Introduction to Data Science

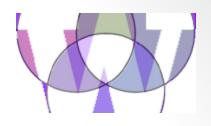
Lecture 03; April 13th, 2015

Ernst Henle

ErnstHe@UW.edu

Skype: ernst.predixion

Agenda



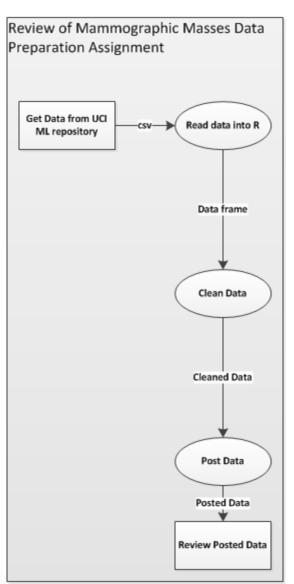
- Social Interactions
 - LinkedIn
 - Encourage Group Homework
- Review
- Quiz on Data Preparation
- Introduction to K-means Clustering
- Break
- Dimensions in Clustering
- Normalization (Clustering vs Linear Regression)
- Break
- Introduction to MATLAB/Octave

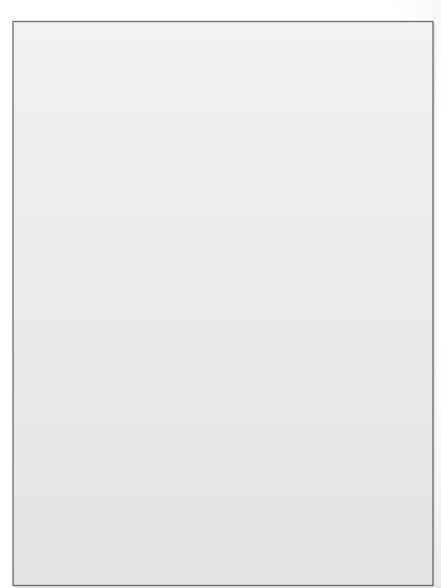
Review

- Optional Class in R
 - DataScience02e.R
- Data Preparation
 - Homework (DataScience02Homework.R)
 - DataScience02f.R (Quiz 03a)
 - Data Preparation Review

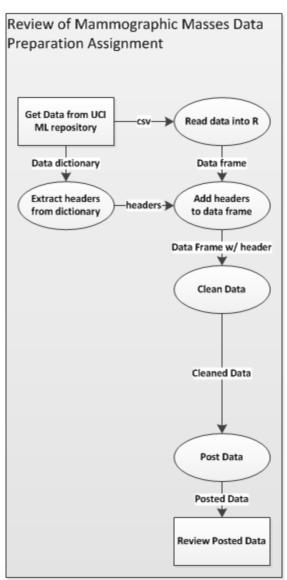
Data Preparation Review

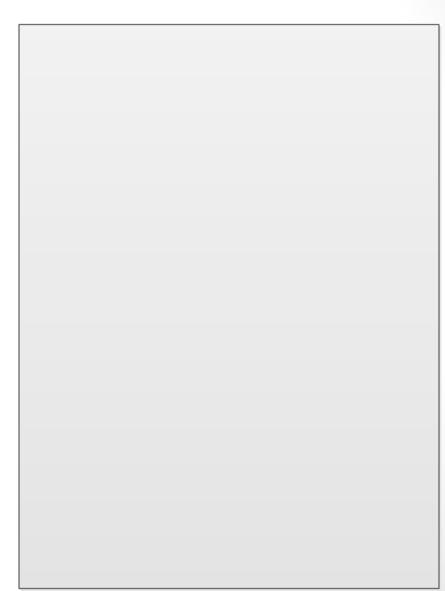
Data Preparation Review (1)



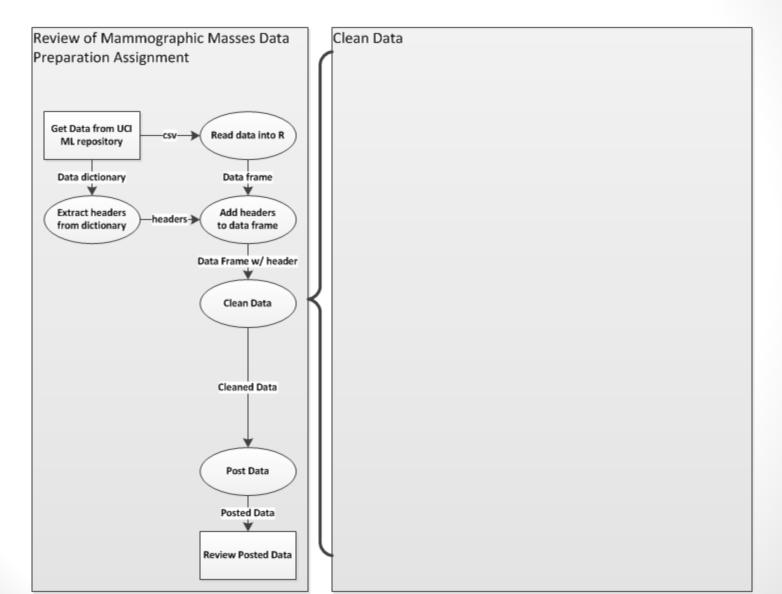


Data Preparation Review (2)

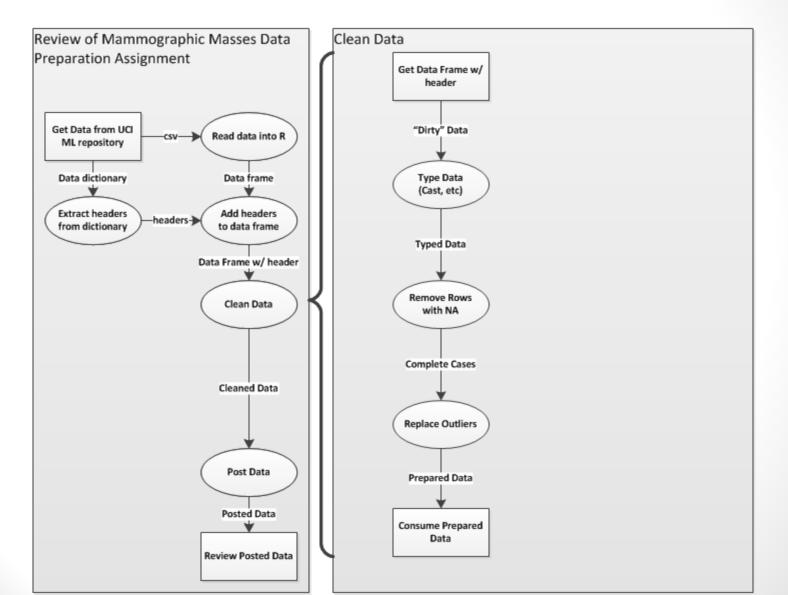




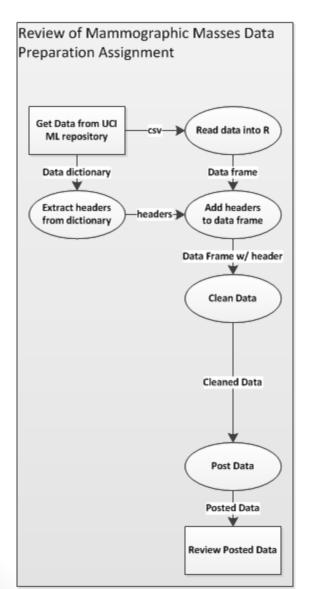
Data Preparation Review (3)

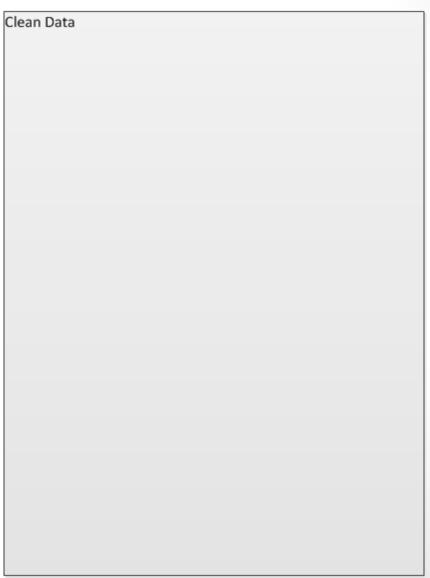


Data Preparation Review (4)

