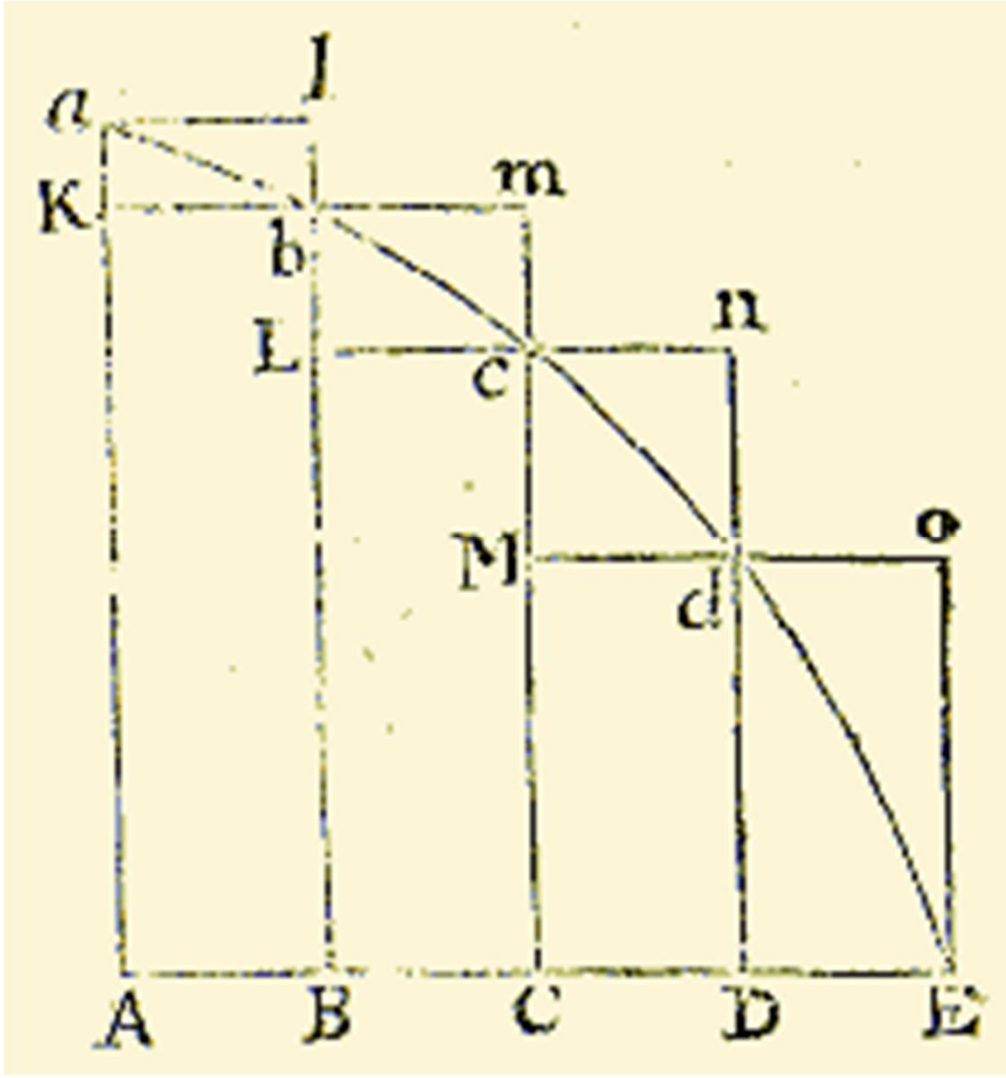
MATLAB Programming

Visualization

Good science requires good visualizations.



Why use MATLAB for plots?

- Produces publication quality plots and images
- Coupled with computation program.

Proper visualization might take exploration.

Simplest plot is a line plot:

```
>> plot(M)
```

Plot each column of M as a separate line.

Sample data sets:

dow.txt : Daily closing value of Dow Jones Industrial Average, an index of the price of 30 stocks on the New York Sock Exchange

sp.txt : Daily closing value of the Standard and Poors 500, a broader index containing 500 stocks.

Data looks like:

Year	Month	Day	Value
1950	01	03	198.89
1950	01	04	200.20
1950	01	05	200.57
1950	01	06	200.96
1950	01	07	201.90

é

Load data:

```
>> dow = importdata('dow.txt');
>> sp = importdata('sp.txt');
```

Simple plot:

```
>> plot(dow(:,4))
```

```
>> plot(dow(:,4))
```

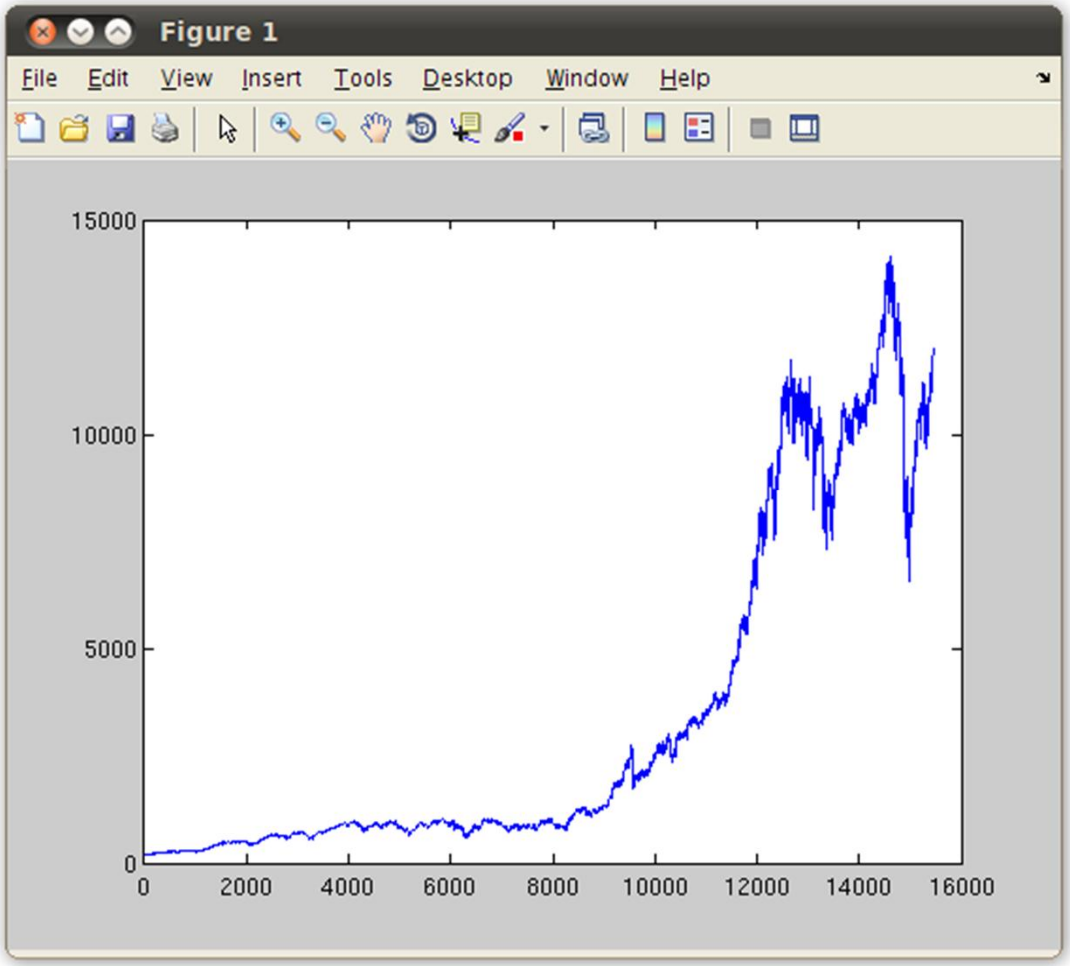


Figure 1
File Edit View Insert Tools Desktop Window Help
15000
10000
5000
0
0 2000 4000 6000 8000 10000 12000 14000 16000

```
>> plot(dow(:,4))
```

