

Graphics and Exportation in Matlab

Tyler Ransom
Duke University

July 16, 2012

Why use graphics?

- Helps readers internalize large amounts of data
- Helps you better understand your data and where variation (and hence, identification) is coming from
- Allows easier identification of outliers that may impact estimation results
- “A picture is worth 1,000 words”

Graphics in Matlab

- Matlab can create plots of data or graphs of mathematical equations
- Easiest way to create figures is on the workspace dropdown menu right above variable names (help comes up when you hover over a command)
- Can also access a similar menu from the main Matlab menu bar ("Graphics" next to "View")

Commands for plotting mathematical functions (2D)

```
1 %plot:      a simple plot of elements of a vector (basically, it ...  
               connects the dots)  
2 %ezplot:    Plots a function  $f(x)$  over range  $[a,b]$   
3 %fplot:     Plots a function  $f(x)$  over range  $[a,b]$   
4 %contour:   graphs contour lines of a matrix  
5 %ezcontour: graphs contour lines of a function  $f(x,y)$  over range  $[ax,bx]$  ...  
               x  $[ay,by]$ 
```

Commands for visualizing data (2D)

```
1 %scatter: classic scatter plot of X on Y
2 %hist:    histogram
3 %bar:     bar graph
4 %barh:    horizontal bar graph
5 %stem:    stem graph
```

Commands for plotting mathematical functions (3D)

```
1 %plot3:    same as plot, but with 3rd dimension
2 %ezplot3:  same as ezplot, but with 3rd dimension
3 %mesh:     wireframe 3D mesh plot of a matrix
4 %ezmesh:   wireframe 3D mesh plot of  $f(x,y)$  over range  $[ax,bx] \times [ay,by]$ 
5 %surf:     3D shaded surface plot of a matrix
6 %ezsurf:   3D shaded surface plot of  $f(x,y)$  over range  $[ax,bx] \times [ay,by]$ 
```

Commands for visualizing data (3D)

```
1 %scatter3
2 %hist3
3 %bar3
4 %bar3h
```

- These do exactly what their 2D counterparts do, except they require one additional input vector

Adding bells and whistles

- The drop-down menu in the workspace only allows for the most basic plot output
- Once a graph is generated, users can interactively add more features to it
- Useful features include: legend, axis labels, axis tick marks, title
- Can also add data summary statistics to plots quite easily

Reproducibility

- Since you will almost always need to change a plot after creating it, reproducibility is the name of the game
- Even though you can edit graphs interactively, you can still produce source code from the finished product so you won't have to manually change the graph each time you produce it
- This is done interactively in the figure window under (File-> Generate M-file)
- Matlab then generates a function m-file which you can call in your script, or paste directly into your script with slight modifications

Overlaying graphs

- Matlab allows users to overlay multiple different plots
- The hold command does this
- Syntax is

```
1 [plot1]
2 hold on
3 [plot2]
4 hold off
5 [plot3]
```

- This plots Plot1 and Plot2 on the same graph and Plot3 on a separate graph
- See help hold for more details—this command is very versatile

Multiple graphs

- If a user wants to display a “matrix” of separate graphs, she should use the subplot command
- `subplot(m,n,p)` or `subplot(mnp)` breaks the figure window into an m-by-n grid and creates an axes object in the pth location for the current plot

Saving and exporting figures

- Figures will likely go into a set of presentation slides or a paper
- Matlab can easily save figures in a variety of formats
- Most publishers prefer .eps (Encapsulated PostScript) files because they can easily be re-sized without losing resolution
- Once a figure has been exported, it can easily be placed in a .tex document
- Since reproducibility is so important, exporting graphs is a simple and effective way to streamline the updating process

Syntax of exporting

- Matlab's `print` command can print and export figures
- I find this command to be highly unwieldy, so I instead downloaded a user-written command called `exportfig`
- Syntax for this is:
- `exportfig(figname, 'myfig.eps')`
- Note that `figname` needs to have been defined prior to a plot command with `figname = figure;`

Conclusion

- Reproducibility is the name of the game
- You should always be searching for ways to automate your tasks
- There's pretty much always a user-written utility out there somewhere that can accomplish what you're looking for