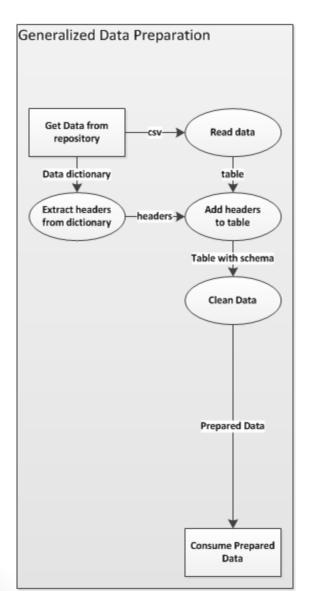


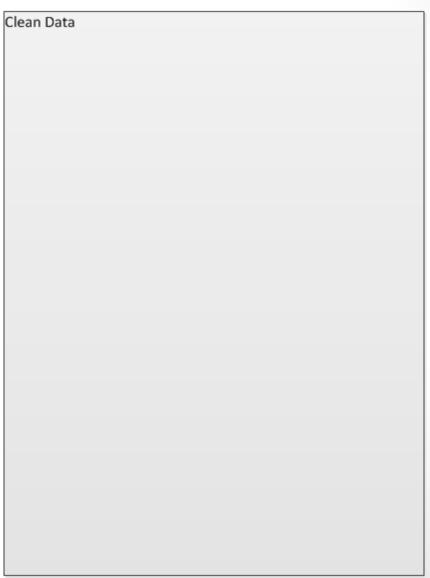
Data Preparation Review (5)



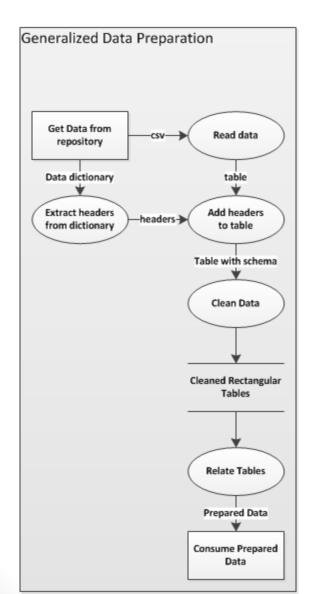


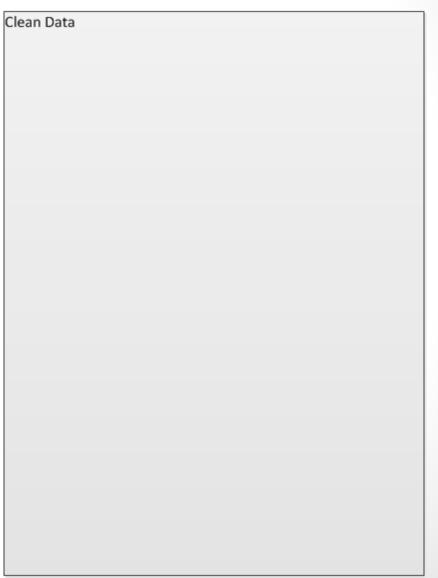
Data Preparation Review (6)



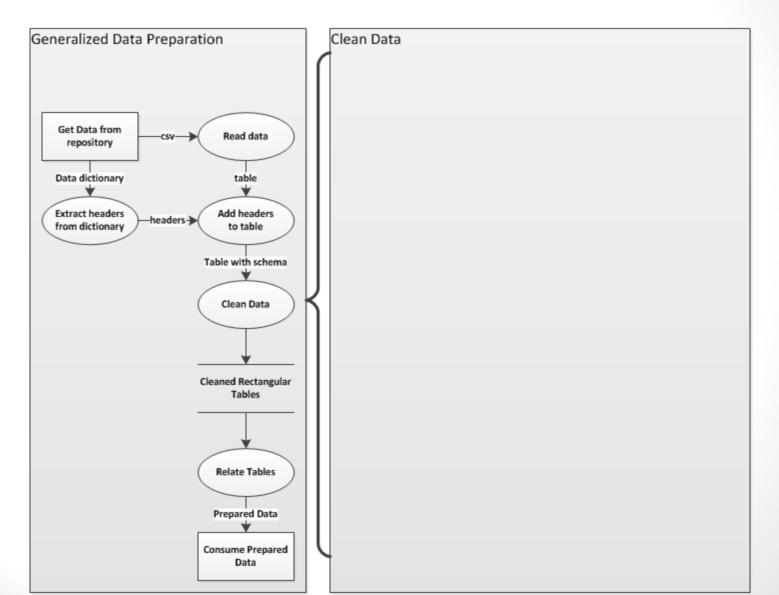


Data Preparation Review (7)

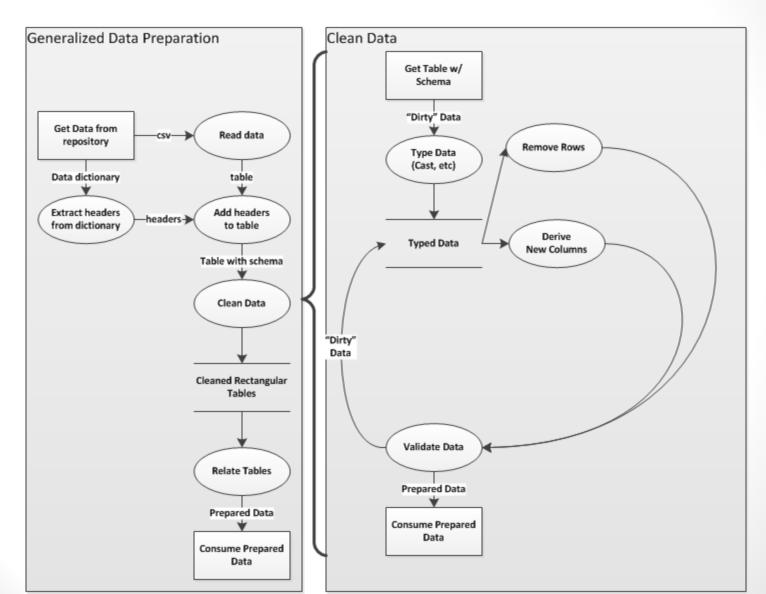




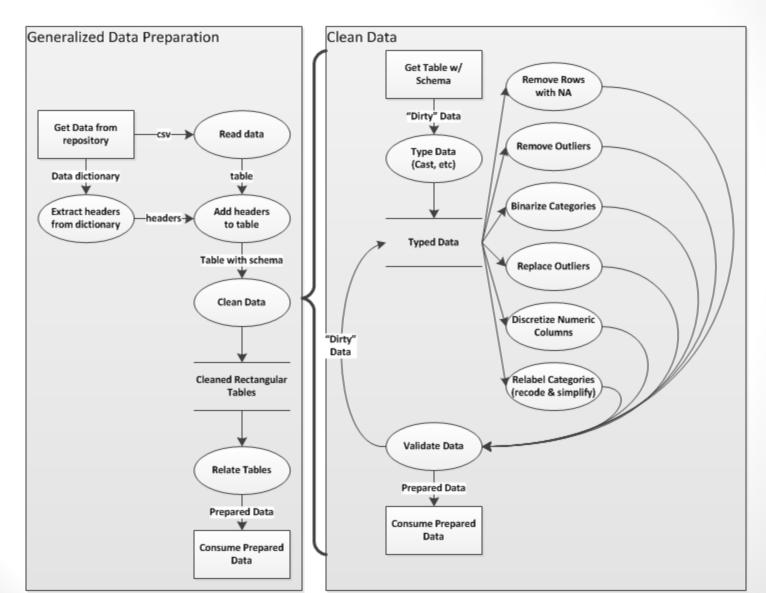
Data Preparation Review (8)



Data Preparation Review (9)



Data Preparation Review (10)



Data Preparation Review

Quiz 03b

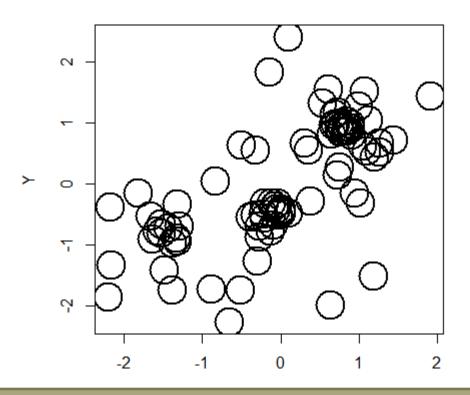
- Data Science UW 2015 Quiz 03b
- https://catalyst.uw.edu/webq/survey/ernsthe/267663

Introduction to K-means Clustering

K-means clustering: Algorithm

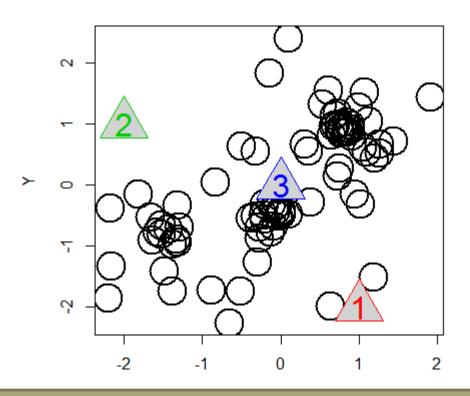
- Pre-requisites
 - 1. Get points in multi-dimensional space.
 - table, matrix, rectangular dataset
 - 2. Specify the number of clusters
 - Weakest point in algorithm (makes algorithm non-deterministic)
 - 3. Get a random center for each cluster
 - Another weak point in the algorithm
- Repeat until convergence:
 - 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)



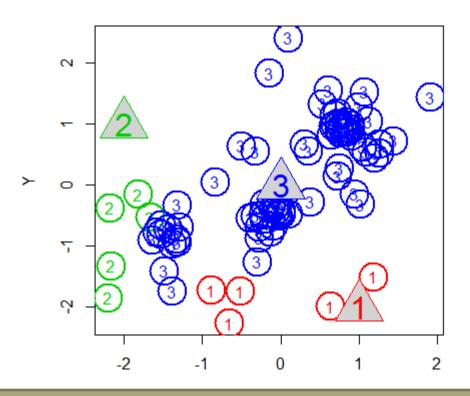
- Clustering starts by getting the data and representing the data as points in space. In this example the space is 2-dimensional.
- Each point describes an observation. An observation is an individual item.
- The dimensions are attributes that describe the item.