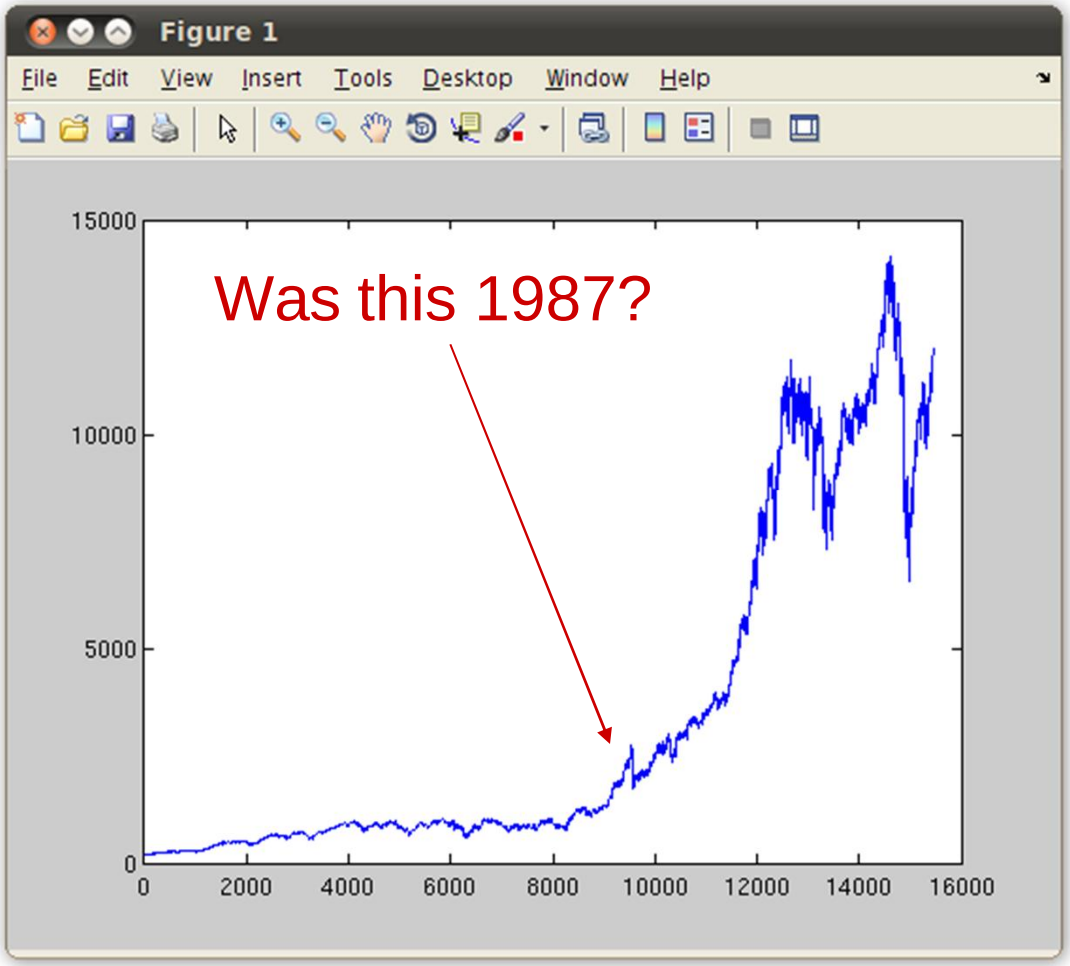


Figure 1
File Edit View Insert Tools Desktop Window Help
Was this 1987?
15000
10000
5000
0
0
2000
4000
6000
8000
10000
12000
14000
16000

```
>> plot(dow(:,4))
```

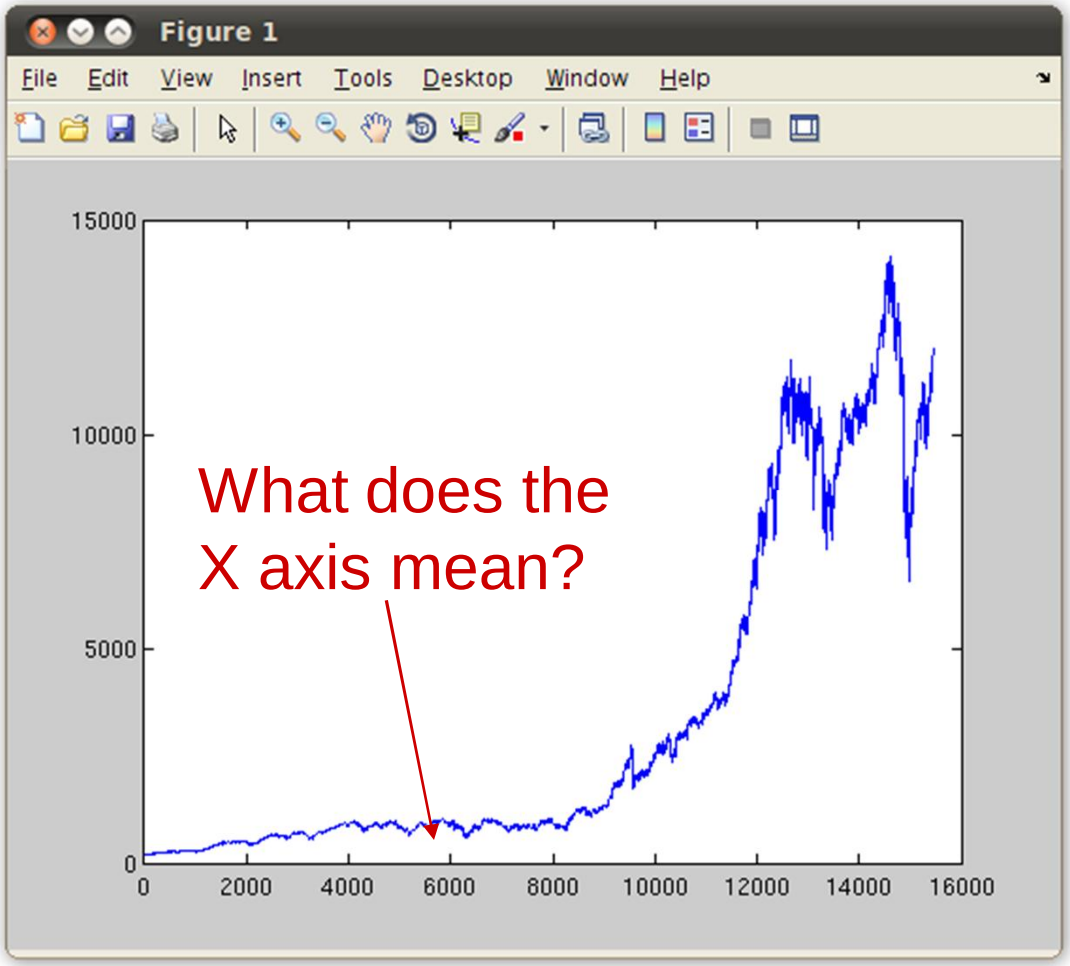
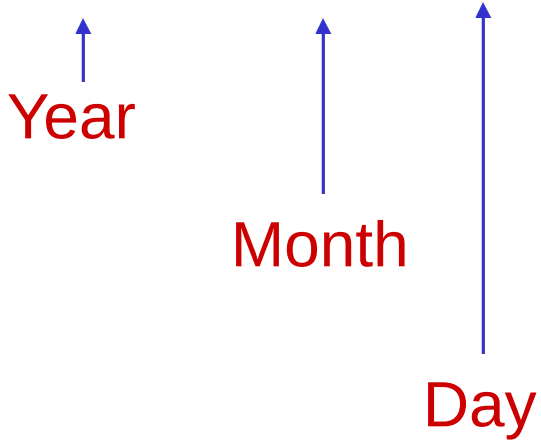


Figure 1
File Edit View Insert Tools Desktop Window Help
15000
10000
5000
0
0 2000 4000 6000 8000 10000 12000 14000 16000
What does the
X axis mean?

```
>> dow(1,1:3)

ans =

    1950     1     3
```

