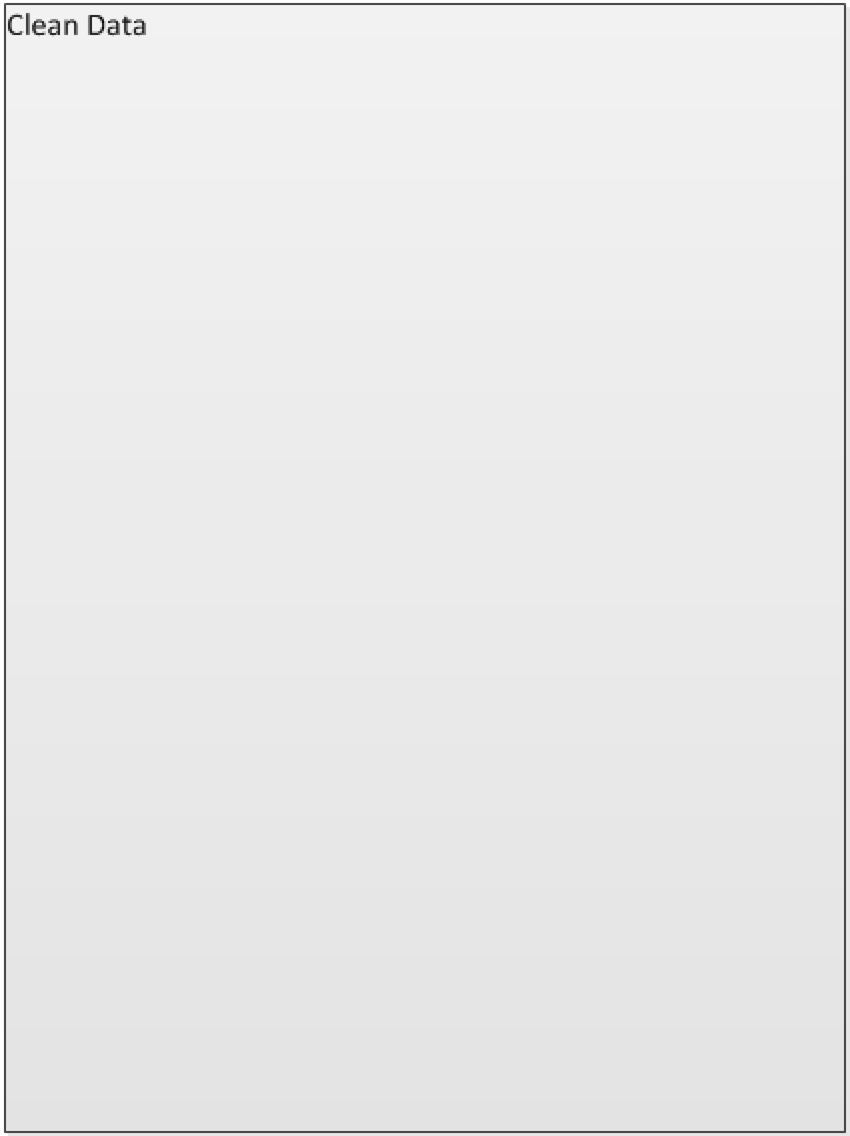


Data Preparation Review (6)

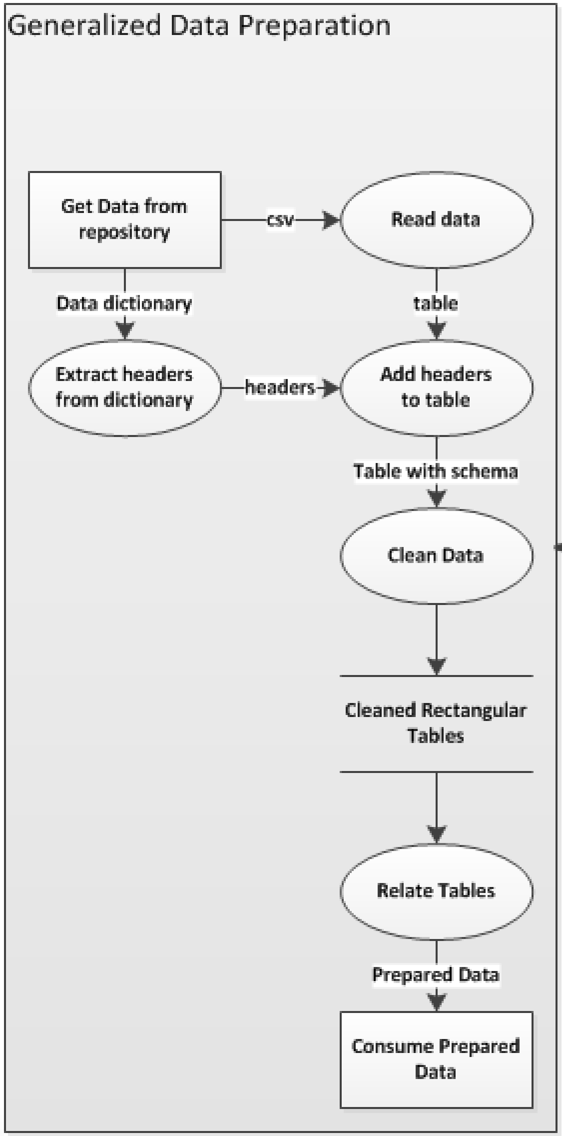


Data Preparation Review (7)



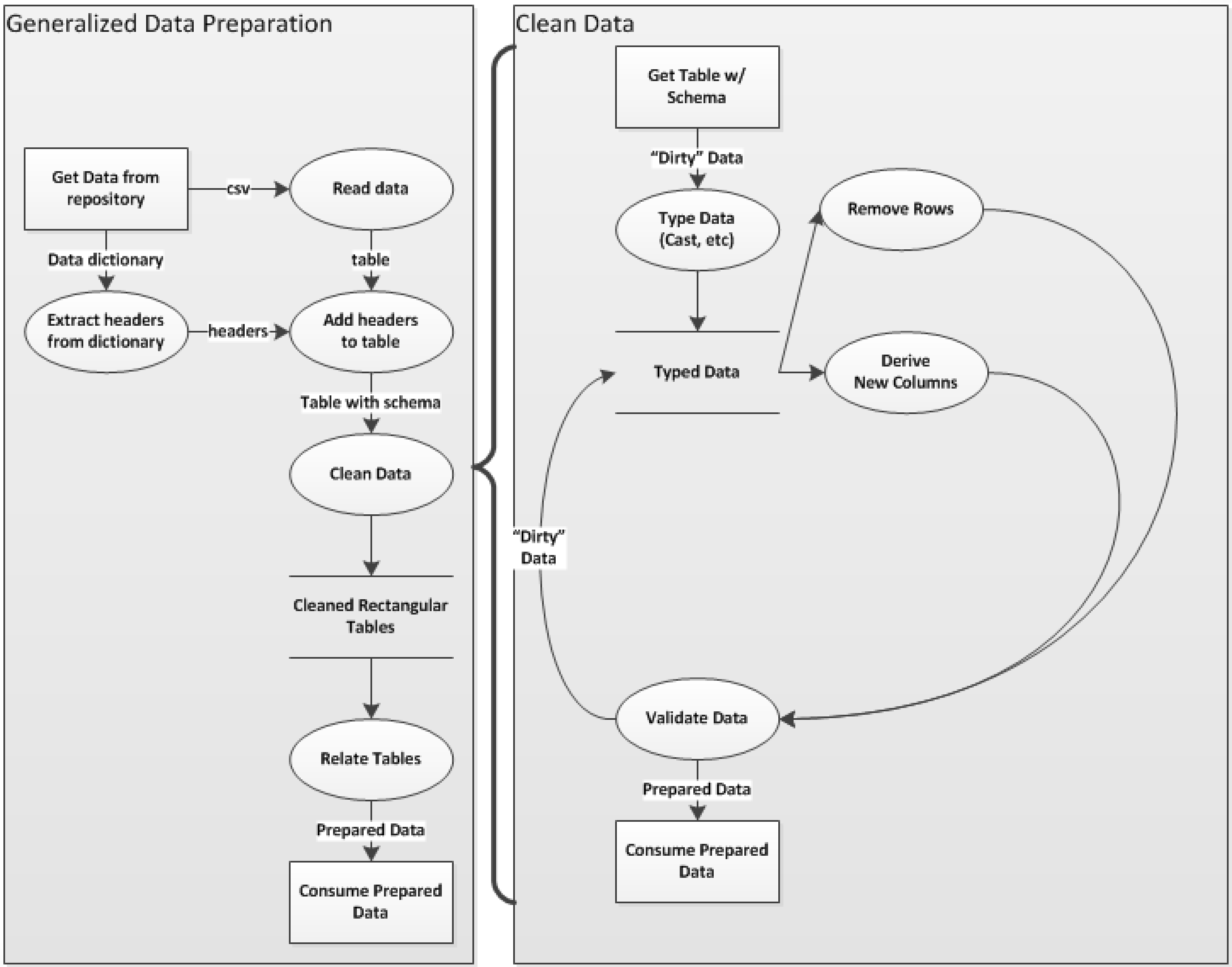


Data Preparation Review (8)