K-Means Clustering (1)



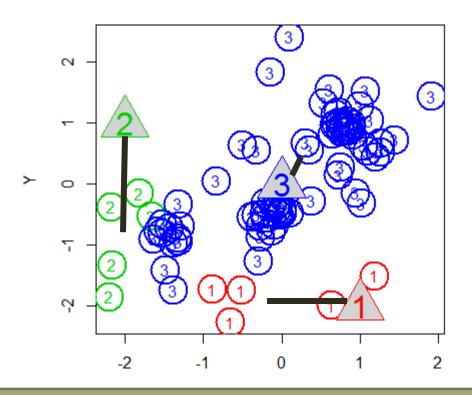
- Clustering continues by guessing, presuming, or specifying a number of clusters.
- Each centroid represents a cluster.
- The centroid positions are determined randomly. The centroids should be within the bounds of the points.

K-Means Clustering (2)



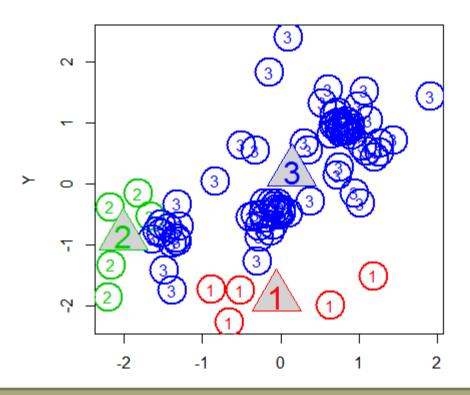
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (2)



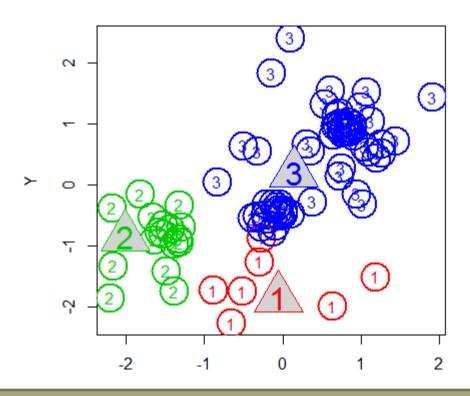
• Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (3)



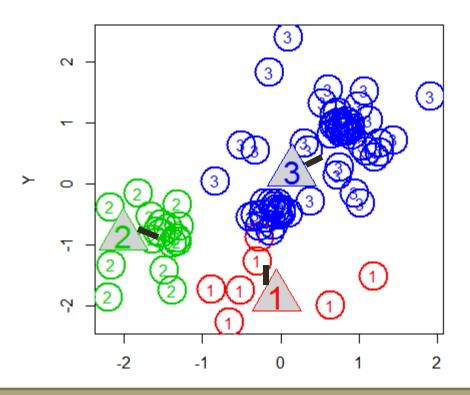
Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (4)



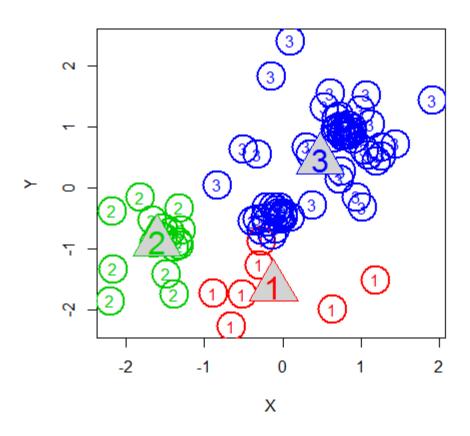
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (4)

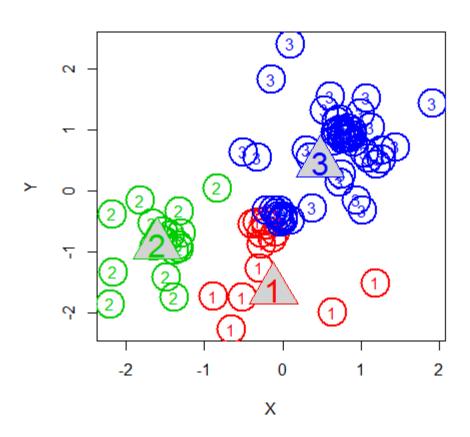


- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

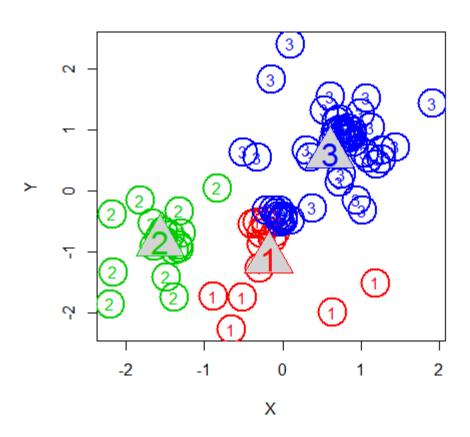
K-Means Clustering (5)



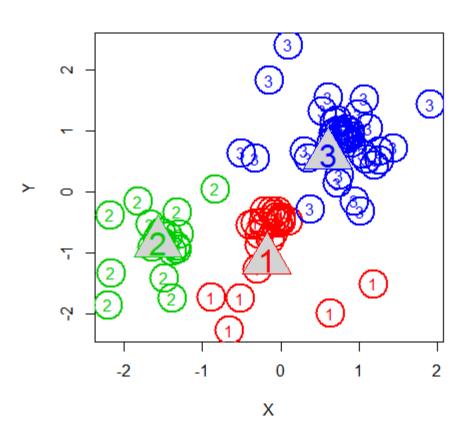
K-Means Clustering (6)



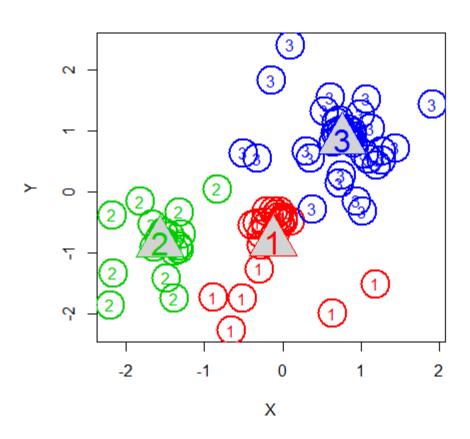
K-Means Clustering (7)



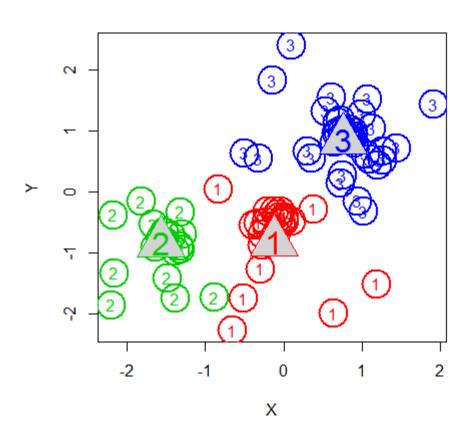
K-Means Clustering (8)



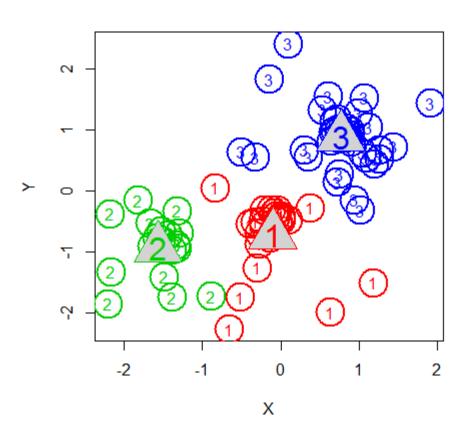
K-Means Clustering (9)



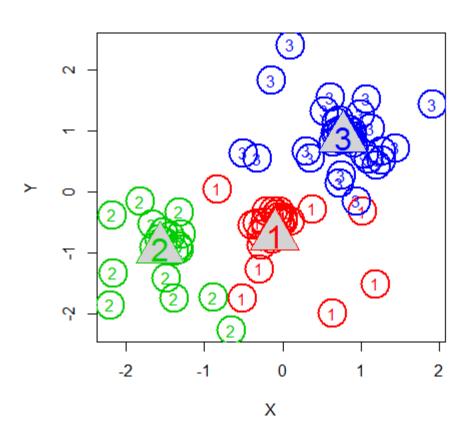
K-Means Clustering (10)



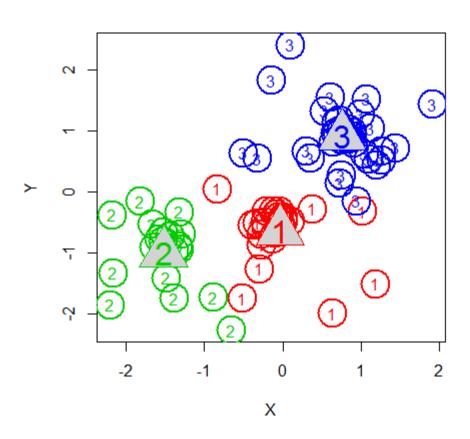
K-Means Clustering (11)



K-Means Clustering (12)



K-Means Clustering (13)



K-means Demo

K-Means Clustering step-by-step Clustering of Patients

K-means

Some Points:

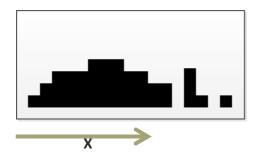
- Normalizations are important to put data on equal terms
- Initial centroid number and placement is an art.
- Categorical Data must be binarized
- K-means is unsupervised because we do not tell the algorithm what outcome was observed or what outcome is desired.