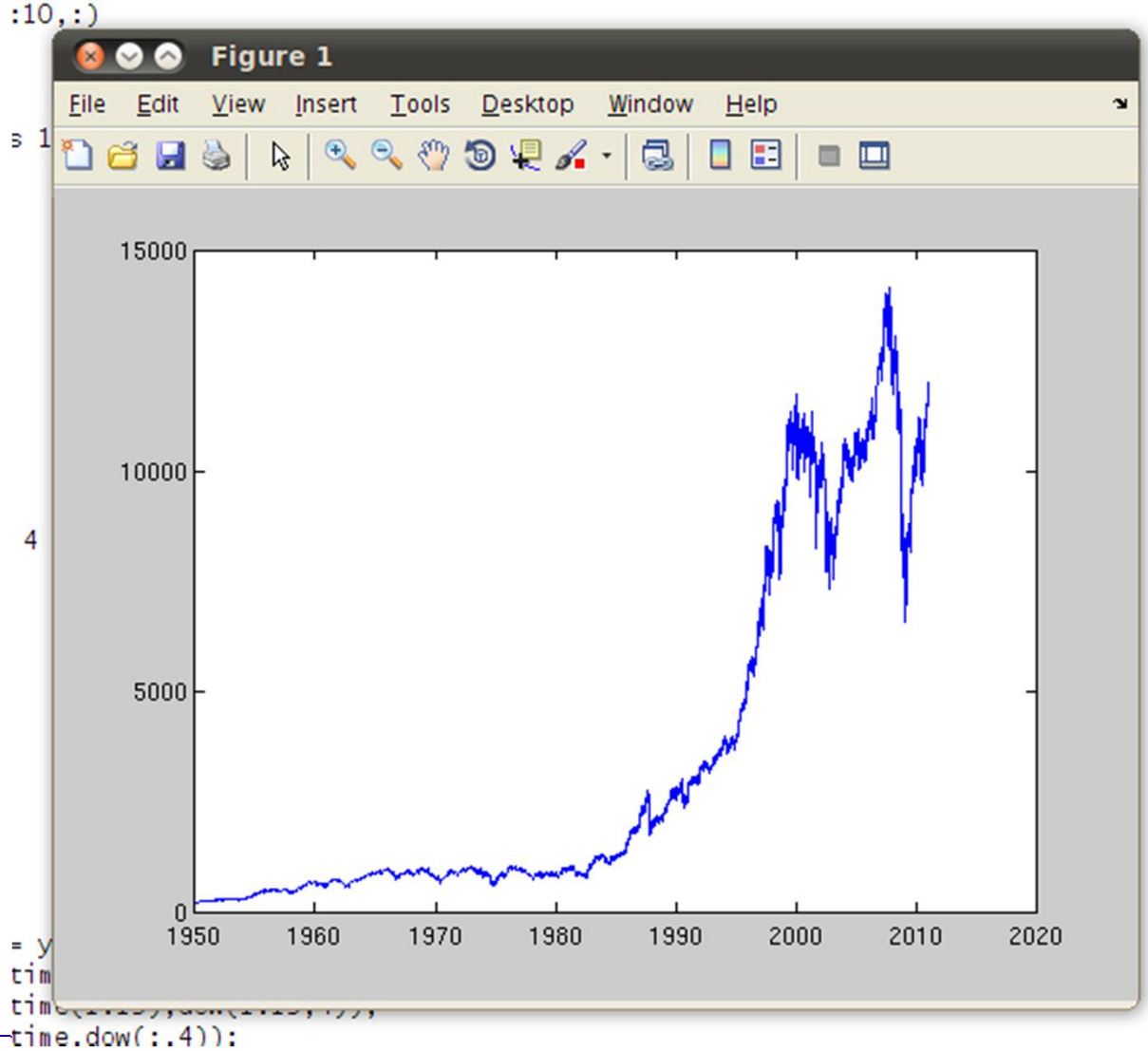


```
>> time = dow(1,:) + (dow(2,:)-1) / 12 +
   (dow(3,:)-1) / 30 / 12
```

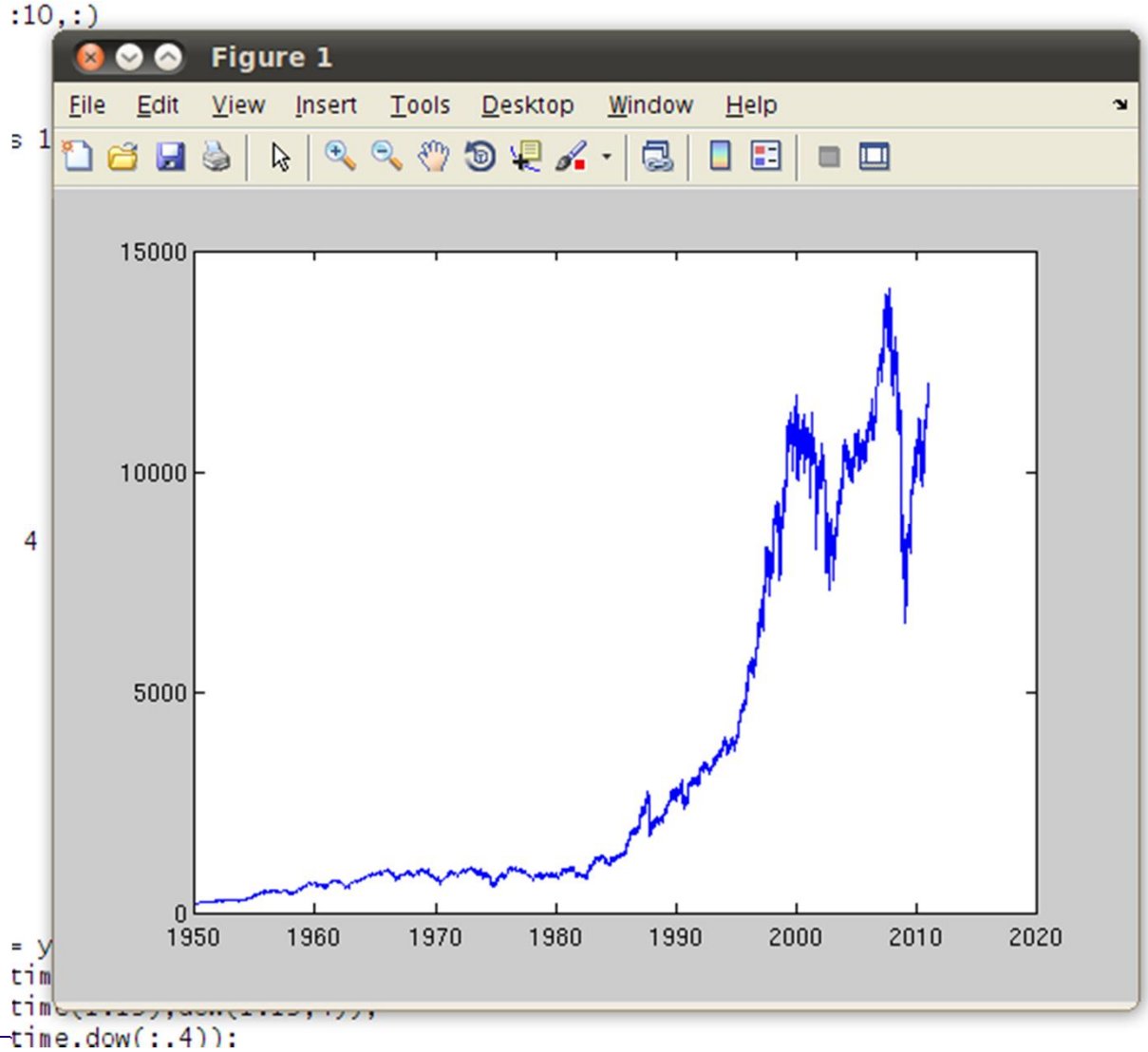
Plot the Dow's value versus time:

```
>> plot(time, dow(:,4))
```

