Introduction to Data Science

Lecture 03; April 13th, 2015

Ernst Henle
ErnstHe@UW.edu
Skype: ernst.predixion

Agenda

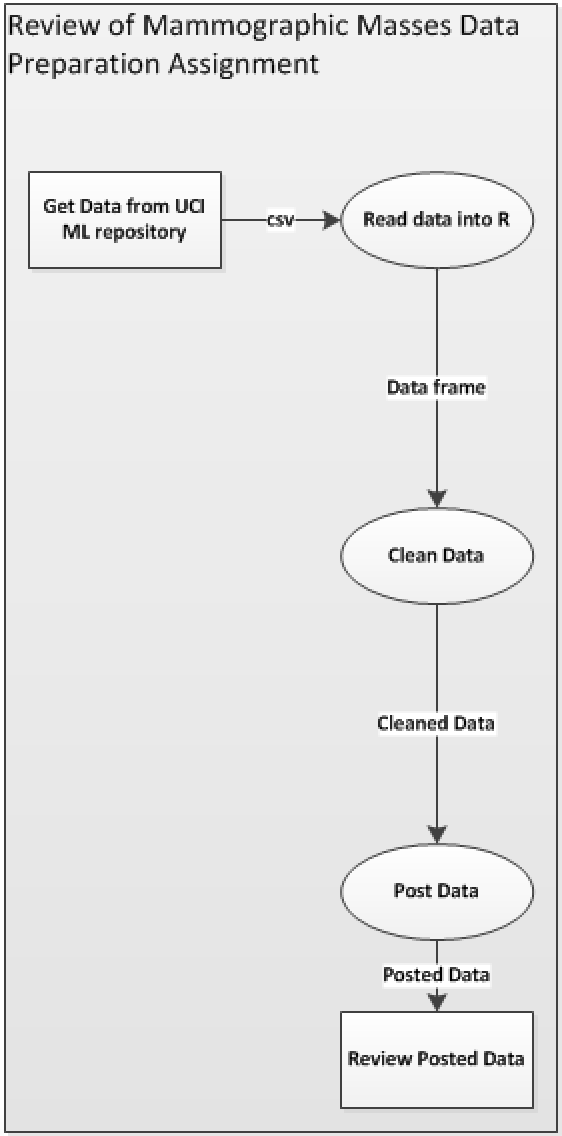
- Social Interactions
  - LinkedIn
  - Encourage Group Homework
- Review
- Quiz on Data Preparation
- Introduction to K-means Clustering
- Break
- Dimensions in Clustering
- Normalization (Clustering vs Linear Regression)
- Break
- Introduction to MATLAB/Octave

Review

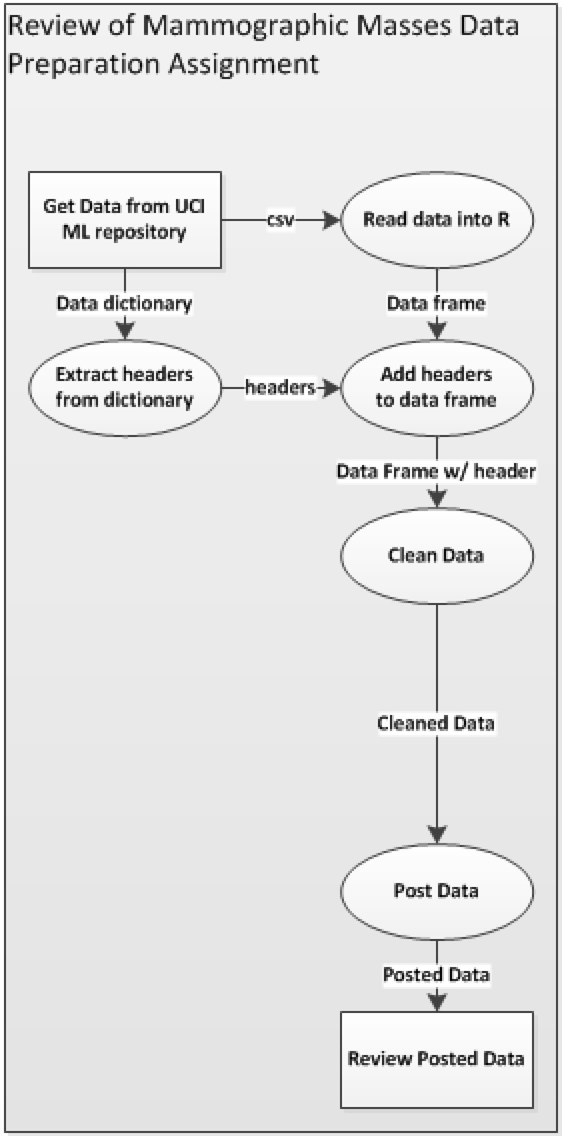
- Optional Class in R
  - DataScience02e.R
- Data Preparation
  - Homework (DataScience02Homework.R)
  - DataScience02f.R (Quiz 03a)
  - Data Preparation Review

Data Preparation Review

Data Preparation Review (1)



Data Preparation Review (2)



Review of Mammographic Masses Data Preparation Assignment

Get Data from UCI ML repository

csv

Read data into R

Data dictionary

Data frame

Extract headers from dictionary

headers

Add headers to data frame

Data Frame w/ header

Clean Data

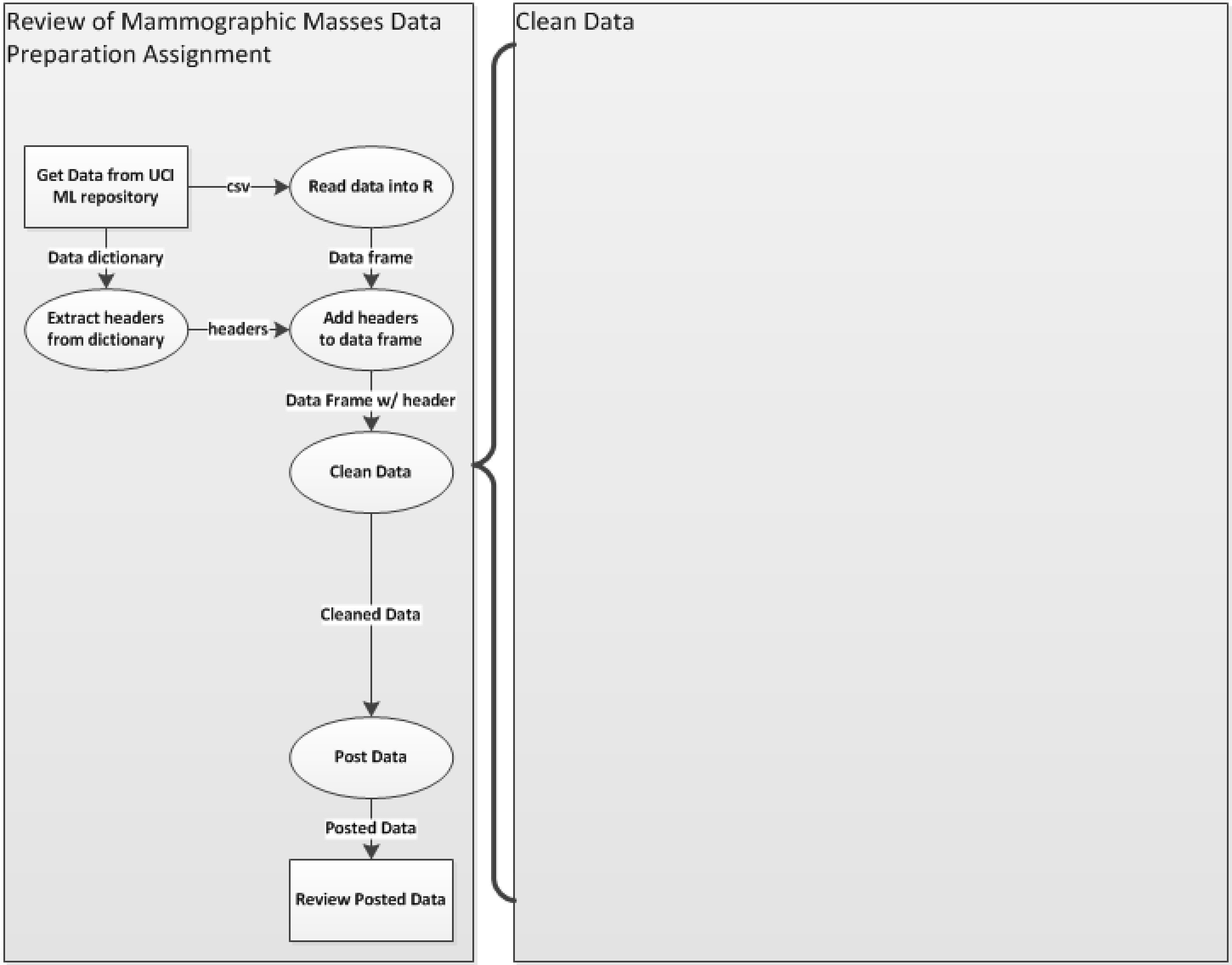
Cleaned Data

Post Data

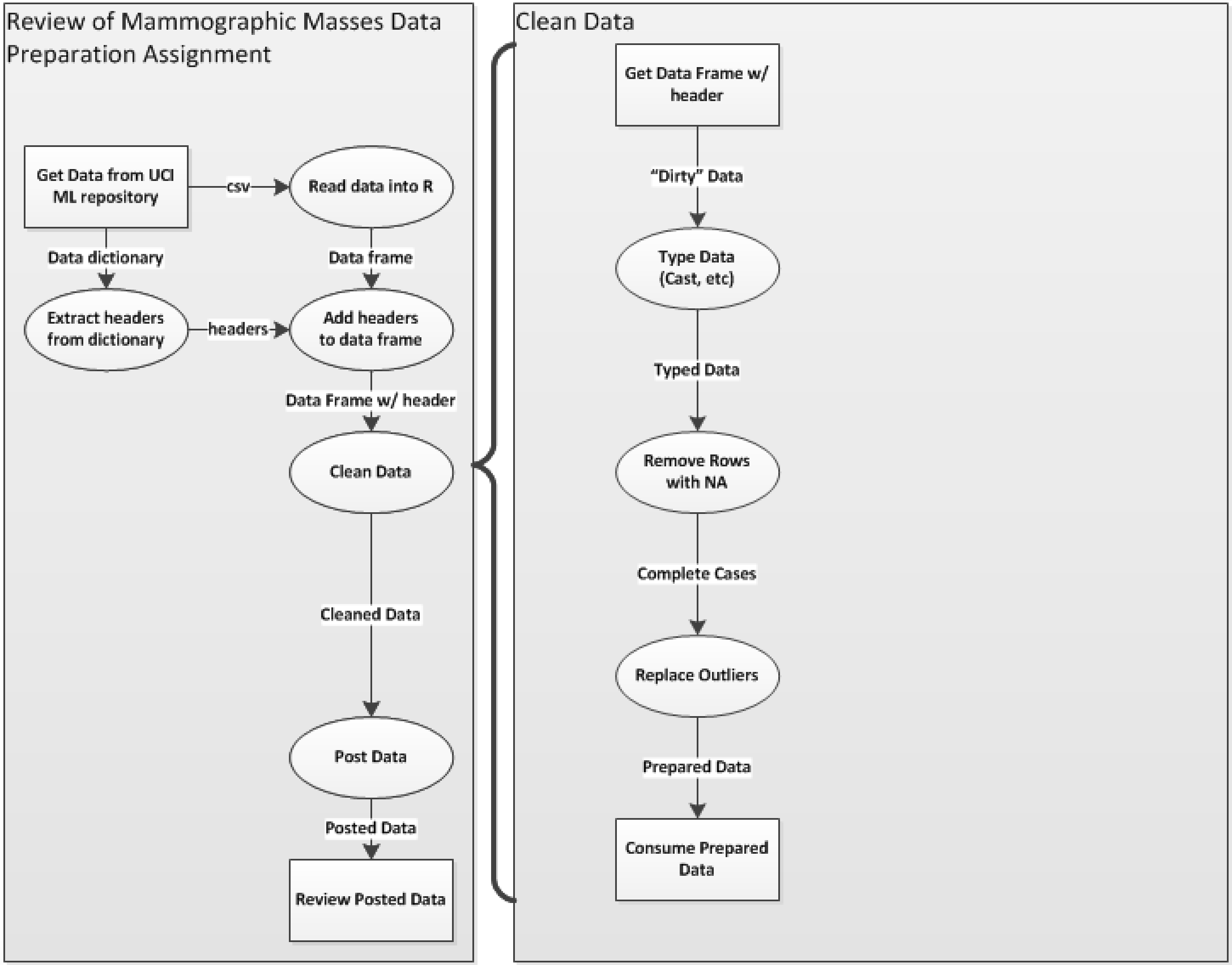
Posted Data

Review Posted Data

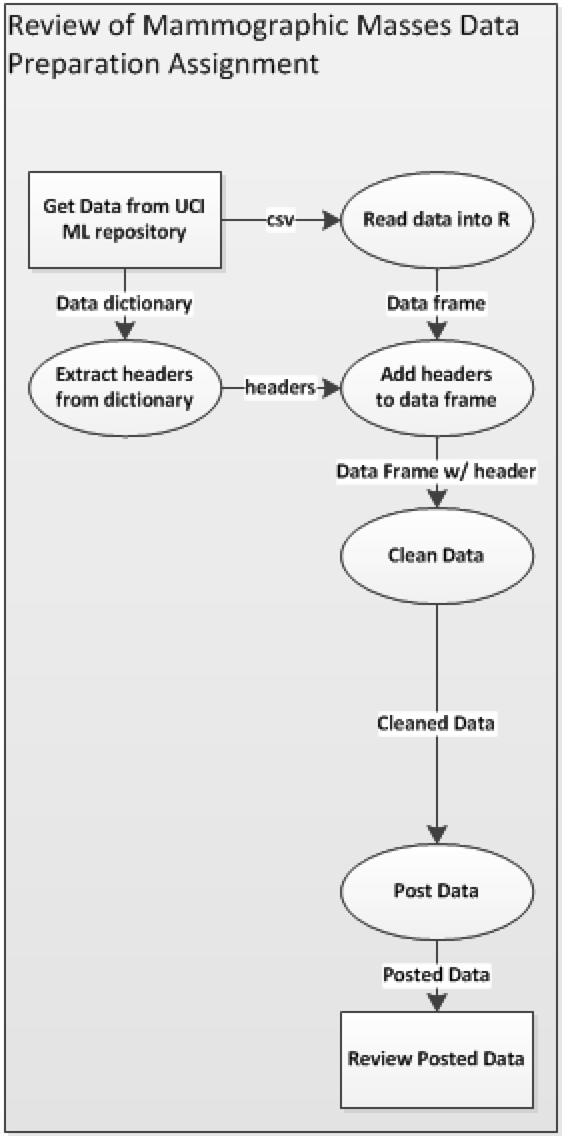
Data Preparation Review (3)


Review of Mammographic Masses Data Preparation Assignment
Get Data from UCI ML repository
csv
Read data into R
Data dictionary
Data frame
Extract headers from dictionary
headers
Add headers to data frame
Data Frame w/ header
Clean Data
Cleaned Data
Post Data
Posted Data
Review Posted Data
Clean Data

Data Preparation Review (4)



Data Preparation Review (5)



Review of Mammographic Masses Data Preparation Assignment

Get Data from UCI ML repository

csv

Read data into R

Data dictionary

Data frame

Extract headers from dictionary

headers

Add headers to data frame

Data Frame w/ header

Clean Data

Cleaned Data

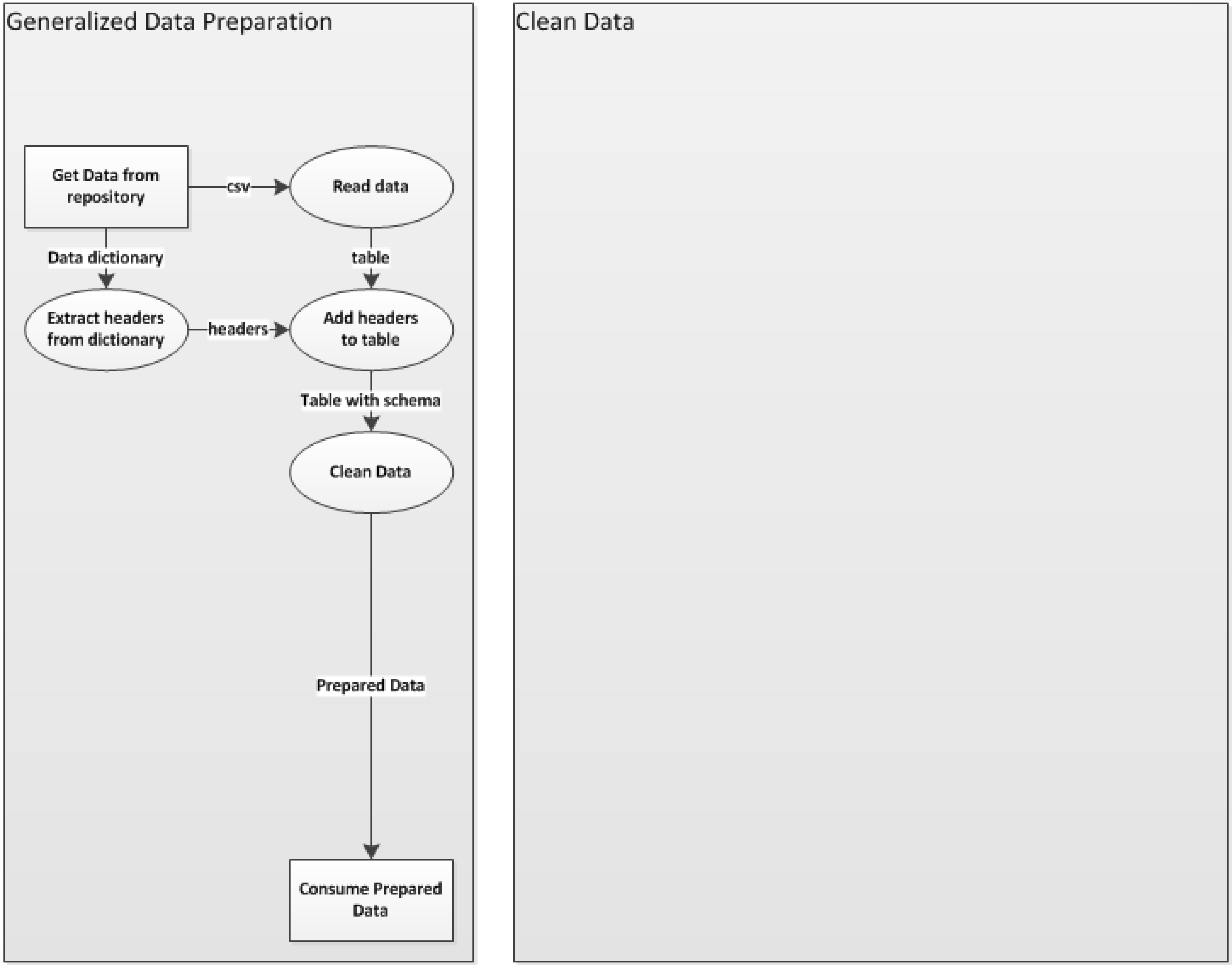
Post Data

Posted Data

Review Posted Data

Clean Data

Data Preparation Review (6)



Generalized Data Preparation

Get Data from repository

csv

Read data

Data dictionary

table

Extract headers from dictionary

headers

Add headers to table

Table with schema

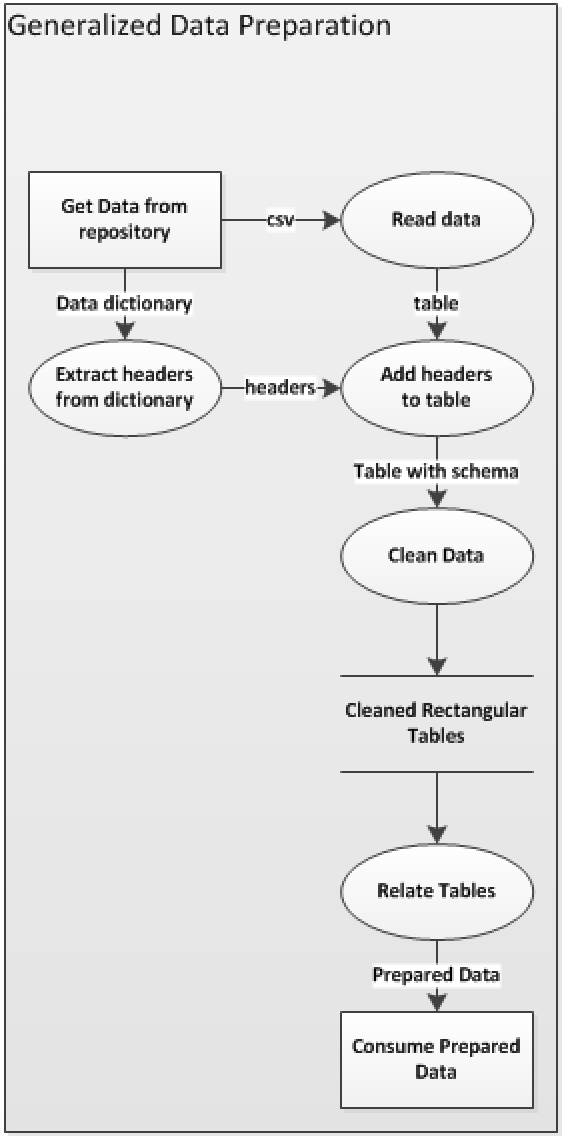
Clean Data

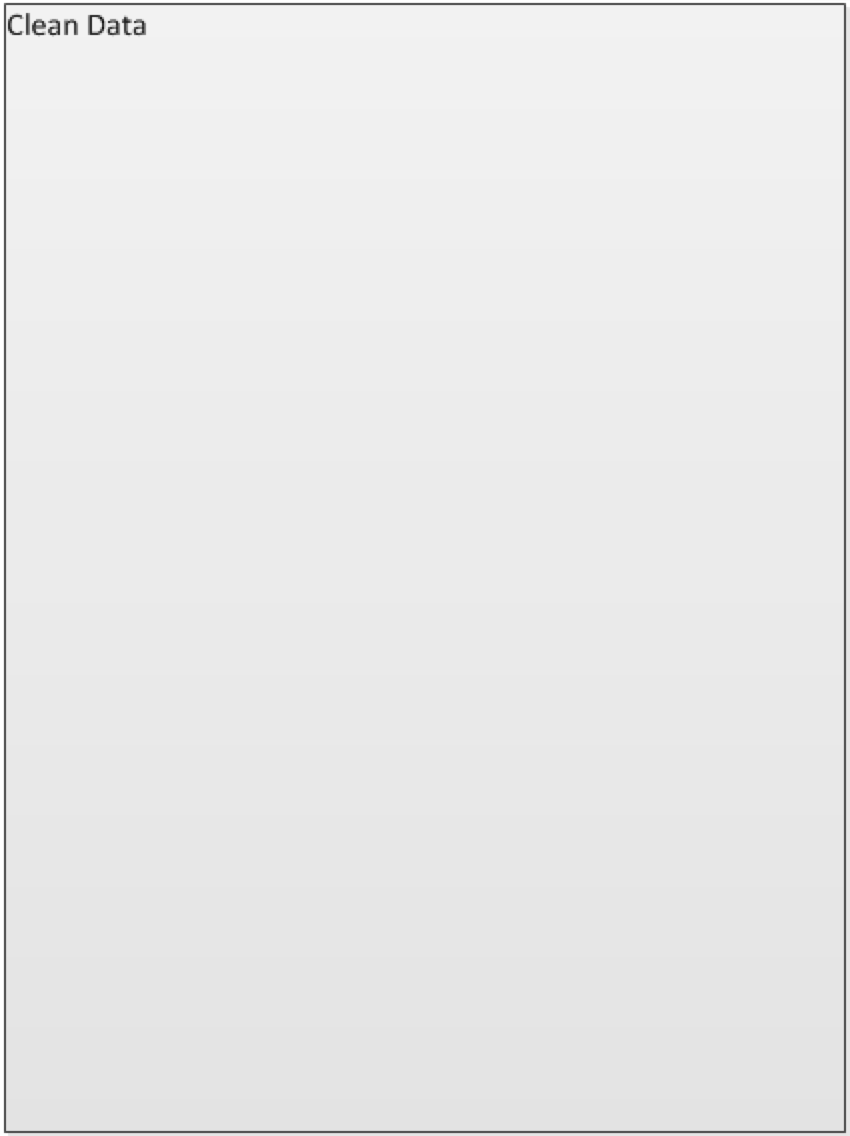
Prepared Data

Consume Prepared Data

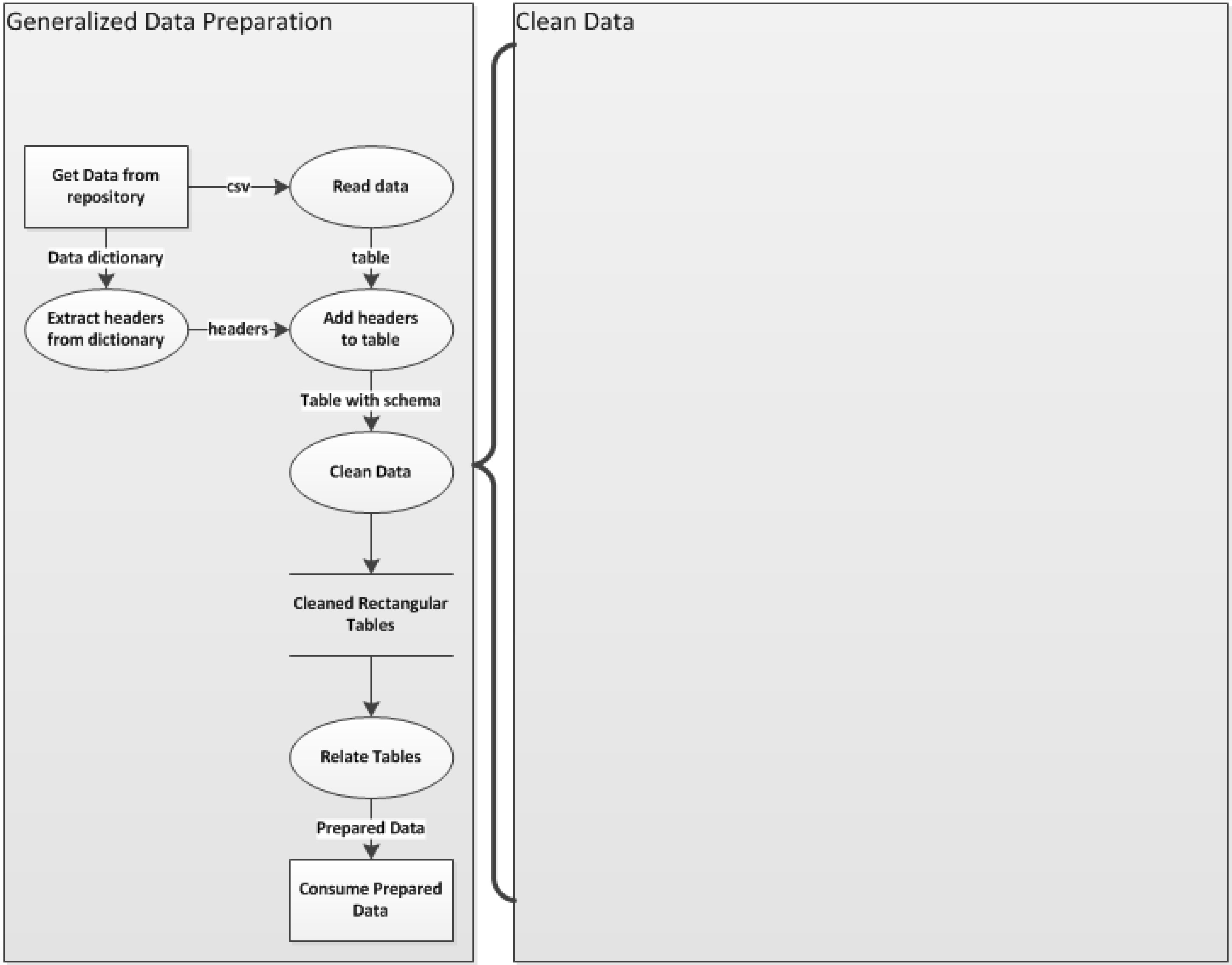
Clean Data

Data Preparation Review (7)