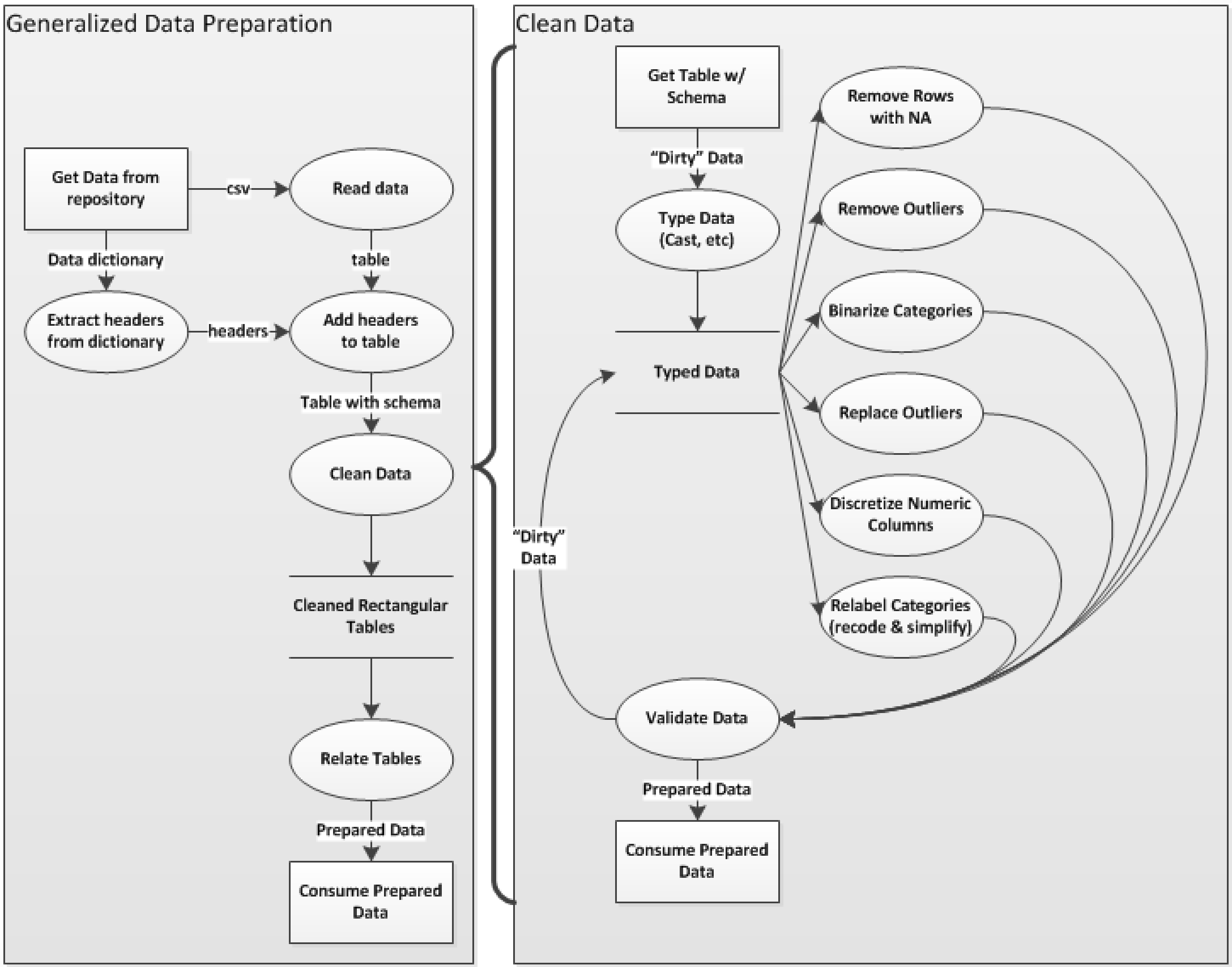


Clean Data

Data Preparation Review (9)



Data Preparation Review (10)



Data Preparation Review

Quiz 02

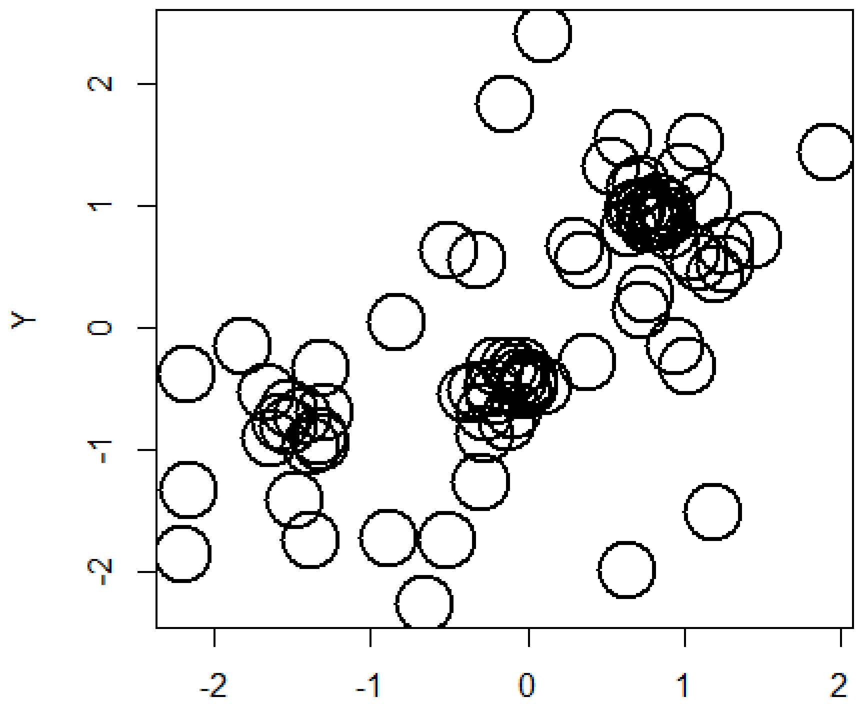
- Quiz available in Canvas

Introduction to K-means Clustering

K-means clustering: Algorithm

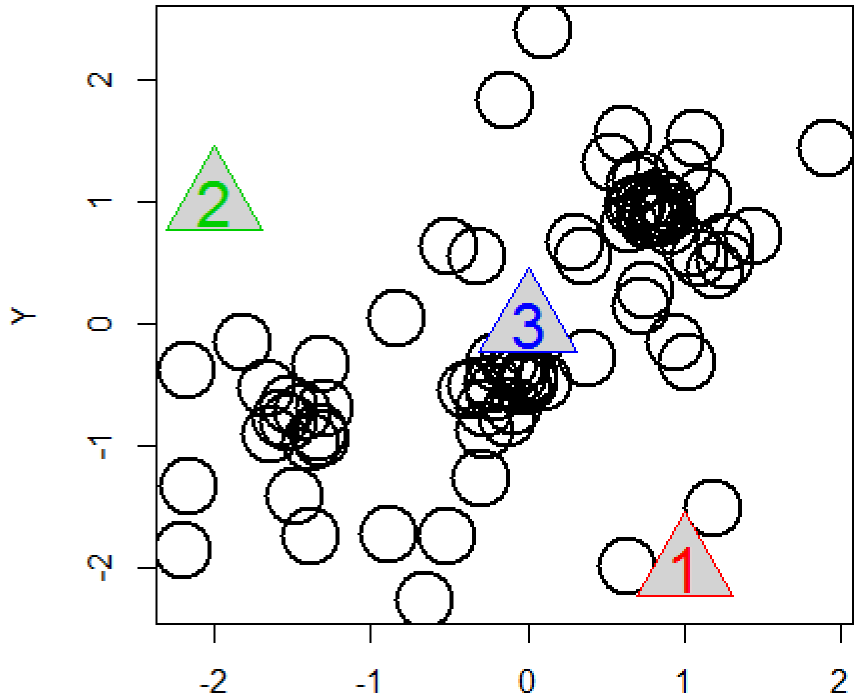
- Pre-requisites
  1. Get points in multi-dimensional space.
     - table, matrix, rectangular dataset
  2. Specify the number of clusters
     - Weakest point in algorithm
     - Get a random center for each cluster (makes algorithm non-deterministic)
     - Another weak point in the algorithm
- Repeat until convergence:
  1. For each point, determine its closest cluster center and assign that point to that cluster
  2. Determine the centroid (mean) for each cluster of points

K-Means Clustering (0)



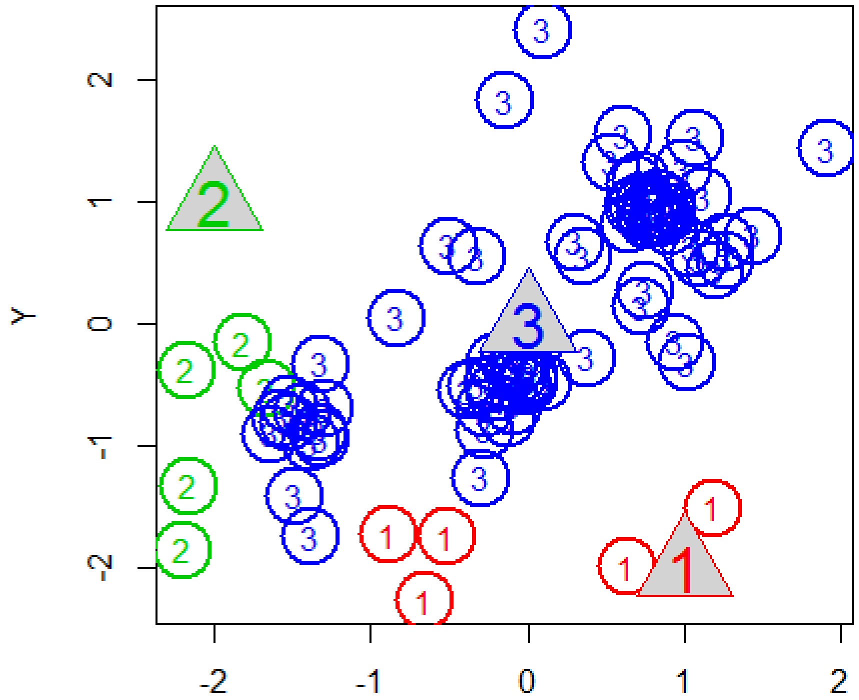
- Clustering starts by getting the data and representing the data as points in space. In this example the space is 2-dimensional.
- Each point describes an observation. An observation is an individual item.
- The dimensions are attributes that describe the item.

K-Means Clustering (1)



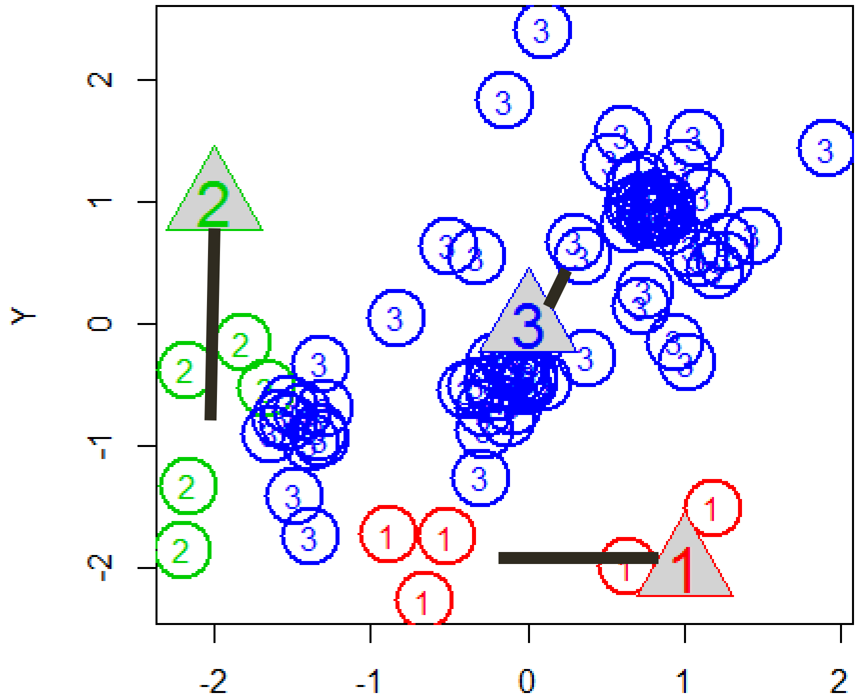
- Clustering continues by guessing, presuming, or specifying a number of clusters.
- Each centroid represents a cluster.
- The centroid positions are determined randomly. The centroids should be within the bounds of the points.