Break

Introduction to K-means Clustering

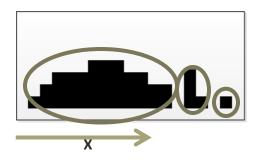
Dimensions in Clustering

Clustering: Dimensions (1)



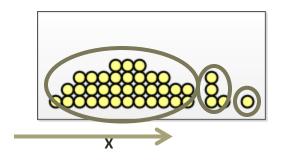
Where are the three clusters?

Clustering: Dimensions (2)



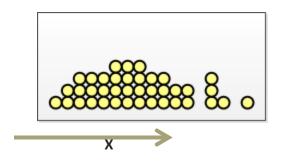
Simple assignment based on a 1D distribution

Clustering: Dimensions (3)



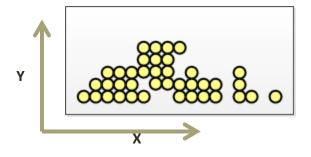
Simple assignment based on a 1D distribution

Clustering: Dimensions (4)



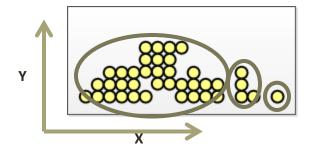
What if this was not a 1D distribution?

Clustering: Dimensions (5)



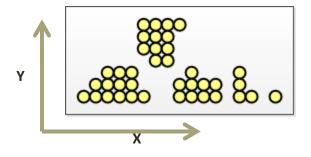
The distribution is in 2D. Some points differ in the 2nd D

Clustering: Dimensions (6)



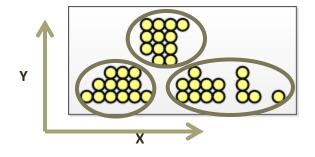
If the difference is minor, we still get the same clusters

Clustering: Dimensions (7)



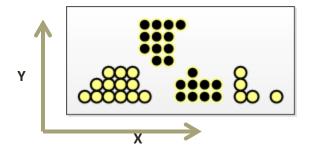
The difference could be significant

Clustering: Dimensions (8)



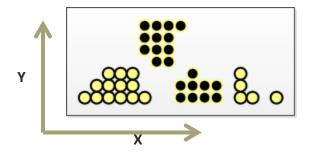
A big difference in the 2nd D can lead to different clusters

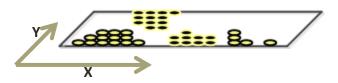
Clustering: Dimensions (9)



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

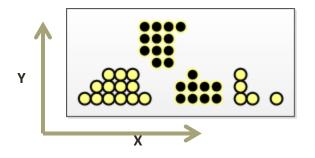
Clustering: Dimensions (10)

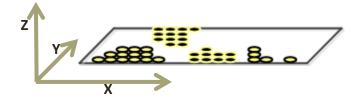




Create a 3rd
Dimansion

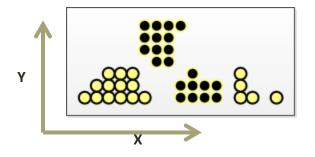
Clustering: Dimensions (11)

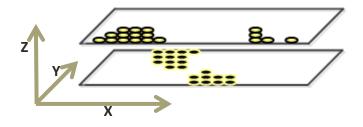




Create a 3rd
Dimansion

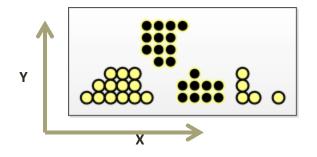
Clustering: Dimensions (12)

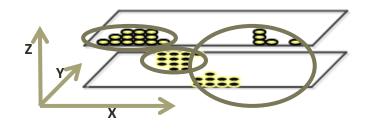


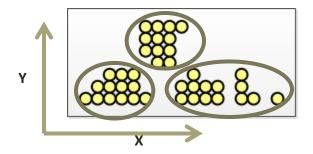


Where are the 3 clusters now?

Clustering: Dimensions (13)

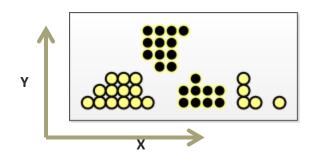


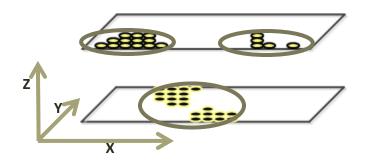


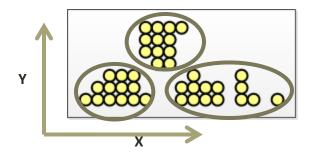


If the 3rd is small, then the clustering is the same as in 2D

Clustering: Dimensions (14)







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

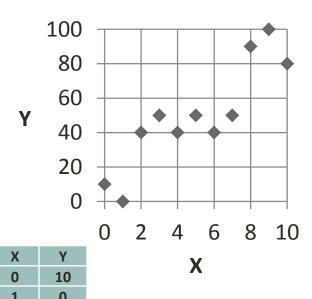
Dimensions in Clustering

Normalization in Clustering

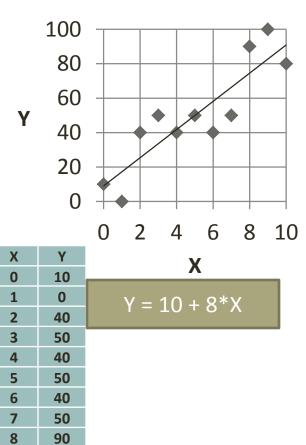
Normalization of a linear relationship (1)

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

Normalization of a linear relationship (2)

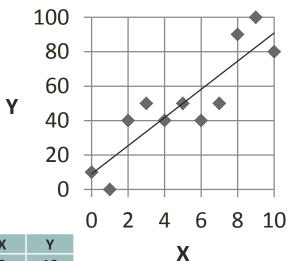


Normalization of a linear relationship (3)

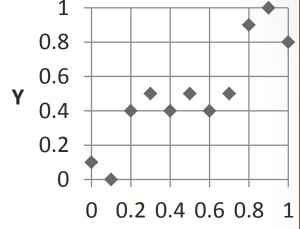


100

Normalization of a linear relationship (4)







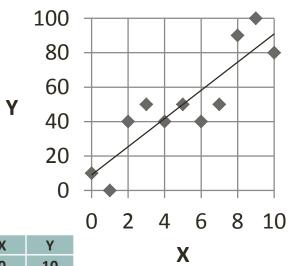
X	Υ
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100

80

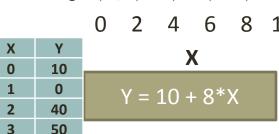


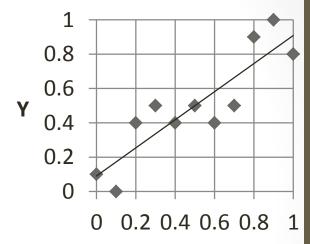
X	X	Υ
	0	0.1
	0.1	0
	0.2	0.4
	0.3	0.5
	0.4	0.4
	0.5	0.5
	0.6	0.4
	0.7	0.5
	0.8	0.9
	0.9	1
	1	0.0

Normalization of a linear relationship (5)







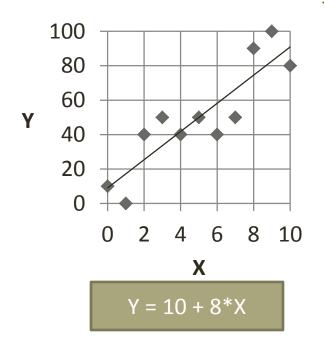


X

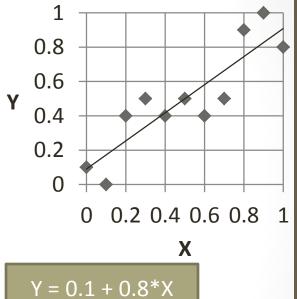
V _	0.1		0.8*X
$\mathbf{Y} = \mathbf{Y}$	\cup . \perp	+	U.O A