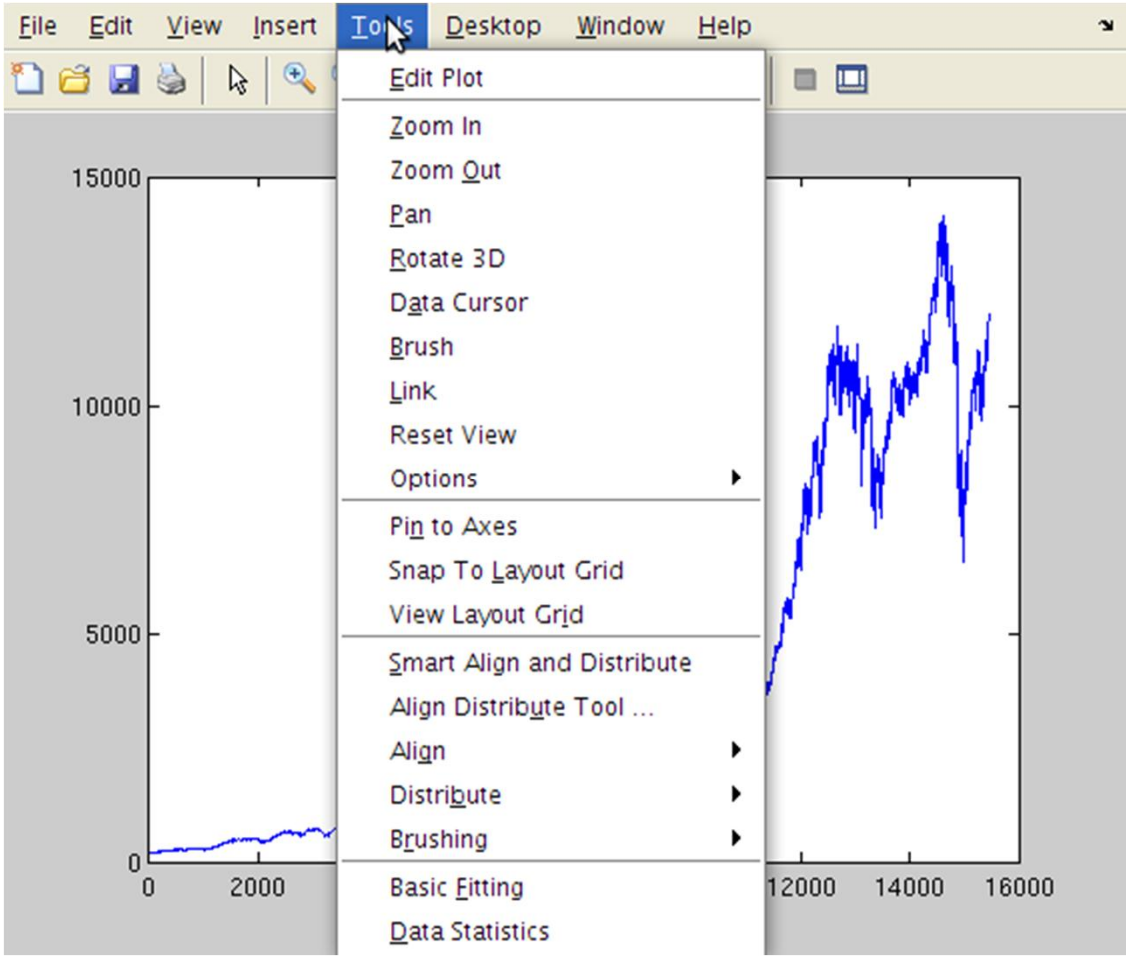
>> plot(time, dow(:,4))



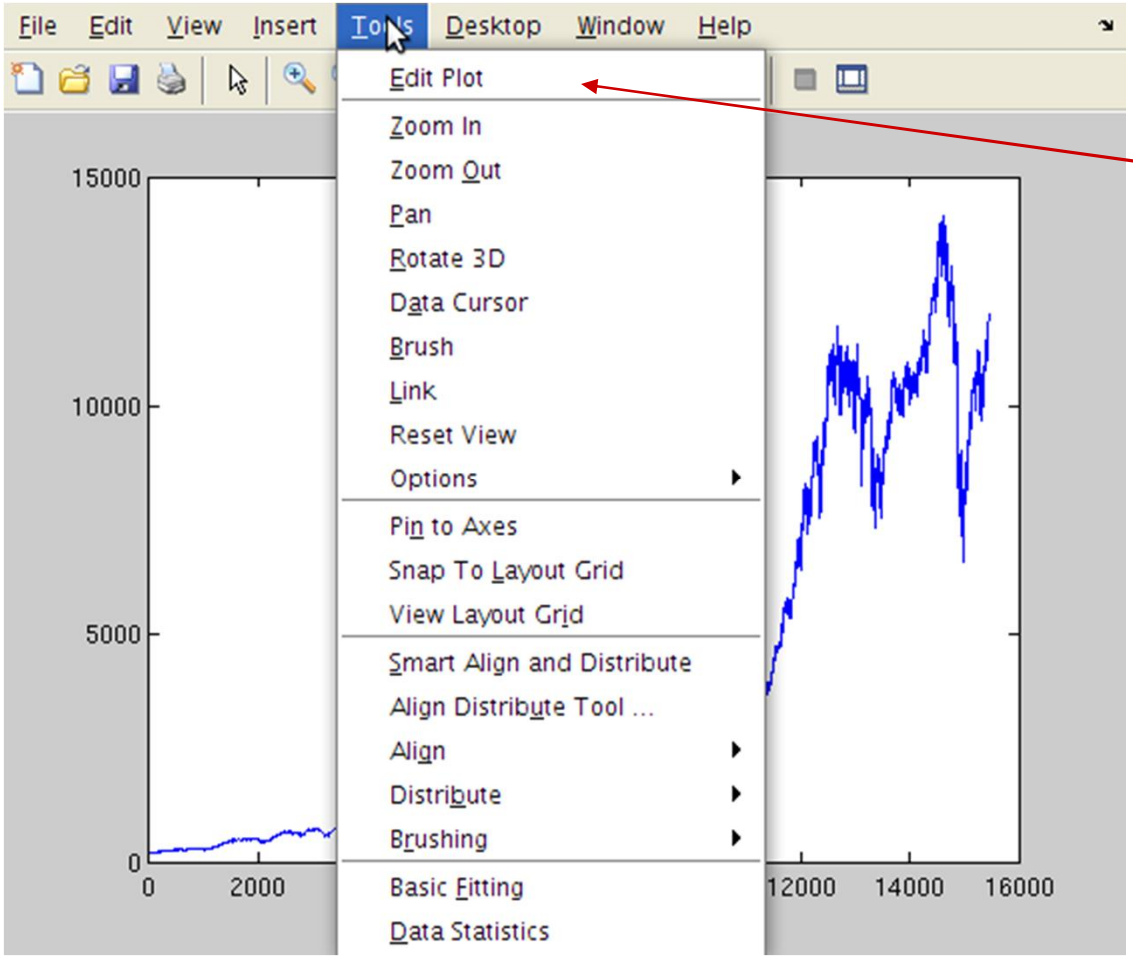
What if we want to edit the plot?



What if we want to edit the plot?

