ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Modeling and Simulating Social Systems with MATLAB

Lecture 2 – Statistics and plotting in MATLAB continued

Tobias Kuhn and Olivia Woolley

Chair of Sociology, in particular of

Modeling and Simulation

Lesson 1: Exercise 1

- **Compute:**

a) $$\frac{18+107}{5\times 25}$$

b) $$\sum_{i=0}^{100} i$$

c) $$\sum_{i=5}^{10} \left(i^2 - i\right)$$

Lesson 1: Exercise 1 – solution

- **Compute:**

a) $$\frac{18+107}{5\times 25}$$

```
>> (18+107)/(5*25)

ans =

    1
```

Lesson 1: Exercise 1 – solution

- **Compute:**

b) $$\sum_{i=0}^{100} i$$

```
>> s=sum(1:100)
or
>> s=sum(1:1:100)
or
>> s=sum(linspace(1,100))
or
>> s=sum(linspace(1,100,100))

s =

    5050
```

default value

default value

Lesson 1: Exercise 1 – solution

- **Compute:**

c) $$\sum_{i=5}^{10}\left(i^2-i\right)$$

```
>> s=0;
>> for i=5:10
>> s=s+i^2-i;
>> end
>> s

s =

    310
```

Lesson 1: Exercise 2

- Solve for **x:**

$$
\begin{aligned}
2x_1 - 3x_2 - x_3 + 4x_4 &= 1 \\
2x_1 + 3x_2 - 3x_3 + 2x_4 &= 2 \\
2x_1 - x_2 - x_3 - x_4 &= 3 \\
2x_1 - x_2 + 2x_3 + 5x_4 &= 4
\end{aligned}
$$

Lesson 1: Exercise 2 – solution

```
>> A=[2 -3 -1 4; 2 3 -3 2; 2 -1 -1 -1; 2 -1 2 5];
>> b=[1; 2; 3; 4];
>> x=A\b

x =
    1.9755
    0.3627
    0.8431
   -0.2549

>> A*x
ans =
    1.0000
    2.0000
    3.0000
    4.0000
```

Ax=b

$$2x_1 - 3x_2 - x_3 + 4x_4 = 1$$
$$2x_1 + 3x_2 - 3x_3 + 2x_4 = 2$$
$$2x_1 - x_2 - x_3 - x_4 = 3$$
$$2x_1 - x_2 + 2x_3 + 5x_4 = 4$$

Lesson 1: Exercise 3

- **Fibonacci sequence:** Write a function which computes the Fibonacci sequence until a given number ***n*** and return the result in a vector.
- The Fibonacci sequence ***F(n)*** is given by :

$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

Lesson 1: Exercise 3 – iterative solution

fibonacci.m:

```
function [v] = Fibonacci(n)

  v(1) = 0;

  if ( n>=1 )
    v(2) = 1;
  end

  for i=3:n+1
    v(i) = v(i-1) + v(i-2);
  end
end
```

$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

```
>> Fibonacci(7)
ans =
     0     1     1     2     3     5     8    13
```