Χ	Υ
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

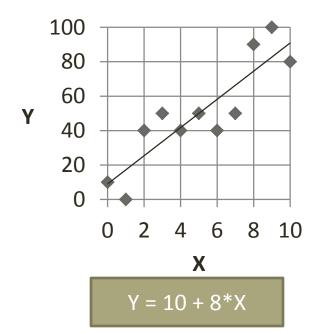
Normalization of a linear relationship (6)







Normalization of a linear relationship (7)

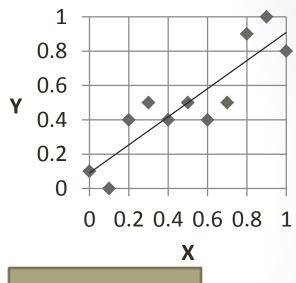


Normalize

Normalize Input $X = 2 \rightarrow X' = 0.2$

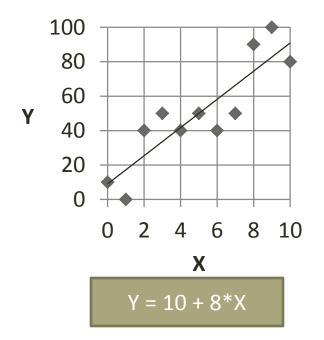
Predict Output X' = 0.2 -> Y' = 0.26

Denormalize Output $Y' = 0.26 \rightarrow Y = 26$



Y = 0.1 + 0.8 * X

Normalization of a linear relationship (8)

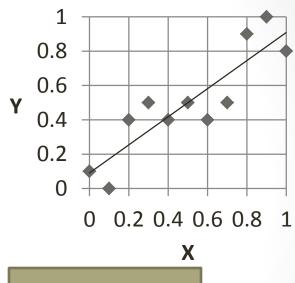


Normalize

Normalize Input $X = 2 \rightarrow X' = 0.2$

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

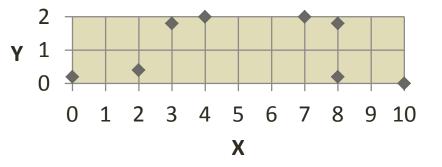
Denormalize Output Y' = 0.26 -> Y = 26



Y = 0.1 + 0.8 * X

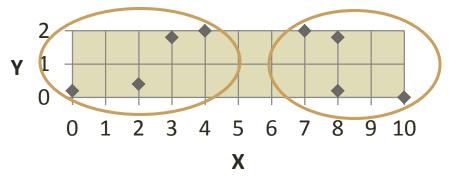
Prediction in Original Space: X = 2 -> Y = 26

Normalization of a non-linear relationship (1)



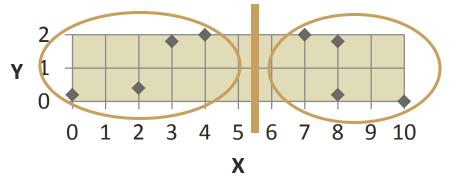
Original data in 2D: Find 2 clusters

Normalization of a non-linear relationship (2)



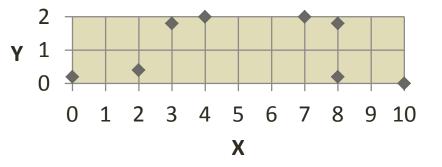
Found 2 Clusters

Normalization of a non-linear relationship (3)



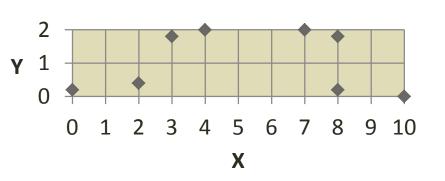
Clusters segment the image

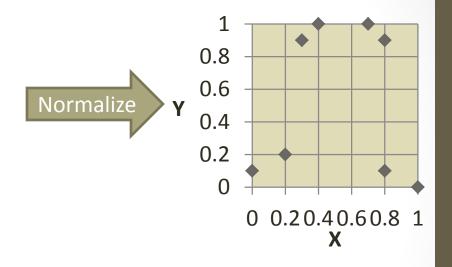
Normalization of a non-linear relationship (4)



Non-normalized 2D data

Normalization of a non-linear relationship (5)

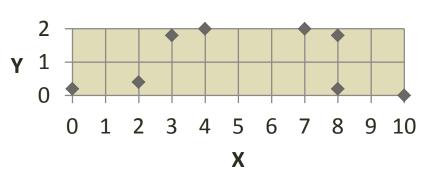


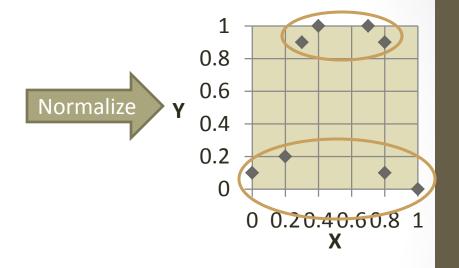


Non-normalized 2D data

Normalize the data: Search for 2 Clusters

Normalization of a non-linear relationship (6)

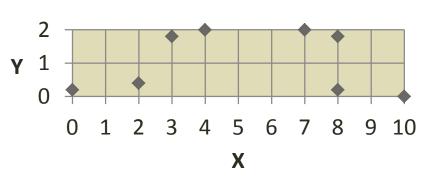


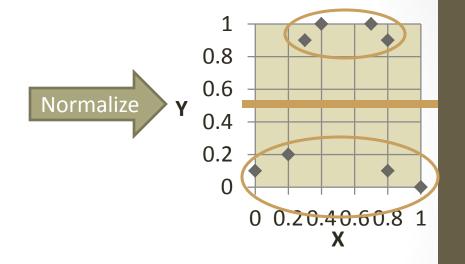


Non-normalized 2D data

Found 2 Clusters in the normalized data

Normalization of a non-linear relationship (6)

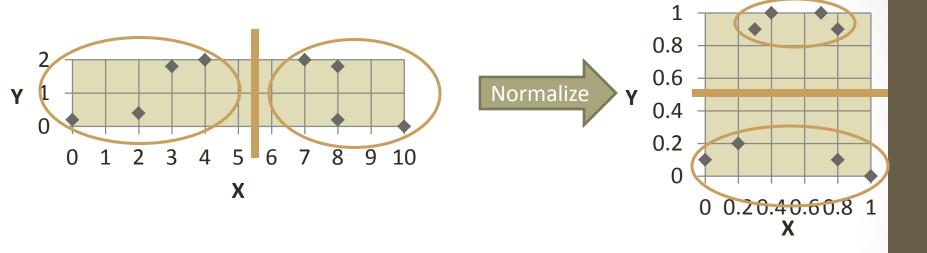




Non-normalized 2D data

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

Normalization in Clustering

Break

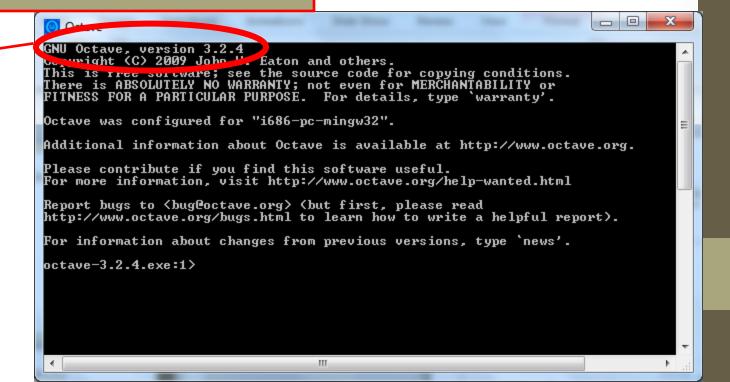
Introduction to Octave

Octave Console (0)

Start Octave



Octave console opens. Note version number



Octave Console (1)

```
% Assignments: Paste this code into the Octave console

a = 17; % simple assignment of a scalar

a % without the semicolon, the result appears in the console

a = [11, 19, 23]; % create a vector using[]

a

a(2) % index using ()

a = 'Hello World'; % assignment of characters

a % simplest Hello World

a(7:11) % 'Hello World ' is a character array and can be indexed

b = [a(7:9) a(11)] % assign and present some of the letters
```

```
octave-3.2.4.exe:1) % Assignments: Paste this code into the Octave console octave-3.2.4.exe:1) a = 17; % simple assignment of a scalar octave-3.2.4.exe:2) a % without the semicolon, the result appears in the console a = 17 octave-3.2.4.exe:3) a = [11, 19, 23]; % create a vector using[] octave-3.2.4.exe:4) a a =

11    19    23

octave-3.2.4.exe:5) a(2) % index using () ans = 19 octave-3.2.4.exe:6) a = 'Hello World'; % assignment of characters octave-3.2.4.exe:7) a % simplest Hello World a = Hello World octave-3.2.4.exe:8) a(7:11) % 'Hello World ' is a character array and can be indexed ans = World octave-3.2.4.exe:9) b = [a(7:9) a(11)] % assign and present some of the letters b = Word octave-3.2.4.exe:10)
```

