

Matlab Visualisation

Yaara Erez

MRC Cognition and Brain Sciences Unit

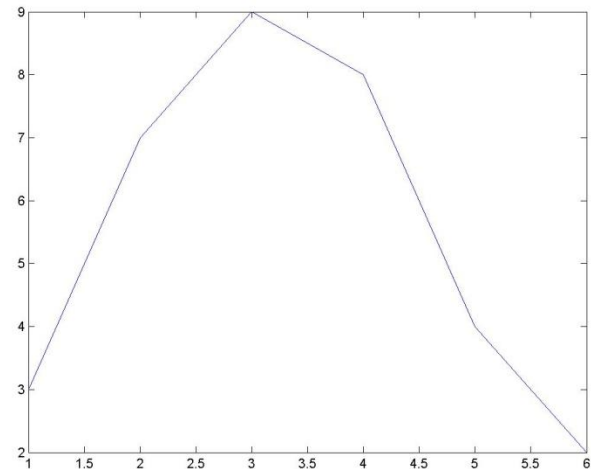
November 2014

Matlab visualisation

- 2D graphics: plot, histogram
- Closing and saving figures
- Accessories
- Copying figures to other documents
- Interactive editing
- 3D graphics
- Handle graphics objects

Plot

- Drawing a curve – Y values versus X values
 - X and Y are variables.
 - Pairs of points: (X1, Y1) (X2, Y2) (X3, Y3)...
- Example:
a = [1 2 3 4 5 6]
b = [3 7 9 8 4 2]
plot(a, b)
- exampleGraphics2D.

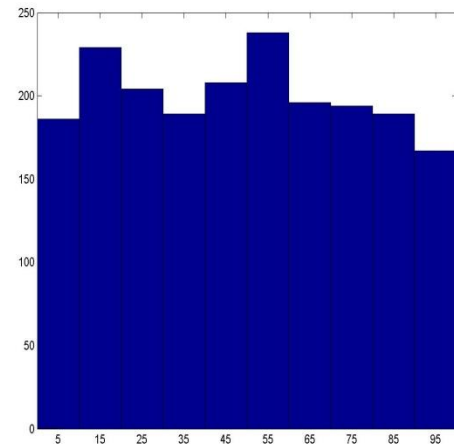


Plot – additional commands

- Hold
- Style
- Subplot

Histogram

- Displaying bars that indicate the number of cases/values in each bin/range.
- Plotting a histogram is a 2-step process:
 - Calculate distribution – how many cases/values fall in each bin.
 - Plot the distribution.



Histogram

- Syntax:

`hist(data)`

- The data is divided into 10 bins according to its range of values.
- Calculates the distribution and plots the diagram

- Another option:

`count = hist(data)`

- The data is divided in the same way.
- The distribution is calculated and returned to 'count' instead of plotting.
- The distribution can be plotted using 'bar'.

Histogram - options

- Set the number of bins to n: `hist(data, n)`
- Set bin centers to c: `hist(data, c)`
 - C is a vector of bin centers.
- Get the bin centers: `[count c] = hist(...)`
- Histogram of a matrix – according to columns (the values of each column are divided into bins).
 - Each bin has a few bars, different color for each column.
- `n = histc(x, edges)`

The figure window

- Matlab displays plots on the current figure window.
- We can add a new figure window, and display the new plot on it.
- 'figure' – opens a new figure window to display plot on.
- 'clf' – clear figure.
- 'close' and 'close all' – close figures.

Additional 2D graphs

- bar, stairs, scatter, pie,...
- help **specgraph**

Practice 1 – basic graphics

- Create 2 vectors of the same size. Plot them one versus the other.
- Create another vector of the same size. Plot it versus the first vector on the same graph (use 'hold on').
- Create a data vector of 50 random grades between 0 and 100.
- Create a new figure and plot a histogram of the grades.
- Plot a histogram of the grades again, this time divide the data into 5 bins.

Accessories

- Endless options...
- title
- xlabel
- ylabel
- axis – change or get the axis.
- grid – on/off/minor.
- legend
 - Syntax: legend('text1', 'text2')
 - Example:
 legend('experiment', 'control')
 - legend off → Remove legend.

Text

- Syntax: `text(x, y, 'text')`
- Text properties:
 - `text(x, y, 'text', 'P-Name', P-Val)`
 - Example:
`text(10, 15, 'peak', 'FontSize', 10)`
- The text properties can be applied also to `title`, `xlabel`, `ylabel`.

Text - Specials

- Subscript letters/digits – use underscore (`_`) before the letter.
Example: `title('mat_1')` → the title will be `'mat1'`
- Superscript letters/digits – use `^` before the letter/digits.
Example: `title('mat^1')` → the title will be `'mat1'`
- An underscore is required before each letter/digit.
- Greek letters – use backslash and the name of the letter.
Example: `title('\alpha')` → the title will be `'α'`

Practice 2 - accessories

- Create 2 vectors of the same size. Plot them one versus the other.
- Add title, x-label, y-label.
- Add grid.
- Create another vector of the same size. Plot it versus the first vector on the same graph (use 'hold on').
- Add legend.

Interactive editing

- A figure can be interactively edited through the figure window.
- Zoom in, zoom out, reset zoom.
- Menu options: xlabel, ylabel, title, figure properties, ...
- Generating m-code.