Lesson 1: Exercise 3 – recursive solution

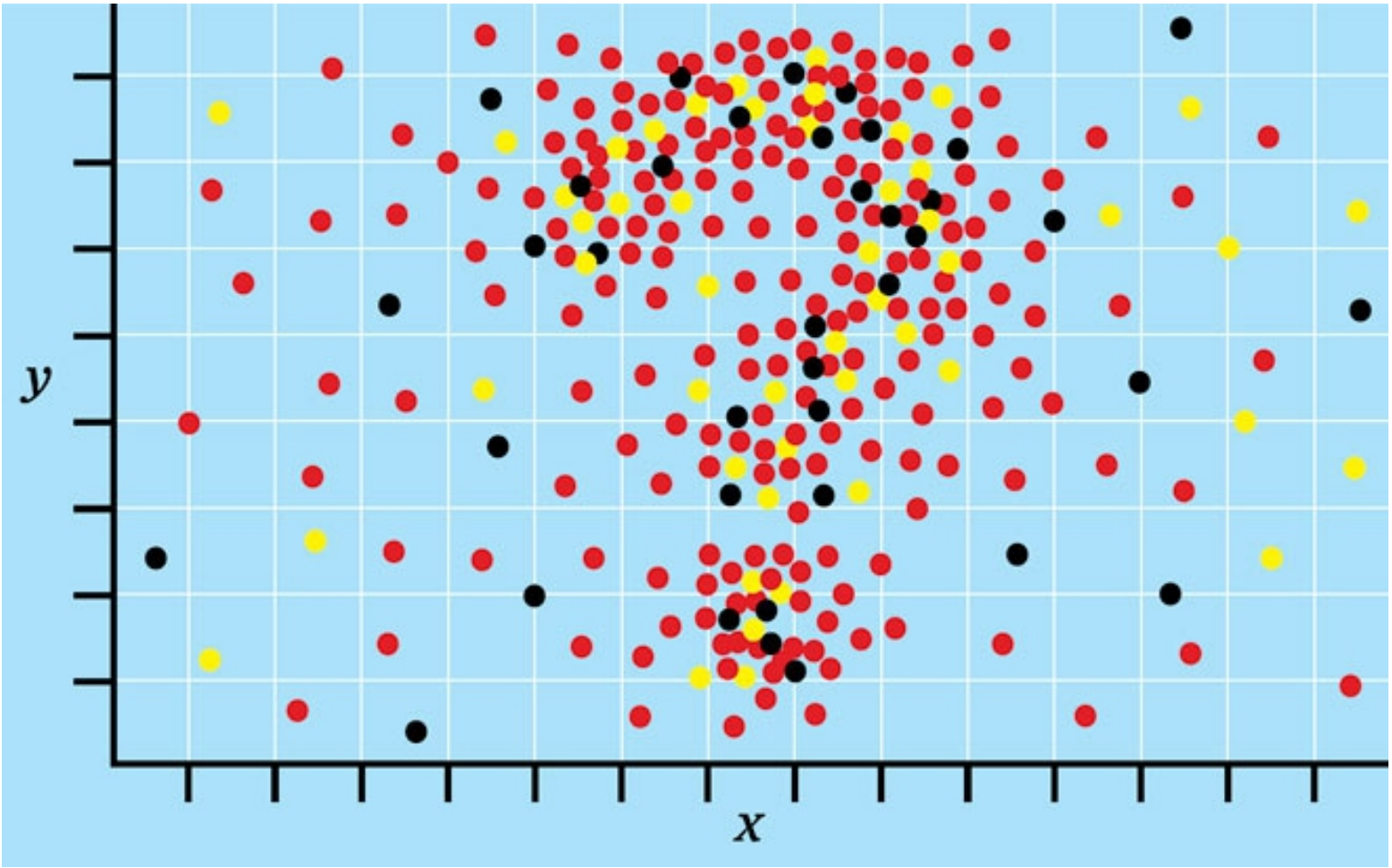
fibo_rec.m:

```
function f = fibo_rec(n)
  if n == 0
    f(1) = 0;

  elseif n == 1
    f(2) = 1;

  elseif n > 1
    f        = fibo_rec(n - 1);
    f(n + 1) = f(n) + f(n - 1);
  end
end
```

$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

```
>> fibo_rec(7)
ans =
     0     1     1     2     3     5     8    13
```

Plotting and Statistics in Matlab Contd.



Back to plotting… details of plot

- An additional parameter can be provided to `plot()` to define how the curve will look like:

  `plot(x, y, 'key')`

  Where key is a string which can contain:

  Color codes: 'r', 'g', 'b', 'k', 'y', …

  Line codes: '-', '--', '.-' (solid, dashed, etc.)