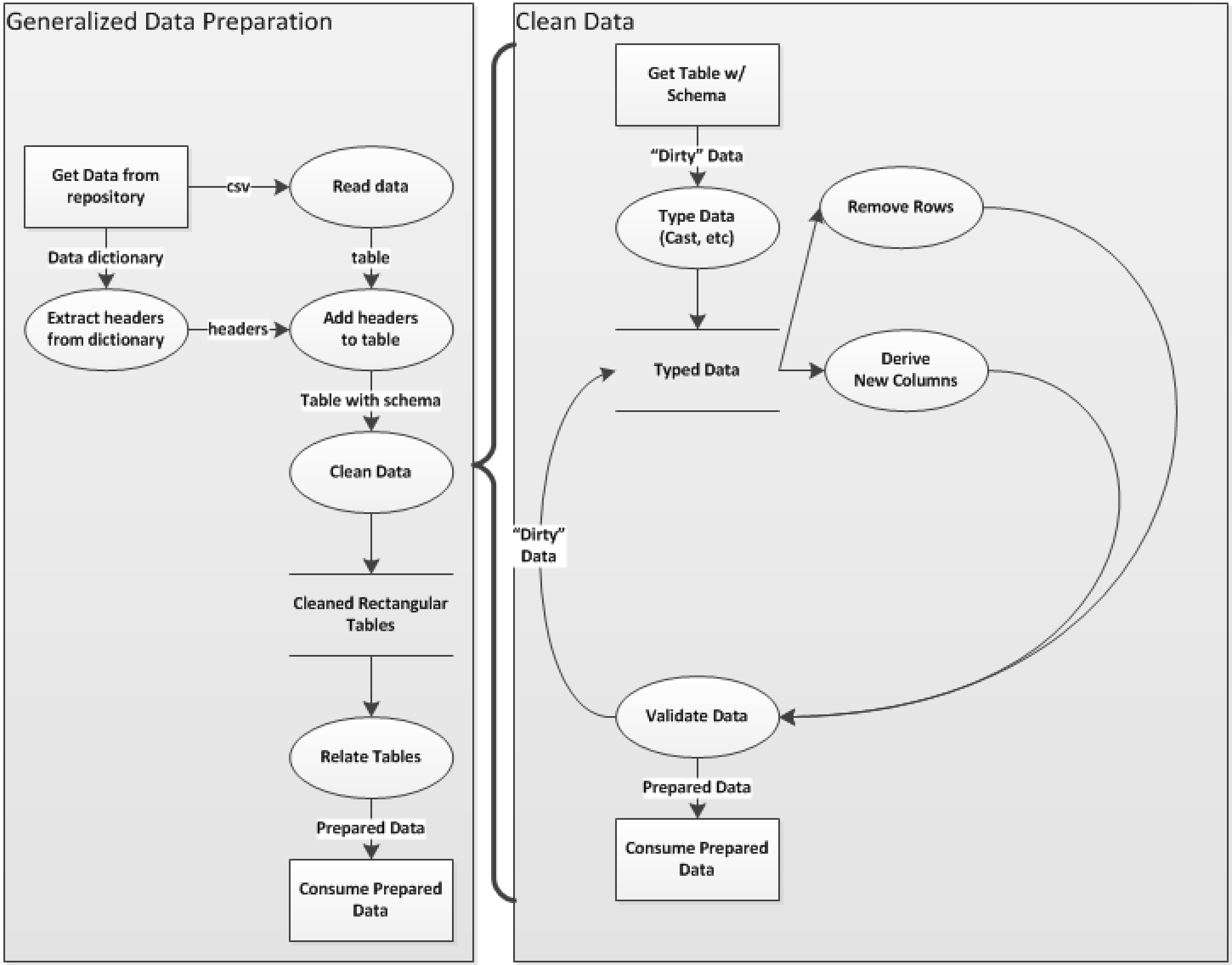




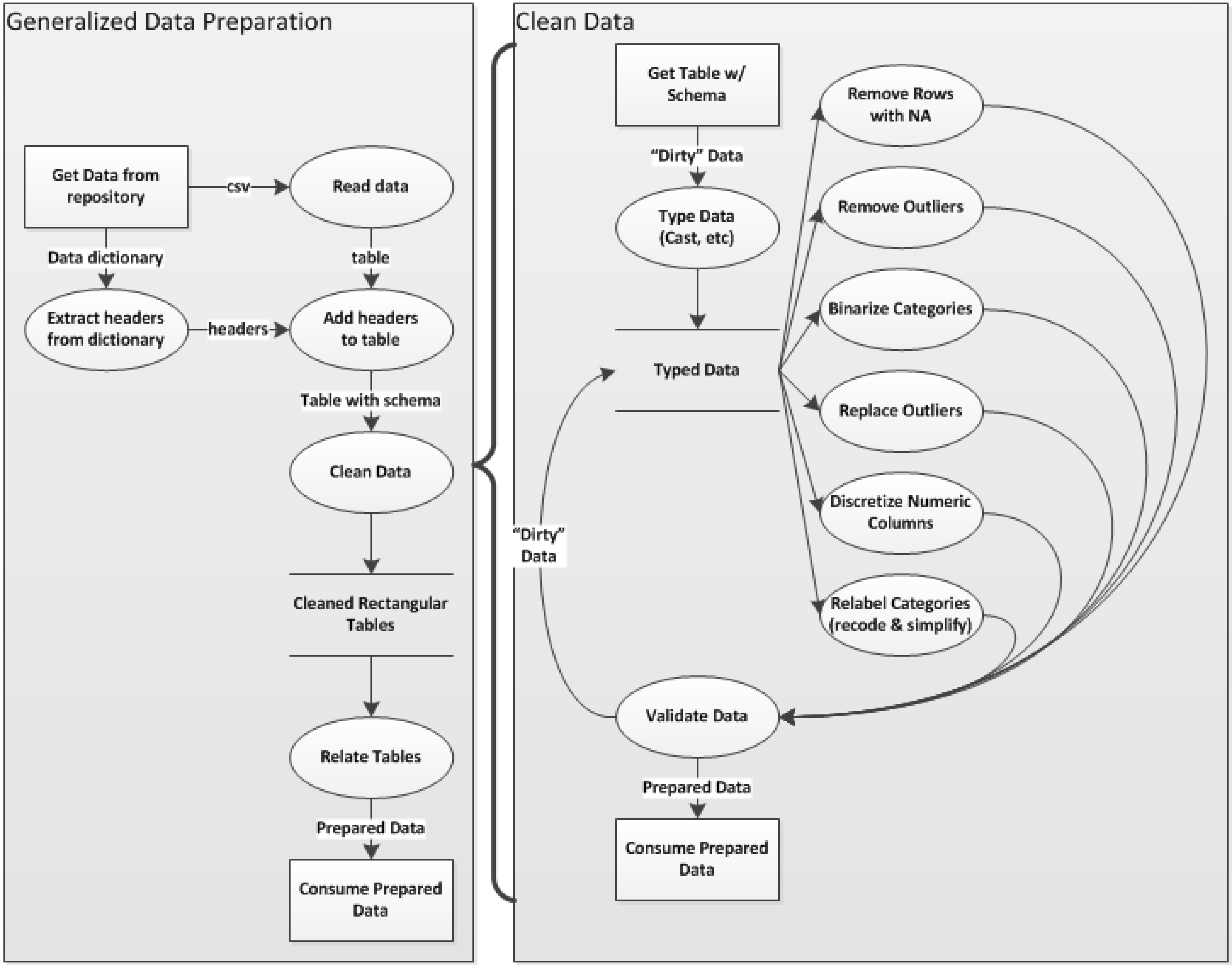
Data Preparation Review (8)



Data Preparation Review (9)



Data Preparation Review (10)



Data Preparation Review

Quiz 03b

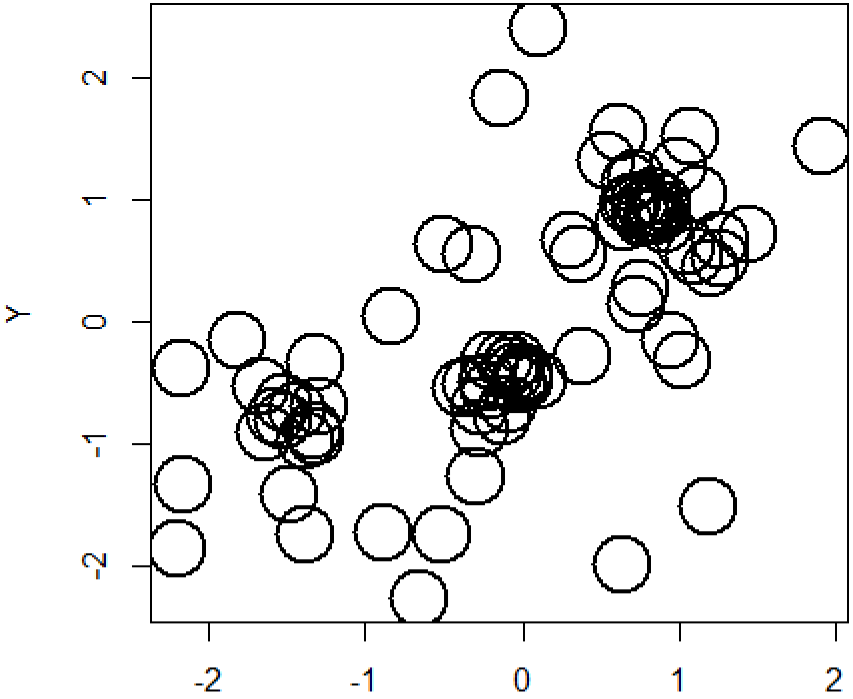
- Data Science UW 2015 Quiz 03b
- https://catalyst.uw.edu/webq/survey/ernsthe/267663

Introduction to K-means Clustering

K-means clustering: Algorithm

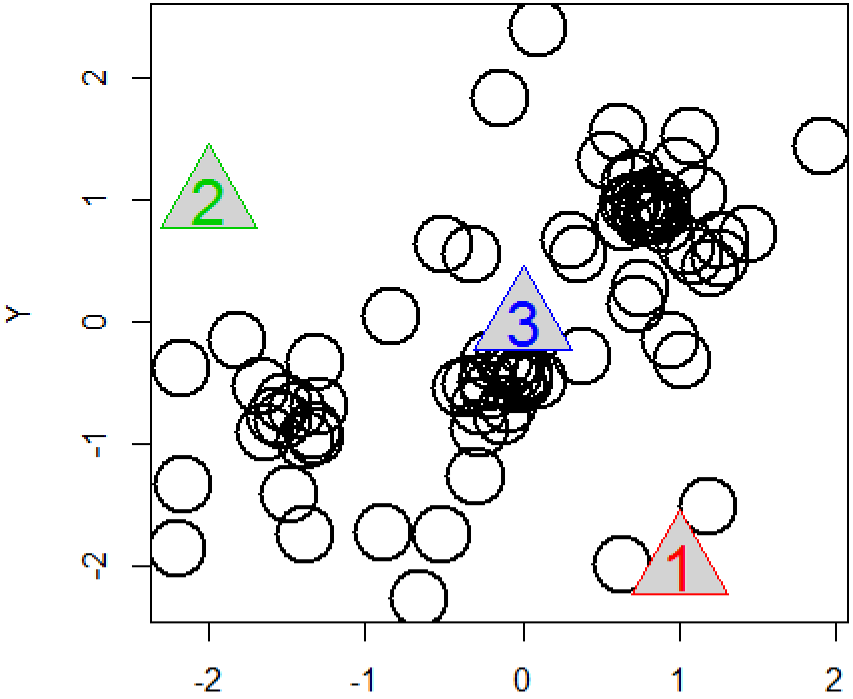
- Pre-requisites
  1. Get points in multi-dimensional space.
     - table, matrix, rectangular dataset
  2. Specify the number of clusters
     - Weakest point in algorithm (makes algorithm non-deterministic)
  3. Get a random center for each cluster
     - Another weak point in the algorithm
- Repeat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



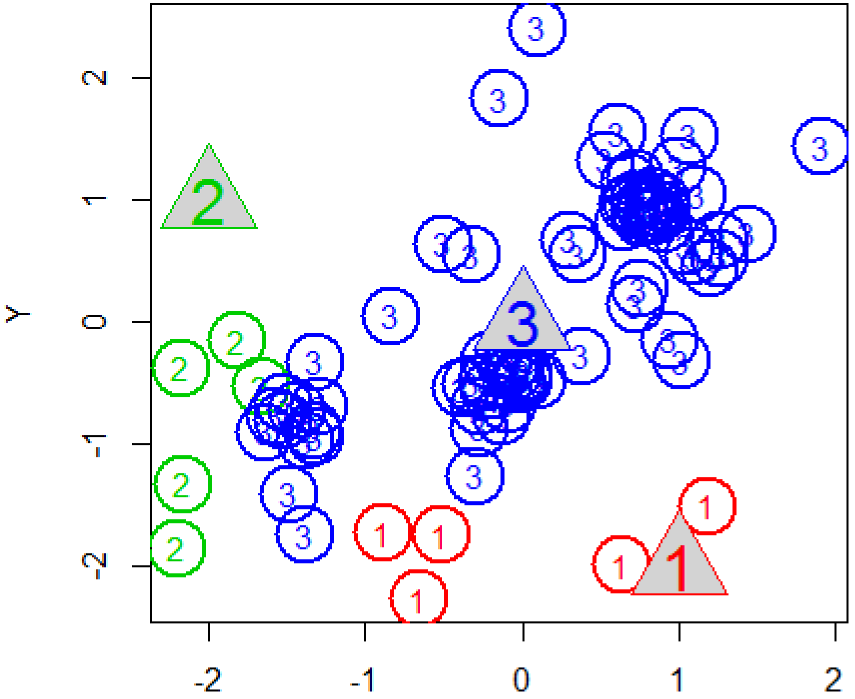
- Clustering starts by getting the data and representing the data as points in space. In this example the space is 2-dimensional.
- Each point describes an observation. An observation is an individual item.
- The dimensions are attributes that describe the item.

K-Means Clustering (1)



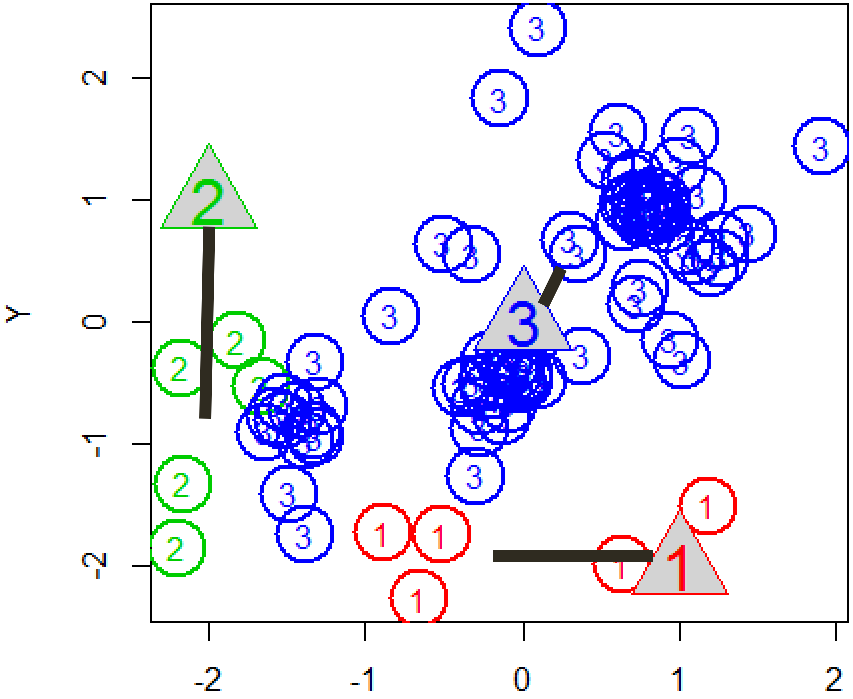
- Clustering continues by guessing, presuming, or specifying a number of clusters.
- Each centroid represents a cluster.
- The centroid positions are determined randomly. The centroids should be within the bounds of the points.

K-Means Clustering (2)



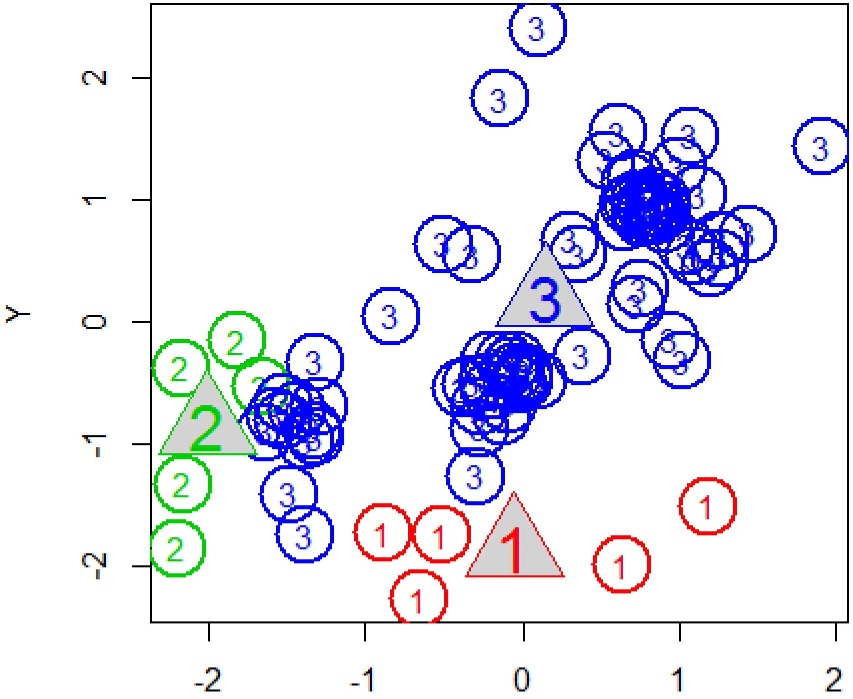
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (2)



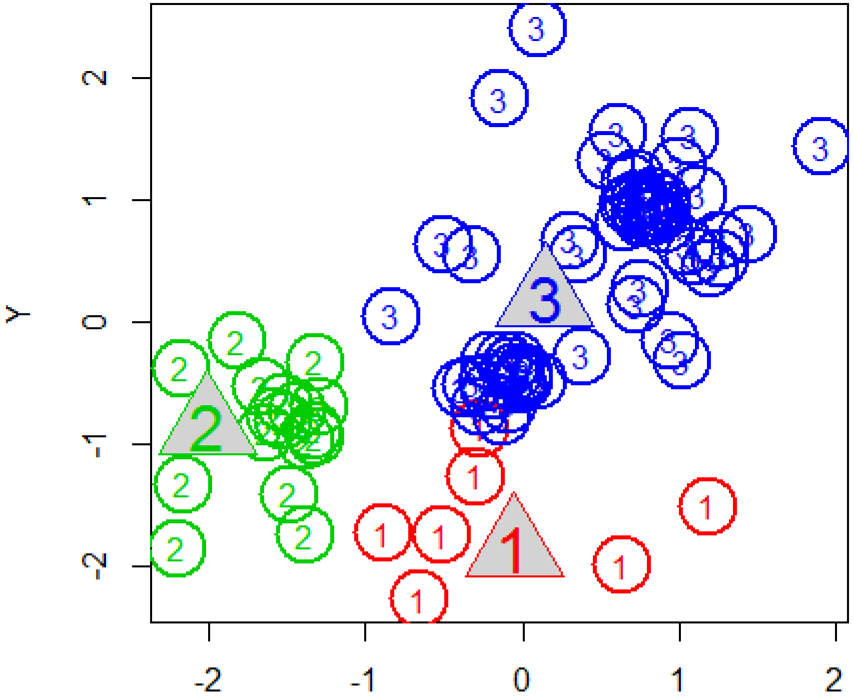
- Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (3)



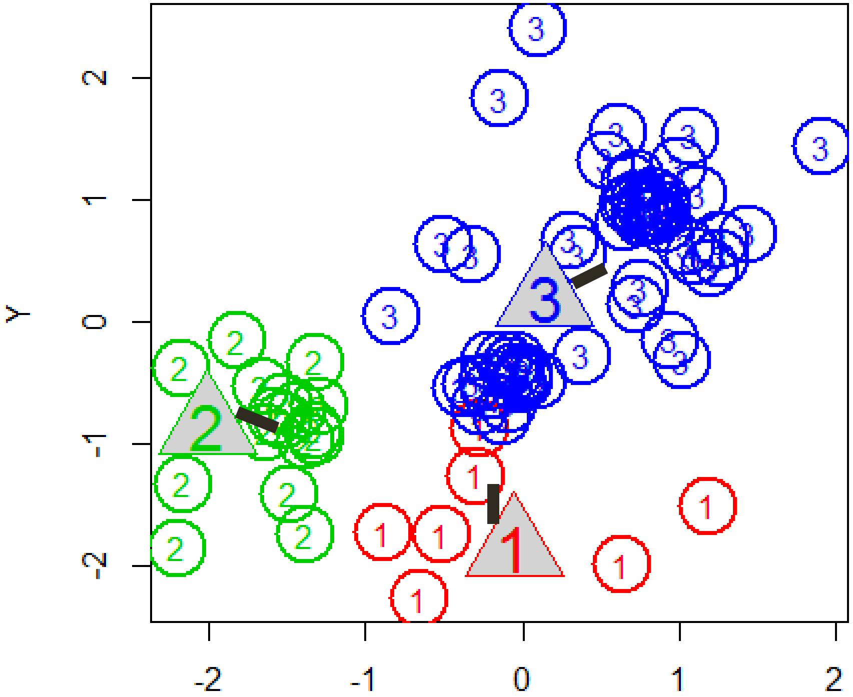
- Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (4)