Octave Console (2)

octave-3.2.4.exe:7>

a is now a

3 X 2 matrix

```
% Basic variables are matrices. Paste these lines into Octave
    a = 19;
    a
    size(a) % the result is the size in two dimensions
    a(3,2) = -7; % Assign a value in a new dimension
    a % Present the matrix
                                                           If the statement has no semi colon (;), then
    size(a) % the result is the size in two dimen
                                                                  the console will print the result
                     Octave
     a = 19
                    octave-3.2.4.exe:1> % Kasic variables are matrices.
                                                                         Paste these lines into Octave
                    octave-3.2.4.exe:1> a = 19;
Results in a 1 X
                           5.2.4.exe:3> size(a) % the result is the size in two dimensions
   1 matrix
                    octave-3.2.4.exe:4> a(3,2) = -7; % Assign a value in a new dimension octave-3.2.4.exe:5> a % Present the matrix
 Extending the
 matrix fills in
      zeros
                    octave—3.2.4.exe:6> size(a) % the result is the size in two dimensions
                    ans =
```

Octave Console (3)

% Distinguish between a row vector and a column vector.

```
a = [11, 19, 23];
```

size(a) % the result is the size in two dimensions

b = a' % Transpose the matrix

size(b) % the result is the size in two dimensions

```
a * b b * a
```

```
Octave
Ksh between a row vector and a column vector. Paste these lines into Octave
octave-3.2.4.exe:7> a = [11, 19, 23];
octave-3.2.4.exe:8> size(a) % the result is the size in two dimensions
      3
octave-3.2.4.exe:9> b = a' % Transpose the matrix
octave-3.2.4.exe:10> size(b) % the result is the size in two dimensions
ans =
octave-3.2.4.exe:11\rangle a * b
ans = 1011
octave-3.2.4.exe:12> b * a
ans =
octave-3.2.4.exe:13>
```

Octave Console (4)

```
% Plot a 1D matrix (aka vector)
     x = -3:0.1:3; % 1D matrix from -3 to 3
     z = x.*x; % note dot (.) for element-by-element operation
     y = \exp(-z);
     plot(x, y)
                                                                                                                               octave-3.2.4.exe:5> % Plot a 1D matrix (aka vector)
octave-3.2.4.exe:5> x = -3:0.1:3; % 1D matrix from -3 to 3
octave-3.2.4.exe:6> z = x.*x; % note dot (.) for element-by-element operation
octave-3.2.4.exe:7> y = exp(-z);
octave-3.2.4.exe:8> plot(x, y)
                        octave-3.2.4.exe:9>
                                                                                                                                    Figure 1
                                                                 8.0
                                                                 0.6
                                                                 0.4
                                                                 0.2
                                                            1.45379, 1.04956
```

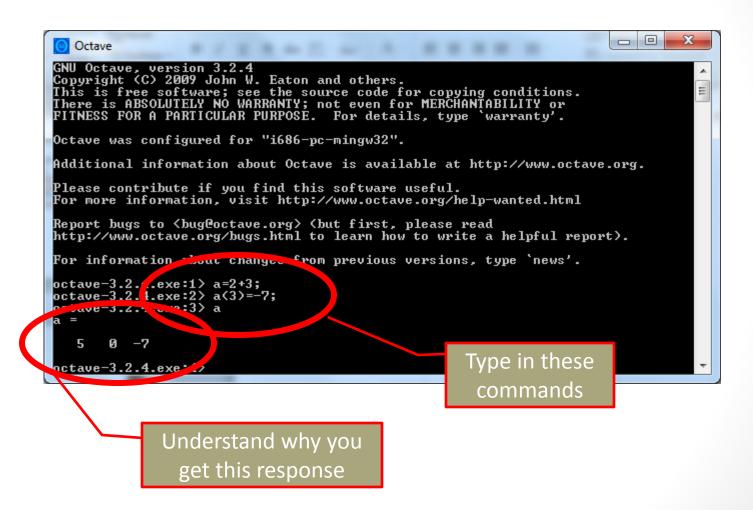
Binary operators in Octave

- Some common operator symbols are for use in matrix algebra and these operators do not do what many people expect.
 - '*' matrix multiplication; Use ".*" for element-by-element multiplication
 - '^' matrix power; Use".^" for element-by-element power
 - '/' matrix right division; Use "./" for element-by-element division

Octave m-files (0)

How to create and use m-files in Octave

Octave m-files (1): Start Octave and Test it

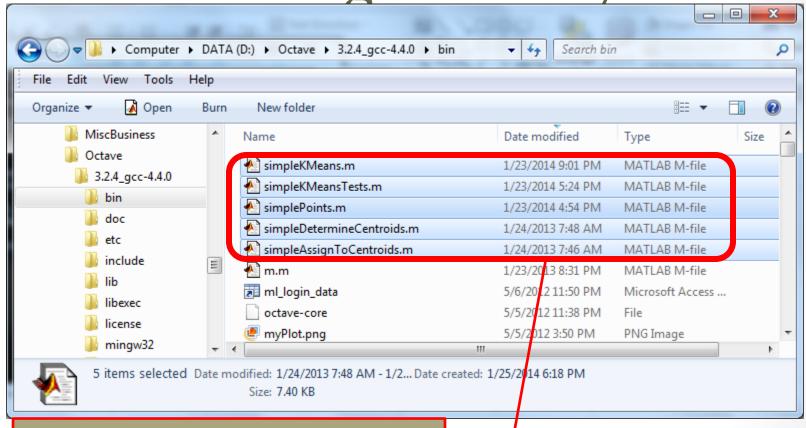


Octave m-files (2): Determine working directory

```
0
   Octave
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.
Octave was configured for "i686-pc-mingw32".
Additional information about Octave is available at http://www.octave.org.
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).
For information about changes from previous versions, type 'news'.
octave-3.2.4.exe:1> a=2+3;
octave-3.2.4.exe:2> a(3)=-7;
octave-3.2.4.exe:3> a
octave-3.2.4.exe:4> pwd
ans = D:\Octave\3.2.4_gcc-4.4.0\bin
octave-3.2.4.exe:5> _
```

Use the pwd command to determine your working directory! Here, my working directory is: D:\Octave\3.2.4_gcc-4.4.0\bin Your working directory may be different.

Octave m-files (3): Place m-files in Working Directory



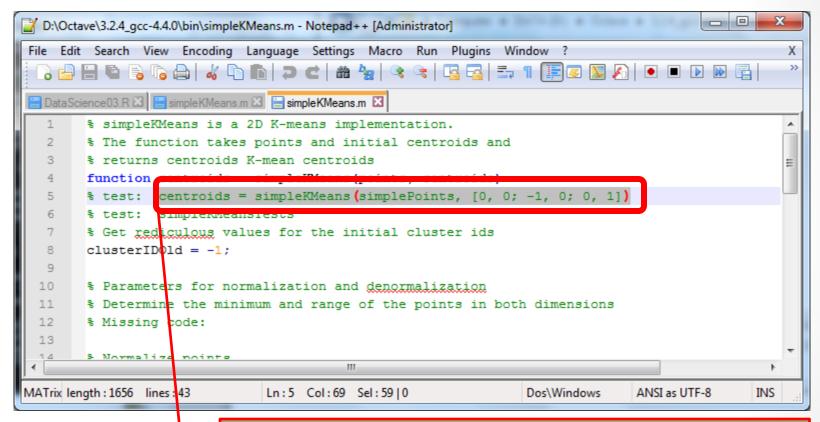
Take these 5 m-files from catalyst and place them in your working directory.

Octave m-files (4): Verify m-files in Working Directory

```
- 0
   Octave
Octave was configured for "i686-pc-mingw32".
Additional information about Octave is available at http://www.octave.org.
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).
For information about changes from previous versions, type 'news'.
octave-3.2.4.exe:1> a=2+3;
octave-3.2.4.exe:2> a(3)=-7;
octave-3.2.4.exe:3 a
octave-3.2.4.exe:4> pwd
  s = N:\^_\uve\3.2.4_gcc-4.4.v\uin
uve-3.2.4.exe:5> dir *.m
simpleAssignToCentroids.m
                            simpleKMeansTests.m
simpleDetermineCentroids.m simplePoints.m
simpleKMeans.m
```

Type in *dir* *.m into the Octave console. The m-files in your working directory will be listed. These m-files must include the m-files from catalyst.