  Marker codes: '*', '.', 's', 'x'

  Examples: `plot(x, y, 'r--')` — — — —

  `plot(x, y, 'g*')` * * * *

Plotting tips

- To make the plots look nicer, the following commands can be used:

  Set label on x axis:

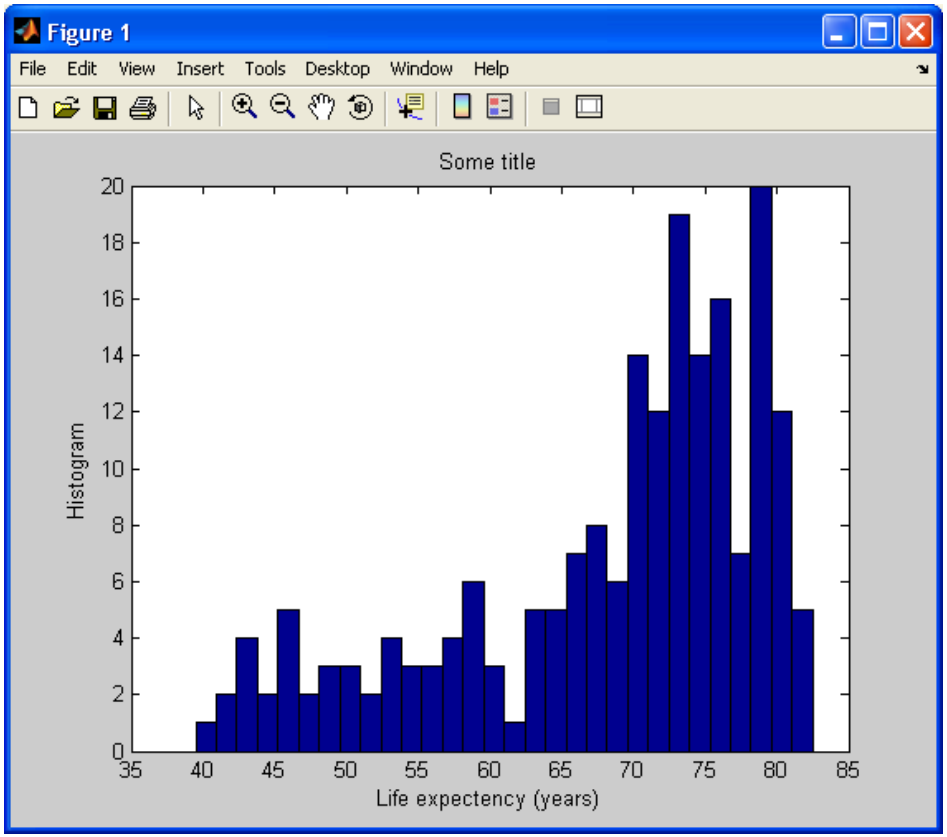
  `xlabel('text')`

  Set label on y axis:

  `ylabel('text')`

  Set title:

  `title('text')`



Plotting tips

- Setting axes limits
  - `xlim([xmin xmax])`
  - `ylim([ymin ymax])`

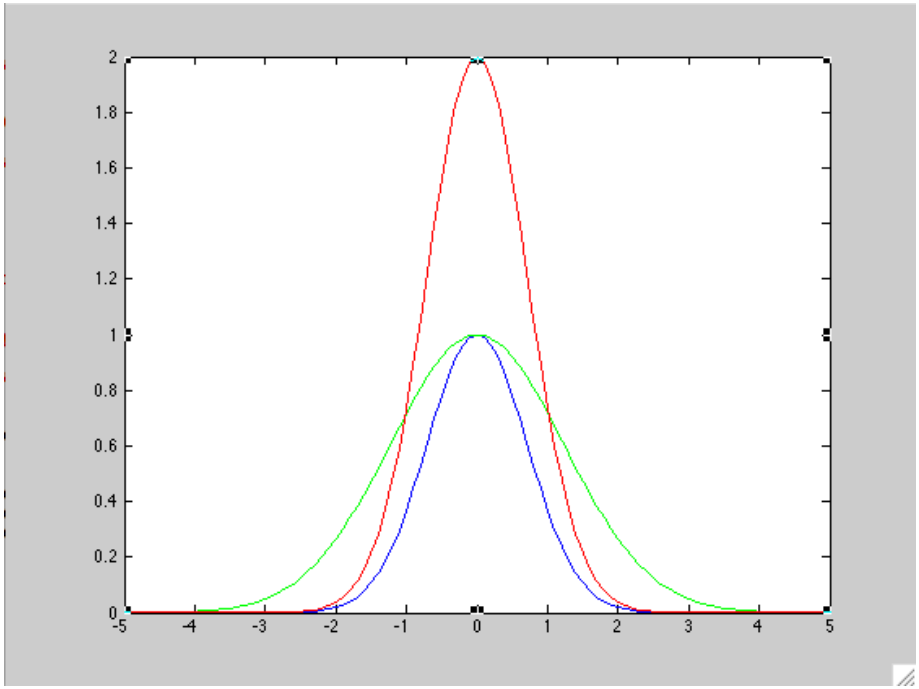
Plotting tips

- Two additional useful commands:
  - `hold on|off`
  - `grid on|off`

```
>> x=[-5:0.1:5];
>> y1=exp(-x.^2);
>> y2=2*exp(-x.^2);
>> y3=exp(-(x.^2)/3);
```