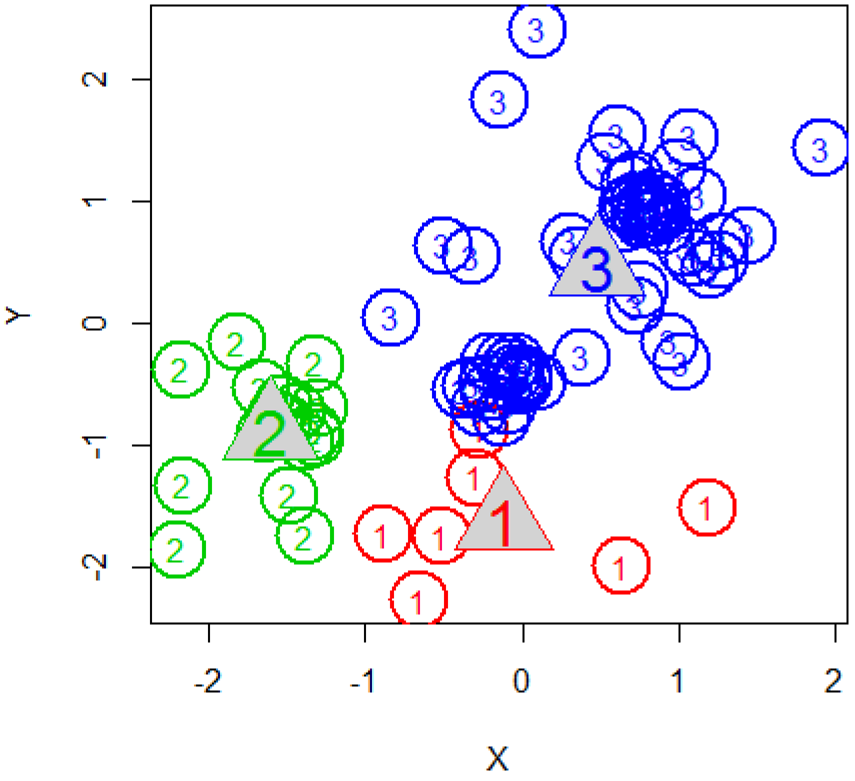
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (4)

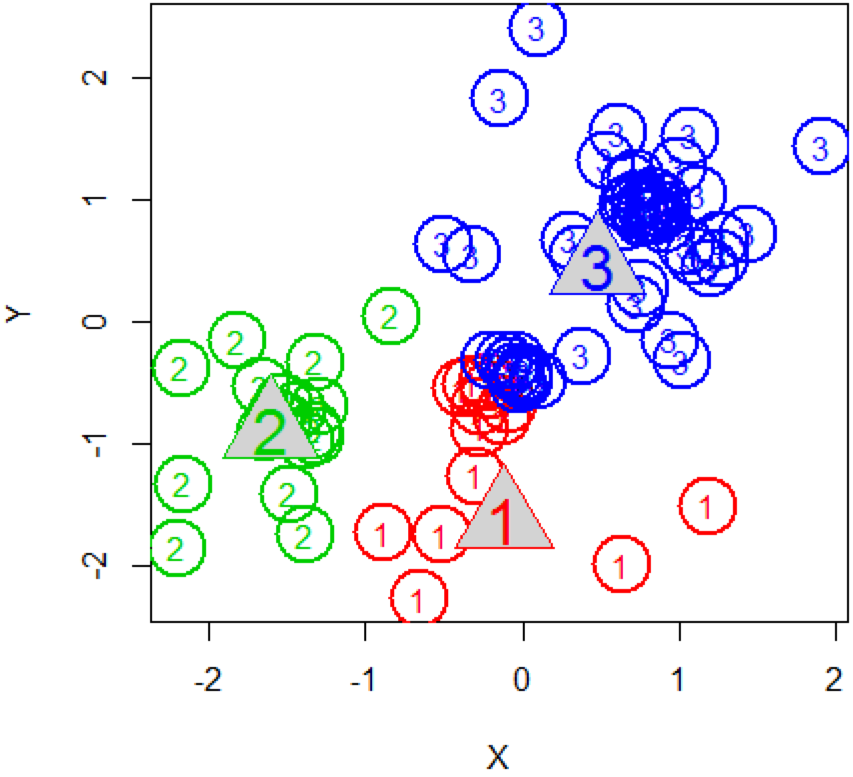


- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

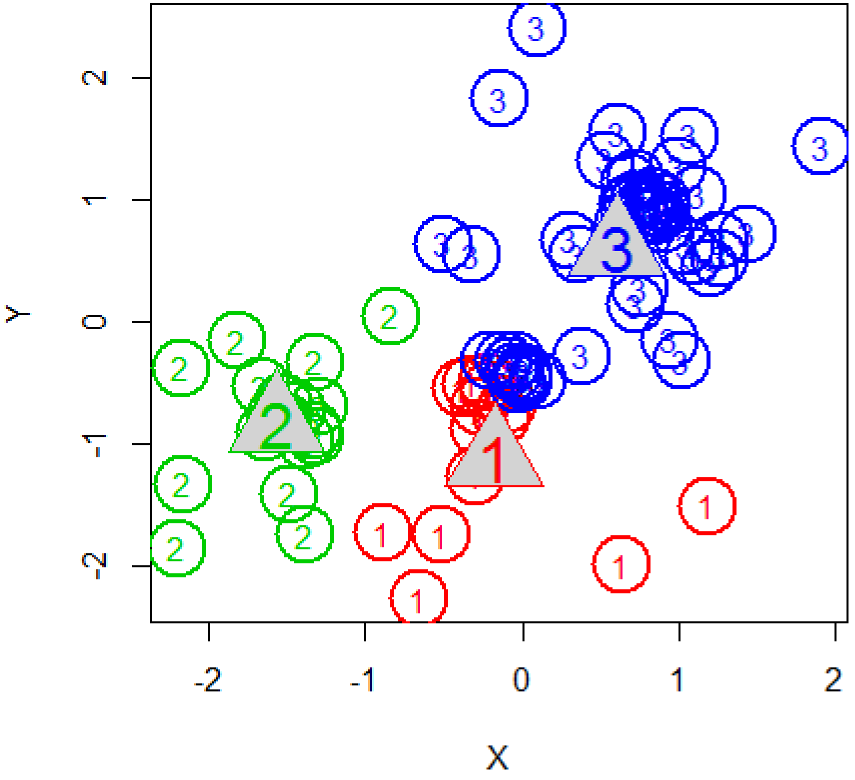
K-Means Clustering (5)



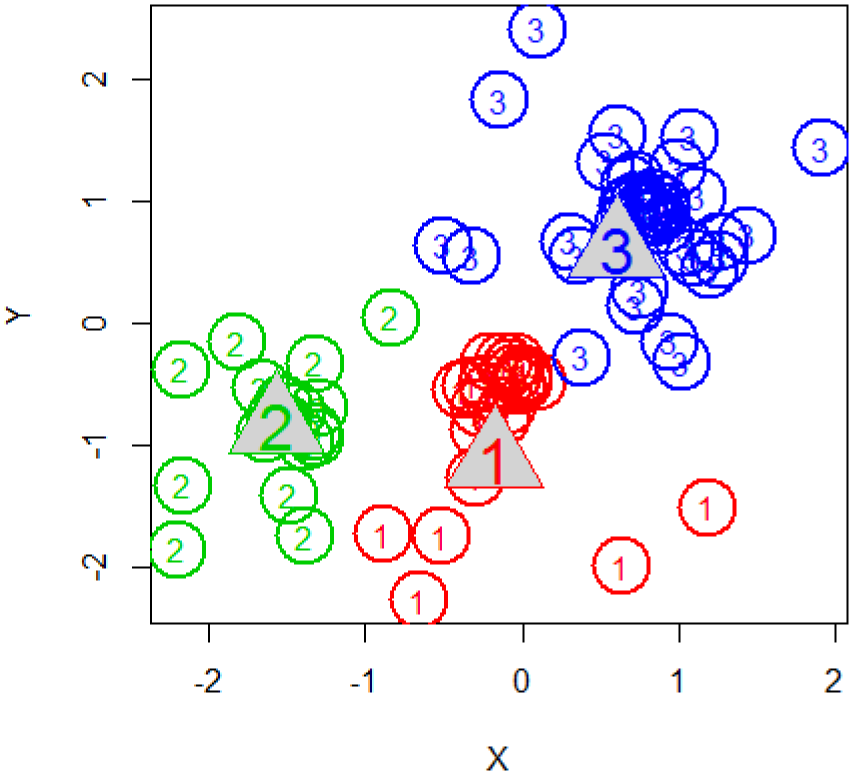
K-Means Clustering (6)



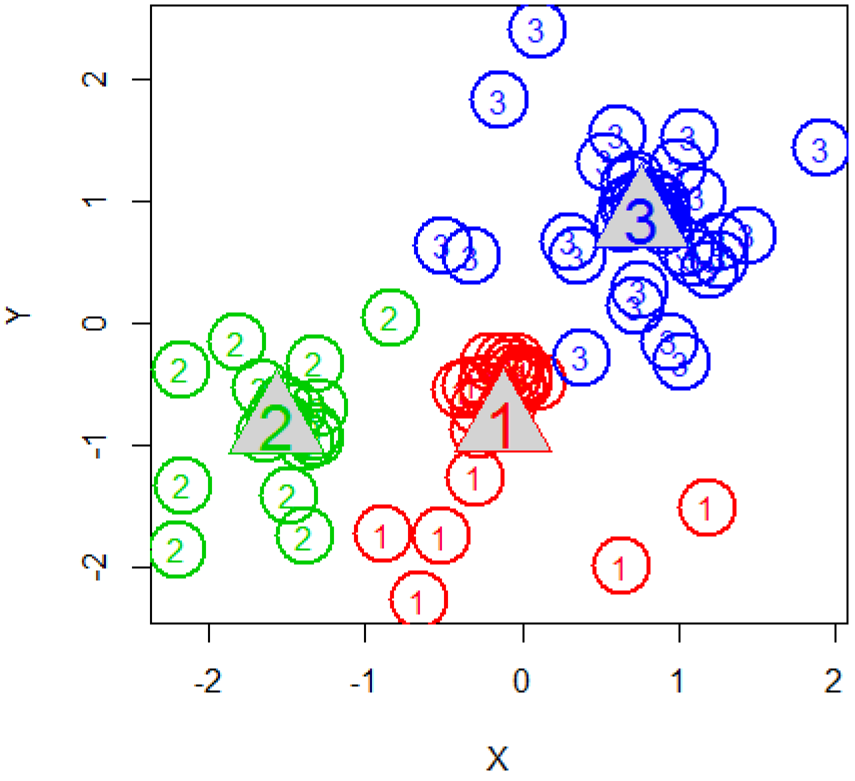
K-Means Clustering (7)



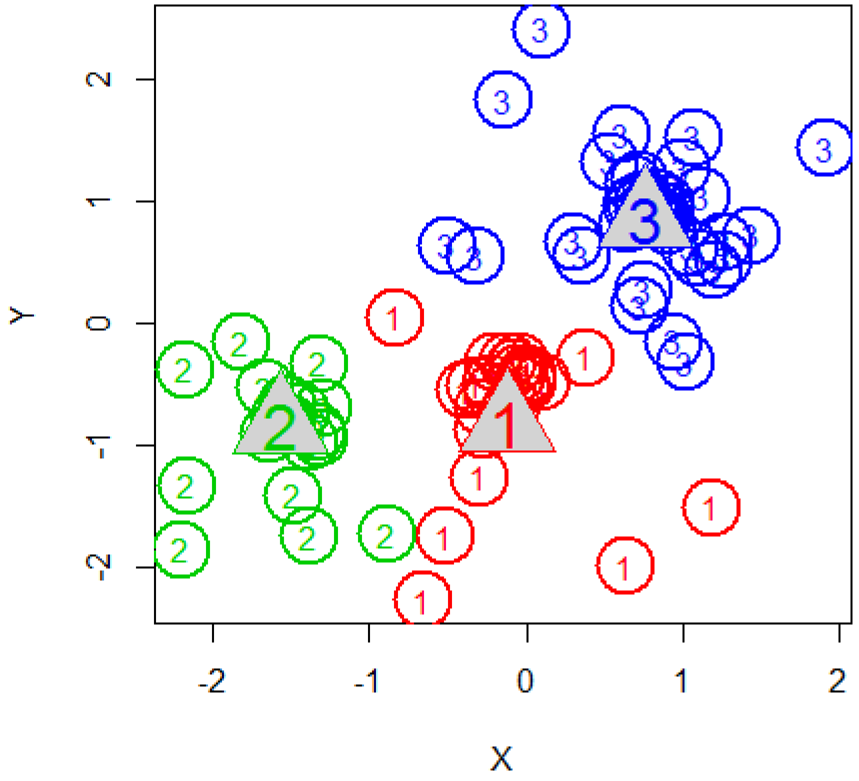
K-Means Clustering (8)



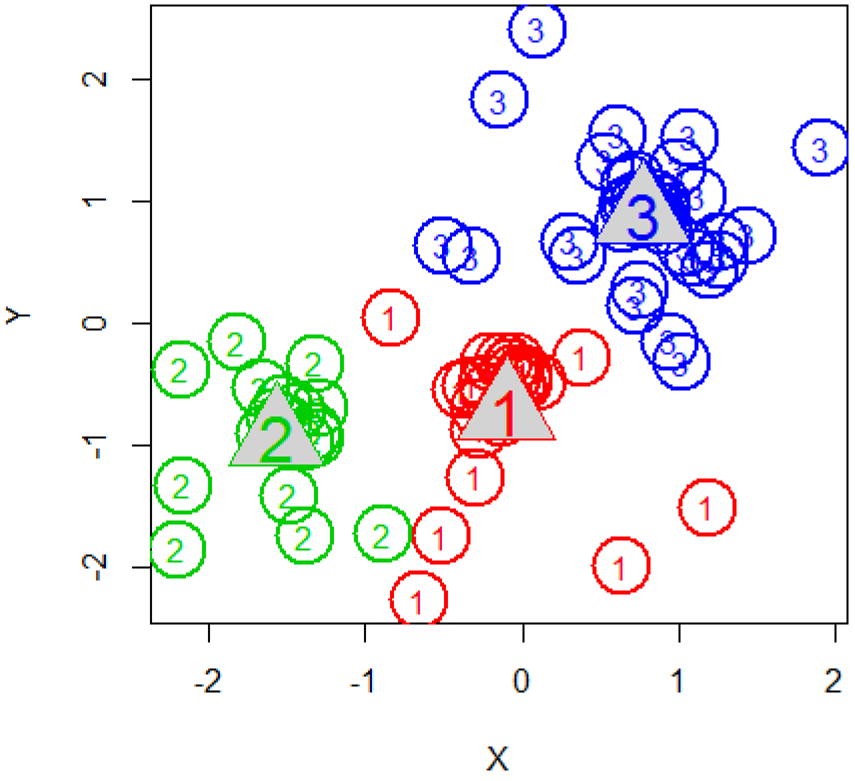
K-Means Clustering (9)



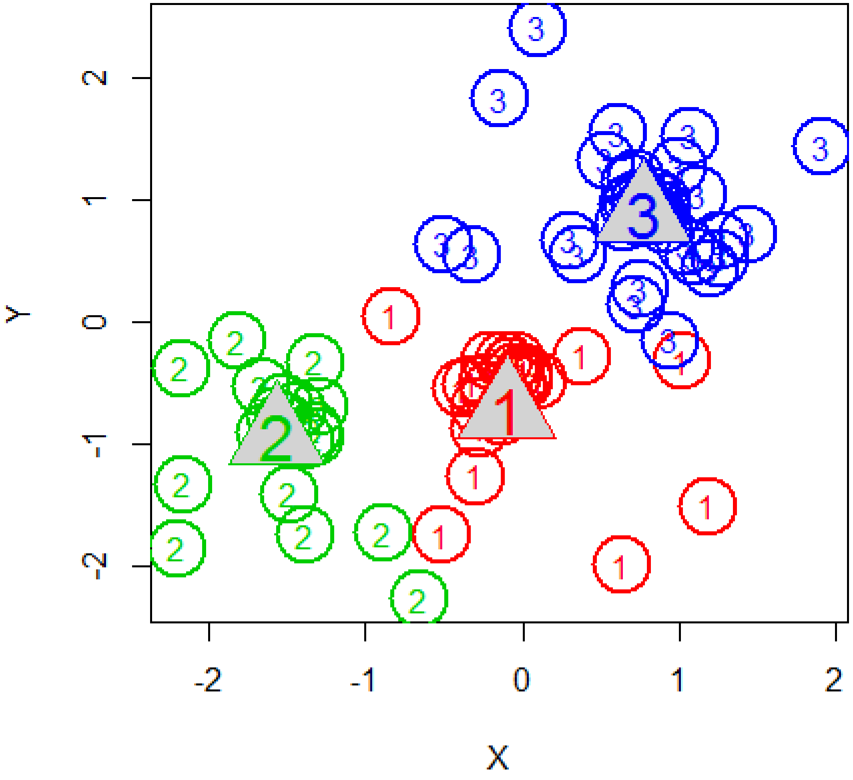
K-Means Clustering (10)



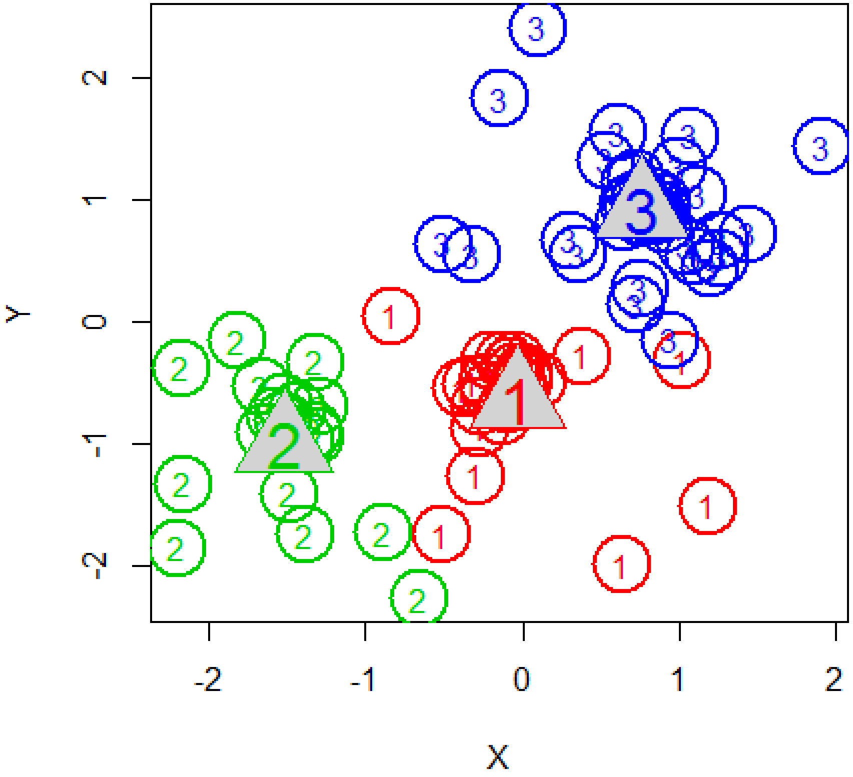
K-Means Clustering (11)



K-Means Clustering (12)



K-Means Clustering (13)



K-means Demo

K-Means Clustering step-by-step

Clustering of Patients

K-means

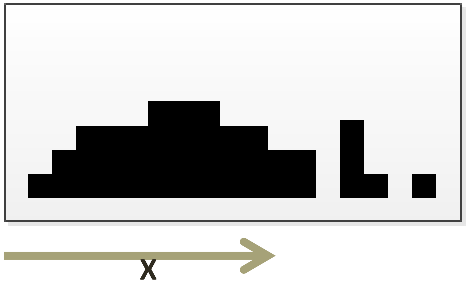
- Some Points:
  - Normalizations are important to put data on equal terms
  - Initial centroid number and placement is an art.
  - Categorical Data must be binarized
  - K-means is unsupervised because we do not tell the algorithm what outcome was observed or what outcome is desired.

Break

Introduction to K-means Clustering

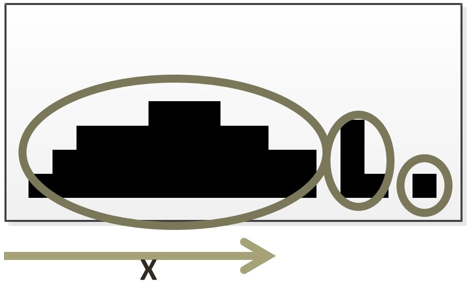
Dimensions in Clustering

Clustering: Dimensions (1)



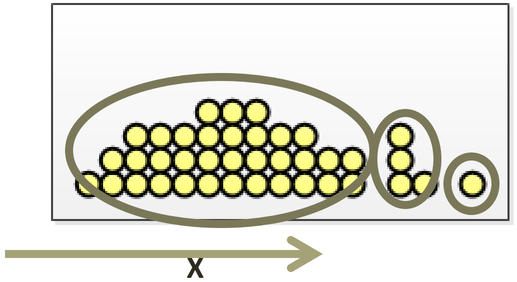
Where are the three clusters?

Clustering: Dimensions (2)



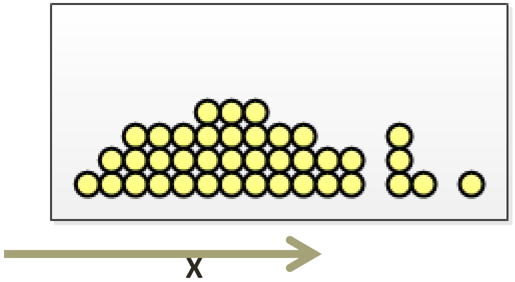
Simple assignment based on a 1D distribution

Clustering: Dimensions (3)



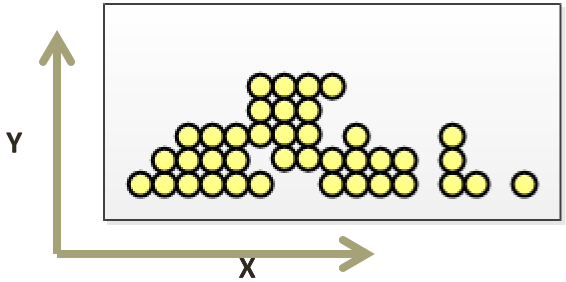
Simple assignment based on a 1D distribution

Clustering: Dimensions (4)



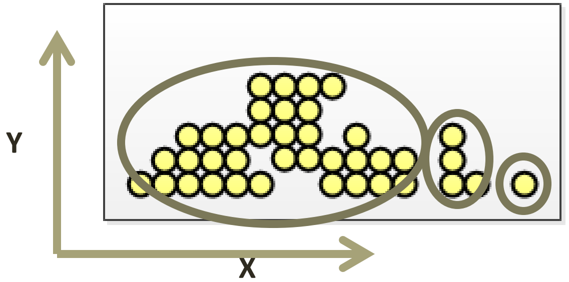
What if this was not a 1D distribution?