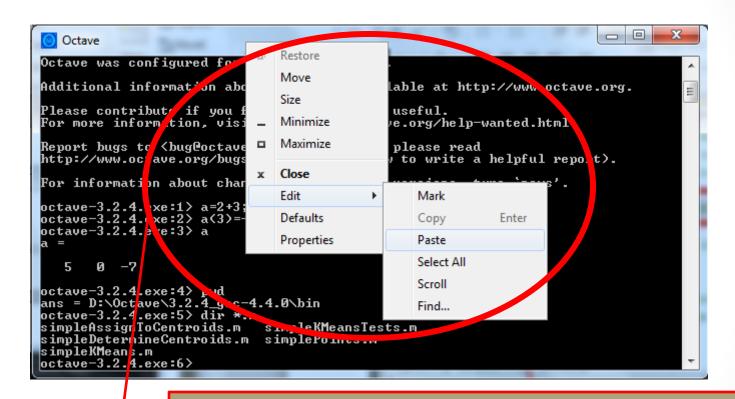
Octave m-files (5): Open m-file with editor



Use an editor, like Notepad++ to open simpleKMeans.m and then copy the following text:

centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])

Octave m-files (6): Paste command into Octave console



Paste the text from the clipboard into the Octave console.

Alternately, type in:

centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])

Octave m-files (7): Paste command into Octave console

```
Octave
Octave was configured for "i686-pc-mingw32".
Additional information about Octave is available at http://www.octave.org.
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug@octave.org> <but first, please read http://www.octave.org/bugs.html to learn how to write a helpful report>.
For information about changes from previous versions, type 'news'.
octave-3.2.4.exe:1> a=2+3;
octave-3.2.4.exe:2> a(3)=-7;
octave-3.2.4.exe:3 a
octave-3.2.4.exe:4> pwd
ans = D:\Octave\3.2.4_gcc-4.4.0\bin
octave-3.2.4.exe:5> dir *.m
                                  simpleKMeans.m
simpleAssignToCentraid
      DecermineCentroids.m simplePoints.m
Kds = simple KMeans(simple Points, [0, 0; -1, 0; 0, 1])_
```

You should see the text appear in the Octave console. If the text is too long for the console, you might not see the first part of the text. In that case, you can see the first part of the pasted text by using the left-arrow on your keyboard.

Octave m-files (8): Paste command into Octave console

```
Octave
Octave was configured for "i686-pc-mingw32".
Additional information about Octave is available at http://www.octave.org.
Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug@octave.org> <but first, please read http://www.octave.org/bugs.html to learn how to write a helpful report>.
For information about changes from previous versions, type 'news'.
octave-3.2.4.exe:1> a=2+3;
octave-3.2.4.exe:2> a(3)=-7;
octave-3.2.4.exe:3 a
octave-3.2.4.exe:4> pwd
ans = D:\Octave\3.2.4_gcc-4.4.0\bin
octave-3.2.4.exe:5> dir *.m
                                simpleKMeans.m
m.m
simpleAssignToCentroid
     octave-3.2.4.exe:6> centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
```

You should see the text appear in the Octave console. If the text is too long for the console, you might not see the first part of the text. In that case, you can see the first part of the pasted text by using the left-arrow on your keyboard.

Octave m-files (9): See results in Octave console

```
Report bugs to \( \text{bug@octave.org} \) \( \text{but first, please read http://www.octave.org/bugs.html to learn how to write a helpful report).} \)

For information about changes from previous versions, type \( \text{news'.} \)

octave=3.2.4.exe:1\( \text{a=2+3;} \)
octave=3.2.4.exe:2\( \text{a} \text{a} \text{a=7;} \)
octave=3.2.4.exe:3\( \text{a} \text{a} \)
a =

5 0 -7

octave=3.2.4.exe:4\( \text{pwd} \)
ans = D:\Octave\( \text{3.2.4.gcc-4.4.0} \)
bin
octave=3.2.4.exe:5\( \text{dir *.m} \)

simpleRssignToCentroids.m \( \text{simplePoints.m} \)
c.ave=3.2.4.exe:6\( \text{controids} \)
centroids =

-1.32500 -0.29000
-15.25789 -0.88368
9.09444 0.69194

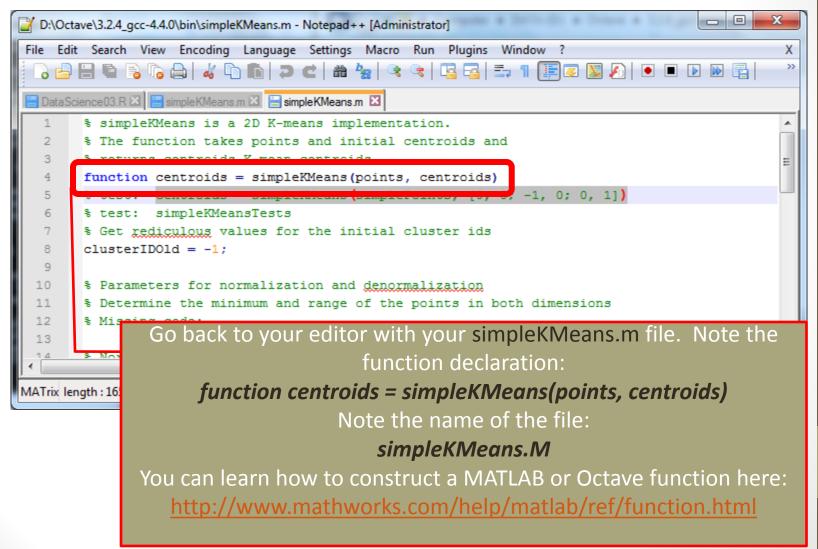
octave=3.2.4.exe:7\( \text{by} \)

octave=3.2.4.exe:7\( \text{by} \)
```

You see the results of the simpleKMeans method. The results are the centroids of the clusters.

For the fun of it, use different starting centroids: centroids = simpleKMeans(simplePoints, [?, ?; ?, ?])

Octave m-files (10): Function Structure



Octave m-files (11): Modify Octave Function

```
D:\Octave\3.2.4_gcc-4.4.0\bin\simpleKMeans.m - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
🖟 🕞 🔒 🖺 🖺 🥦 🥱 🚵 | 🔏 🐚 🖍 🦶 🖎 😅 🗥 🦢 | 🤏 👒 🖳 🚍 🚍 🖺 👭 💽 💌 🕖 | 🗨 🗷
 🗎 Data Science 03. R 🖾 📙 simple KMeans.m 🖾 📙 simple KMeans.m 🗵
        % simpleKMeans is a 2D K-means implementation.
        % The function takes points and initial centroids and
        % returns centroids K-mean centroids
        function centroids = simpleKMeans(points, centroids)
        % test: centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
        % test: simpleKMeansTests
        % Get rediculous values for the initial cluster ids
        clusterIDOld = -1;
   9
  10
        % Parameters for normalization and denormalization
  11
        % Determine the minimum and range of the points in both dimensions
       'This string is Test Code'
MATrix length: 1682 lines .43
                               Ln:13 Col:27 Sel:0|0
                                                                  Dos\Windows
                                                                                ANSI as UTF-8
                                                                                                INS
```

Make a change to the code by introducing a simple string. A good place where you can introduce a string is after the first occurrence of *Missing code:*. In this example, I introduced 'This string is Test Code' on line 13.

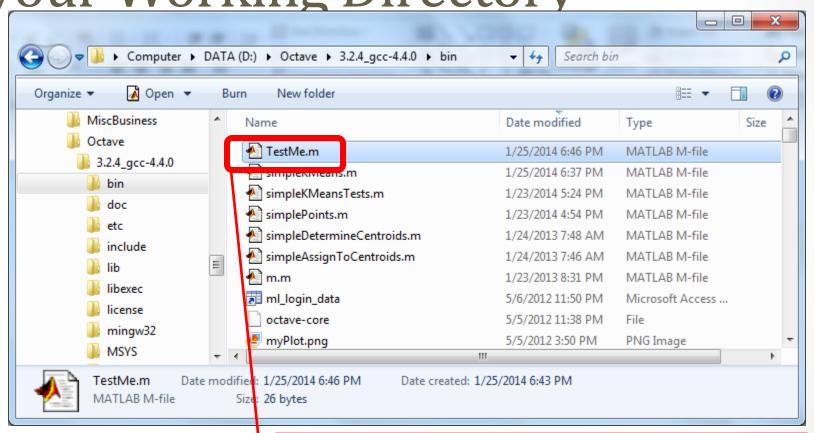
Octave m-files (12): Note the effects of your modification

```
Octave
octave-3.2.4.exe:4> pwd
ans = D:\Octave\3.2.4_gcc-4.4.0\bin
octave-3.2.4.exe:5 dir *.m
                               simpleKMeans.m
                              simpleKMeansTests.m
simpleAssignToCentroids.m
simpleDetermineCentroids.m simplePoints.m
octave-3.2.4.exe:6> centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
centroids =
   -1.32500
               -0.29000
               -0.88368
octave-3.2.4.exe:7> centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
ans = This string is Test Code
cen--
               -0.29000
   -1.32500
               -0.88368
    9.09444
                0.69194
octave-3.2.4.exe:8> _
```

Run simpleKMeans again. You now see the effect of your change to the code.