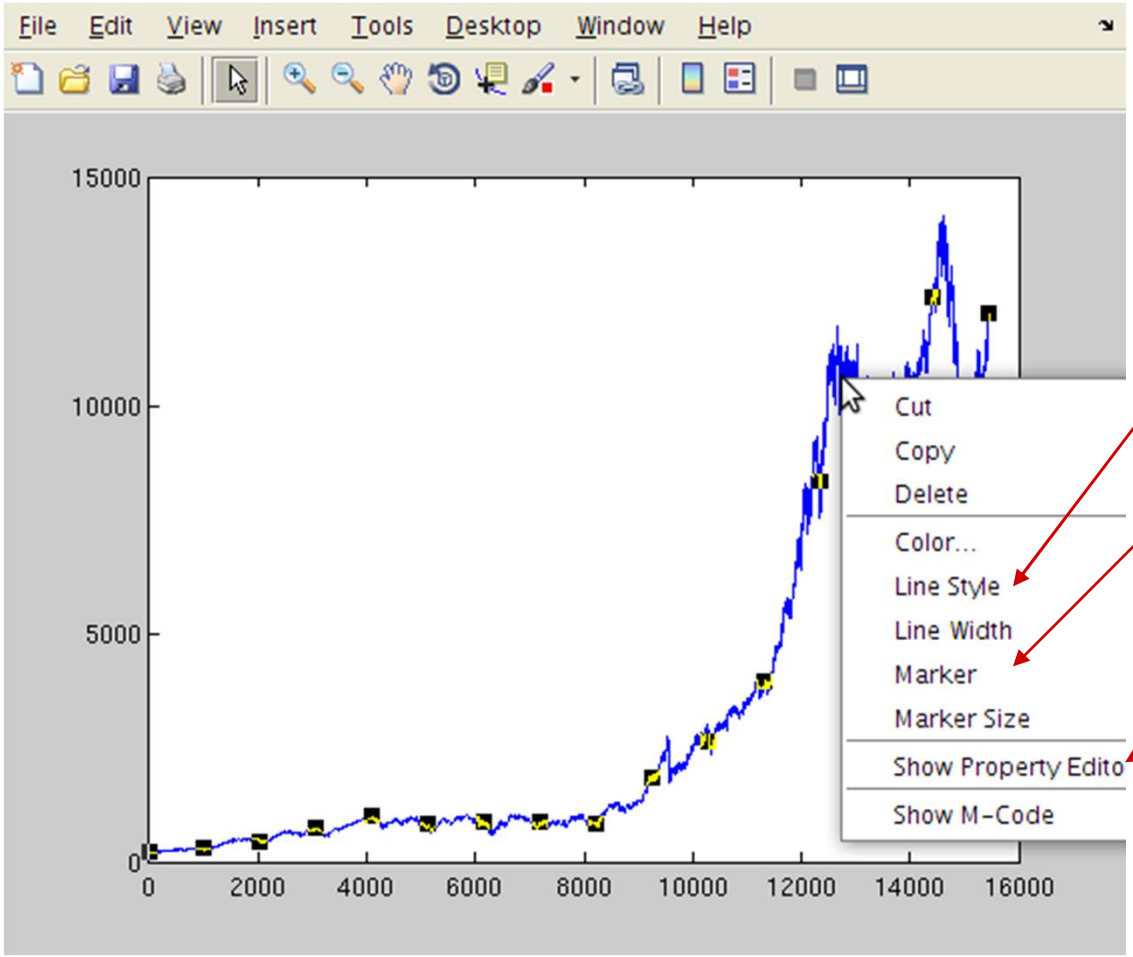


What if we want to edit the plot?



Provides access to plot details

What if we want to edit the plot?



Change line

Add markers

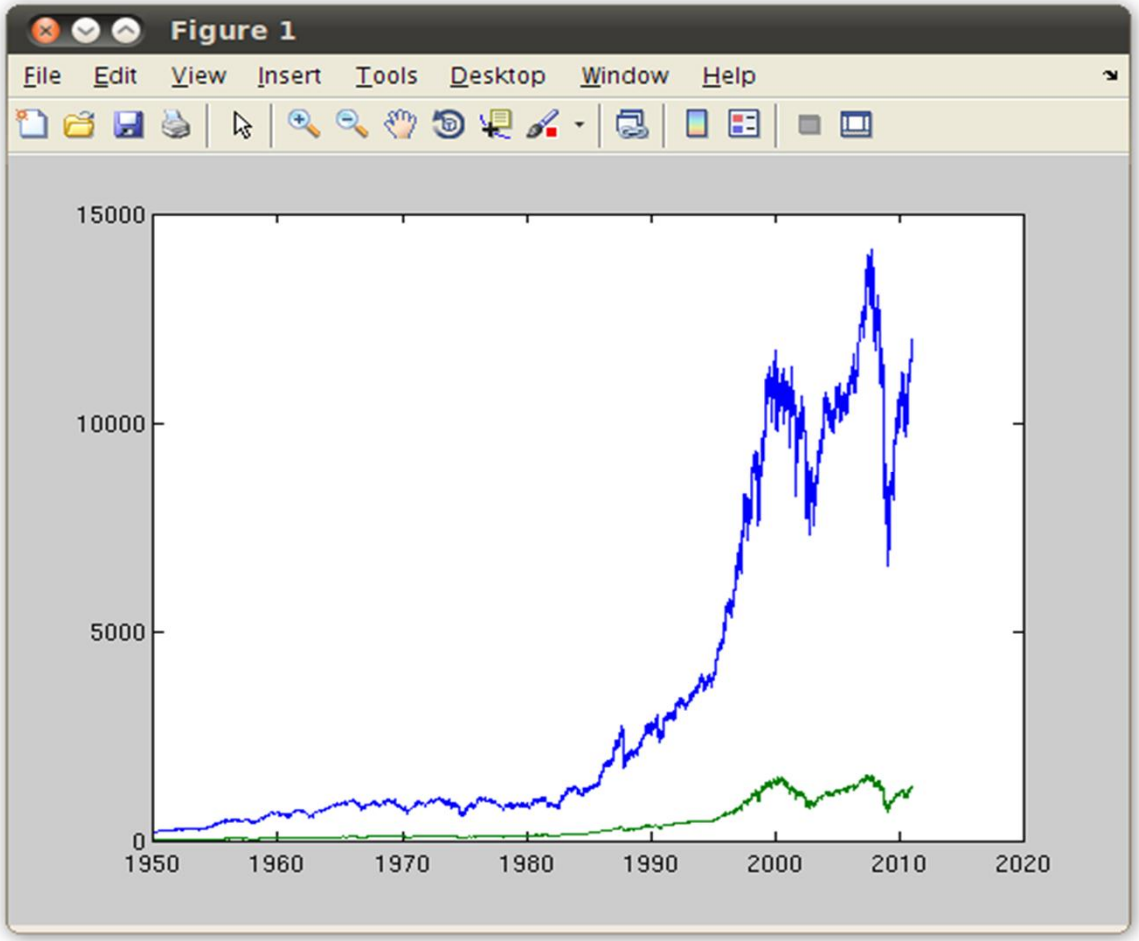
Many other options.

Compare the Dow to the S&P:

```
>> stocks = [dow(:,4) sp(:,4)];
>> plot(time, stocks);
```

Plotting a matrix (stocks) against a vector(time) plots each column of the matrix with the shared X-axis.

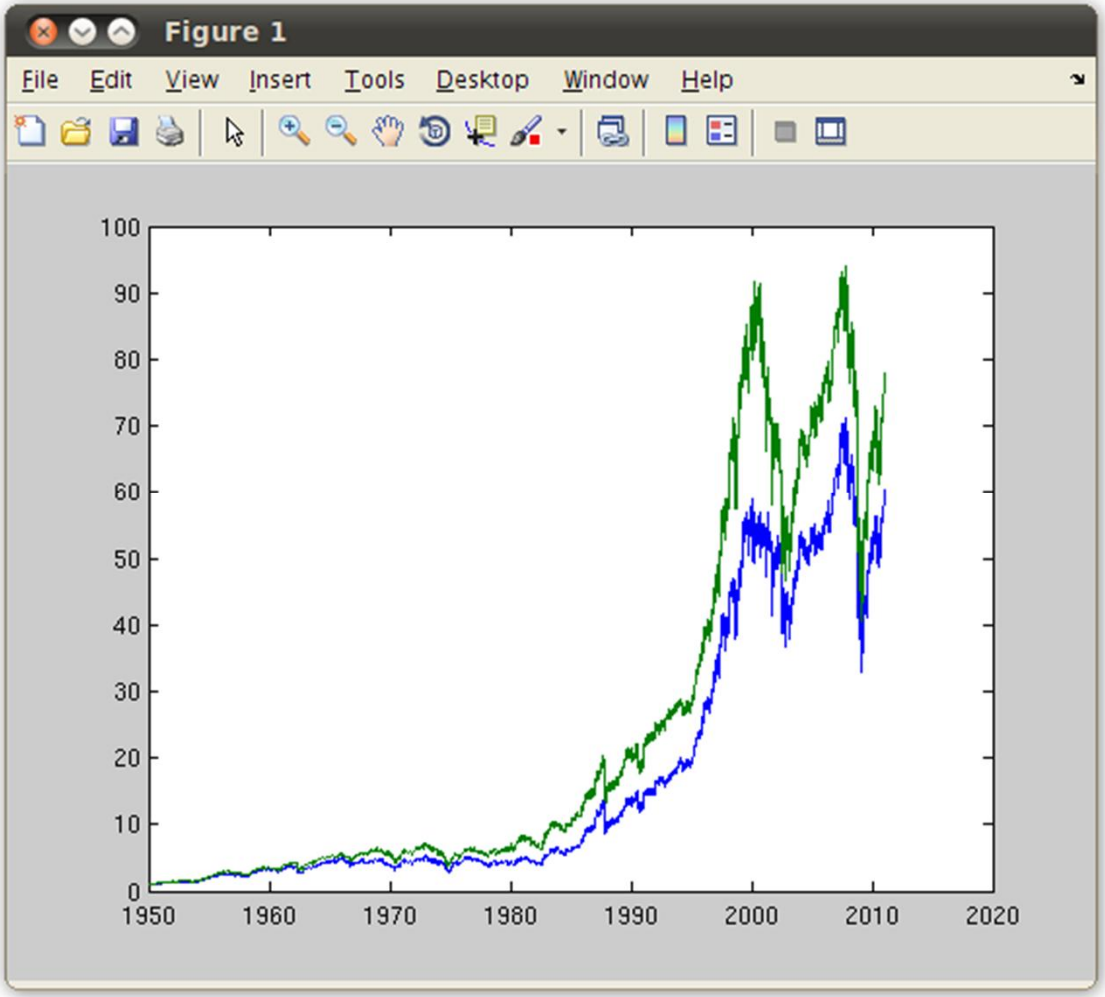
```
>> plot(time, stocks);
```



Rescale the indices to start at the same place:

```
>> d2 = dow(:,4) / dow(1:4);
>> s2 = sp(:,4) / sp(1,4);
>> plot(time, [d2 s2]);
```

```
>> plot(time, [d2 s2]);
```



MATLAB has utilities to plot many kinds of data:

- hist: histograms
  - pie: pie charts
    - bar, barh: bar charts

Even many kinds of 3D charts:

- pie3
  - bar3
    - pareto

A lot of data is one dimensionalé

é what about 2-D data?