Clustering: Dimensions (5)



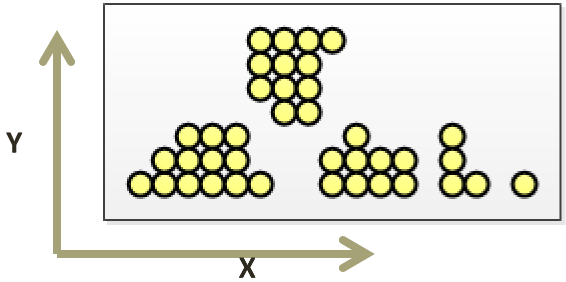
The distribution is in 2D. Some points differ in the 2nd D

Clustering: Dimensions (6)



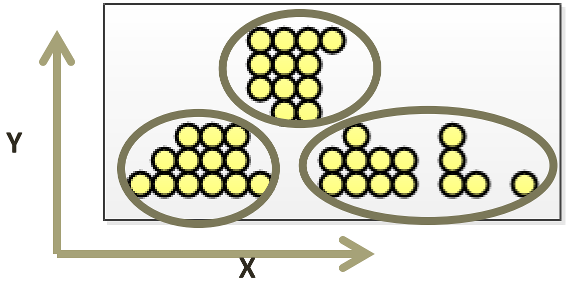
If the difference is minor, we still get the same clusters

Clustering: Dimensions (7)



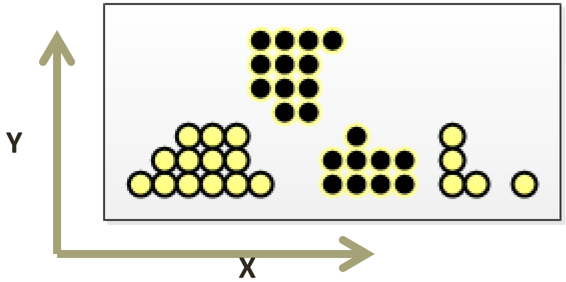
The difference could be significant

Clustering: Dimensions (8)



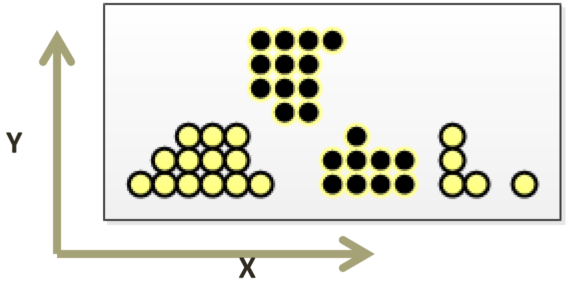
A big difference in the 2nd D can lead to different clusters

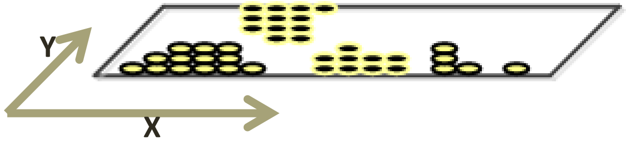
Clustering: Dimensions (9)



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

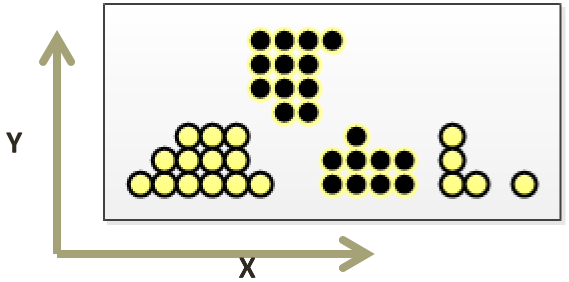
Clustering: Dimensions (10)

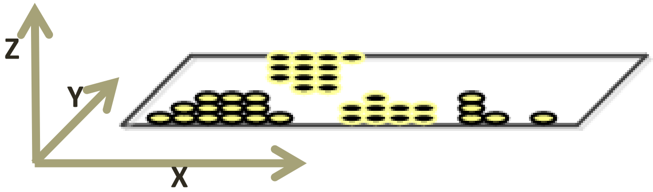




Create a 3rd Dimansion

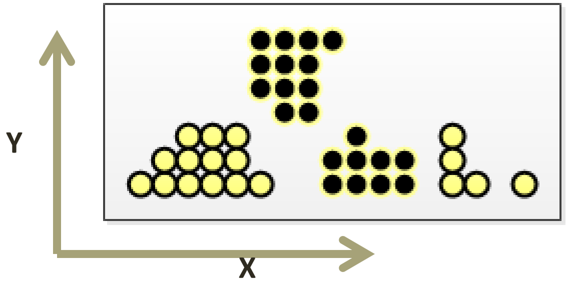
Clustering: Dimensions (11)

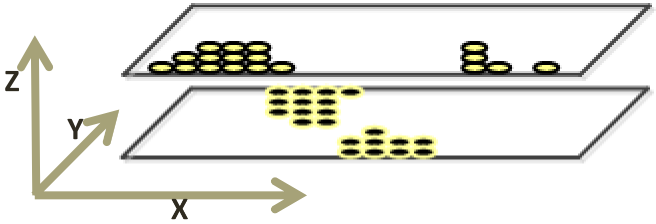




Create a 3rd Dimansion

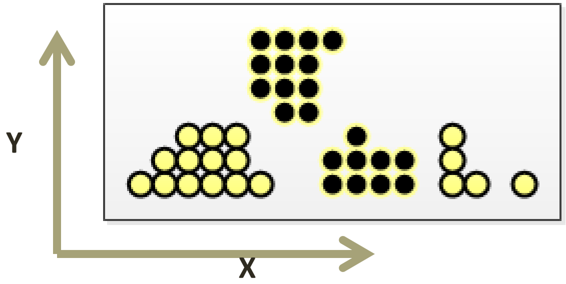
Clustering: Dimensions (12)

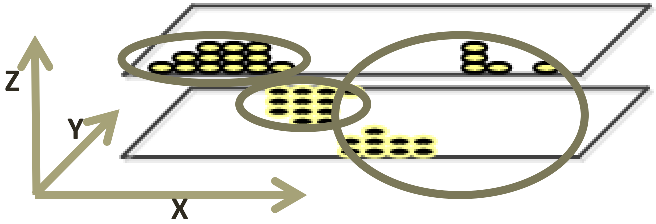


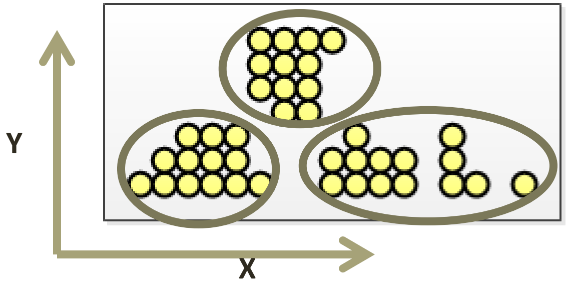


Where are the 3 clusters now?

Clustering: Dimensions (13)

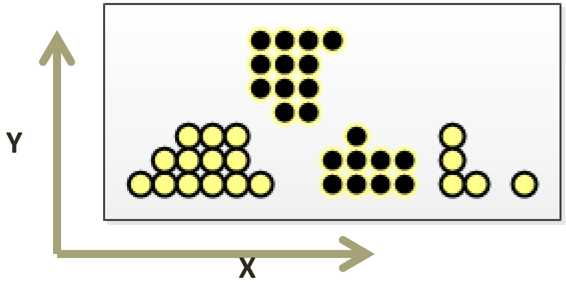


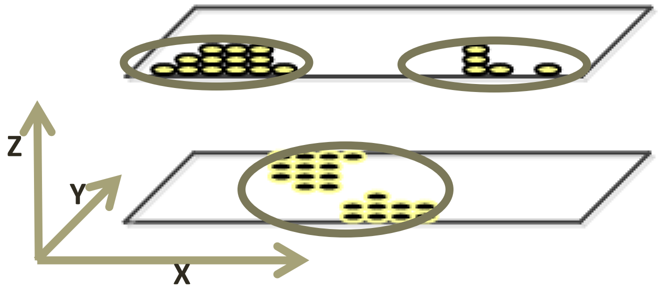


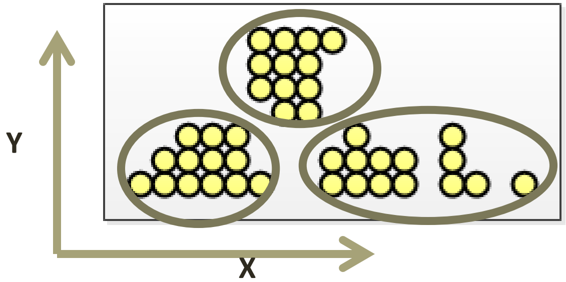


If the 3rd is small, then the clustering is the same as in 2D

Clustering: Dimensions (14)







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

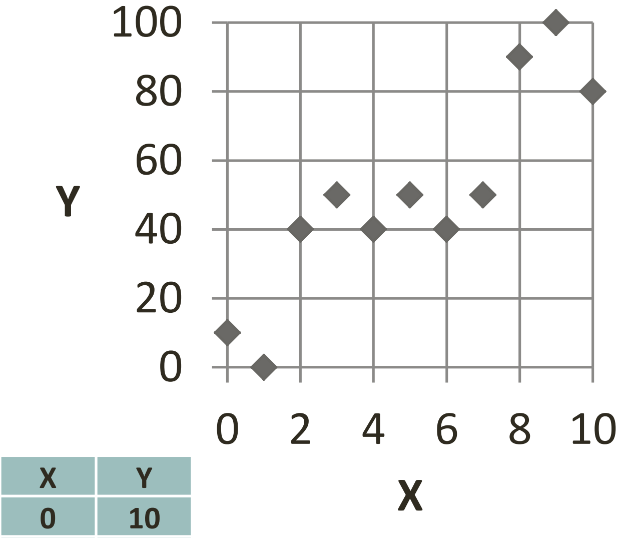
Dimensions in Clustering

Normalization in Clustering

Normalization of a linear relationship (1)

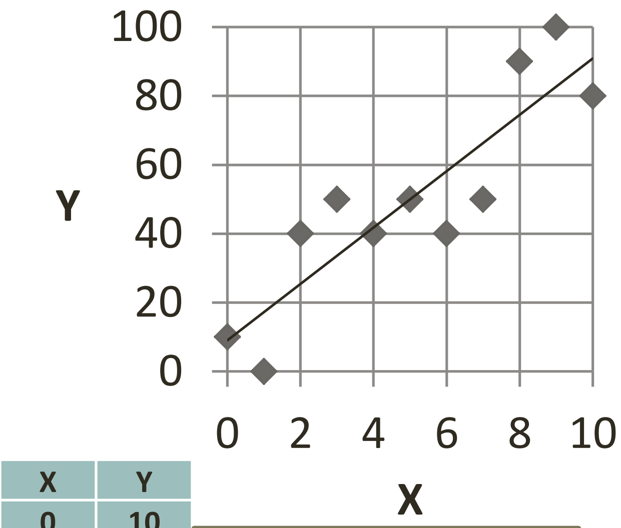
X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

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

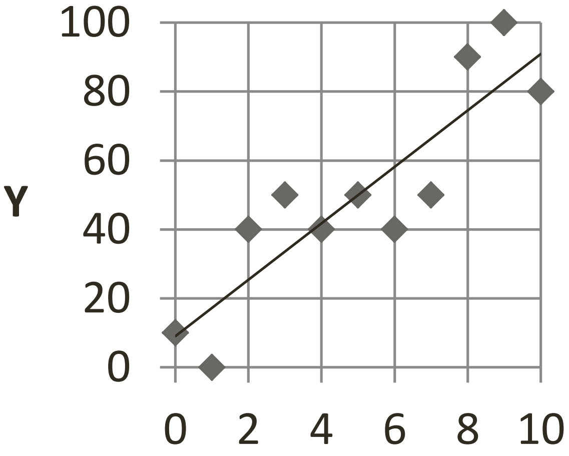
Normalization of a linear relationship (3)



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

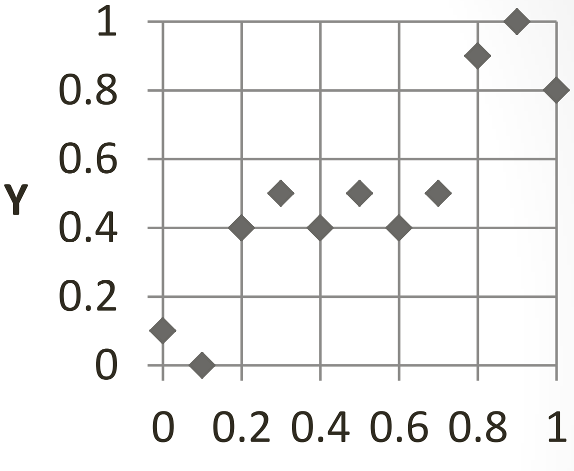
$Y = 10 + 8*X$

Normalization of a linear relationship (4)



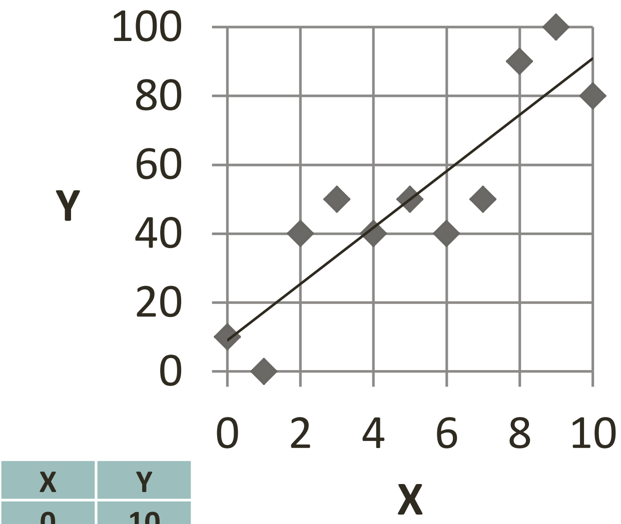
X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

Y = 10 + 8*X



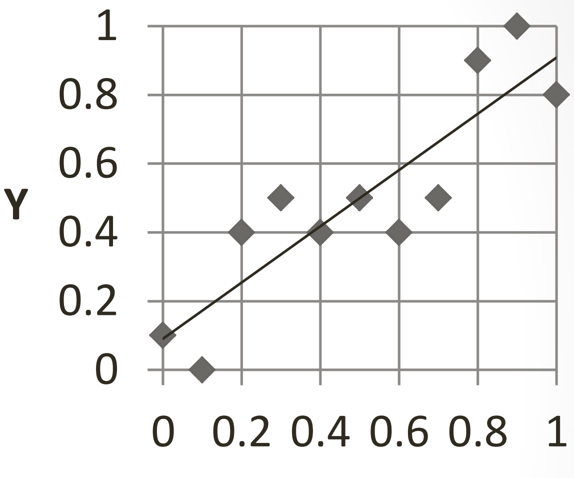
X	Y
0	0.1
0.1	0
0.2	0.4
0.3	0.5
0.4	0.4
0.5	0.5
0.6	0.4
0.7	0.5
0.8	0.9
0.9	1
1	0.8

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

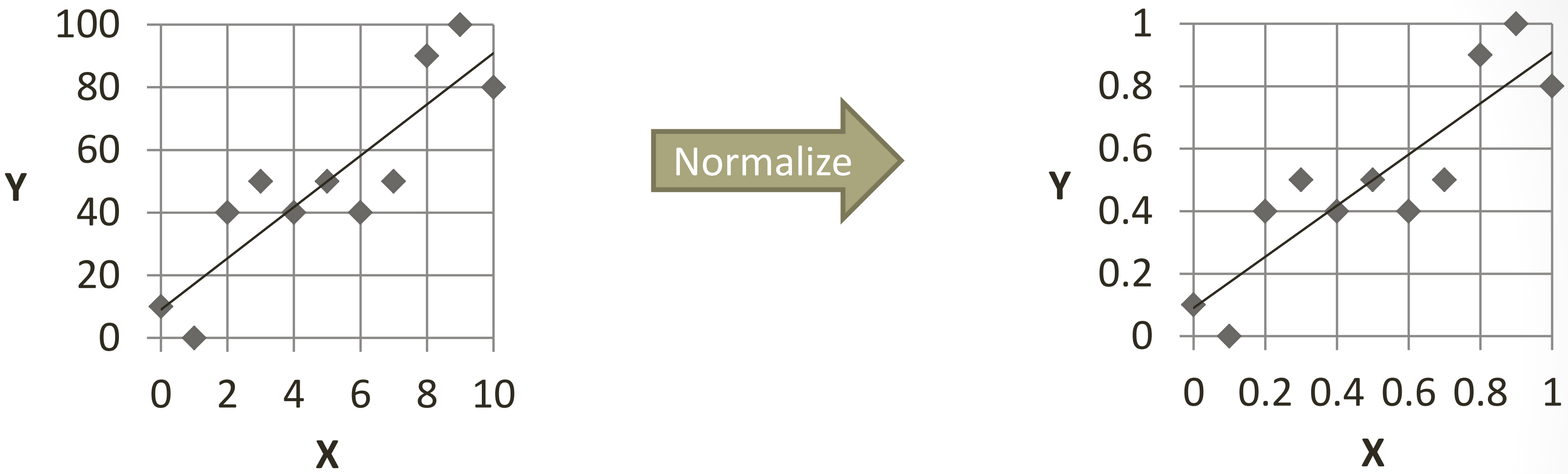
Y = 10 + 8*X



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

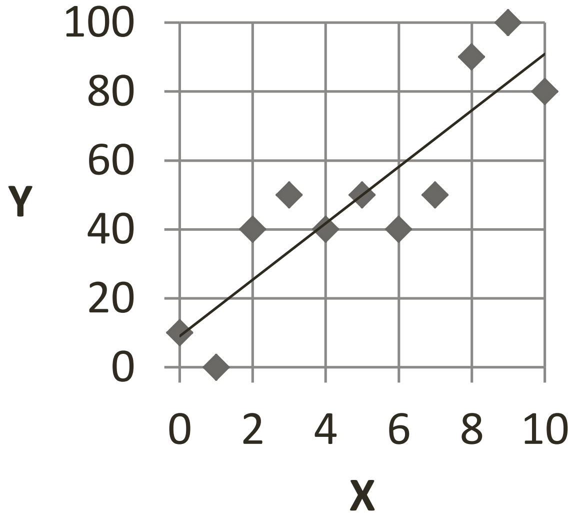
Normalization of a linear relationship (6)



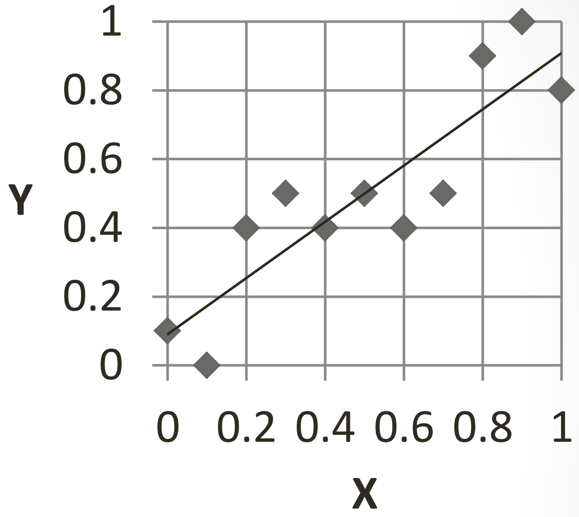
Y = 10 + 8*X

Y = 0.1 + 0.8*X

Normalization of a linear relationship (7)



Normalize



Y = 10 + 8*X

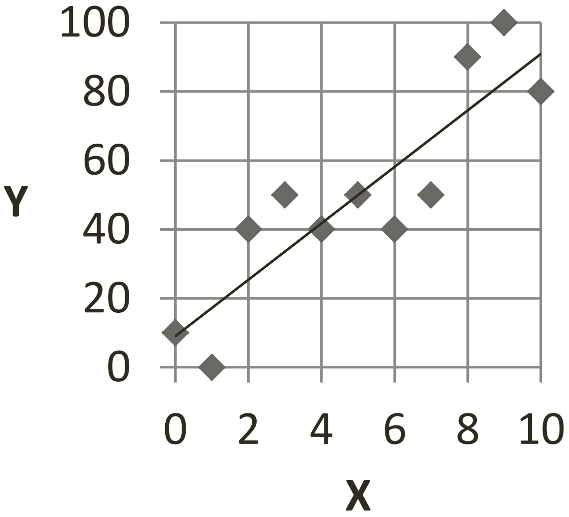
Normalize Input
X = 2 -> X’ = 0.2

Y = 0.1 + 0.8*X

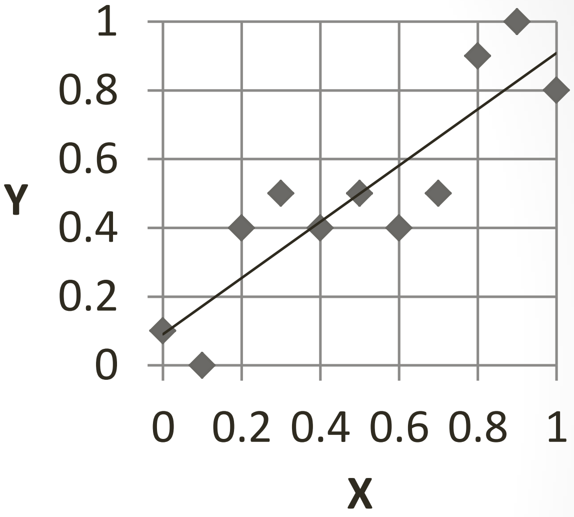
Predict Output
X’ = 0.2 -> Y’= 0.26

Denormalize Output
Y’= 0.26 -> Y = 26

Normalization of a linear relationship (8)



Normalize



Y = 10 + 8*X

Normalize Input
X = 2 -> X’ = 0.2

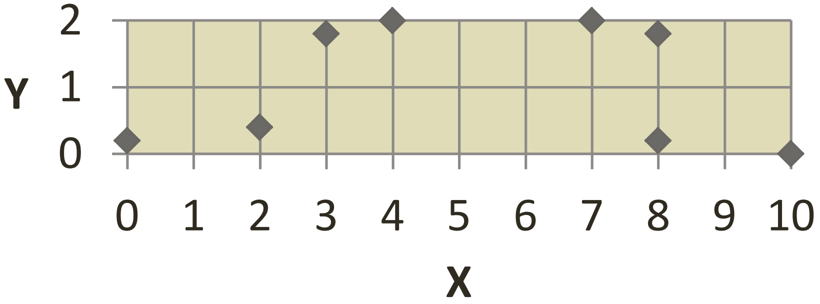
Y = 0.1 + 0.8*X

Predict Output
X’ = 0.2 -> Y’= 0.26

Denormalize Output
Y’= 0.26 -> Y = 26

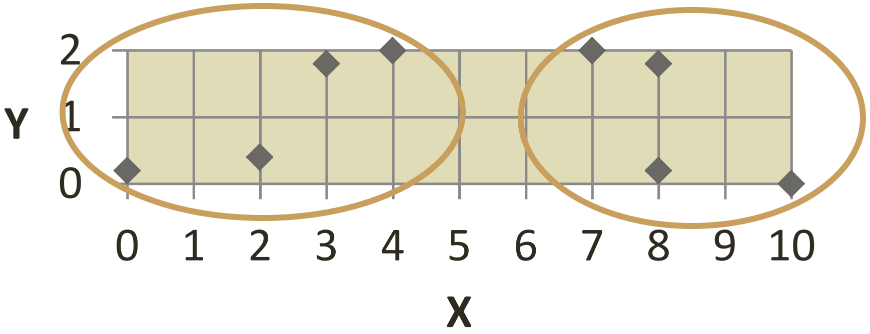
Prediction in Original Space:
X = 2 -> Y = 26

Normalization of a non-linear relationship (1)



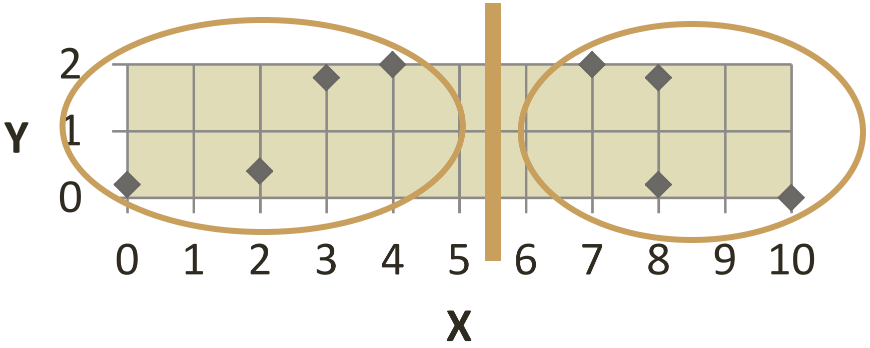
Original data in 2D:
Find 2 clusters

Normalization of a non-linear relationship (2)



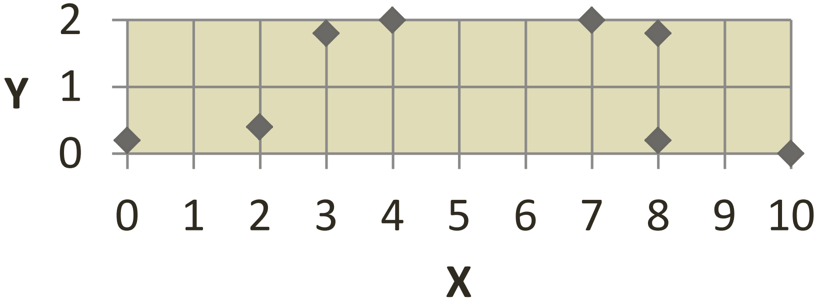
Found 2 Clusters

Normalization of a non-linear relationship (3)



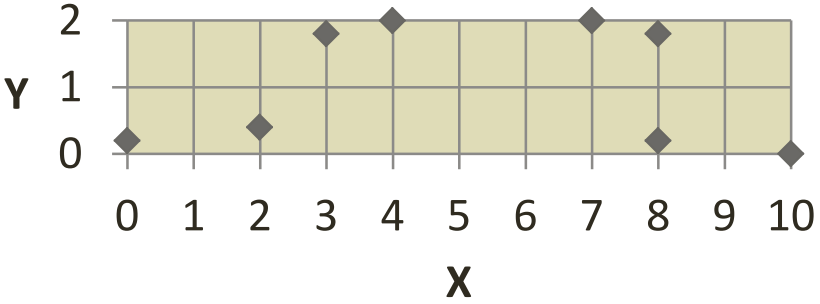
Clusters segment the image

Normalization of a non-linear relationship (4)