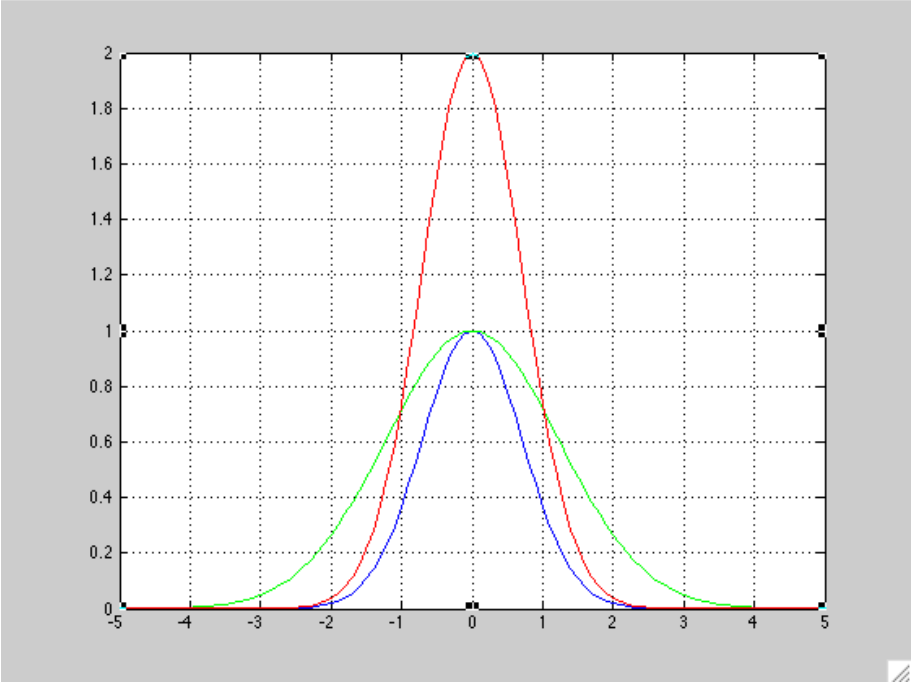
Plotting tips

```
>> plot(x,y1);

>> hold on

>> plot(x,y2,'r');
>> plot(x,y3,'g');
```



Plotting tips

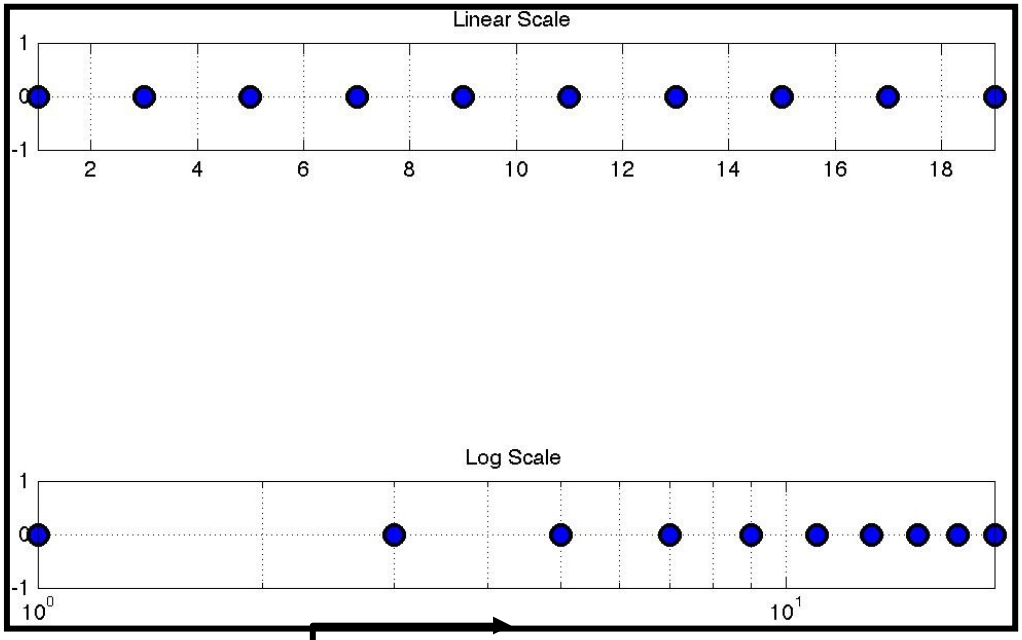
```
>> plot(x,y1);

>> hold on

>> plot(x,y2,'r');
>> plot(x,y3,'g');
>> grid on
```