K-Means Clustering (2)



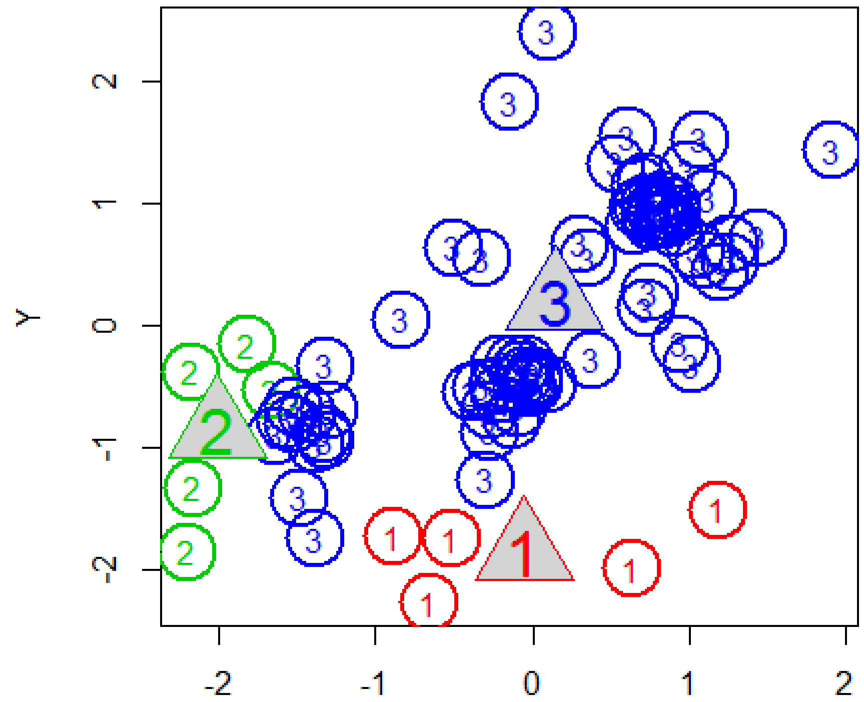
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (2)



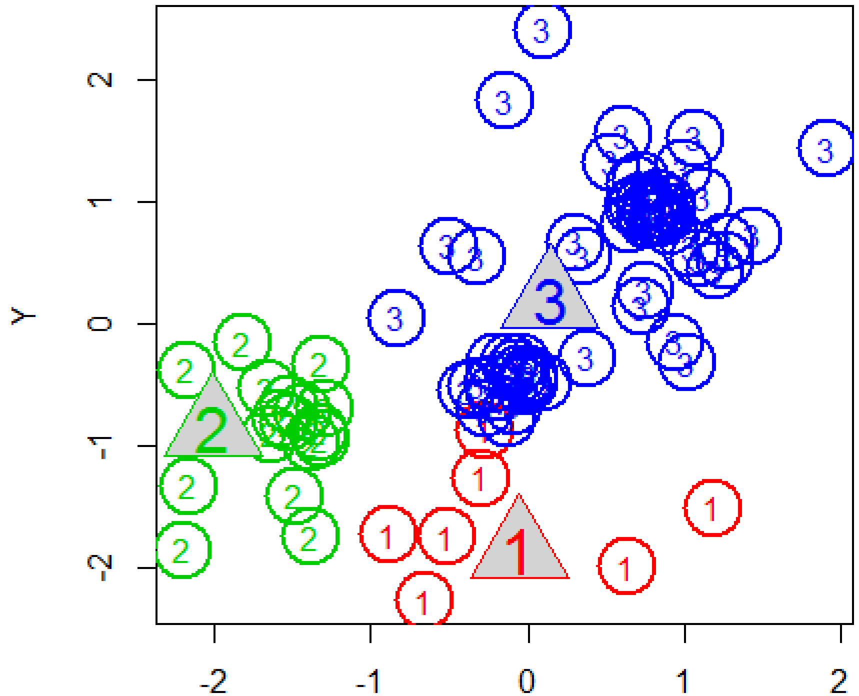
- Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (3)



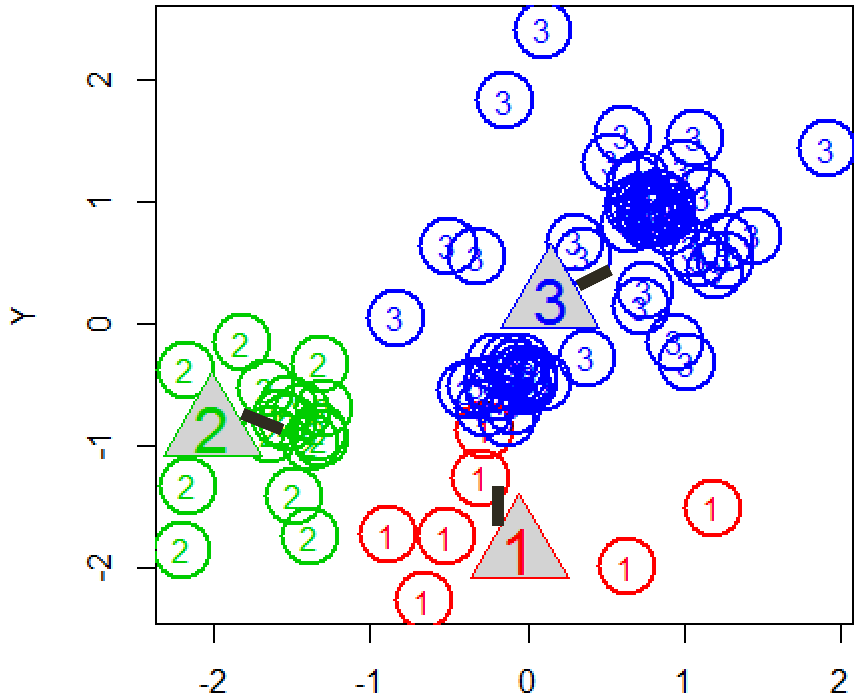
- Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (4)



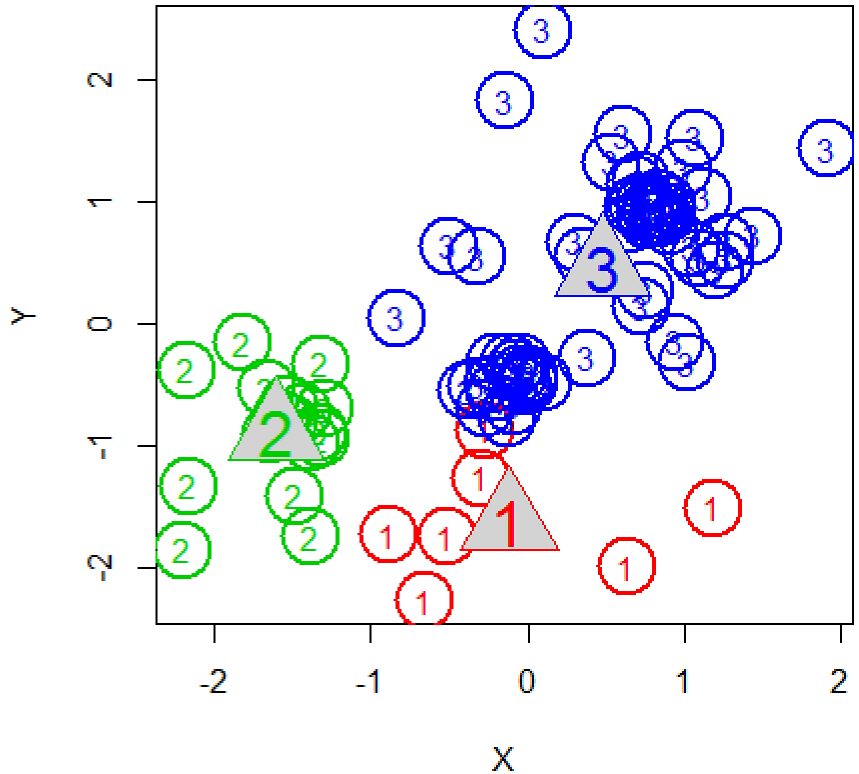
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (4)

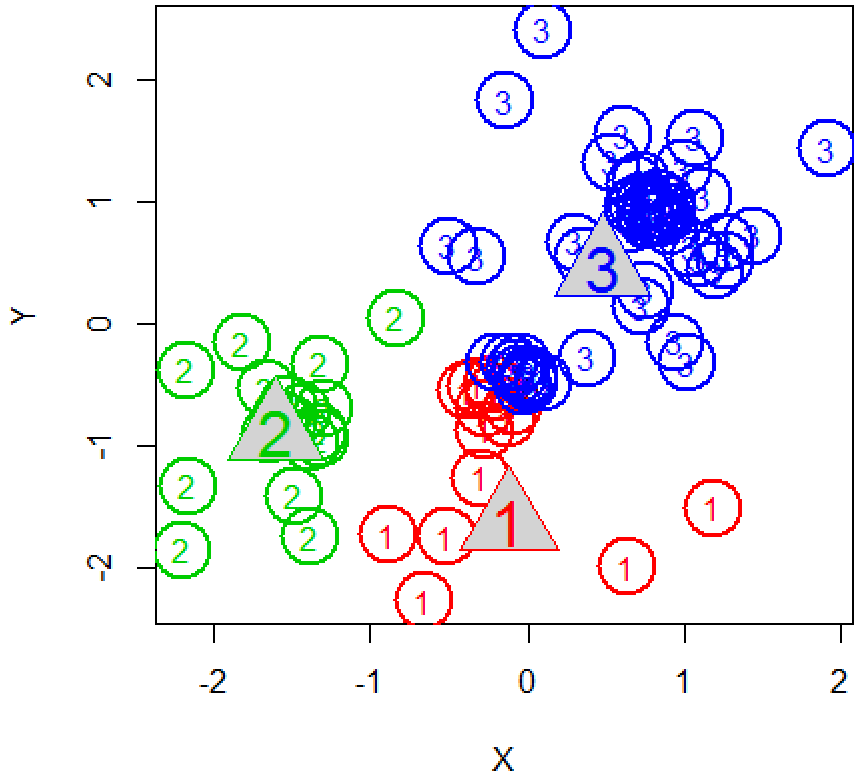


- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

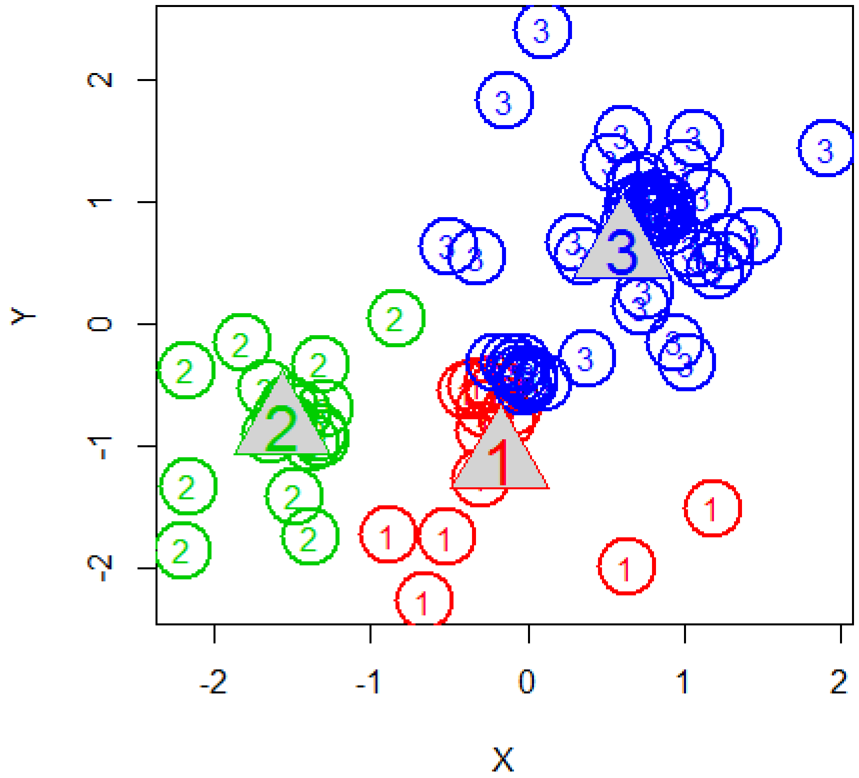
K-Means Clustering (5)