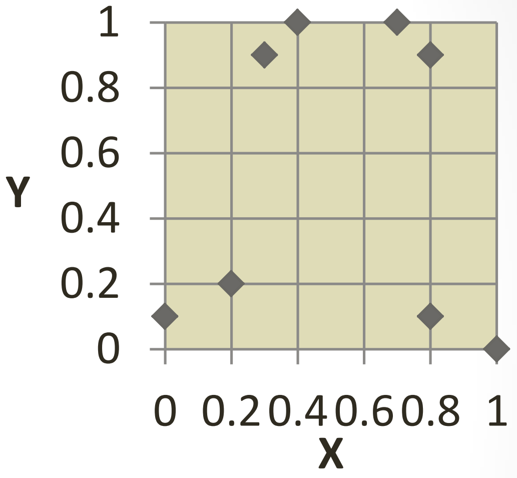


Non-normalized 2D data

Normalization of a non-linear relationship (5)



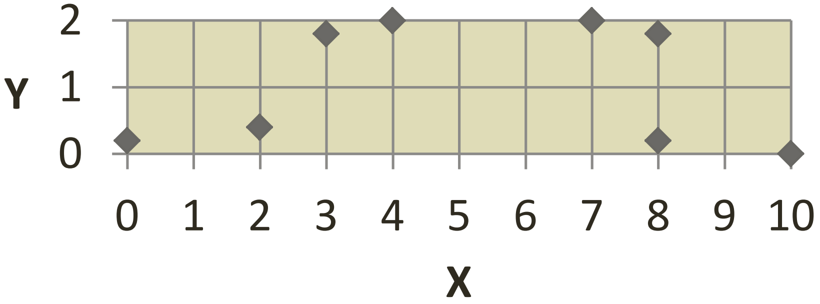
Normalize



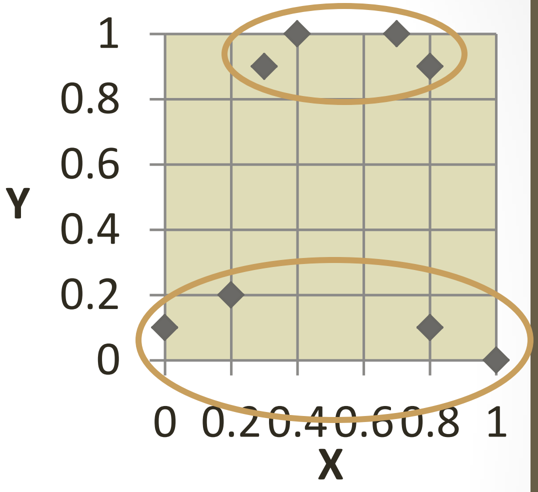
Non-normalized 2D data

Normalize the data:
Search for 2 Clusters

Normalization of a non-linear relationship (6)



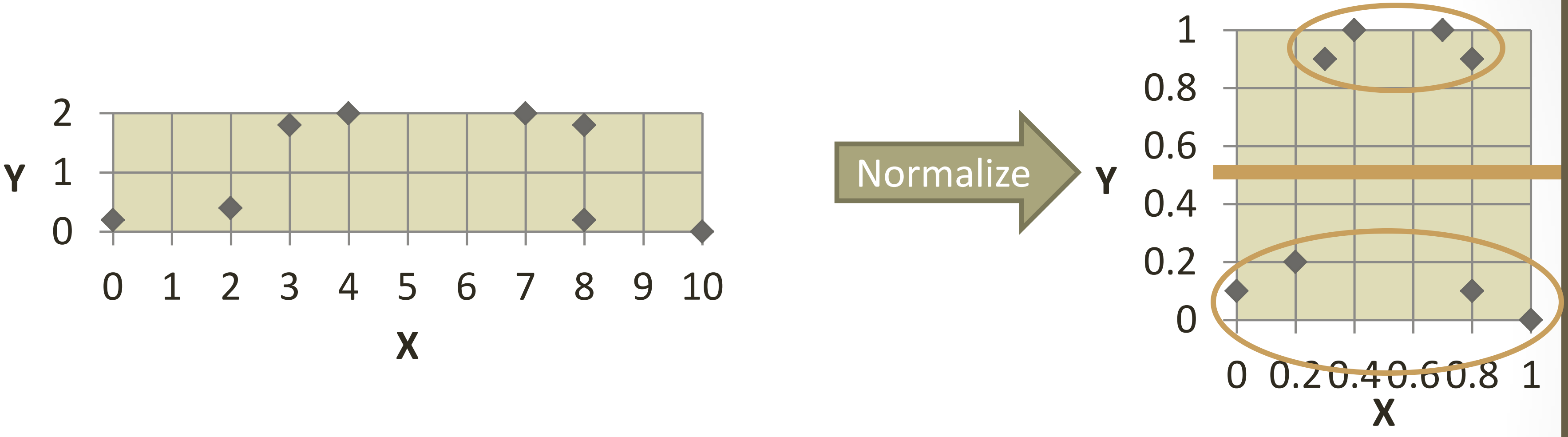
Normalize



Non-normalized 2D data

Found 2 Clusters in the normalized data

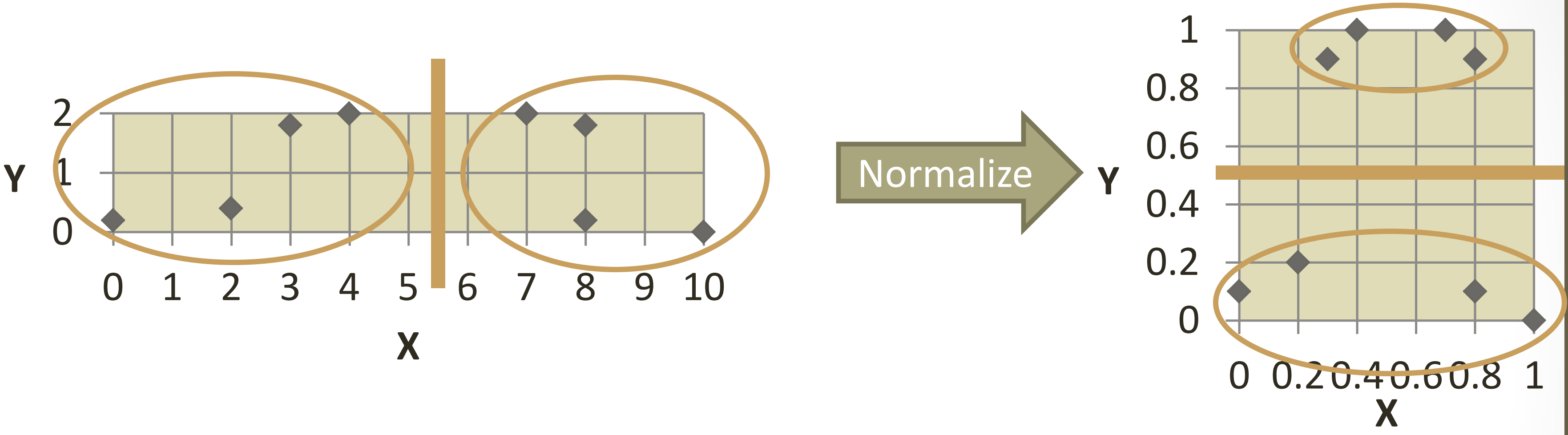
Normalization of a non-linear relationship (6)



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

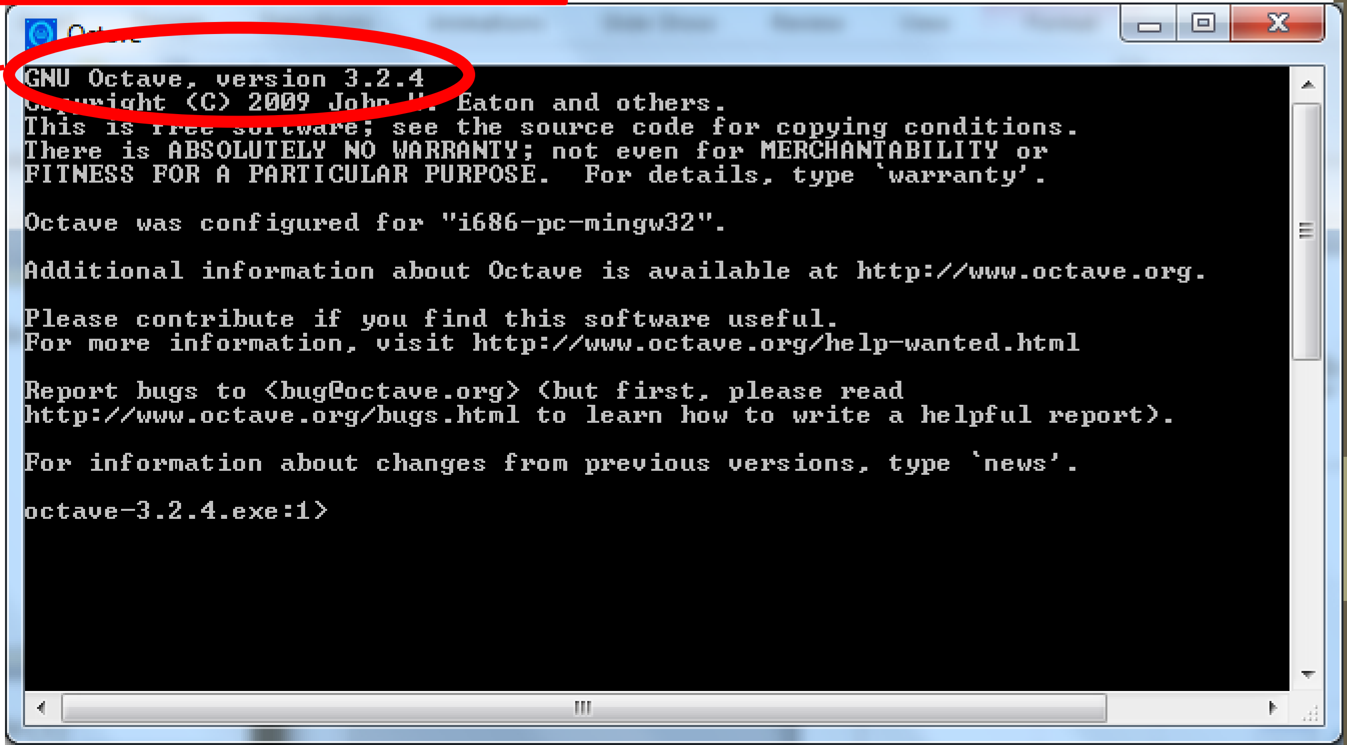
Introduction to Octave

Octave Console (0)

Start Octave

Octave

Octave console opens. Note version number

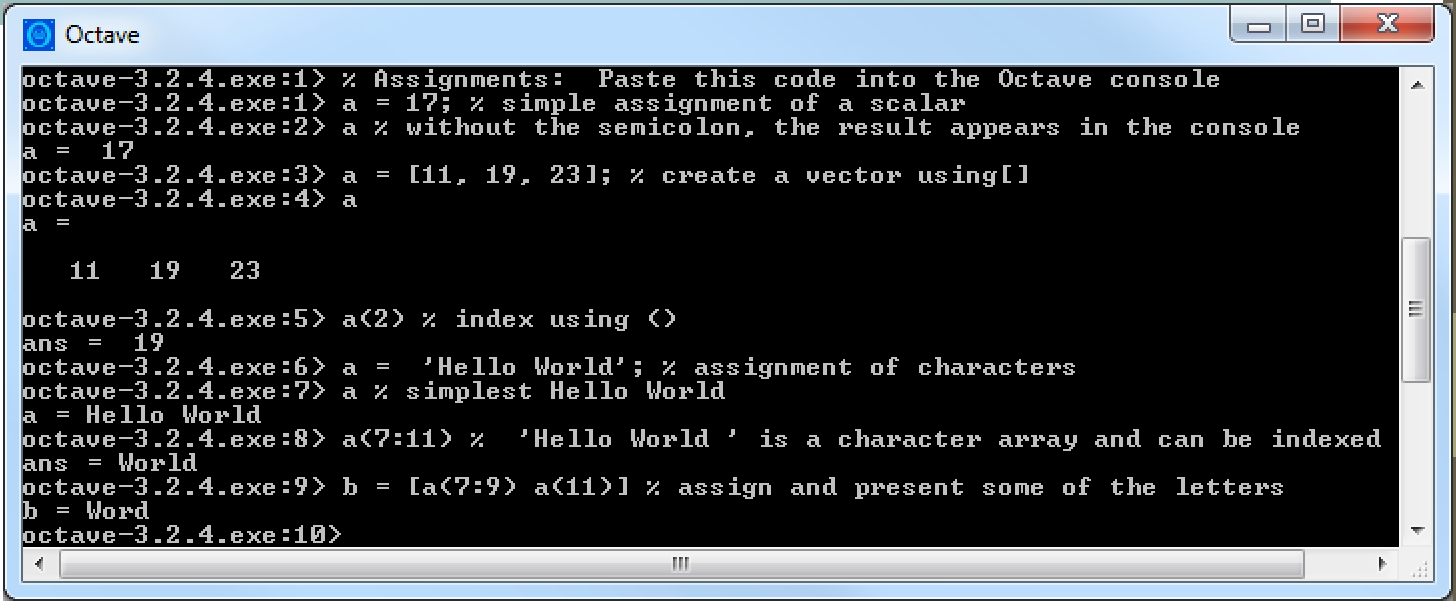


```
GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type `warranty'.

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>
```

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
```

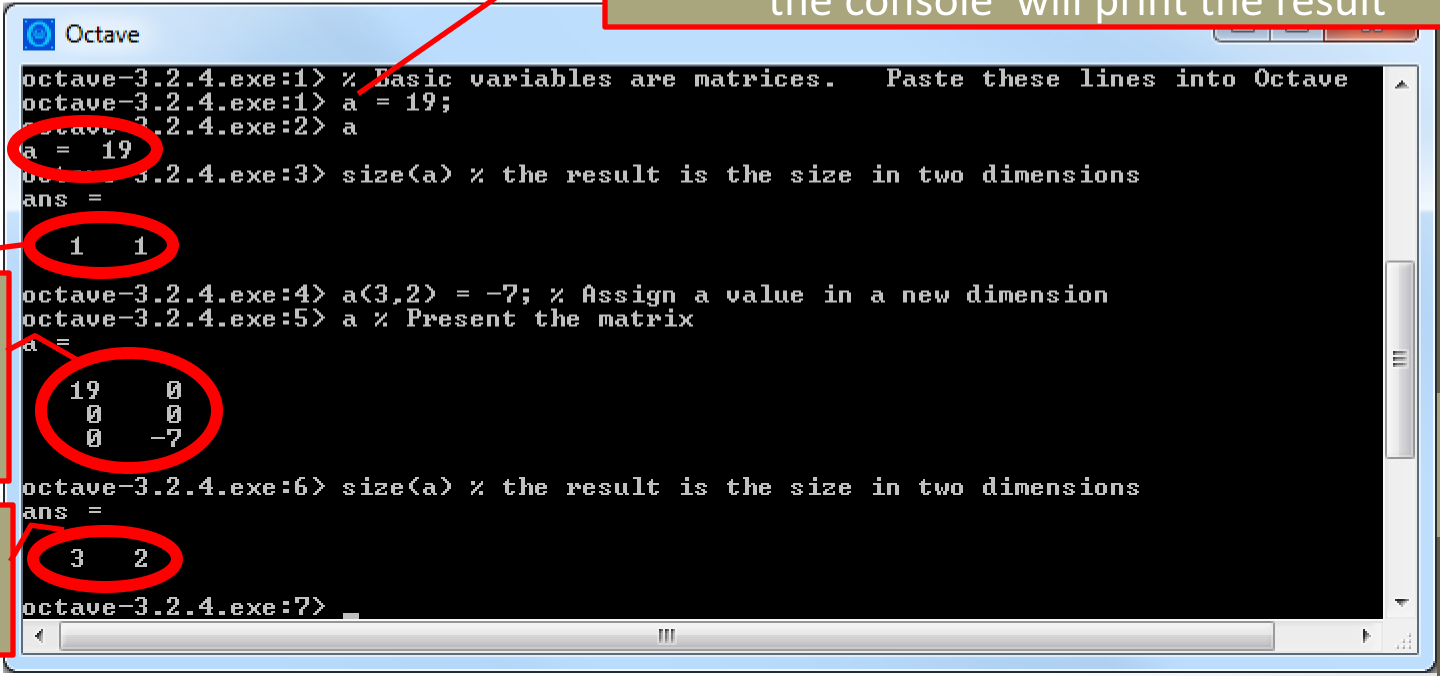


```
octave-3.2.4.exe:1> % Assignments:  Paste this code into the Octave console
octave-3.2.4.exe:1> a = 17; % simple assignment of a scalar
octave-3.2.4.exe:2> a % without the semicolon, the result appears in the console
a =  17
octave-3.2.4.exe:3> a = [11, 19, 23]; % create a vector using[]
octave-3.2.4.exe:4>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

```
% 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
    size(a) % the result is the size in two dimen
```

If the statement has no semi colon (;), then the console will print the result



```
octave-3.2.4.exe:1> % Basic variables are matrices.   Paste these lines into Octave
octave-3.2.4.exe:1> a = 19;
octave-3.2.4.exe:2> a
a =  19
octave-3.2.4.exe:3> size(a) % the result is the size in two dimensions
ans =

   1   1

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
a =

   19    0
    0    0
    0   -7

octave-3.2.4.exe:6> size(a) % the result is the size in two dimensions
ans =

   3   2

octave-3.2.4.exe:7> _
```

a = 19
Results in a 1 X 1 matrix

Extending the matrix fills in zeros

a is now a
3 X 2 matrix

Octave Console (3)

% Distinguish between a row vector and a column vector.

a = [11, 19, 23];

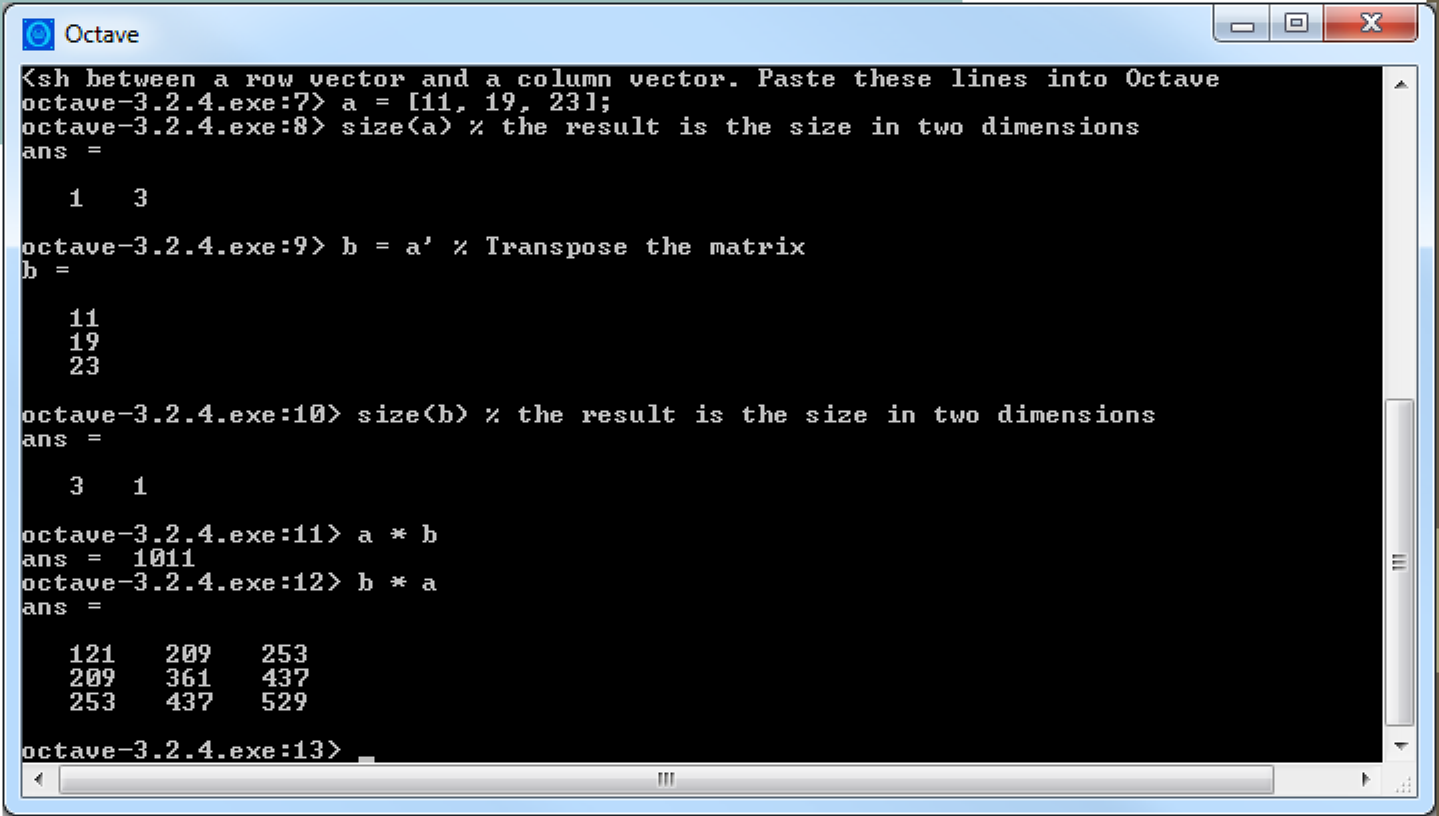
size(a) % the result is the size in two dimensions

b = a' % Transpose the matrix

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

a * b

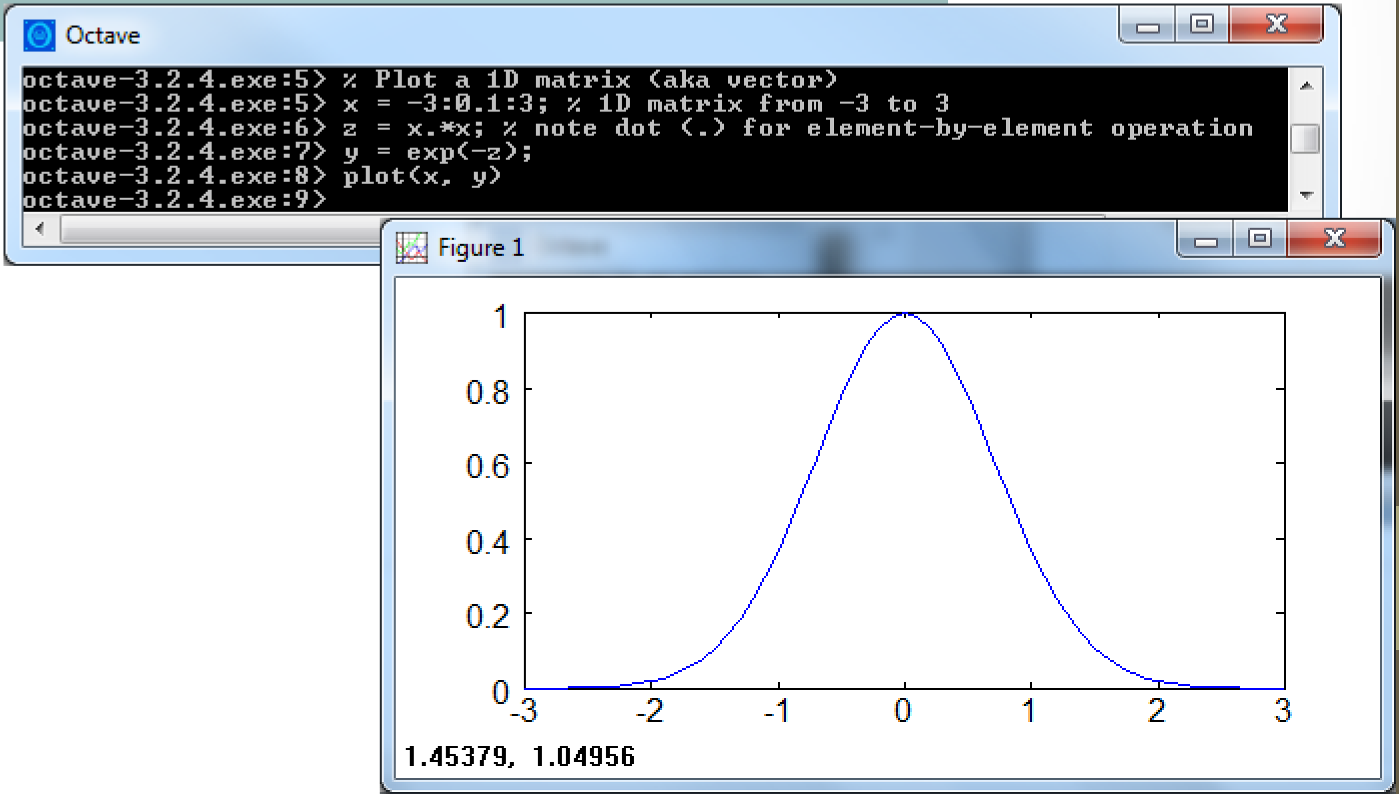
b * a



```
<sh 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
ans =

   1   3

octave-3.2.4.exe:9> b = a' % Transpose the matrix
b =

   11
   19
   23

octave-3.2.4.exe:10> size(b) % the result is the size in two dimensions
ans =

   3   1

octave-3.2.4.exe:11> a * b
ans =  1011
octave-3.2.4.exe:12> b * a
ans =

   121   209   253
   209   361   437
   253   437   529

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>
```

```
1.45379, 1.04956
```