Example: geographically oriented data.

Ever wondered where people tweet the most?

Question: what places in Toronto are the most popular locations for people to send a geo-located tweet?

Data collection:

Record all geolocated tweets for 2 months.

Divide the city into a grid and count the number of tweets in each cell of the grid.

Question: what places in Toronto are the most popular locations for people to send a geo-located tweet?

Data collection:

Record all geolocated tweets for 2 months.

Divide the city into a grid and count the number of tweets in each cell of the grid.

Data: a matrix of grid centers and the relative number of tweets in that spot.

Question: what places in Toronto are the most popular locations for people to send a geo-located tweet?

Data collection:

Record all geolocated tweets for 2 months.

Divide the city into a grid and count the number of tweets in each cell of the grid.

Data: a matrix of grid centers and the relative number of tweets in that spot.

>> image(data);

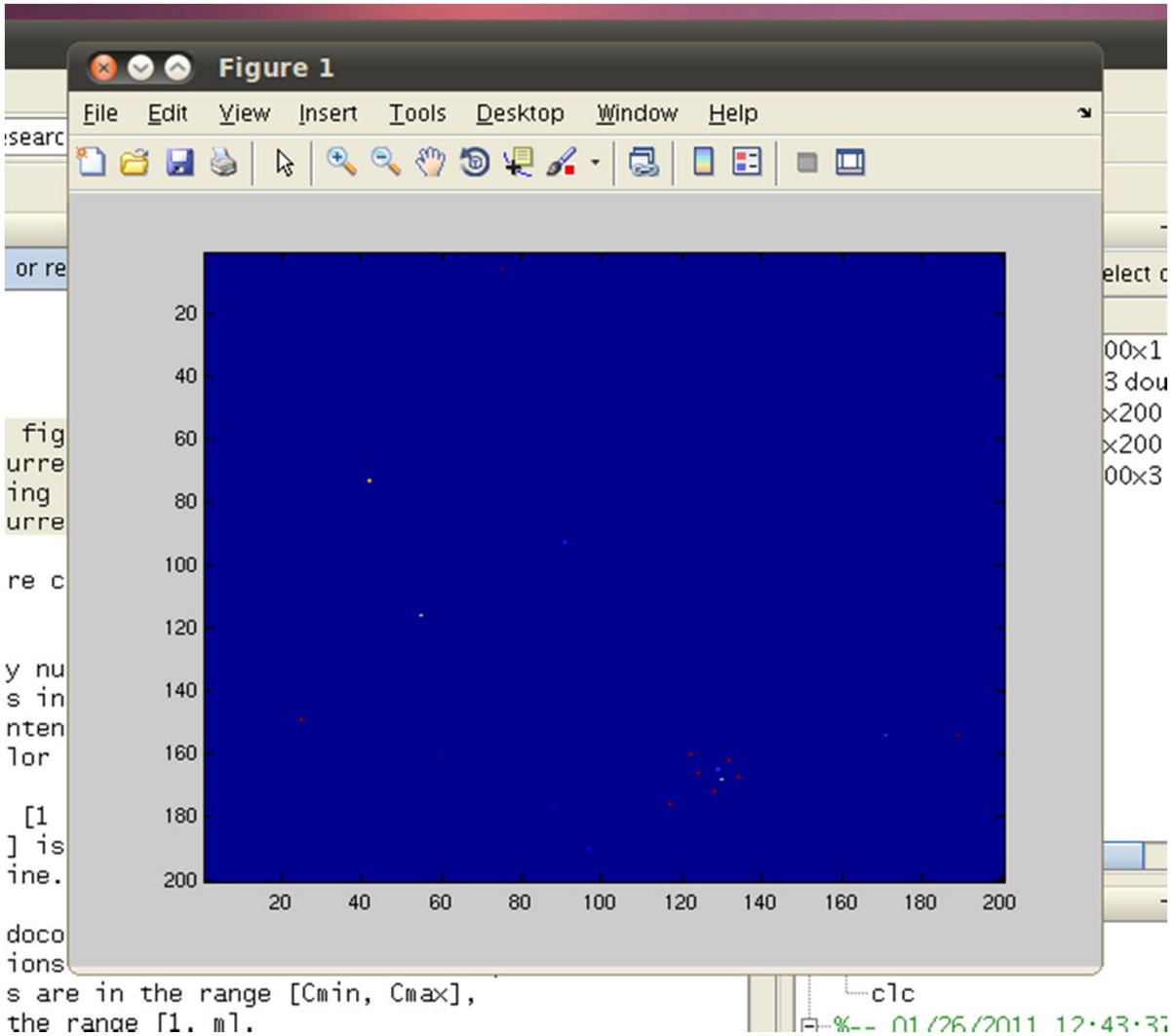


Figure 1
File Edit View Insert Tools Desktop Window Help
20
40
60
80
100
120
140
160
180
200
20 40 60 80 100 120 140 160 180 200
s are in the range [Cmin, Cmax],
clc

>> image(data);