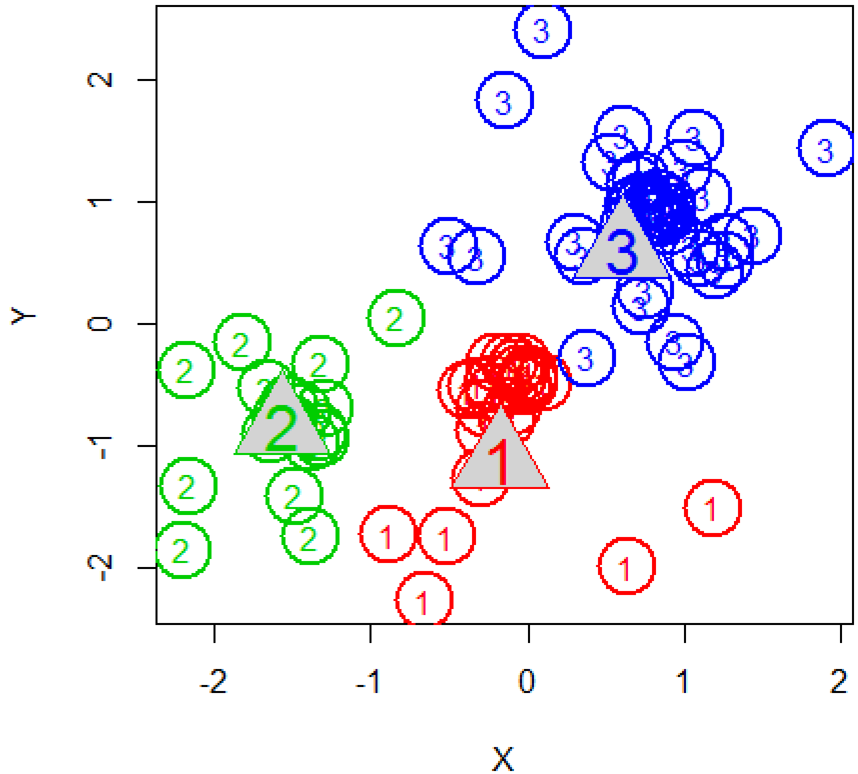
K-Means Clustering (6)



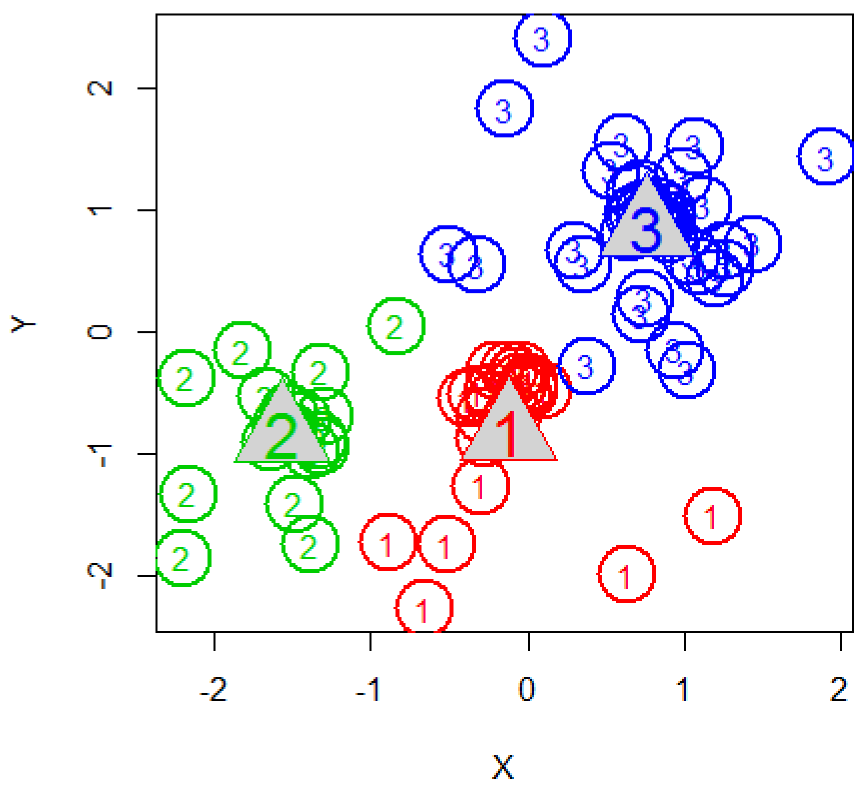
K-Means Clustering (7)



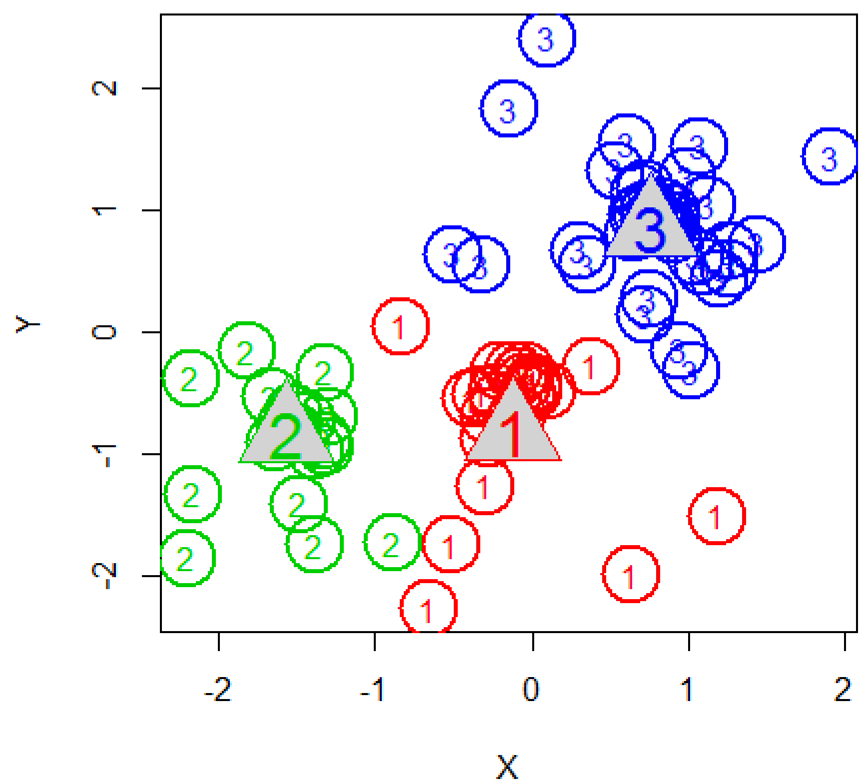
K-Means Clustering (8)



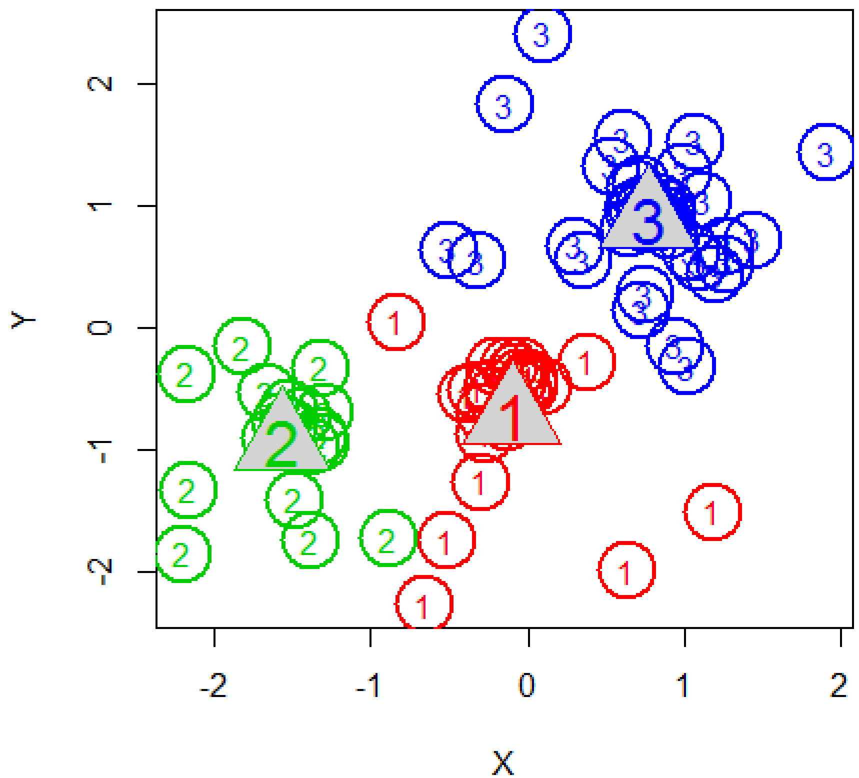
K-Means Clustering (9)