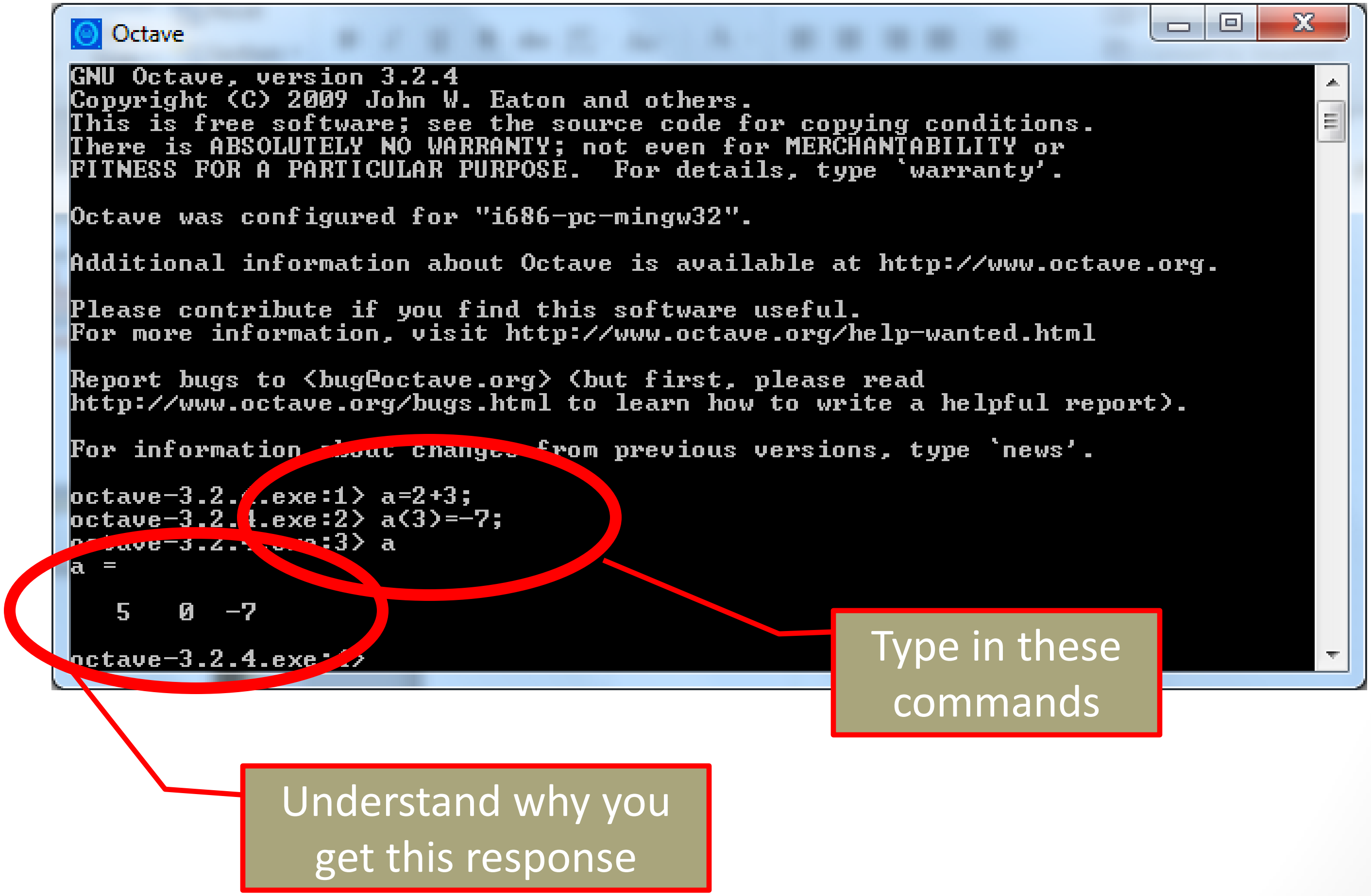
Binary operators in Octave

- 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

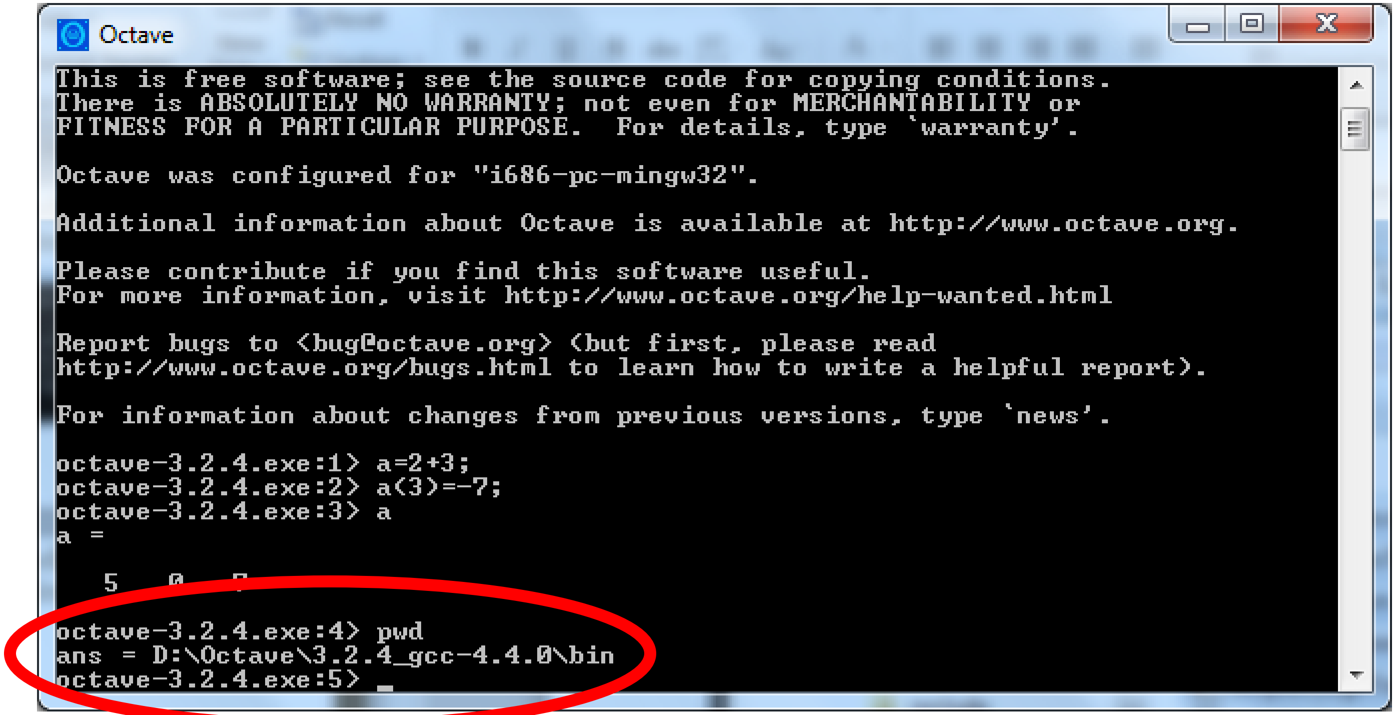


```
GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type `warranty'.

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
a =

   5   0  -7

octave-3.2.4.exe:4>
```

Type in these commands

Understand why you get this response

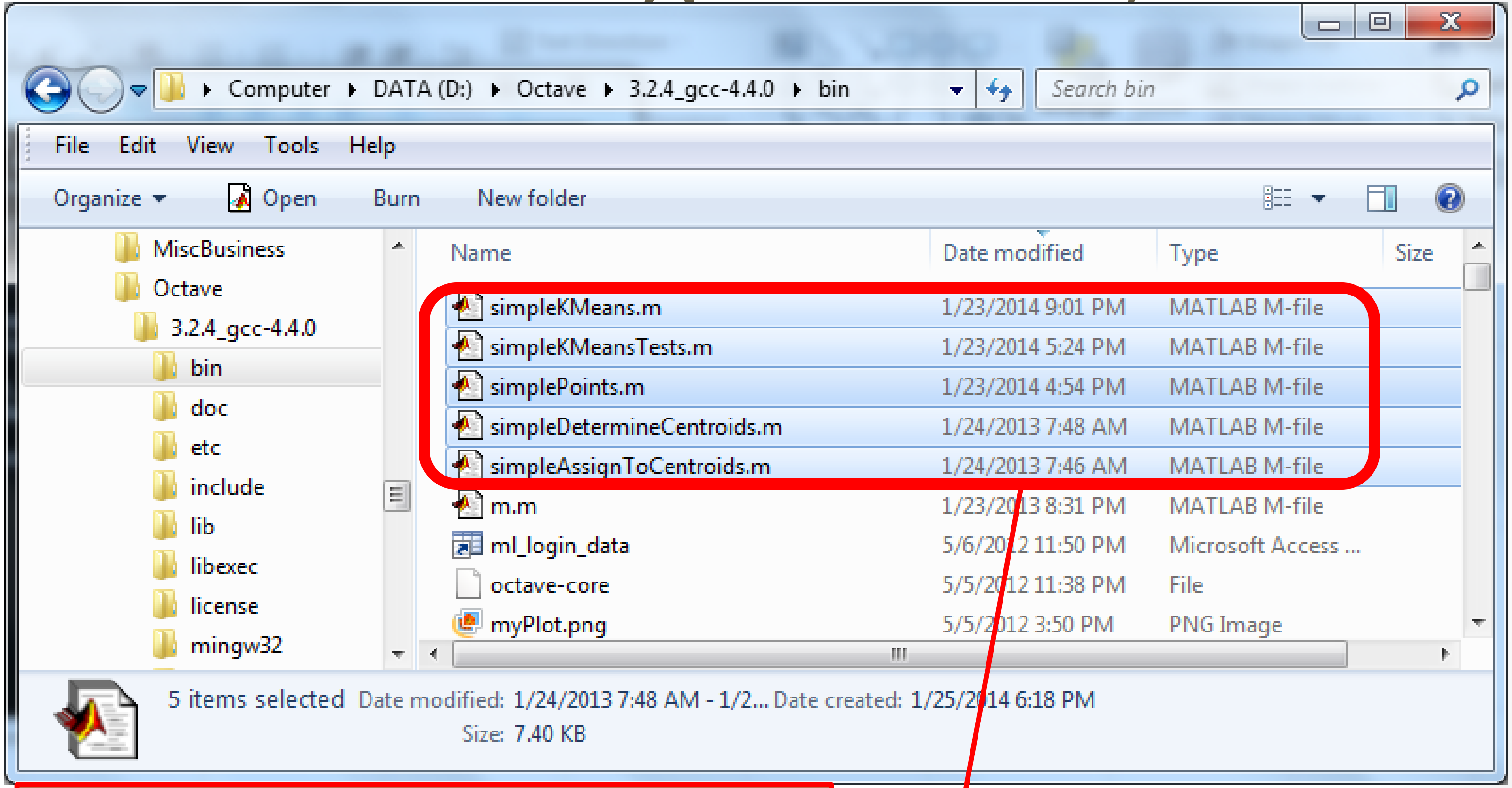
Octave m-files (2): Determine working directory



```
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
a =

   5   0  -7

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

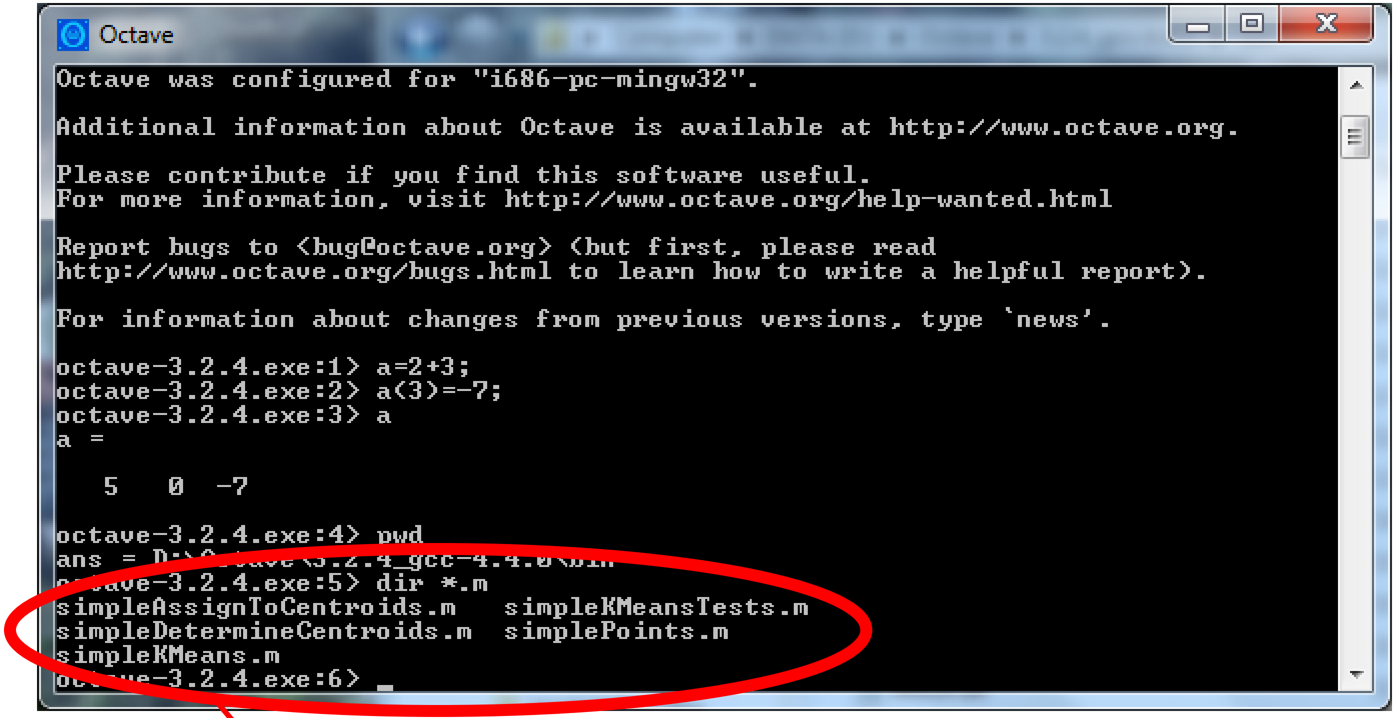


Name	Date modified	Type	Size
simpleKMeans.m	1/23/2014 9:01 PM	MATLAB M-file	
simpleKMeansTests.m	1/23/2014 5:24 PM	MATLAB M-file	
simplePoints.m	1/23/2014 4:54 PM	MATLAB M-file	
simpleDetermineCentroids.m	1/24/2013 7:48 AM	MATLAB M-file	
simpleAssignToCentroids.m	1/24/2013 7:46 AM	MATLAB M-file	
m.m	1/23/2013 8:31 PM	MATLAB M-file	
ml_login_data	5/6/2012 11:50 PM	Microsoft Access ...	
octave-core	5/5/2012 11:38 PM	File	
myPlot.png	5/5/2012 3:50 PM	PNG Image	

5 items selected Date modified: 1/24/2013 7:48 AM - 1/2... Date created: 1/25/2014 6:18 PM
Size: 7.40 KB

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

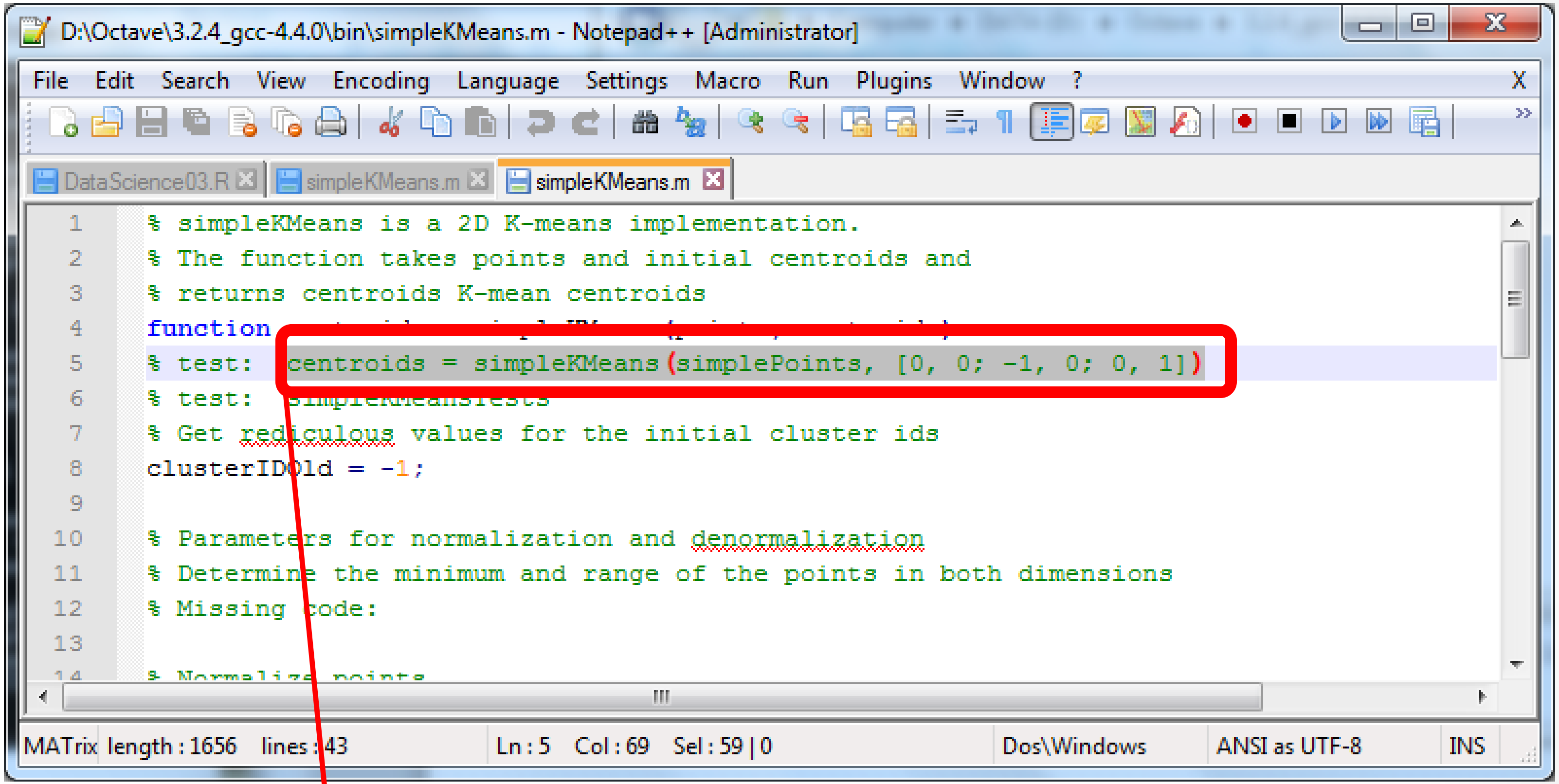
Octave m-files (4): Verify m-files in Working Directory



```
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
a =

   5   0  -7

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
simpleAssignToCentroids.m   simpleKMeansTests.m
simpleDetermineCentroids.m  simplePoints.m
simpleKMeans.m
octave-3.2.4.exe:6> _
```

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



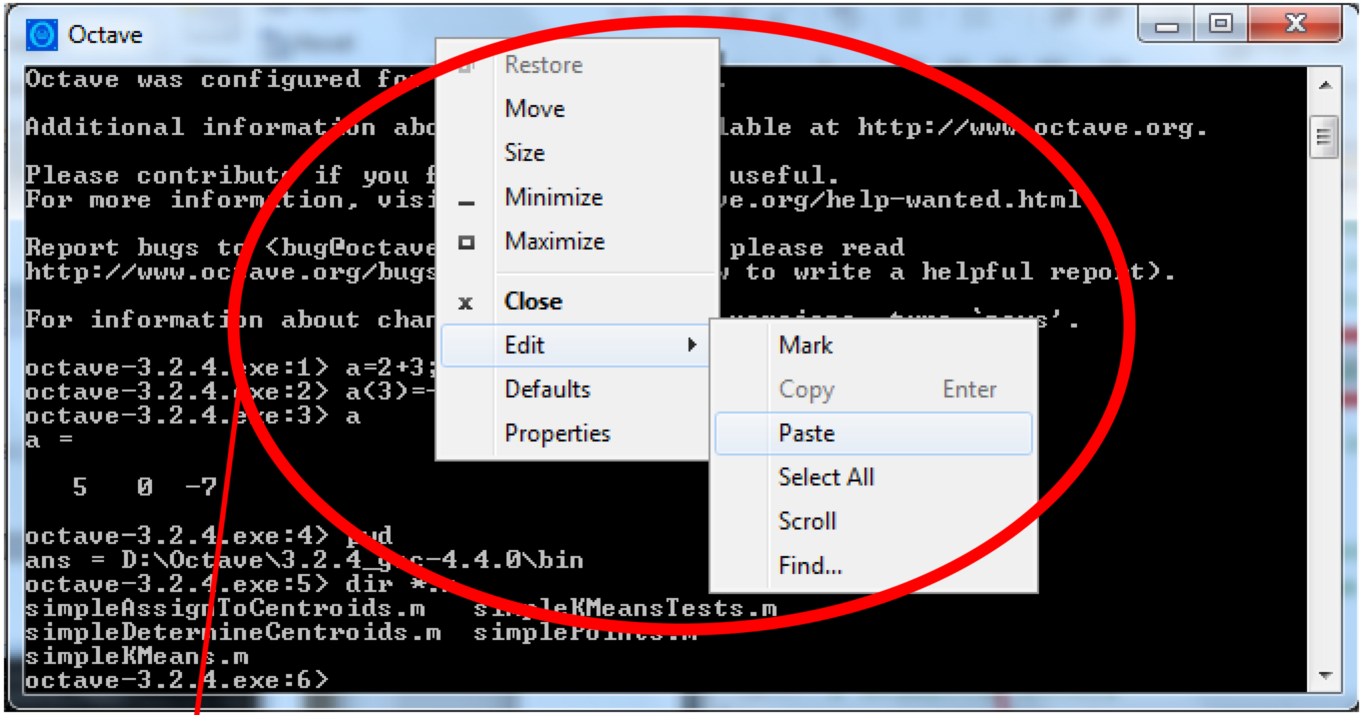
D:\Octave\3.2.4_gcc-4.4.0\bin\simpleKMeans.m - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
DataScience03.R
simpleKMeans.m
simpleKMeans.m

```
% 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;

% Parameters for normalization and denormalization
% Determine the minimum and range of the points in both dimensions
% Missing code:

% Normalize points
```

MATrix length : 1656 lines : 43 Ln : 5 Col : 69 Sel : 59 | 0 Dos\Windows ANSI as UTF-8 INS

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

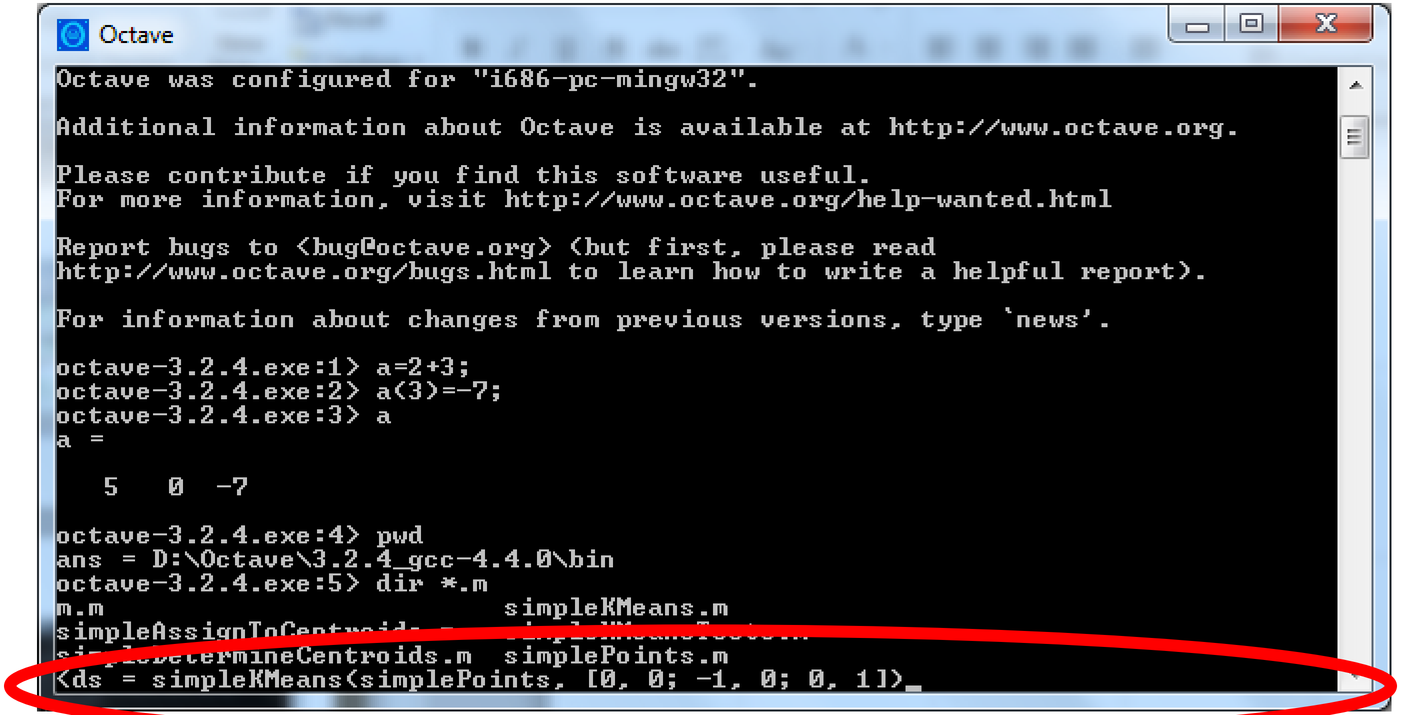


```
Octave was configured fo                  .