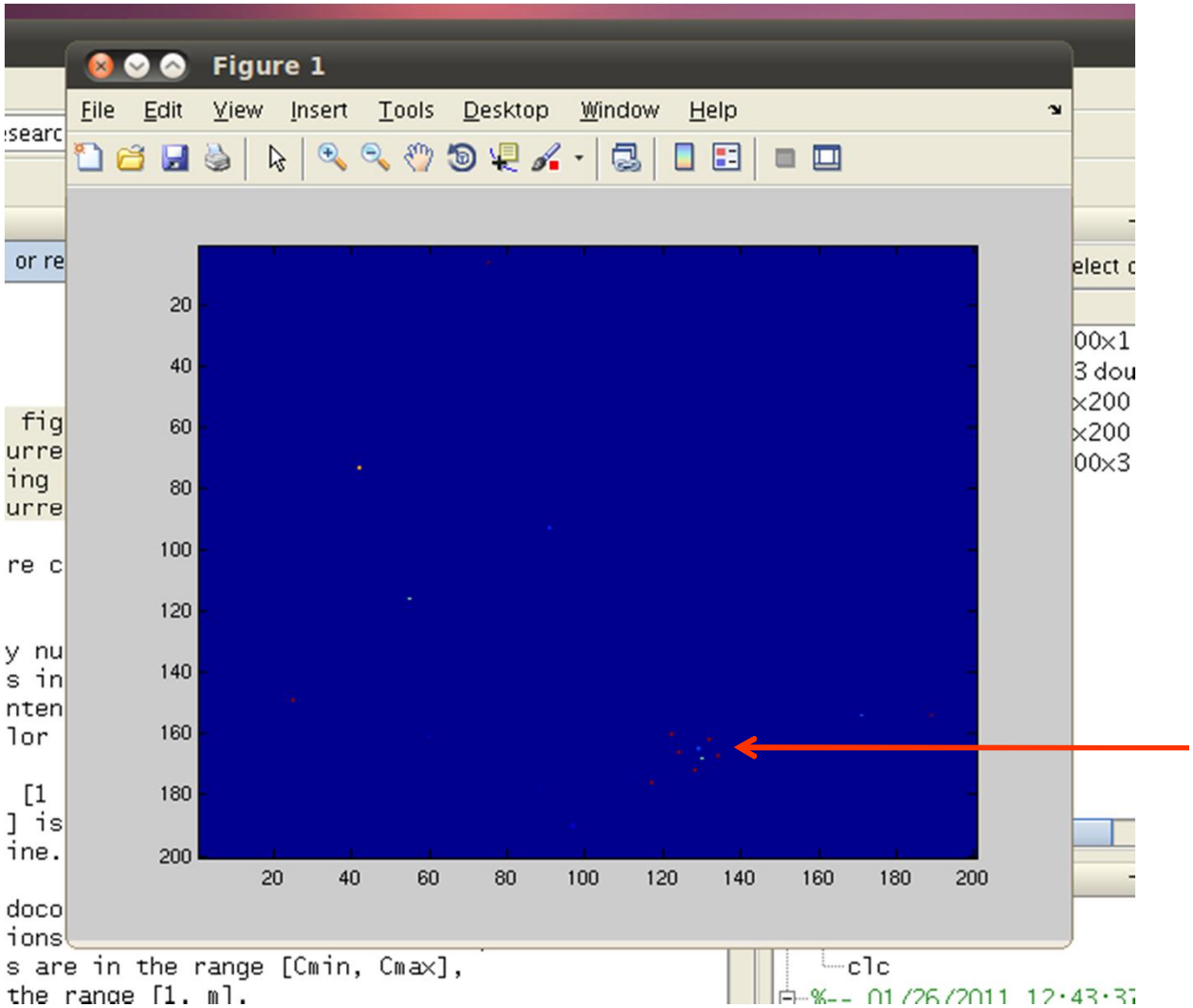


Figure 1
File Edit View Insert Tools Desktop Window Help
20
40
60
80
100
120
140
160
180
200
20 40 60 80 100 120 140 160 180 200

image():

Take either an N X M or N X M X 3 array.

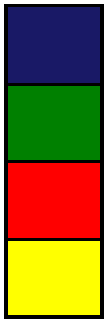
Third dimension is for three channels of a color image.

Map each location a color using a colormap.

Data

45	**1**	**1**	**1**
46	1	1	1
46	1	1	1
46	45	45	45

Colormap



Only use first column
because Data is 4x4

Image

A colormap is a color guide that maps the values 0.0 to 64.0 to colors.

Many colormapsé

Just checké

>> help colormaps

é for all the options.

A colormap is a color guide that maps the values 0.0 to 64.0 to colors.

Many colormapsé

Just checké

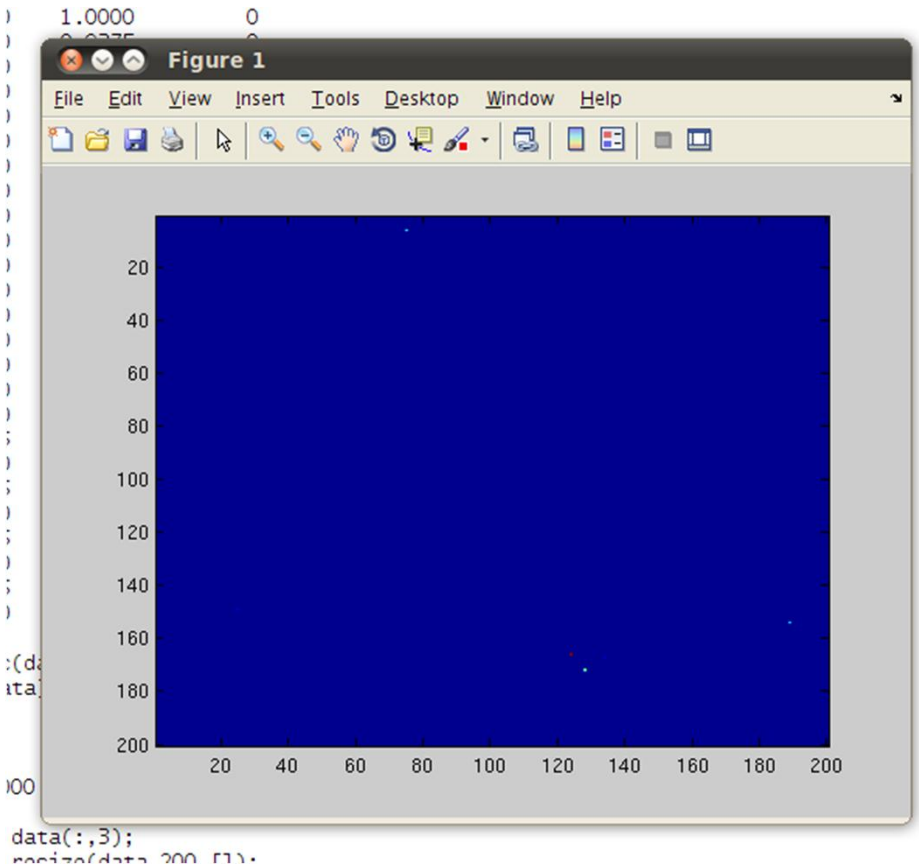
>> help colormaps

é for all the options.

What if our matrix has a different range?

```
>> imagesc(data);
```

Scales the matrix to use the entire colormap.



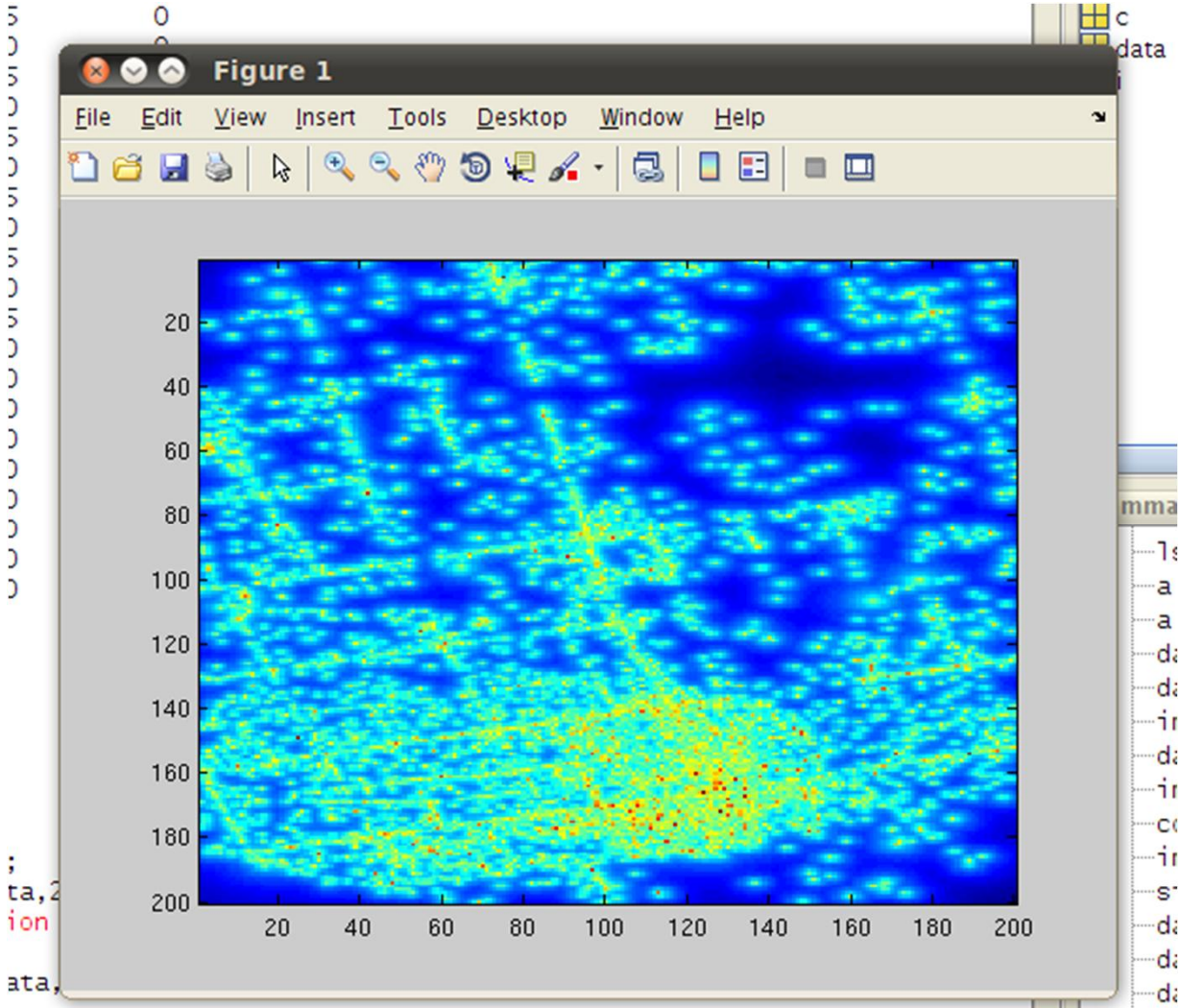
Key: imagesc scales the data linearly.