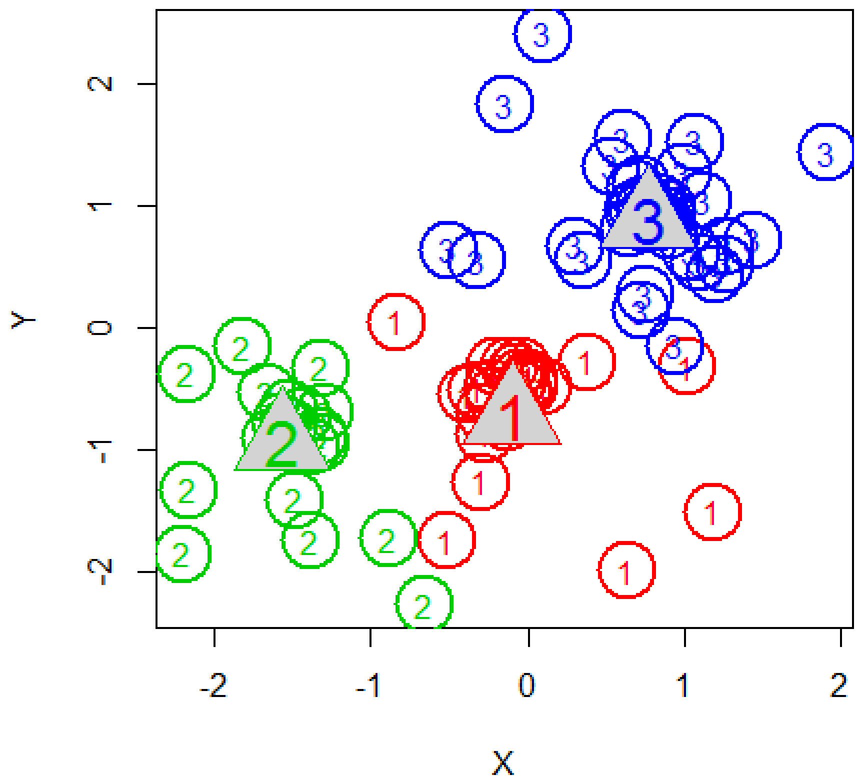
K-Means Clustering (10)



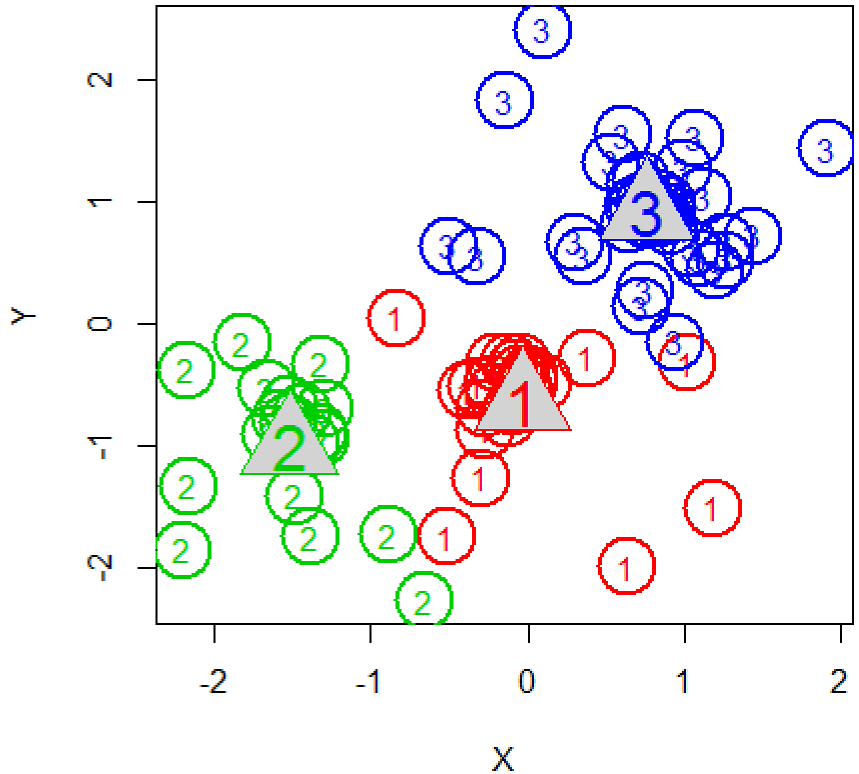
K-Means Clustering (11)



K-Means Clustering (12)



K-Means Clustering (13)



K-means

- Some Points:
  - Normalizations are important to put data on equal terms
  - Initial centroid number and placement is an art.
  - Categorical Data must be binarized
  - K-means is unsupervised because we do not tell the algorithm what outcome was observed or what outcome is desired.

Break

In-Class Exercise and Homework Assignment

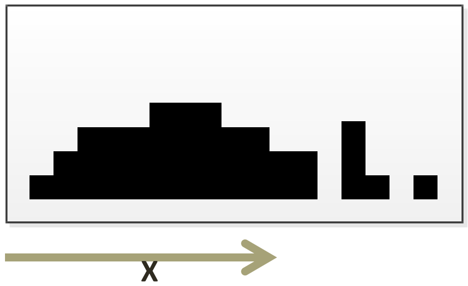
Write K-Means in R

- Download KMeansIncomplete.R and open it in RStudio
- Complete the function KMeans: replace all lines that say: "**Put code in place of this line**". Execute the built in tests and verify that your code works:
  - **ClusterPlot()**
  - **findLabelOfClosestCluster()**
  - **calculateClusterCenters()**
  - **KMeans()**

Rename the completed script KMeans.R

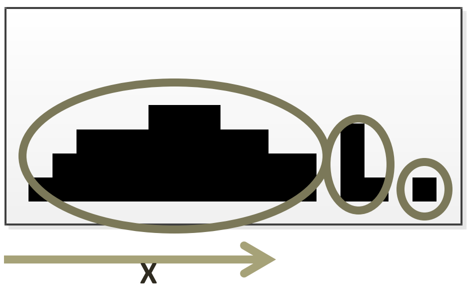
Dimensions in Clustering

Clustering: Dimensions (1)



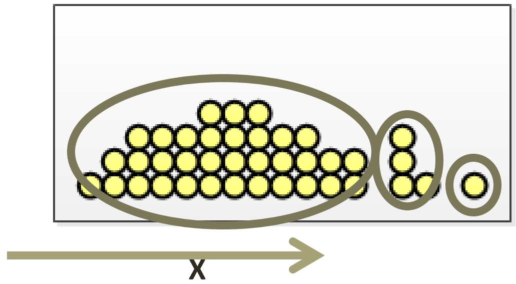
Where are the three clusters?

Clustering: Dimensions (2)



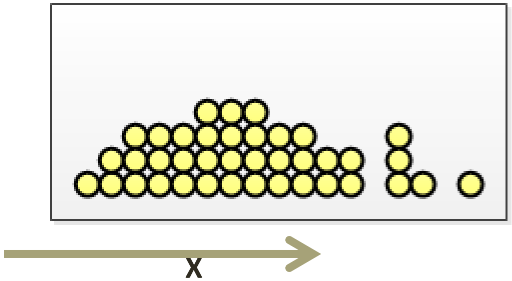
Simple assignment based on a 1D distribution

Clustering: Dimensions (3)



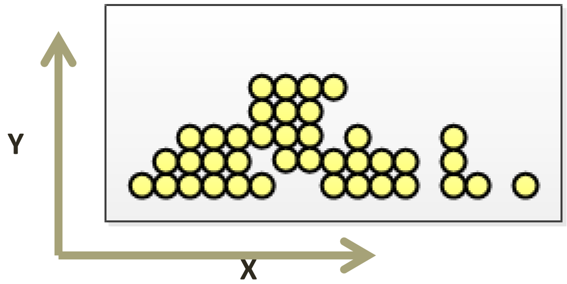
Simple assignment based on a 1D distribution

Clustering: Dimensions (4)



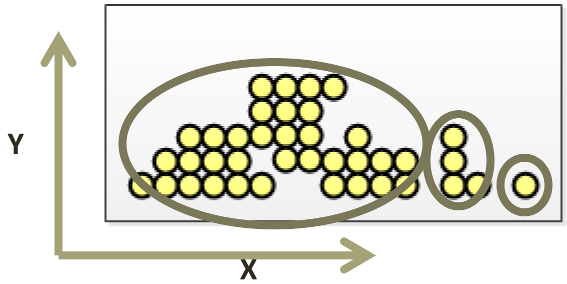
What if this was not a 1D distribution?

Clustering: Dimensions (5)



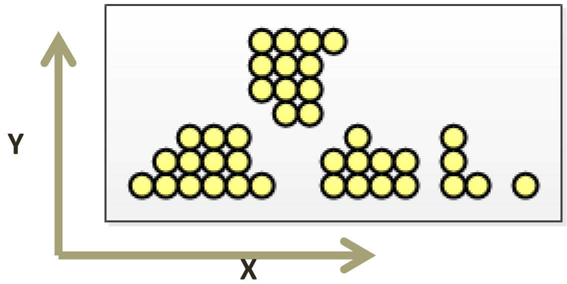
The distribution is in 2D. Some points differ in the 2nd D

Clustering: Dimensions (6)



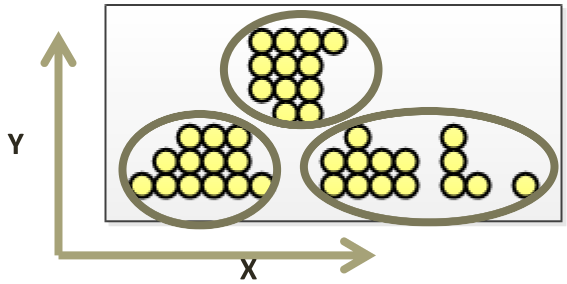
If the difference is minor, we still get the same clusters

Clustering: Dimensions (7)



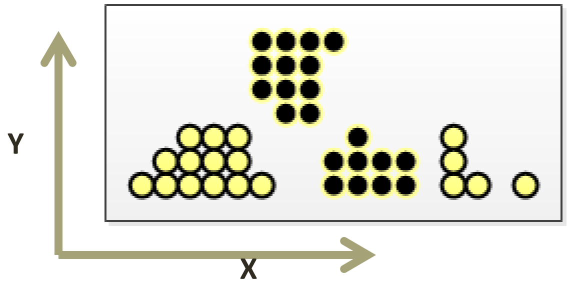
The difference could be significant

Clustering: Dimensions (8)



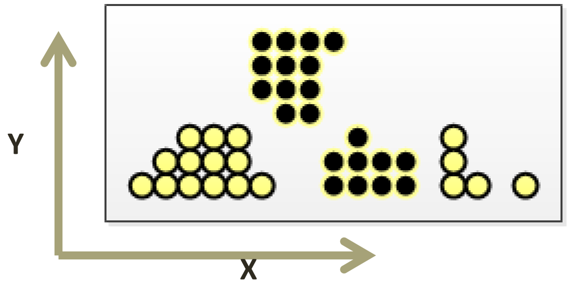
A big difference in the 2nd D can lead to different clusters

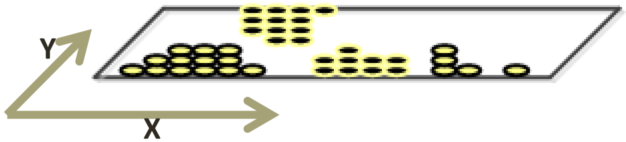
Clustering: Dimensions (9)