Subplot()

- **`subplot(m,n,p)`** breaks the figure window into an m-by-n matrix and selects the pth axes object for the current plot.
- The axes are counted along the top row of the figure window, then the second row, etc.

Subplot() by the way…



We have already seen an example of a chart created using subplot last lecture. Can you reproduce it?

Handles, figure(), set() and get()

- `h = figure(...)` adds a new figure window to the screen. It also returns the handle to it, so it can be further processed.

- Figure handle: `gcf` Axes handle: `gca`

- `get(handle,'PropertyName')`
- `set(handle,'PropertyName',PropertyValue,...)`

e.g. `set(gcf, 'Position', [100 100 150 150])`

`[left bottom width height];`

Saving and printing figures

- `saveas(handle,'myimage.fig');`
- `saveas(handle,'myimage.jpg');`

- `export_fig` from MathWorks File Exchange
  http://www.mathworks.com/matlabcentral/fileexchange/23629-exportfig

```
export_fig('/plots/myimage.pdf', '-pdf',
'-nocrop')
```

Exercises: working with datasets

- Two datasets for statistical plotting can be found on the course web page (with NETHZ login)

  http://www.soms.ethz.ch/matlab

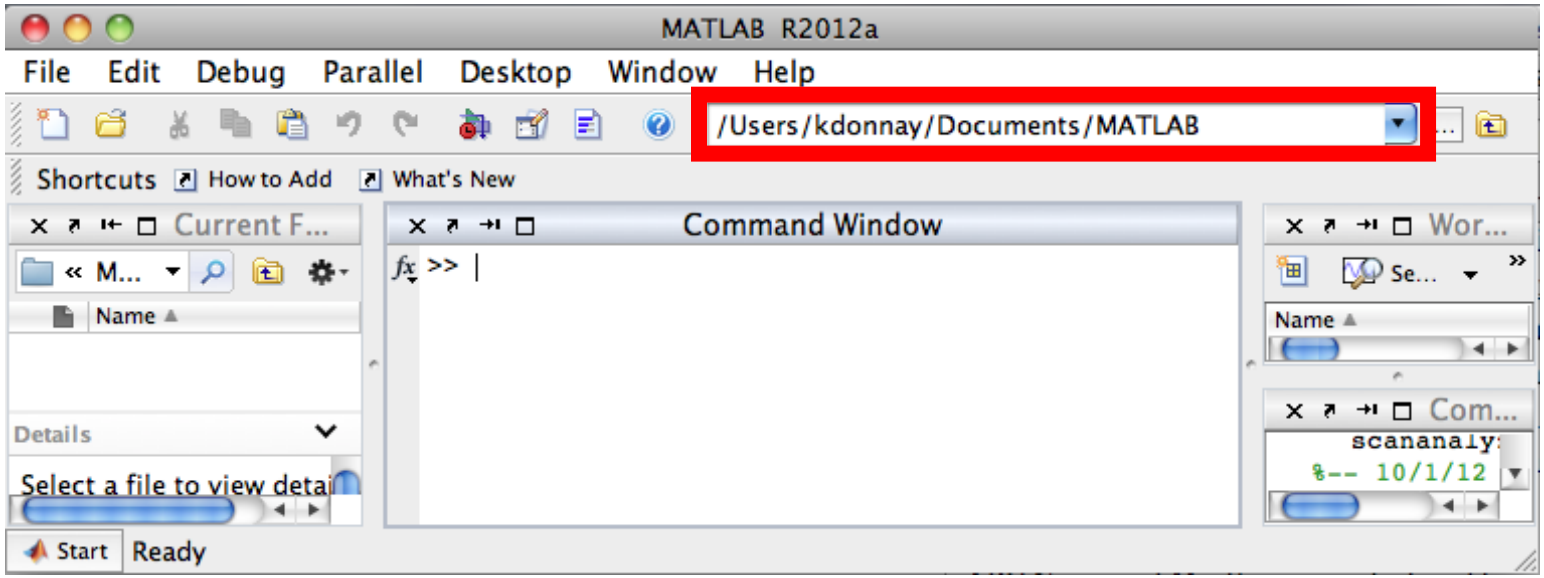
  you will find the files:

  `countries.m`

  `cities.m`

Exercises: Datasets

- Download the files **`countries.m`** and **`cities.m`** and save them in the working directory of MATLAB.



Exercises: Datasets – countries

- This dataset **`countries.m`** contains a matrix **`A`** with the following specification:
- Rows: Different countries
- Column 1: Population
- Column 2: Annual growth (%)
- Column 3: Percentage of youth
- Column 4: Life expectancy (years)
- Column 5: Mortality

Exercises: Datasets – countries

- Most often, we want to access complete columns in the matrix. This can be done by

  `A(:, index)`

  For example if you are interested in the life-expectancy column, it is recommended to do:

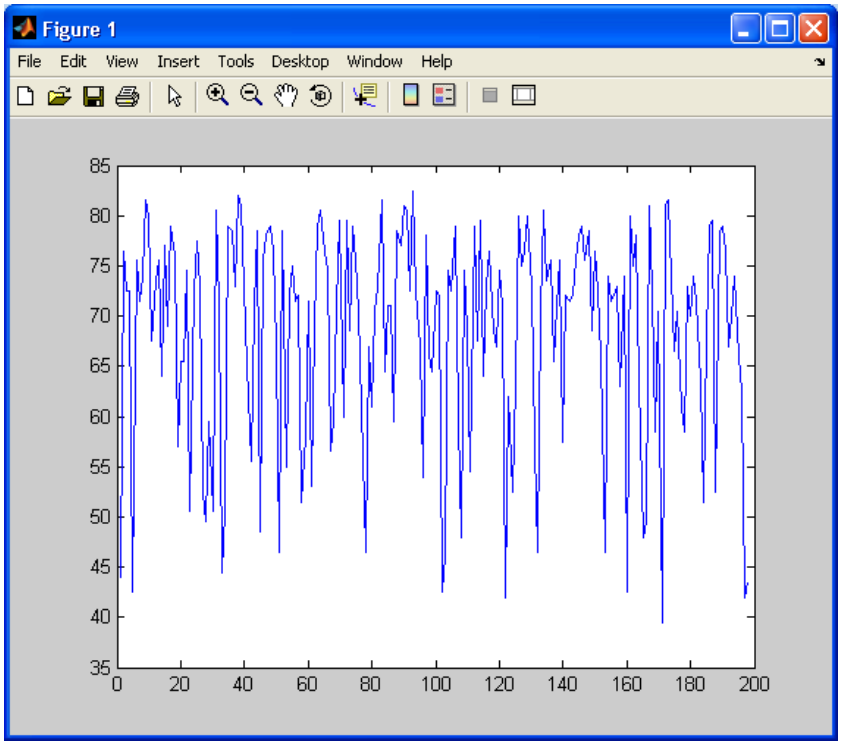
  `>> life = x(:,4);`

  and then the vector **`life`** can be used to access the vector containing all life expectancies.

Exercises: Datasets – countries

- The `sort()` function can be used to sort all items of a vector in inclining order.