Our data:

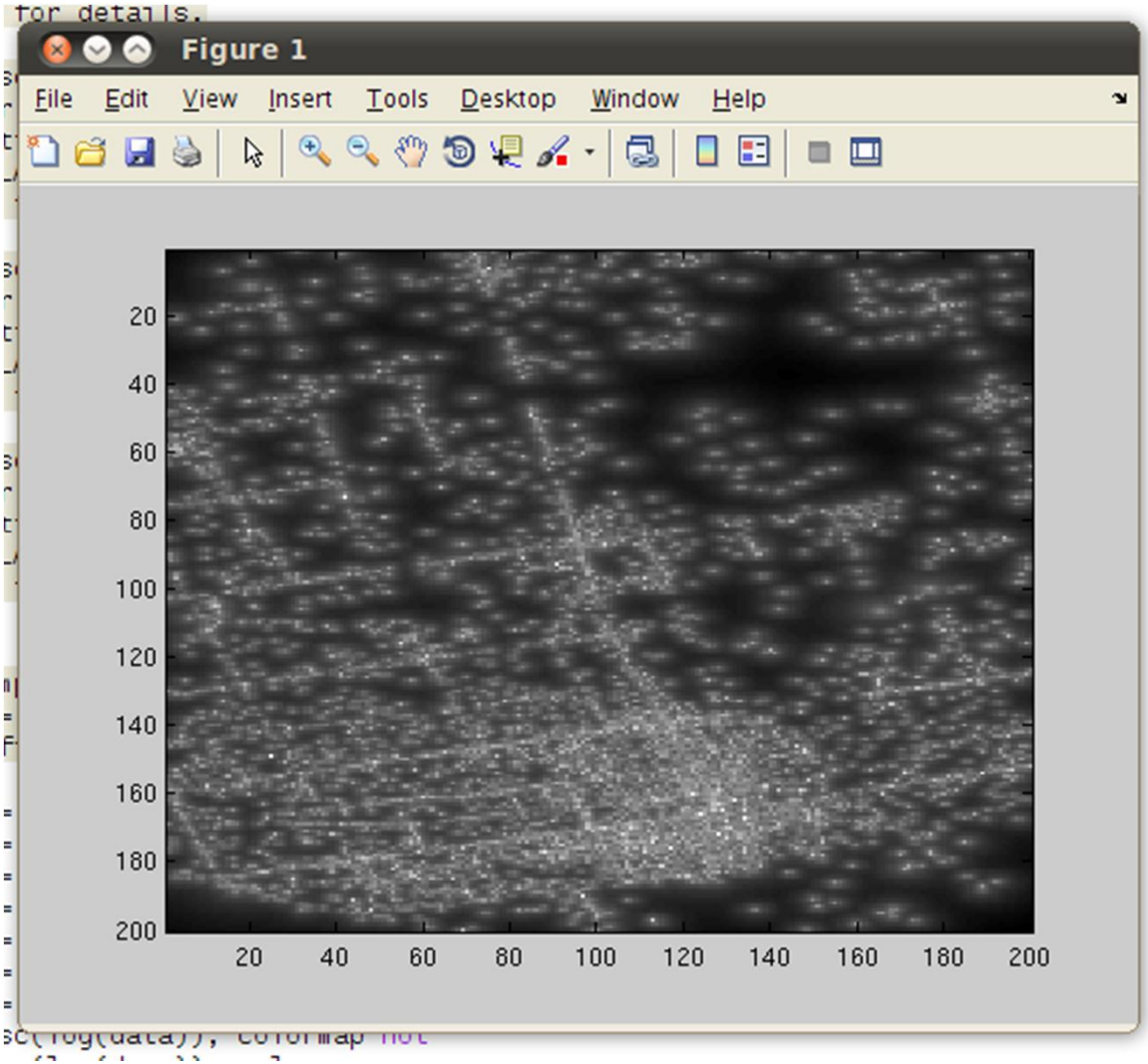
```
>> max(max(data))
1.93 e+04
>> min(min(data))
2.05 e-24
>> mean(mean(data))
1.1190
```

Our data is scaled exponentiallyé

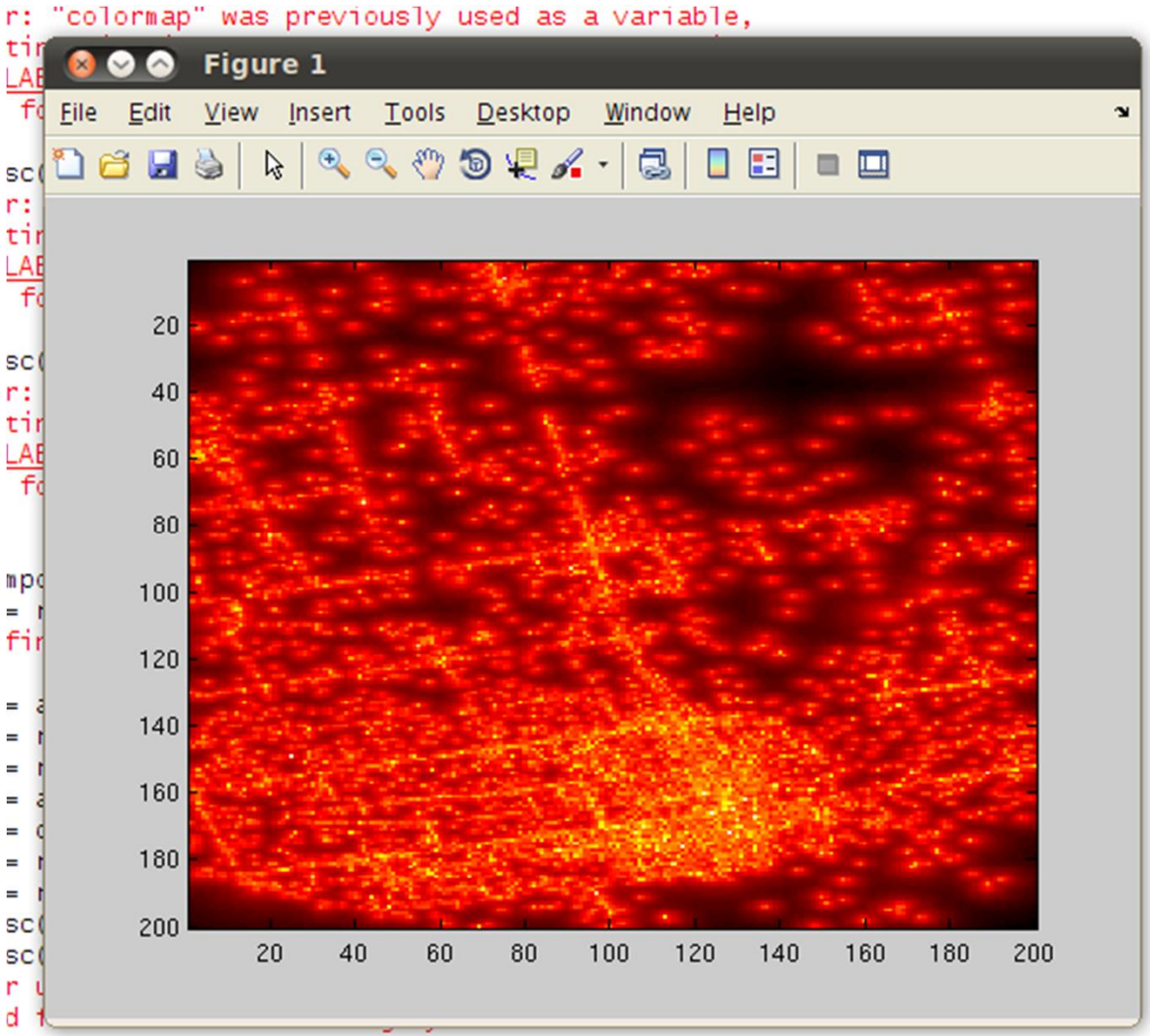
```
>> imagesc(log(data));
```



```
>> imagesc(log(data)), colormap gray;
```



>> imagesc(log(data)), colormap hot;



Conclusion:

Imaging is a powerful way to explore dataé

é but be sure to take full advantage of the pattern in that data.

created by

Richard T. Guy

February 2011