Additional informat on abo                lable at http://www octave.org.

Please contribut  if you f                useful.
For more inform tion, visi                e.org/help-wanted.html

Report bugs to <bug@octave                please read
http://www.oc ave.org/bugs                 to write a helpful repo t).

For informati n about char

octave-3.2.4. xe:1> a=2+3;
octave-3.2.4. xe:2> a(3)=-
octave-3.2.4. e:3> a
a =

   5   0  -7

octave-3.2.4 exe:4>  pwd
ans = D:\Oct ave\3.2.4_g c-4.4.0\bin
octave-3.2.4.exe:5> dir *.
simpleAssign ToCentroids.m   s    leKMeansTests.m
simpleDeter ineCentroids.m  simple
simpleKMean s.m
octave-3.2. 4.exe:6>
```

Paste the text from the clipboard into the Octave console. Alternately, type in:

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

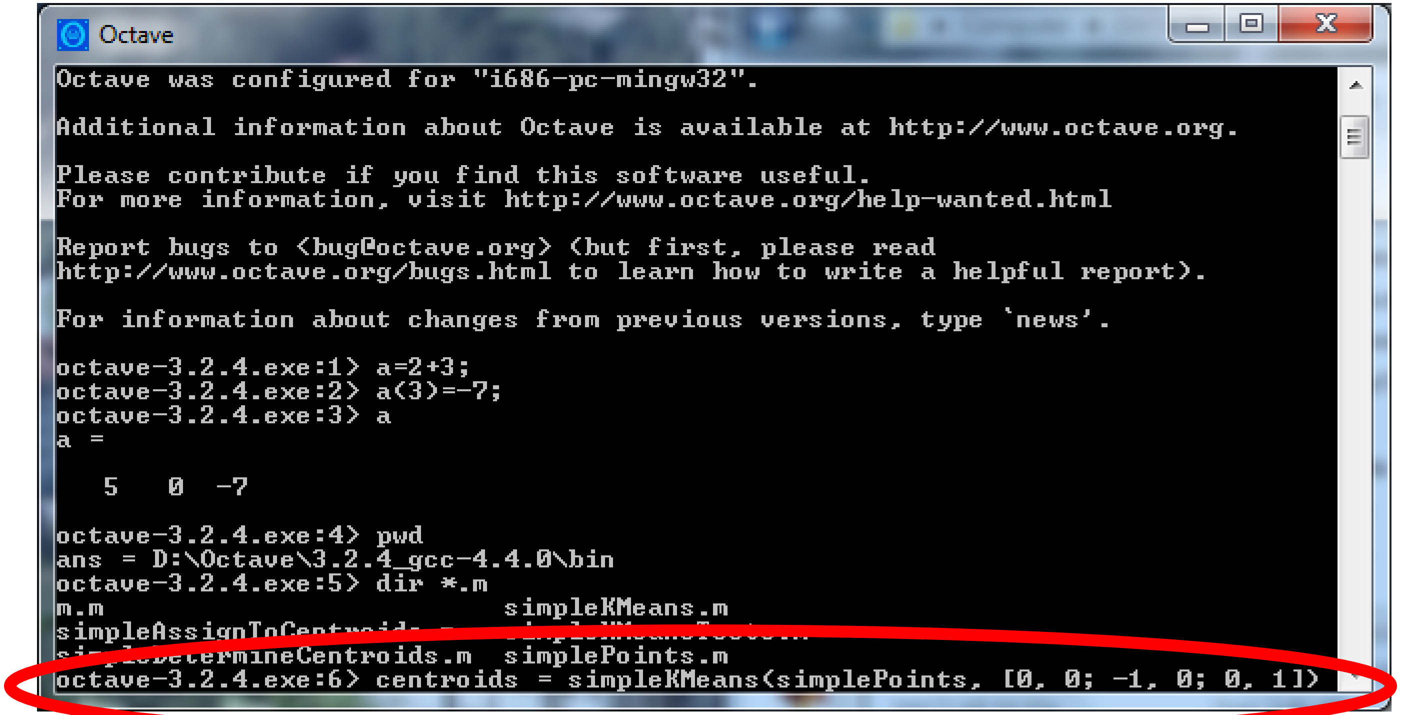
Octave m-files (7): Paste command into Octave console



```
Octave 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
a =

   5   0  -7

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
m.m                         simpleKMeans.m
simpleAssignToCentroids.m   simpleKMeansTest.m
simpleDetermineCentroids.m  simplePoints.m
(ds = 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.

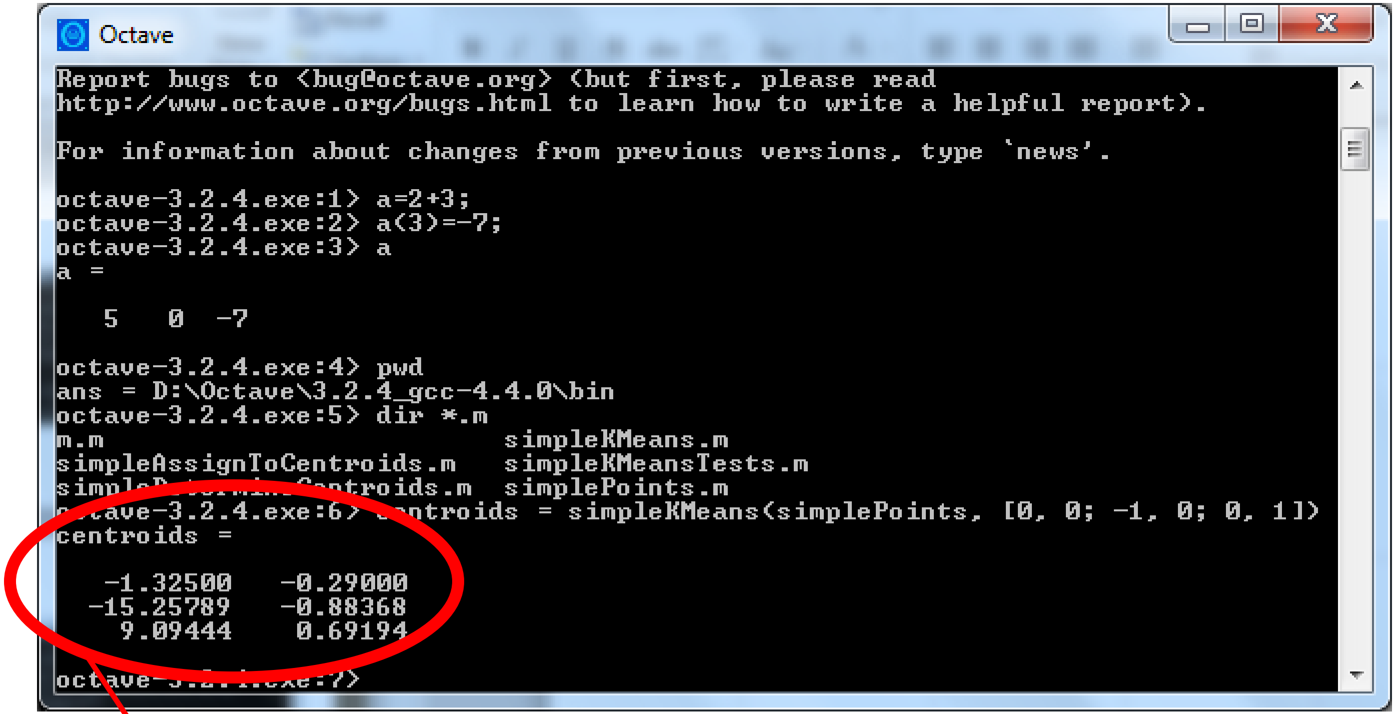
Octave m-files (8): Paste command into Octave console



```
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
a =

   5   0  -7

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
m.m                         simpleKMeans.m
simpleAssignToCentroids.m   simpleKMeansTest[illegible]
simpleDetermineCentroids.m  simplePoints.m
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

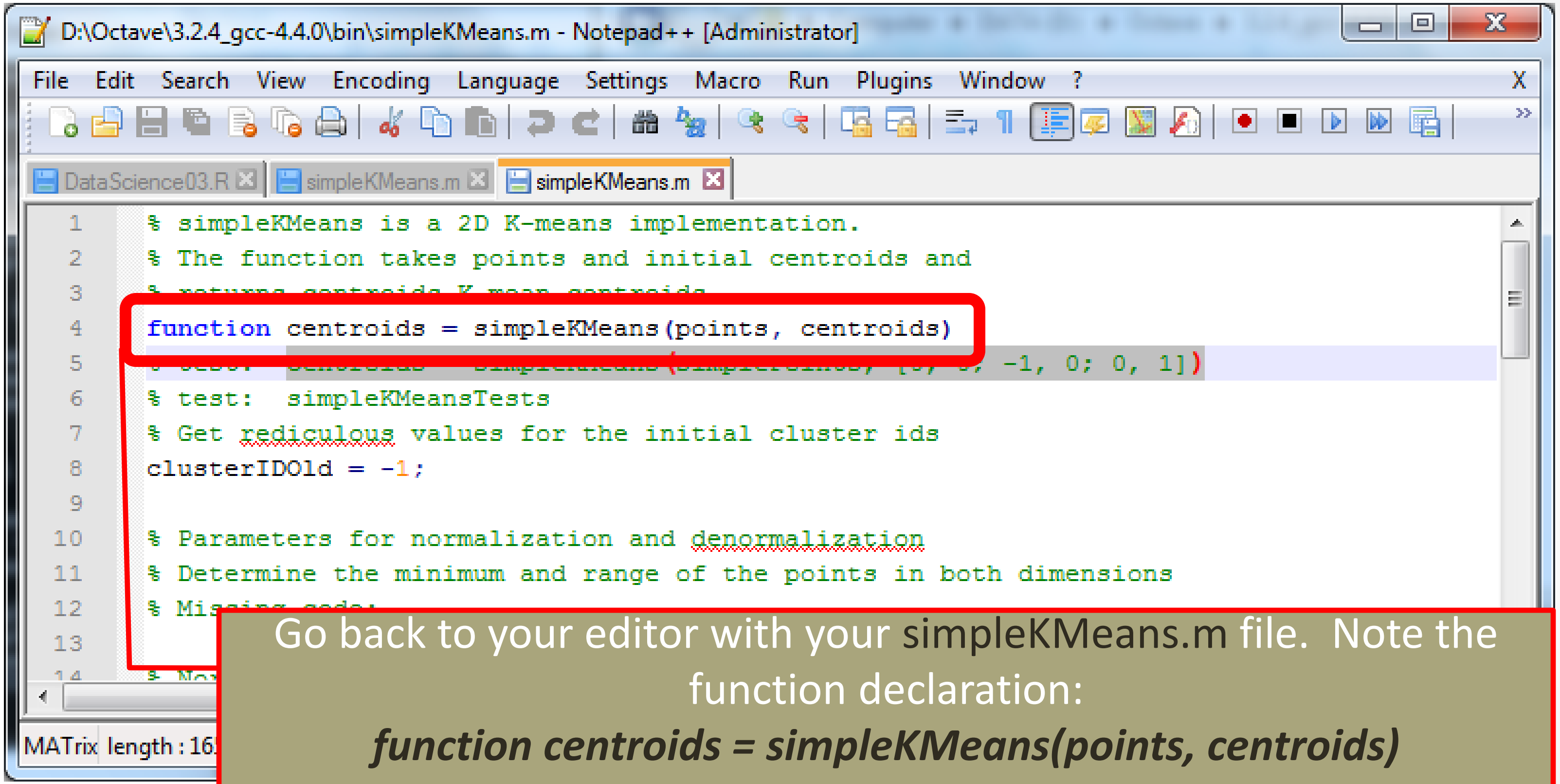


```
Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

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
a =

   5   0  -7

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
m.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    0.69194

octave-3.2.4.exe:7>
```

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



```
% 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;

% Parameters for normalization and denormalization
% Determine the minimum and range of the points in both dimensions
% Missing code:

% No
```

Go back to your editor with your simpleKMeans.m file. Note the function declaration:

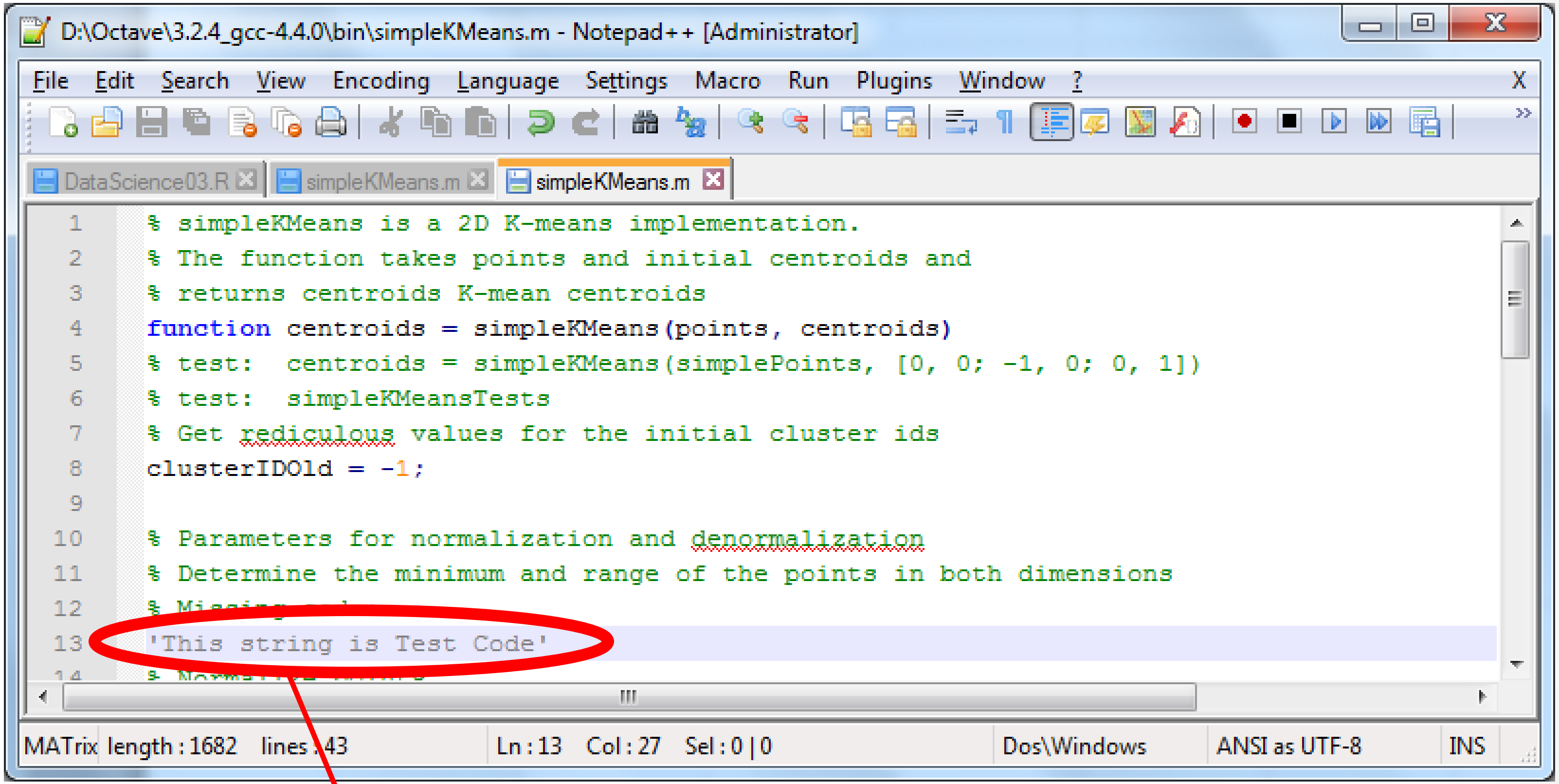
function centroids = simpleKMeans(points, centroids)

Note the name of the file:

simpleKMeans.M

You can learn how to construct a MATLAB or Octave function here: http://www.mathworks.com/help/matlab/ref/function.html

Octave m-files (11): Modify Octave Function



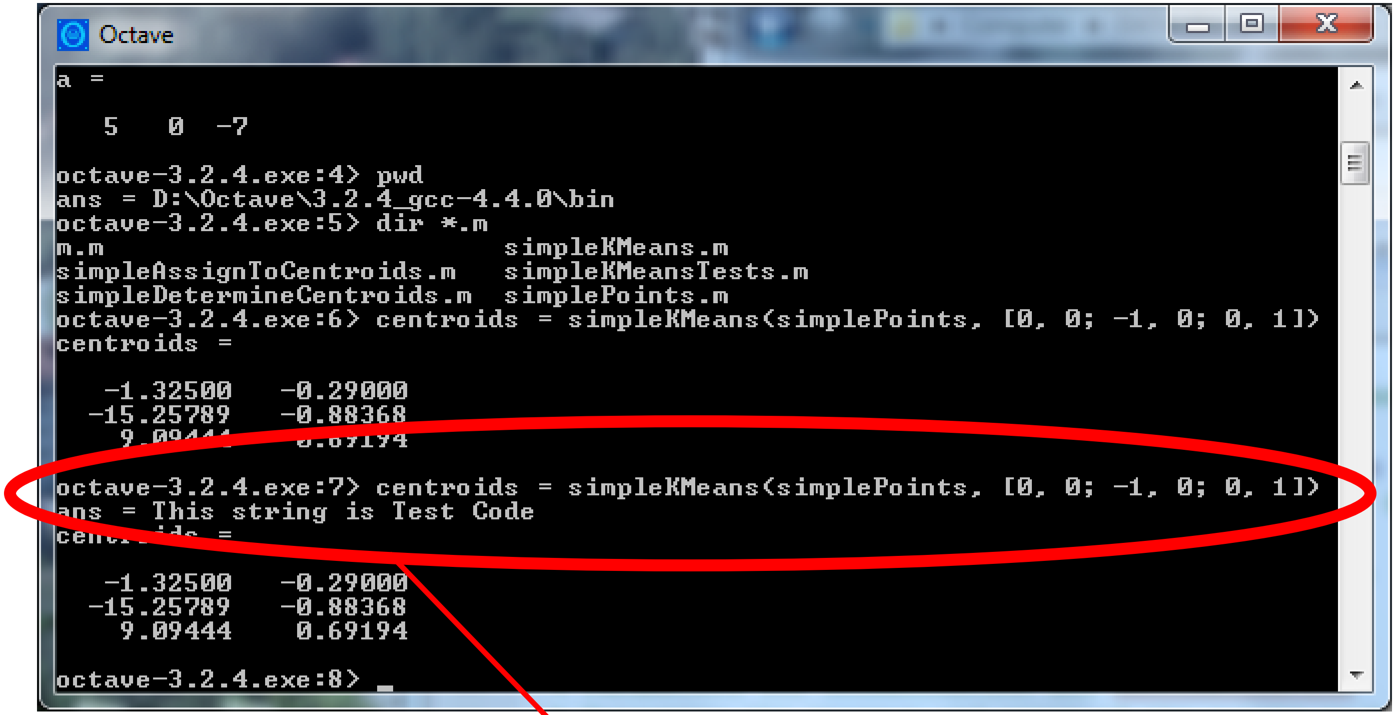
D:\Octave\3.2.4_gcc-4.4.0\bin\simpleKMeans.m - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
DataScience03.R
simpleKMeans.m
simpleKMeans.m

```
1  % simpleKMeans is a 2D K-means implementation.
2  % The function takes points and initial centroids and
3  % returns centroids K-mean centroids
4  function centroids = simpleKMeans(points, centroids)
5  % test:  centroids = simpleKMeans(simplePoints, [0, 0; -1, 0; 0, 1])
6  % test:  simpleKMeansTests
7  % Get rediculous values for the initial cluster ids
8  clusterIDOld = -1;
9
10 % Parameters for normalization and denormalization
11 % Determine the minimum and range of the points in both dimensions
12 % Missing code:
13 'This string is Test Code'
14 % Normalize points
```

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

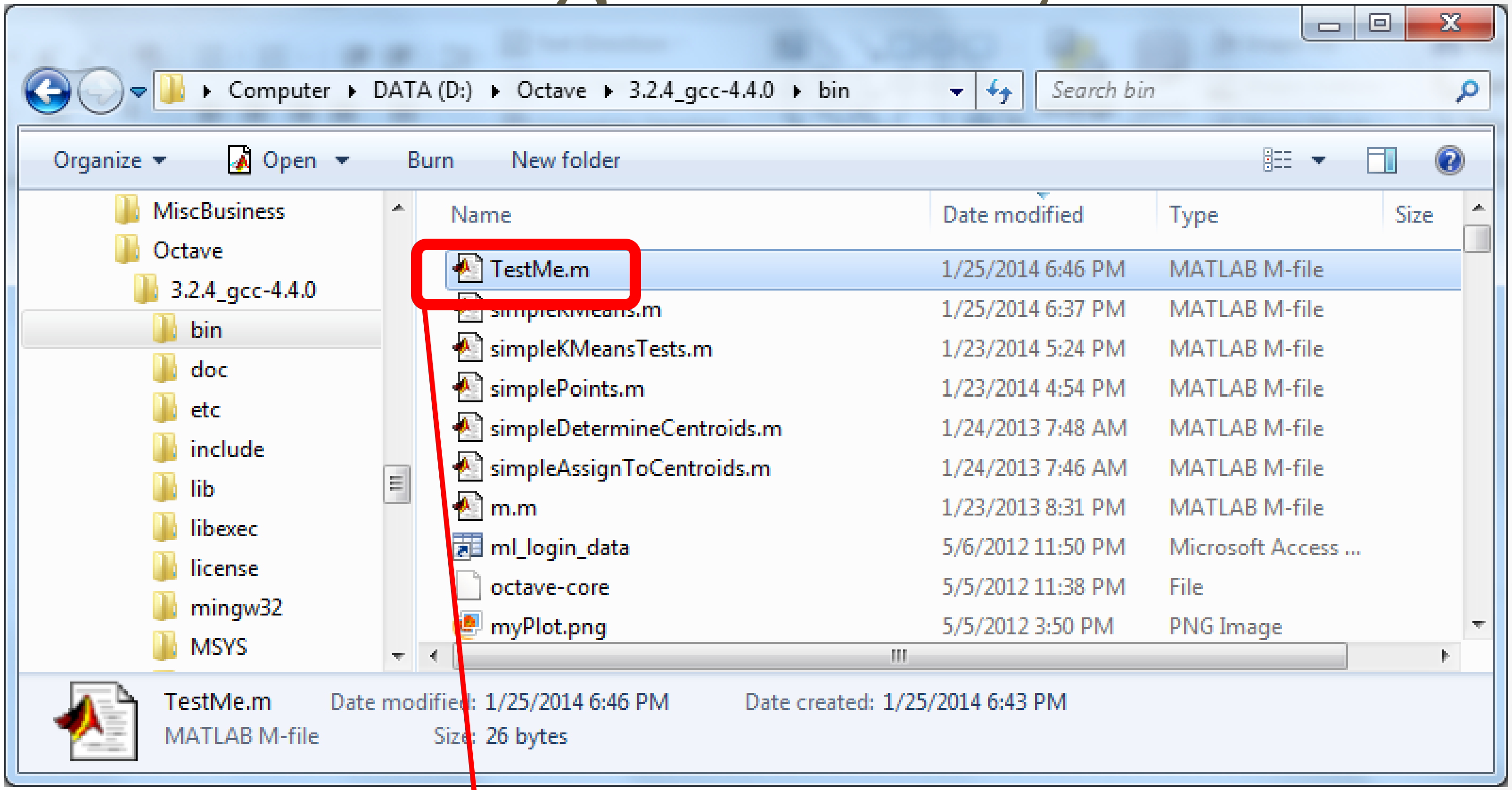