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

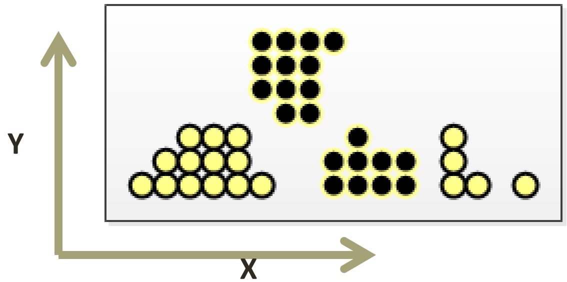
Clustering: Dimensions (10)

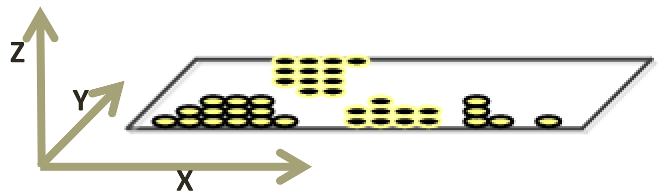




Create a 3rd Dimansion

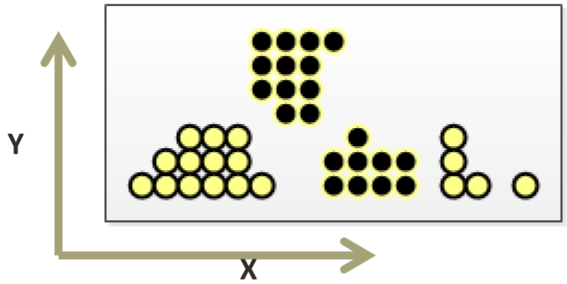
Clustering: Dimensions (11)

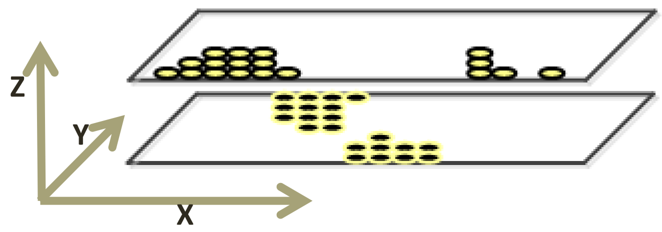




Create a 3rd Dimansion

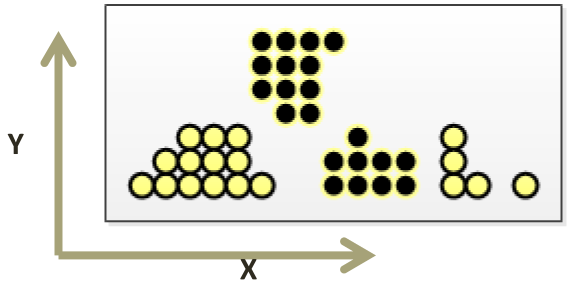
Clustering: Dimensions (12)

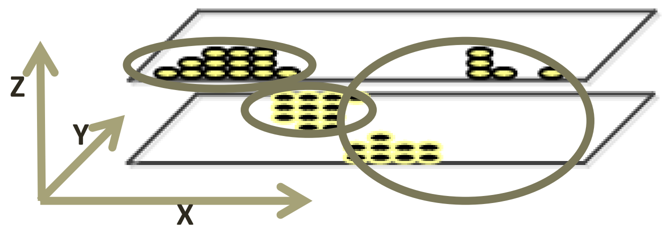


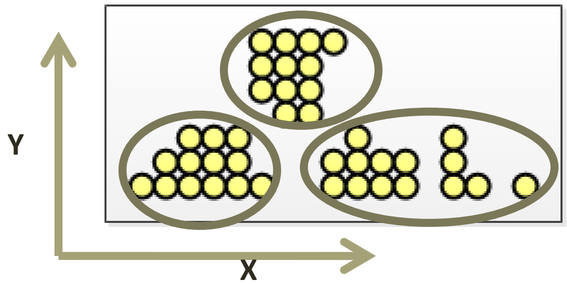


Where are the 3 clusters now?

Clustering: Dimensions (13)

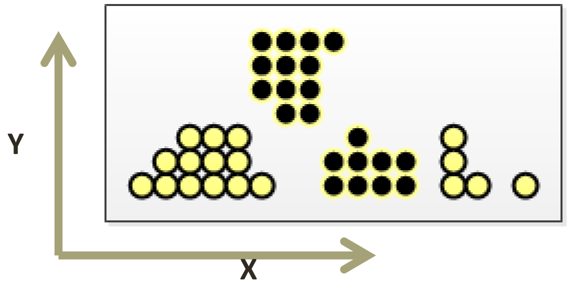


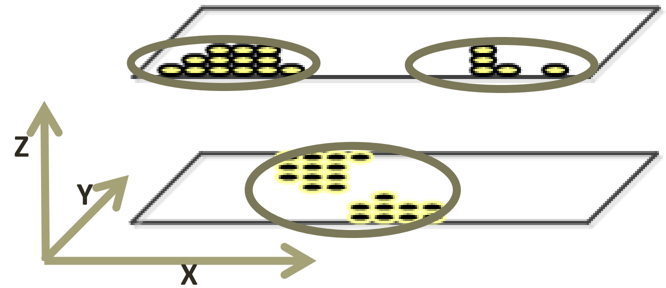


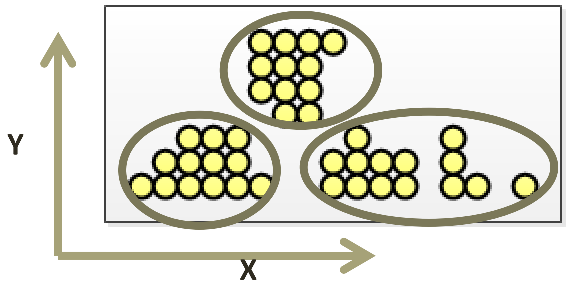


If the 3rd is small, then the clustering is the same as in 2D

Clustering: Dimensions (14)







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

Dimensions in Clustering

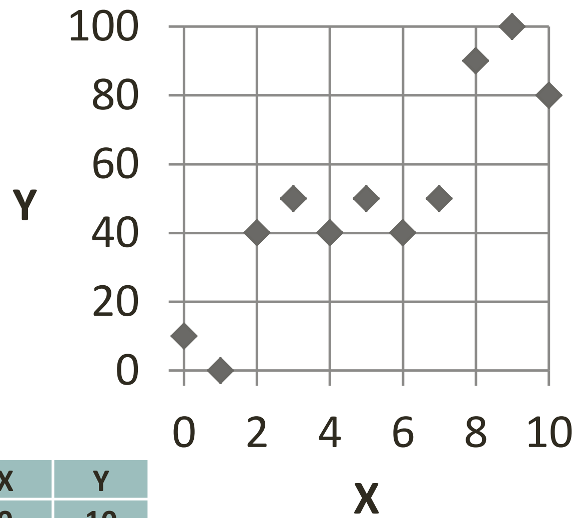
Break

Normalization in Clustering

Normalization of a linear relationship (1)

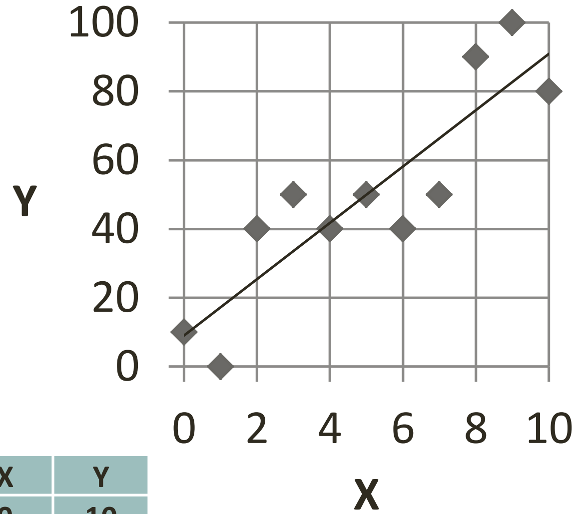
X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

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

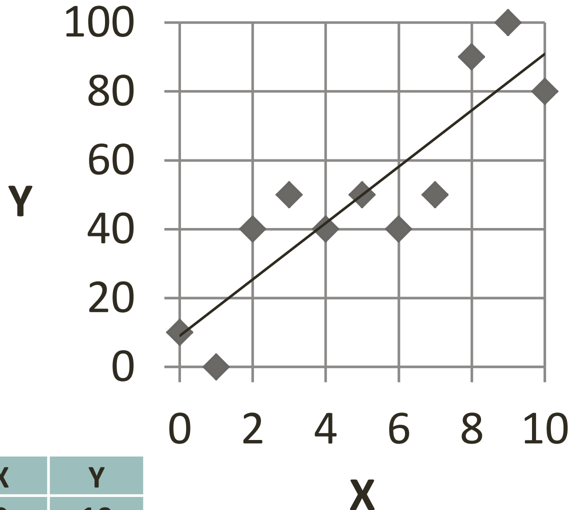
Normalization of a linear relationship (3)



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

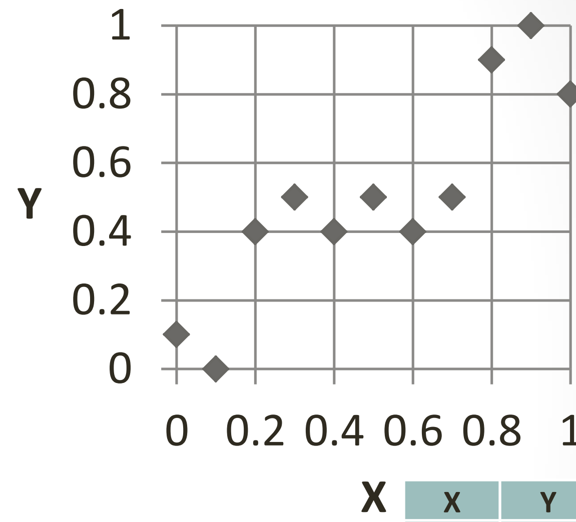
Normalization of a linear relationship (4)