Printing and Saving Plots

- Print
 - ‘print’ icon
 - File → Print
- Save
 - ‘save’ icon
 - File → Save (or ‘Save As’)
 - ‘saveas’ function (next lesson)
- When saving, default extension is ‘.fig’ – can be opened only from Matlab.
 - A different extension can be chosen: jpg, bmp, pdf.
- One can copy and paste a figure to another program.
 - Edit → Copy Figure

3D Graphics

- Display 3 dimensional data.
- `exampleGraphics3D.m`
- `help graph3d`

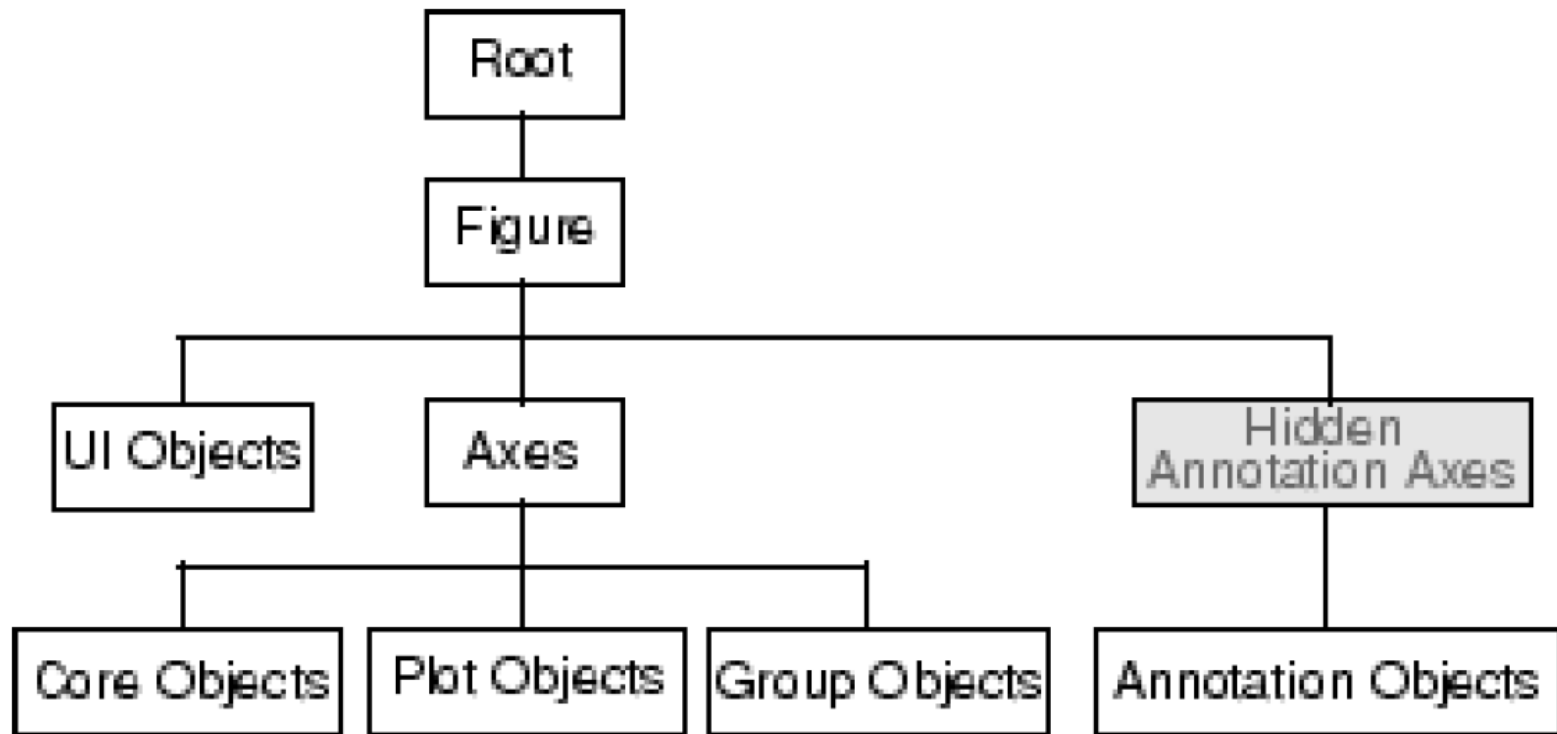
Handle graphics objects

- Figures and graphs can be viewed as **objects** in Matlab.
 - Types of graphic objects: line, figure, axes, text, etc.
- We can gain direct access to objects and manipulate their **properties** by using **handles**.
- A **handle** is a unique number that acts as an “**ID**” to an object.
 - A handle is meaningful only as long as the object exists.
- Example: `h = figure();`

Properties

- Each type of object has many properties.
- Properties constitute of **name & value** pairs.
 - Property names are their identifier and are always strings.
 - Property values may be of any data type, depending on the property, including other handles.
- Access to properties:
 - `get(h)` → all properties of h are displayed
 - `get(h,propName)` → display a specific property of h
 - `set(h,propName,propValue)` → set a new value to a property of h
- `graphicsHandles.m`

Handle hierarchy



Practice 3 – graphics handles

- Create a new empty figure and get its handle.
- Create 2 vectors of the same size. Plot them one versus the other on that figure using the figure handle, and get the handle for the plot line.
- Change the plot line colour using its handle.
- Get the handle to the axes and change the limits of the x axis.
- Using the axes handle, add x-label and title to the graph.
- Using the axes handle, change the tick labels of the y axis.

Function List

- figure, clf, close, subplot, hold
- plot, pie, hist, histc, bar, stairs, scatter
- Image, imagesc, plot3, mesh, surf.
- title, xlabel, ylabel, axis, legend, grid, text, gtext, line, colorbar
- For graphics handles: get, set