```
a =

   5   0  -7

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
m.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     0.69194

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     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

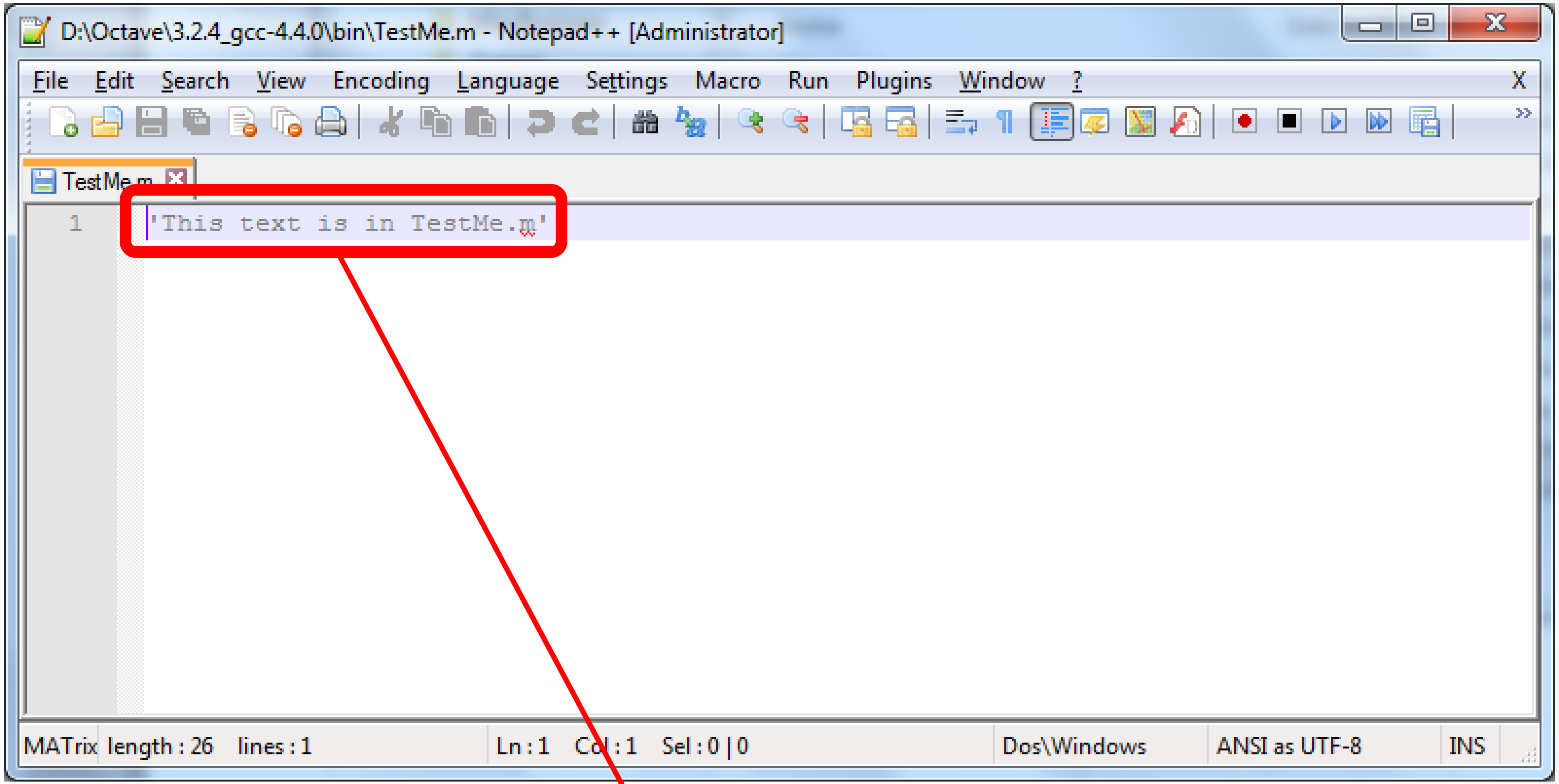


Name	Date modified	Type	Size
TestMe.m	1/25/2014 6:46 PM	MATLAB M-file	
simpleKMeans.m	1/25/2014 6:37 PM	MATLAB M-file	
simpleKMeansTests.m	1/23/2014 5:24 PM	MATLAB M-file	
simplePoints.m	1/23/2014 4:54 PM	MATLAB M-file	
simpleDetermineCentroids.m	1/24/2013 7:48 AM	MATLAB M-file	
simpleAssignToCentroids.m	1/24/2013 7:46 AM	MATLAB M-file	
m.m	1/23/2013 8:31 PM	MATLAB M-file	
ml_login_data	5/6/2012 11:50 PM	Microsoft Access ...	
octave-core	5/5/2012 11:38 PM	File	
myPlot.png	5/5/2012 3:50 PM	PNG Image	

TestMe.m Date modified: 1/25/2014 6:46 PM Date created: 1/25/2014 6:43 PM
MATLAB M-file Size: 26 bytes

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.

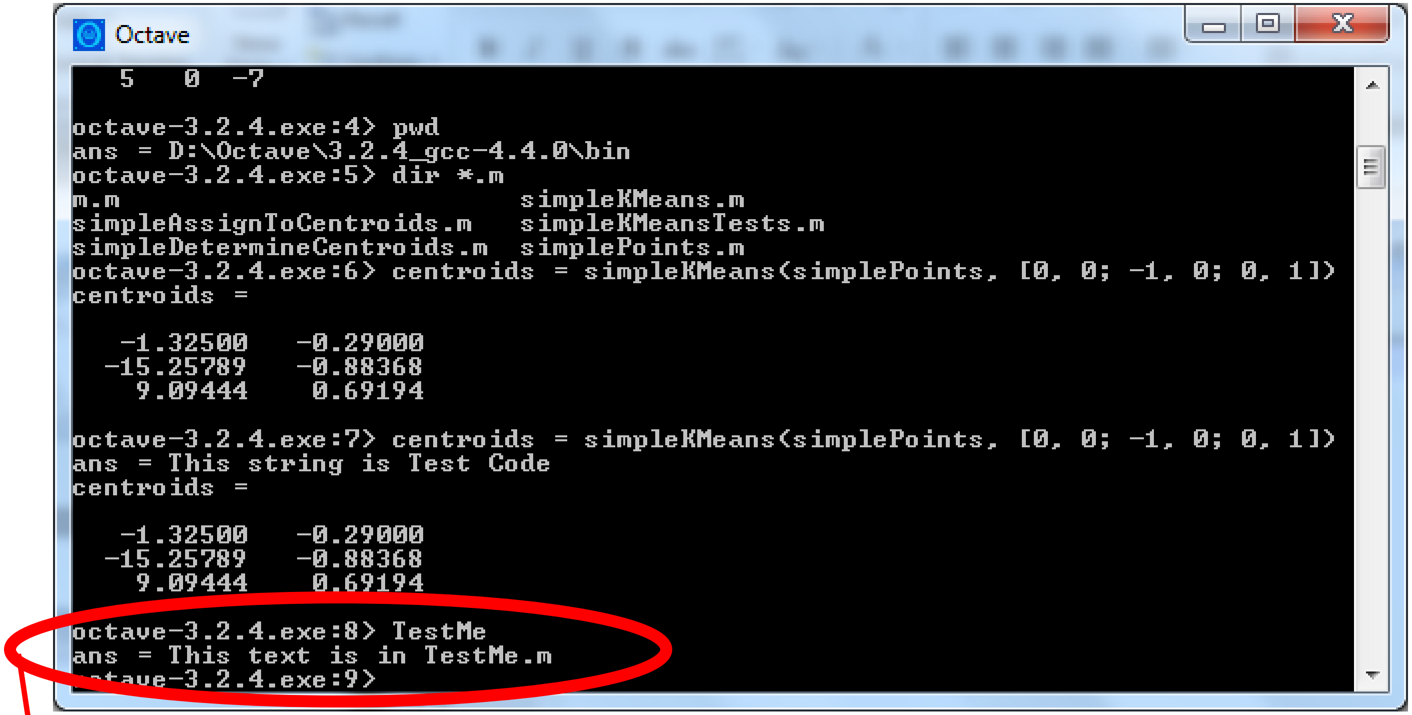

D:\Octave\3.2.4_gcc-4.4.0\bin\TestMe.m - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
TestMe.m

```
'This text is in TestMe.m'
```

MATrix length : 26 lines : 1 Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows ANSI as UTF-8 INS

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



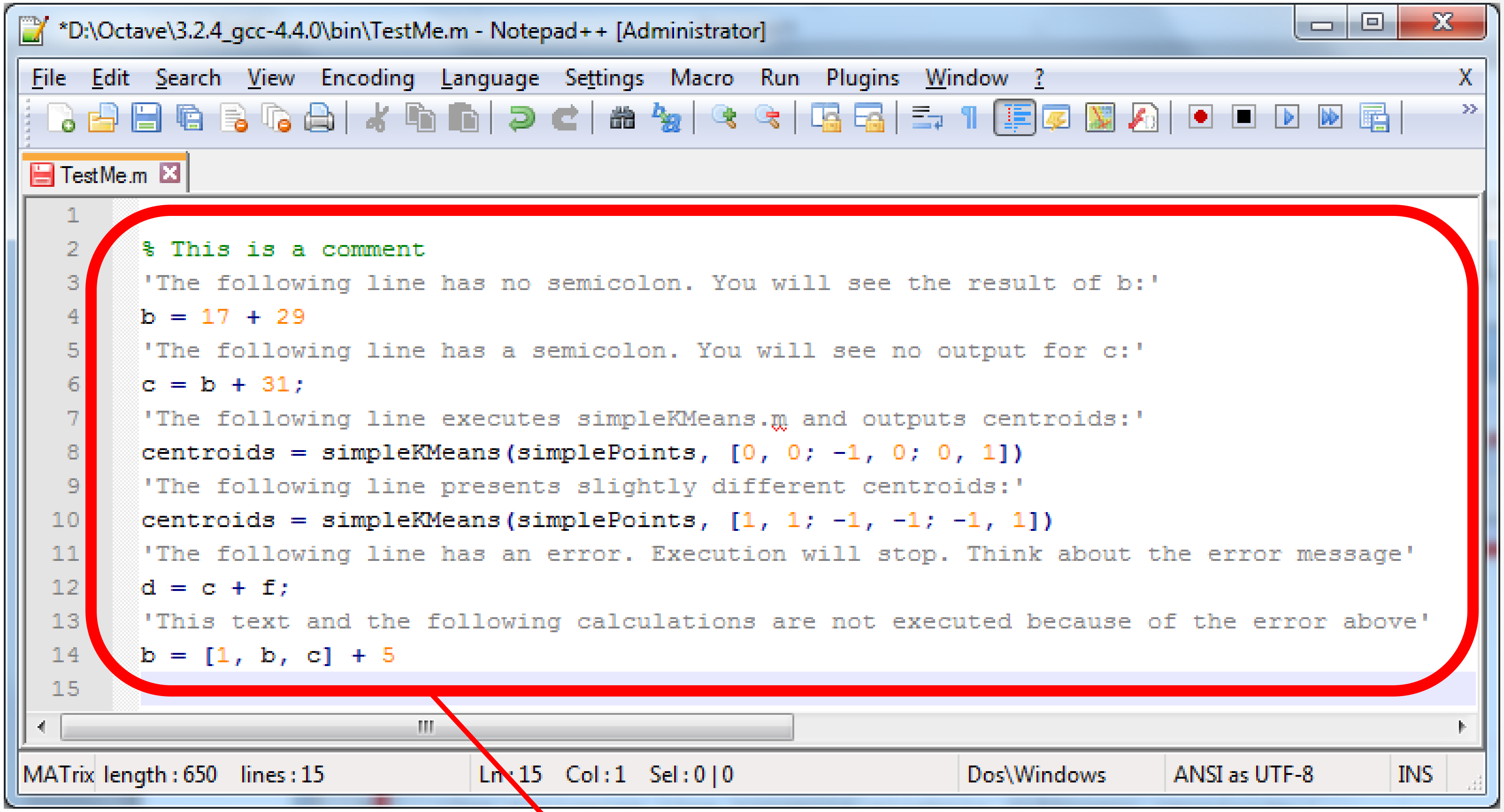
```
   5   0  -7

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
m.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    0.69194

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    0.69194

octave-3.2.4.exe:8> TestMe
ans = This text is in TestMe.m
octave-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
```

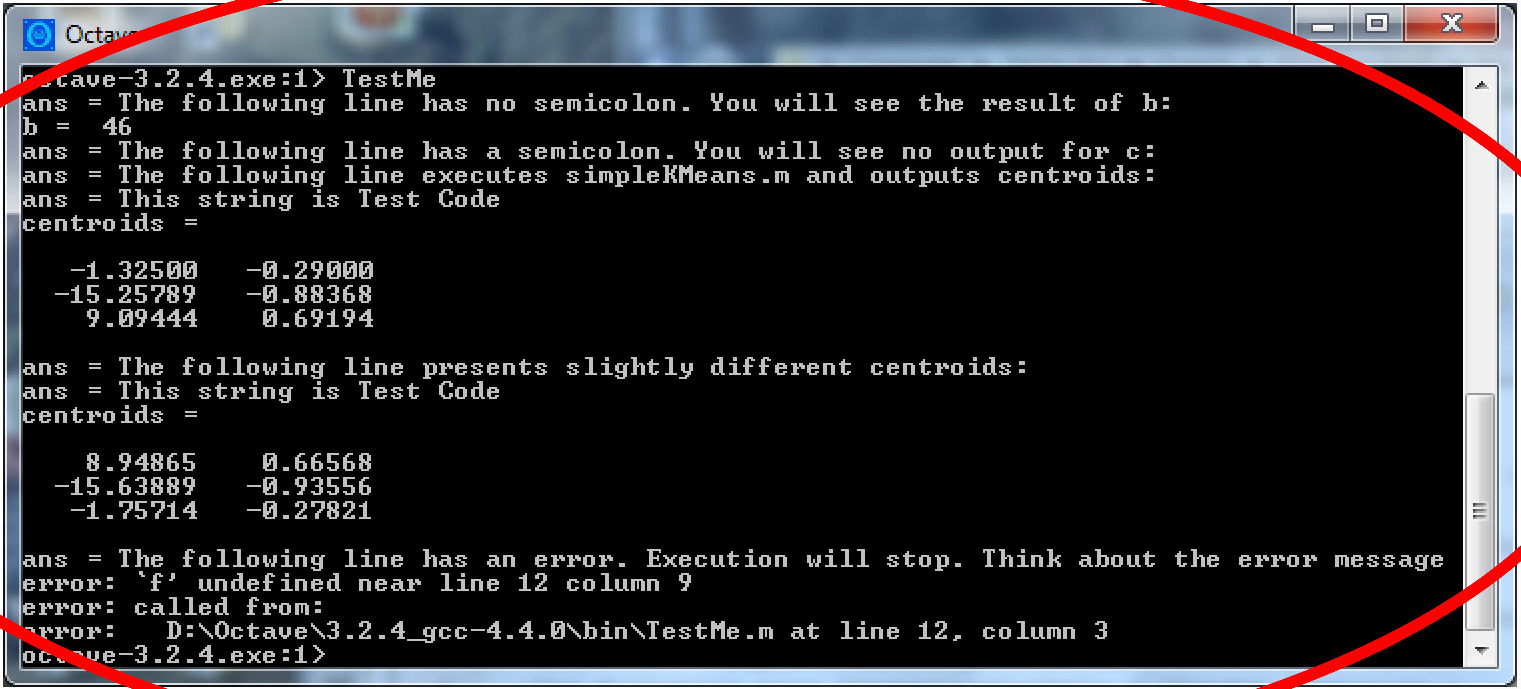
Octave (5): Edit your m-file


*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
MATrix length : 650 lines : 15
Ln : 15 Col : 1 Sel : 0 | 0
Dos\Windows
ANSI as UTF-8
INS

```

% 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 has an error. Execution will stop. Think about the error message'
d = c + f;
'This text and the following calculations are not executed because of the error above'
b = [1, b, c] + 5

```

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

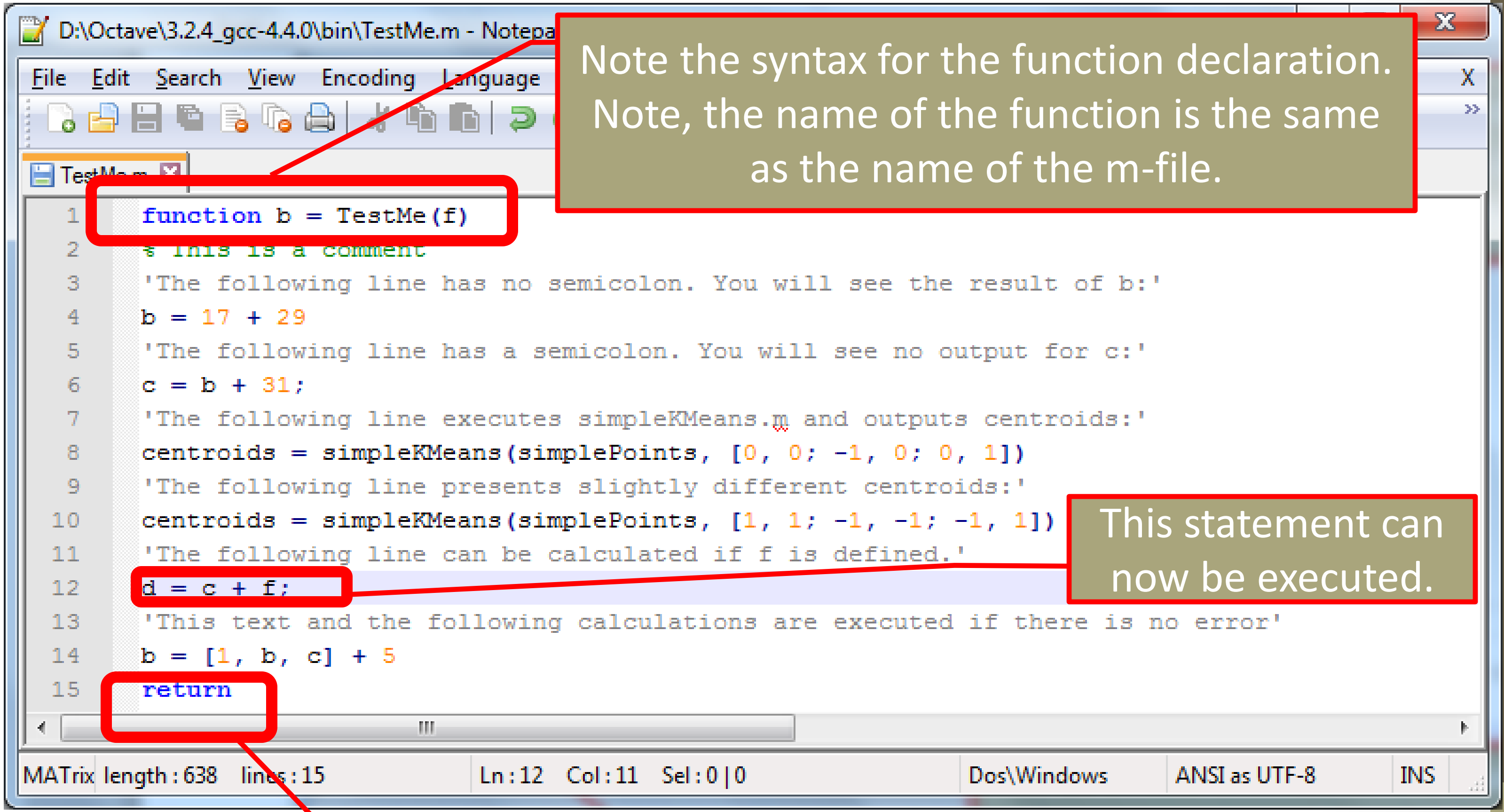
Octave (6): Run your m-file. Read the error code.



```
octave-3.2.4.exe:1> TestMe
ans = The following line has no semicolon. You will see the result of b:
b =  46
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
  -15.63889   -0.93556
   -1.75714   -0.27821

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:
error:    D:\Octave\3.2.4_gcc-4.4.0\bin\TestMe.m at line 12, column 3
octave-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


D:\Octave\3.2.4_gcc-4.4.0\bin\TestMe.m - Notepa
File Edit Search View Encoding Language
TestMe.m
MATrix length : 638 lines : 15
Ln : 12 Col : 11 Sel : 0 | 0
Dos\Windows
ANSI as UTF-8
INS

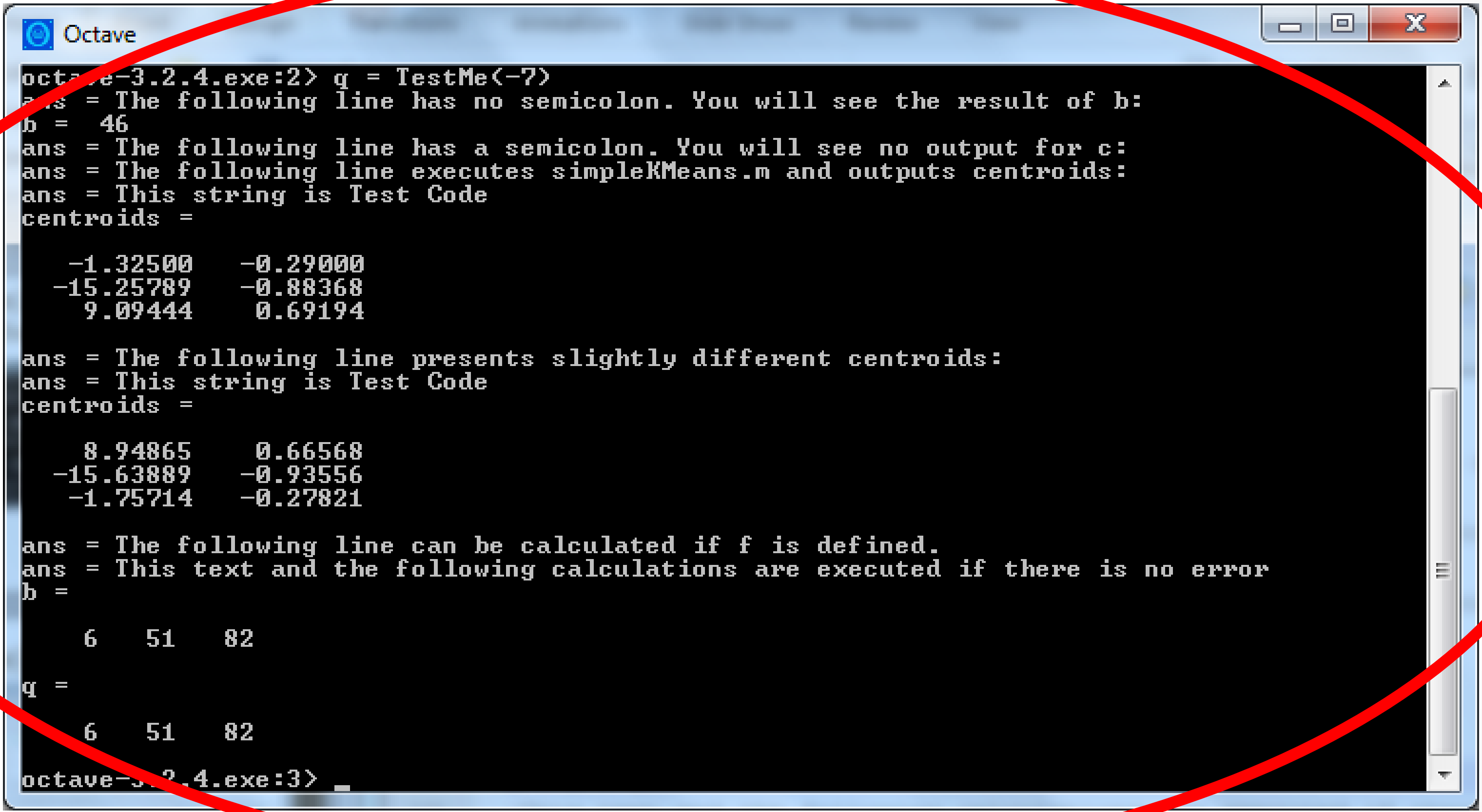
```
function b = TestMe(f)
% This is a comment
'The following line has no semicolon. You will see the result of b:'
b = 17 + 29
'The following line has a semicolon. You will see no output for c:'
c = b + 31;
'The following line executes simpleKMeans.m and outputs centroids:'
centroids = simpleKMeans(simplePoints, [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
return
```

Note the syntax for the function declaration. Note, the name of the function is the same as the name of the m-file.

This statement can now be executed.

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

Octave (8): Relate Output to Code



```
octave-3.2.4.exe:2> q = TestMe(-7)
ans = The following line has no semicolon. You will see the result of b:
b =  46
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
  -15.63889   -0.93556
   -1.75714   -0.27821

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
b =

    6   51   82

q =

    6   51   82

octave-3.2.4.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 an input argument? 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