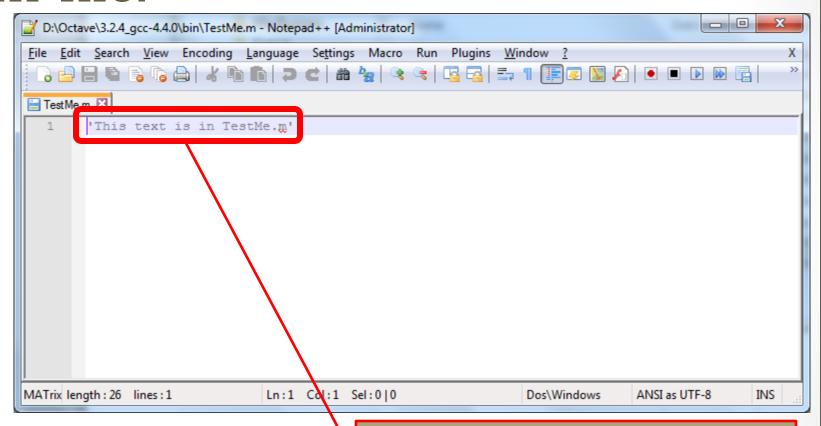
Octave (0) new m-file

Octave (1): Add an m-file to your Working Directory



Add an m-file to your working directory. For instance, I created a new text file and renamed it *TestMe.m*. This m-file has no text, yet.

Octave (2): Add a string to your m-file.



Open the new m-file in your working directory. Add a proper string like: 'This text is in TestMe.m' Save the m-file!

Octave (3): Run your m-file

```
Octave
octave-3.2.4.exe:4> pwd
ans = D:\Octave\3.2.4_gcc-4.4.0\bin
octave-3.2.4.exe:5> dir *.m
                            simpleKMeans.m
simpleAssignToCentroids.m
                            simpleKMeansTests.m
simpleDetermineCentroids.m simplePoints.m
octave-3.2.4.exe:6> centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1]>
centroids =
   -1.32500
              -0.29000
  -15.25789
              -0.88368
    9.09444
octave-3.2.4.exe:7> centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1]>
ans = This string is Test Code
centroids =
   -1.32500
              -0.29000
  -15.25789
              -0.88368
    9.09444
octave-3.2.4.exe:8> TestMe
ans = This text is in TestMe.m
 staue-3.2.4.exe:9>
```

Run TestMe.m by typing TestMe into the Octave console and hitting Enter. Note that console presents the string in TestMe.m.

Octave (4): Paste this Code into TestMe.m

```
% This is a comment
'The following line has no semicolon. You will see the result of b:'
b = 17 + 29
'The following line has a semicolon. You will see no output for c:'
c = b + 31;
'The following line executes simpleKMeans.m and outputs centroids:'
centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
'The following line presents slightly different centroids:'
centroids = simpleKMeans(simplePoints, [1, 1; -1, -1; -1, 1])
'The following line can be calculated if f is defined.'
d = c + f:
'This text and the following calculations are executed if there is no
error'
b = [1, b, c] + 5
```

Octave (5): Edit your m-file

```
*D:\Octave\3.2.4_gcc-4.4.0\bin\TestMe.m - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
TestMe.m
       % This is a comment
       'The following line has no semicolon. You will see the result of b:'
       b = 17 + 29
       'The following line has a semicolon. You will see no output for c:'
       c = b + 31:
       'The following line executes simpleKMeans.m and outputs centroids:'
  8
       centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
       'The following line presents slightly different centroids:'
  9
 10
       centroids = simpleKMeans(simplePoints, [1, 1; -1, -1; -1, 1])
 11
       'The following line has an error. Execution will stop. Think about the error message'
       d = c + f:
 13
       'This text and the following calculations are not executed because of the error above'
 14
       b = [1, b, c] + 5
  15
MATrix length: 650 lines: 15
                             Ln:15 Col:1 Sel:0|0
                                                                         ANSI as UTF-8
                                                            Dos\Windows
                                                                                       INS
```

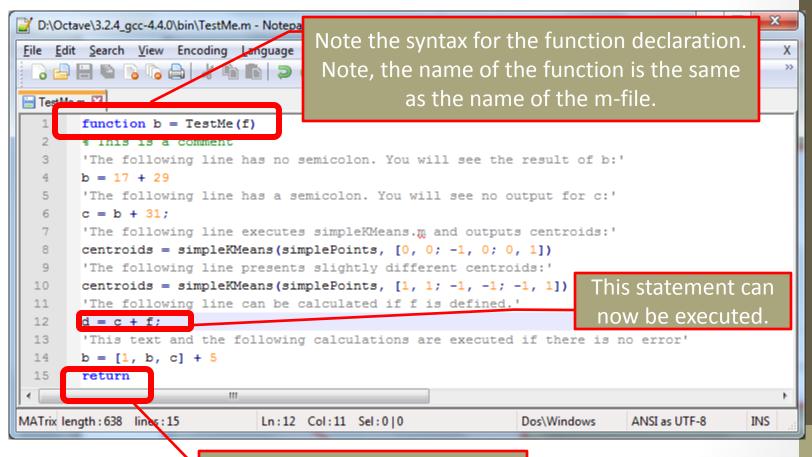
Edit and save your m-file. Pay attention to the text in the m-file and the console.

Octave (6): Run your m-file. Read the error code.

```
Octa
 cave-3.2.4.exe:1> TestMe
ans = The following line has no semicolon. You will see the result of b:
ans = The following line has a semicolon. You will see no output for c:
ans = The following line executes simpleKMeans.m and outputs centroids:
ans = This string is Test Code
centroids =
   -1.32500
              -0.29000
  -15.25789
              -0.88368
               0.69194
ans = The following line presents slightly different centroids:
ans = This string is Test Code
centroids =
    8.94865
               0.66568
  -15.63889
              -0.93556
ans = The following line has an error. Execution will stop. Think about the error message
error: `f' undefined near line 12 column 9
error: called from:
         D:\Octave\3.2.4_gcc-4.4.0\bin\TestMe.m at line 12, column 3
oc. ve-3.2.4.exe:1>
```

Run your m-file. Relate the console's output to the code. Read the error code.

Octave (7): Create a Function



The return statement is usually not necessary. But, I like it.

Octave (8): Relate Output to Code

```
oct = e-3.2.4.exe:2> q = TestMe(-7)
 s = The following line has no semicolon. You will see the result of b:
ans = The following line has a semicolon. You will see no output for c:
ans = The following line executes simpleKMeans.m and outputs centroids:
ans = This string is Test Code
centroids =
   -1.32500
              -0.29000
  -15.25789
              -0.88368
    9.09444
               0.69194
ans = The following line presents slightly different centroids:
ans = This string is Test Code
centroids =
    8.94865
               0.66568
              -Ø.93556
  -15.63889
   -1.75714
ans = The following line can be calculated if f is defined.
ans = This text and the following calculations are executed if there is no error
        51
             82
        51
             82
octave-5.2.4.exe:3>
```

Execute your function. Note that you can supply an input argument and a return value. Why do you see the return value twice? What happens if you do not supply a return value?

Octave (9): Matrix-Oriented Programming

MatrixOrientedProgrammining.m

```
MA = [1, 2, 3; 8, 9, 0];
numberOfRows = size(MA, 1);
meanMA = mean(MA);
% Loop
for rowNumber = 1:numberOfRows
       MAplusMean(rowNumber, :) = MA(rowNumber, :) + meanMA;
end % end the for loop
MAplusMean
% Vectorized approach
meanMAs = repmat(meanMA, numberOfRows, 1);
MAplusMean = MA + meanMAs
% Broadcasting in Octave 3.2.6
MAplusMean = MA + meanMA
```

Introduction to Octave

K-Means Normalization in MATLAB

- Create simpleKMeansFinished.m from simpleKMeans.m by adding a z-Score Normalization
 - Get mean of points. Use the mean function
 - Get standard deviation of points. Use the max function
 - Normalize points based on mean and standard deviation of points
 - Normalize centroids based on mean and standard deviation of points
 - Let the existing code determine the centroids in normalized space
 - De-normalize the centroids (Don't bother de-normalizing points)

Assignment

- 1. Answer these questions:
 - a. Why is normalization important in K-means clustering?
 - b. How do you encode categorical data in a K-means clustering?
 - c. Why is clustering un-supervised learning as opposed to supervised learning?
- 2. Given the following: simpleAssignToCentroids assigns the 17th point to a centroid by measuring the distance of the 17th point to each centroid. The centroid with the smallest distance to the 17th point is the point's centroid. How does simpleKMeans know which centroid was chosen for the 17th point? (Answer in one sentence or less by describing the data structure)
- 3. Given the following: simpleDetermineCentroids determines centroid for cluster 2 by finding the mean of all points that belong to cluster 2. How does simpleKMeans know which returned centroid is the one for cluster 2? (Answer in one sentence or less by describing the data structure)
- 4. Normalization in simpleKMeansFinished.m
 - 1. Copy simpleKMeans.m to simpleKMeansFinished.m
 - 2. Complete simpleKMeansFinished.m by adding code to normalize the inputs and to de-normalize the output. The directions on what to do is in the code and in the slide: K-Means Normalization in MATLAB
- 5. Write answers to item 1, 2, & 3 into the completed simpleKMeansFinished.m and submit to Catalyst by Saturday 11:00 PM.

Introduction to Data Science