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

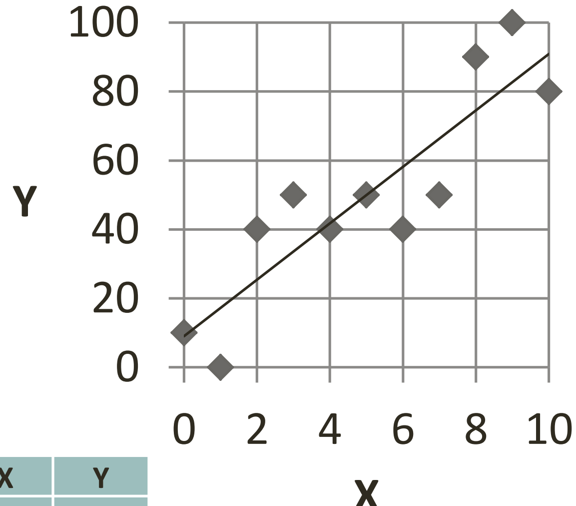
Y = 10 + 8*X

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

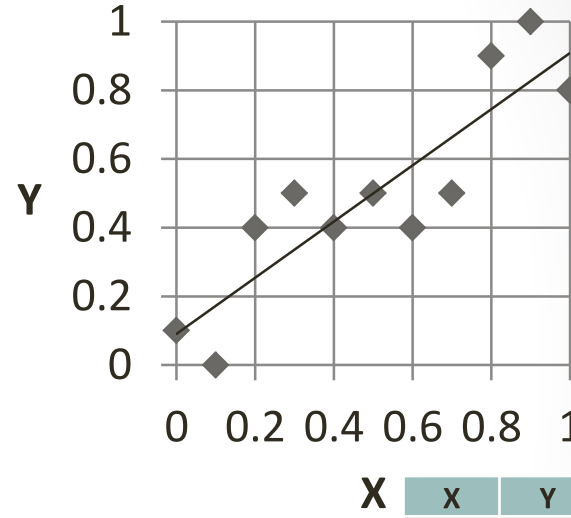
Normalization of a linear relationship (5)



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

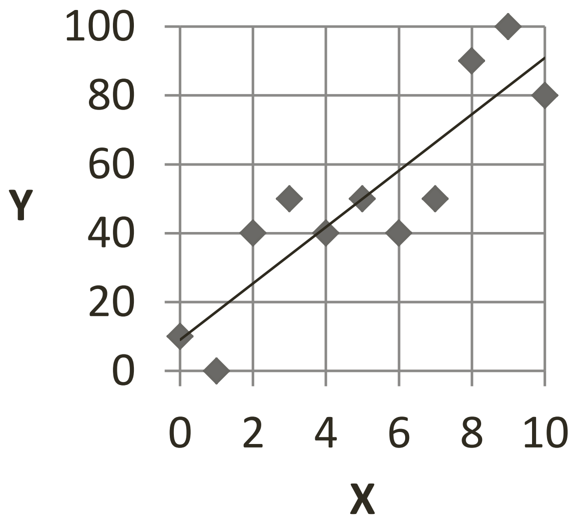
Normalize



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

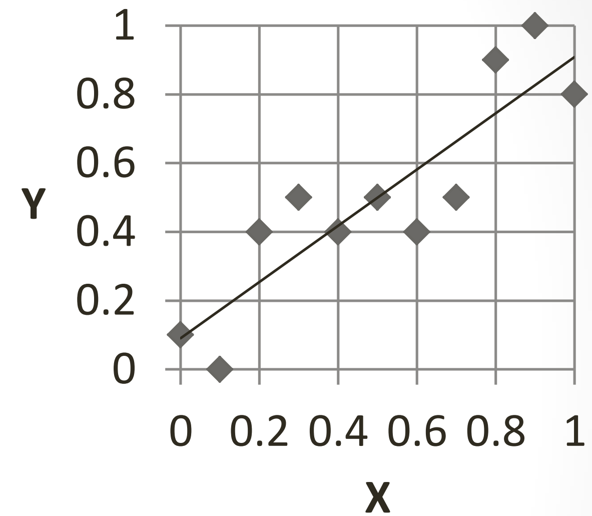
Y = 0.1 + 0.8*X

Normalization of a linear relationship (6)



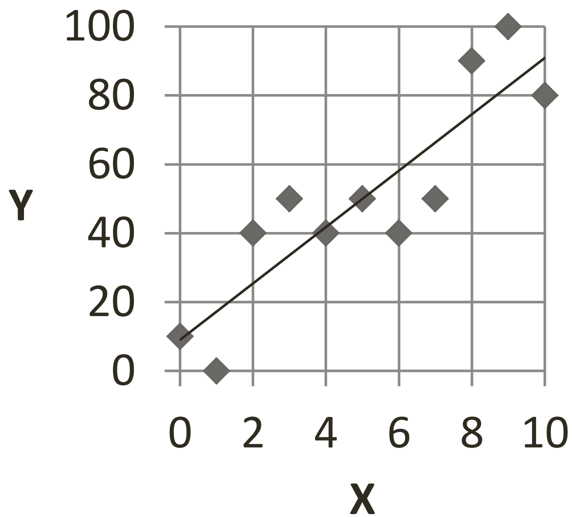
Y = 10 + 8*X

Normalize



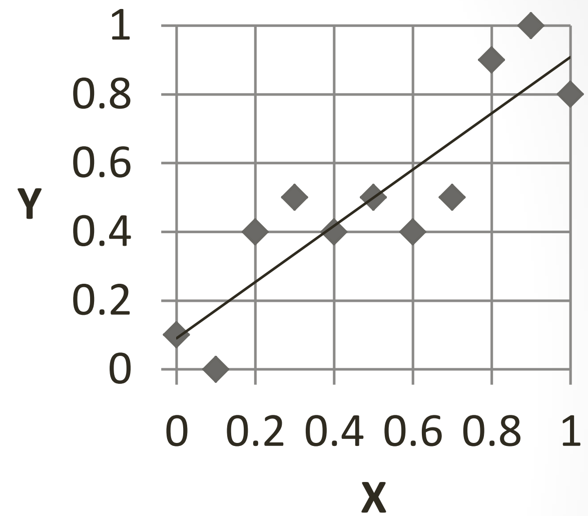
Y = 0.1 + 0.8*X

Normalization of a linear relationship (7)



Y = 10 + 8*X

Normalize



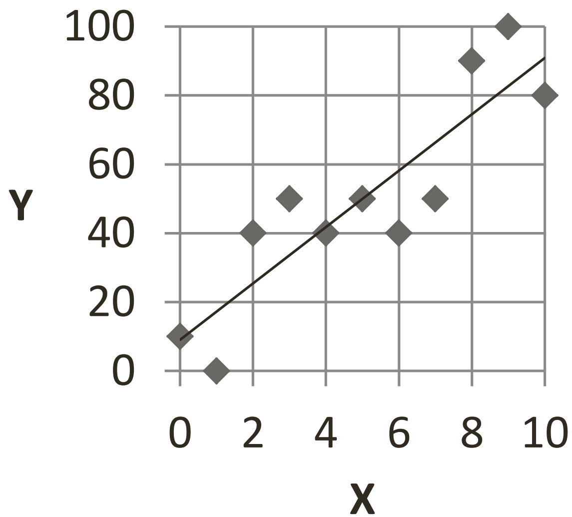
Y = 0.1 + 0.8*X

Normalize Input
X = 2 -> X’ = 0.2

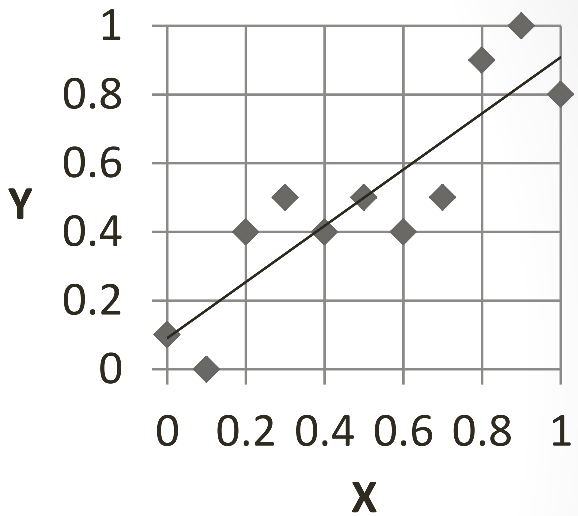
Predict Output
X’ = 0.2 -> Y’= 0.26

Denormalize Output
Y’= 0.26 -> Y = 26

Normalization of a linear relationship (8)



Normalize



Y = 10 + 8*X

Y = 0.1 + 0.8*X

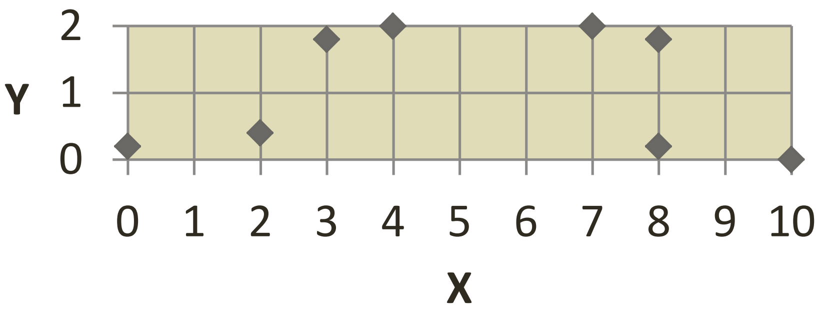
Normalize Input
X = 2 -> X' = 0.2

Predict Output
X' = 0.2 -> Y'= 0.26

Denormalize Output
Y'= 0.26 -> Y = 26

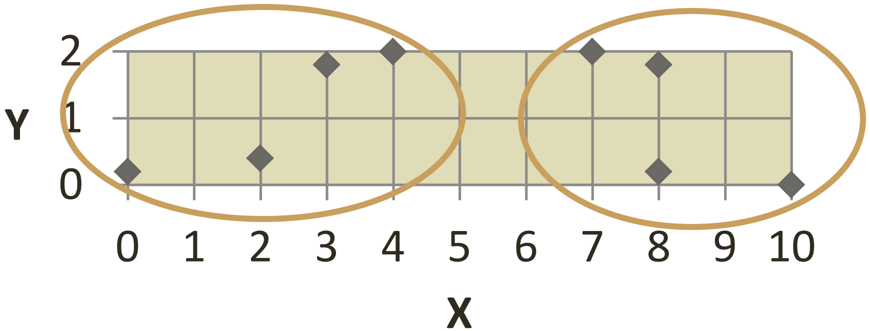
Prediction in Original Space:
X = 2 -> Y = 26

Normalization of a non-linear relationship (1)



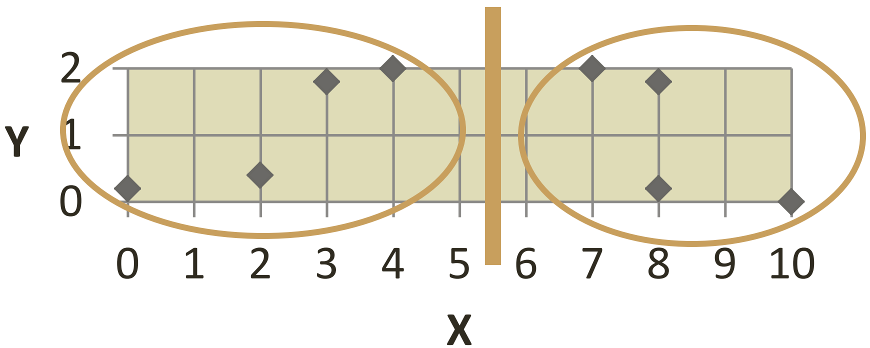
Original data in 2D:
Find 2 clusters

Normalization of a non-linear relationship (2)



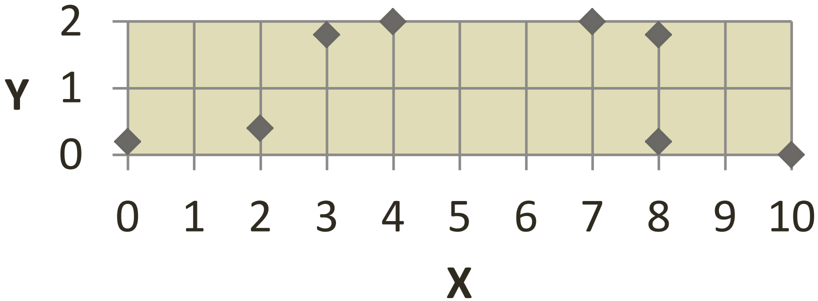
Found 2 Clusters

Normalization of a non-linear relationship (3)



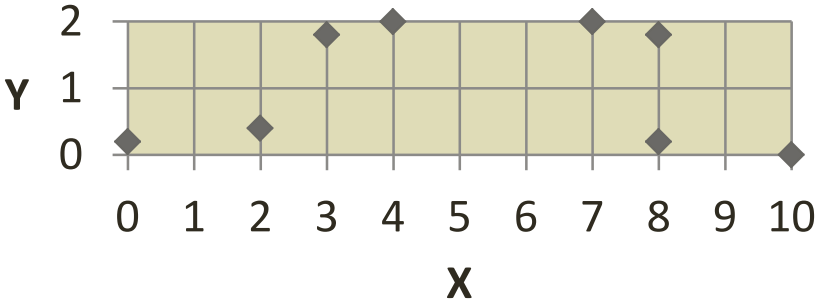
Clusters segment the image

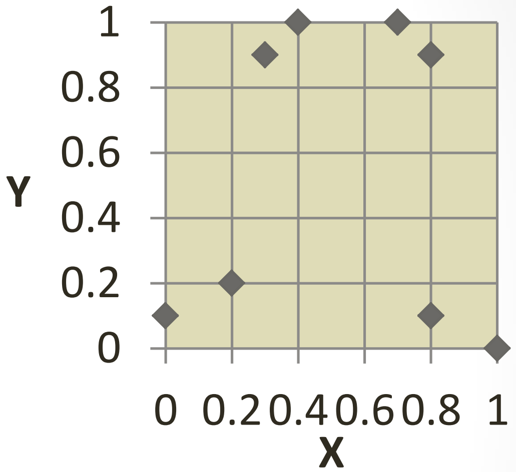
Normalization of a non-linear relationship (4)



Non-normalized 2D data

Normalization of a non-linear relationship (5)

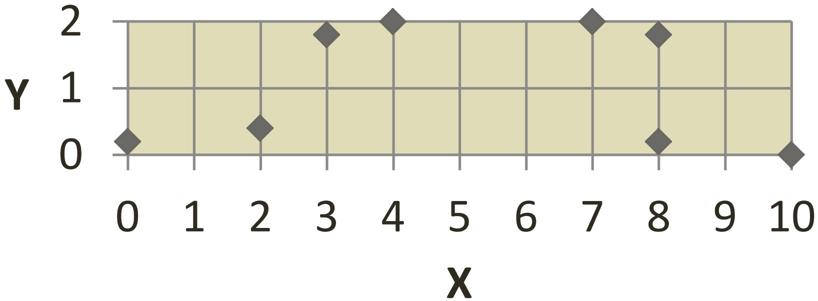




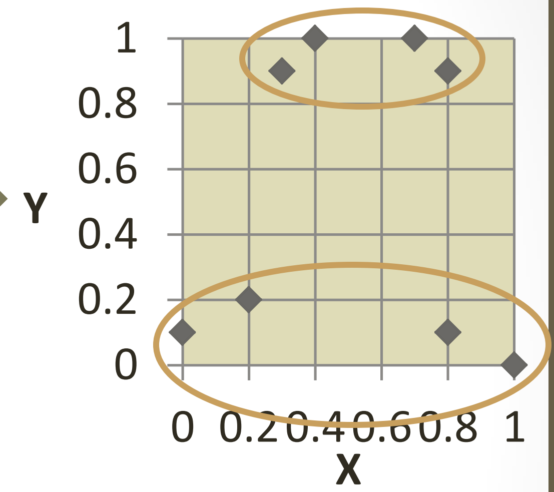
Non-normalized 2D data

Normalize the data:
Search for 2 Clusters

Normalization of a non-linear relationship (6)



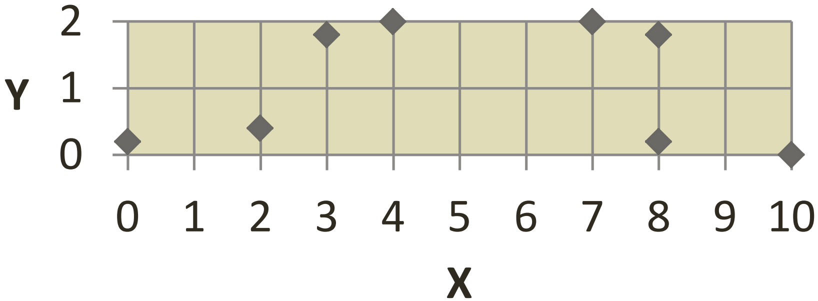
Normalize



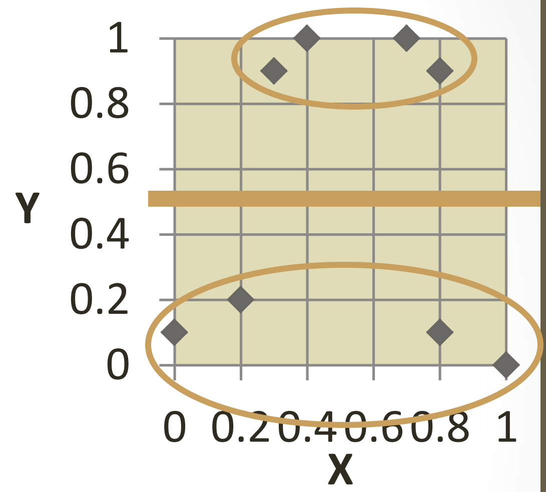
Non-normalized 2D data

Found 2 Clusters in the normalized data

Normalization of a non-linear relationship (6)



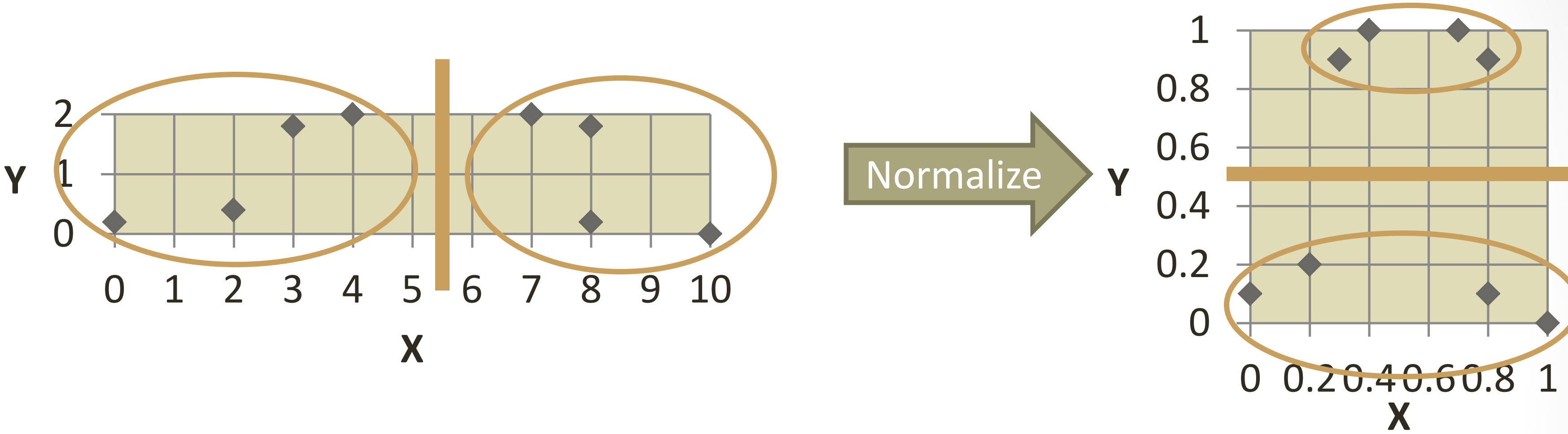
Normalize



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

Assignment (0)

- All assignment items from all assignment slides are due by Saturday 11:57.

Assignment (1)

1. Download KMeansIncomplete.R, KMeansHelper.R, TestObservations.csv, KMeansNormTest.R, and KMeansNorm.R from Canvas. KMeansIncomplete.R and KMeansNorm.R in Canvas are incomplete.
2. Complete the function KMeans in KMeansIncomplete.R and rename KMeansIncomplete.R to KMeans.R. Do not submit KMeans.R
3. Complete the function KMeansNorm in KMeansNorm.R by adding code to z-normalize the input points and centroids and to de-normalize the output centroids. In the function KMeansNorm, each dimension can be individually normalized/de-normalized.
   a. Get mean and standard deviation of point dimensions. Use the mean and sd functions
   b. Z-Normalize points and centroid guesses based on distribution of points
   c. Let the KMeans function in Kmeans.R determine the centroids in normalized space
   d. De-normalize the centroids and return the de-normalized centroids
4. Run KMeansNormTest.R with TestObservations and TestCenters. Do not submit these scripts or their results. Answer the following questions and add those answers as comments at the bottom of the completed KMeansNorm.R. Label each answer with its assignment item (4a etc)
   a) What is the single most obvious difference between the distributions of the first and second dimensions?
   b) Does clustering in Test 1 occur along one or two dimensions? Which dimensions? Why?
   c) Does clustering in Test 2 occur along one or two dimensions? Which dimensions? Why?
   d) Does clustering in Test 3 occur along one or two dimensions? Which dimensions? Why?
   e) Does clustering in Test 4 occur along one or two dimensions? Which dimensions? Why?

Assignment (2)

5. Why is normalization important in K-means clustering? Add answer as a comment to bottom of the completed KMeansNorm.R. Label the answer with its assignment item
6. How do you encode categorical data in a K-means clustering? Add answer as a comment to bottom of the completed KMeansNorm.R. Label the answer with its assignment item
7. Why is clustering un-supervised learning as opposed to supervised learning? Add answer as a comment to bottom of the completed KMeansNorm.R. Label the answer with its assignment item
8. Submit only the KMeansNorm.R, which also contains the answers to the above questions. Submit to Canvas by Saturday 11:57 PM.
9. Start a discussion, or make a comment on an existing discussion in the LinkedIn group.
10. Reading Assignment: See this week's and last week's preview section

Introduction to Data Science