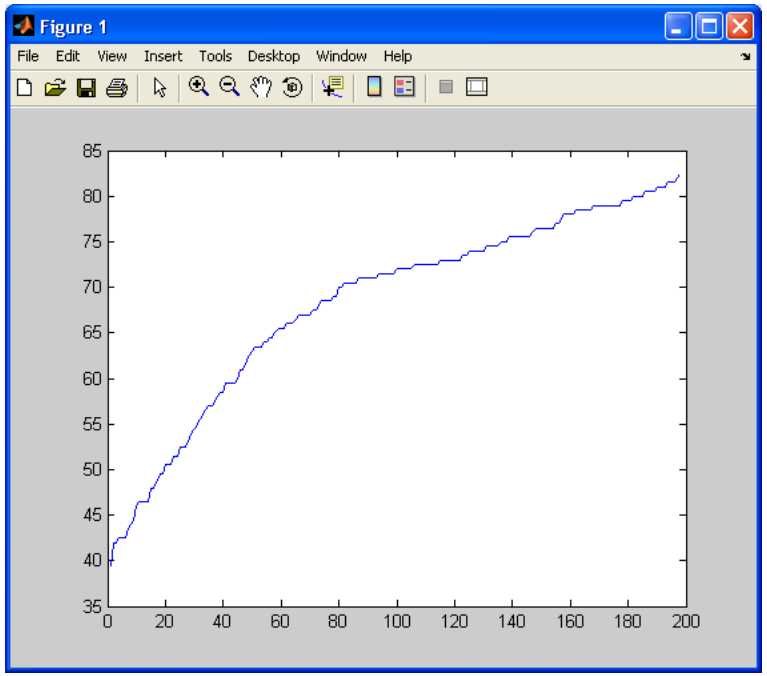
```
>> life = A(:, 4);
>> plot(life)
```



Exercises: Datasets – countries

- The `sort()` function can be used to sort all items of a vector in inclining order.

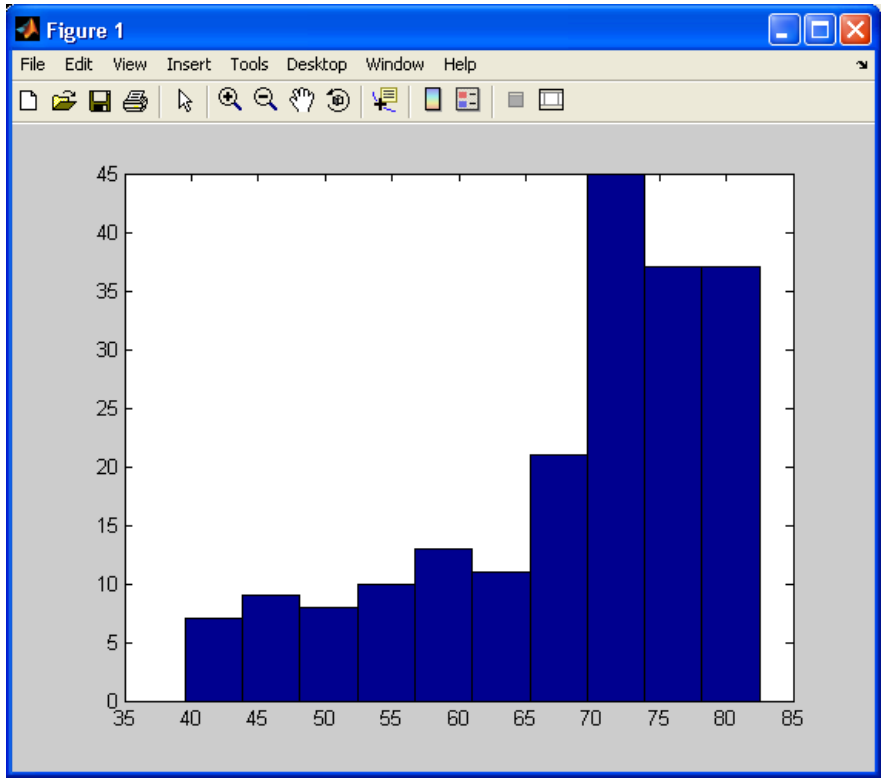
```
>> life = A(:, 4);
>> lifeS = sort(life)
>> plot(lifeS)
```



Exercises: Datasets – countries

- The histogram **`hist()`** is useful for getting the distribution of the values of a vector.

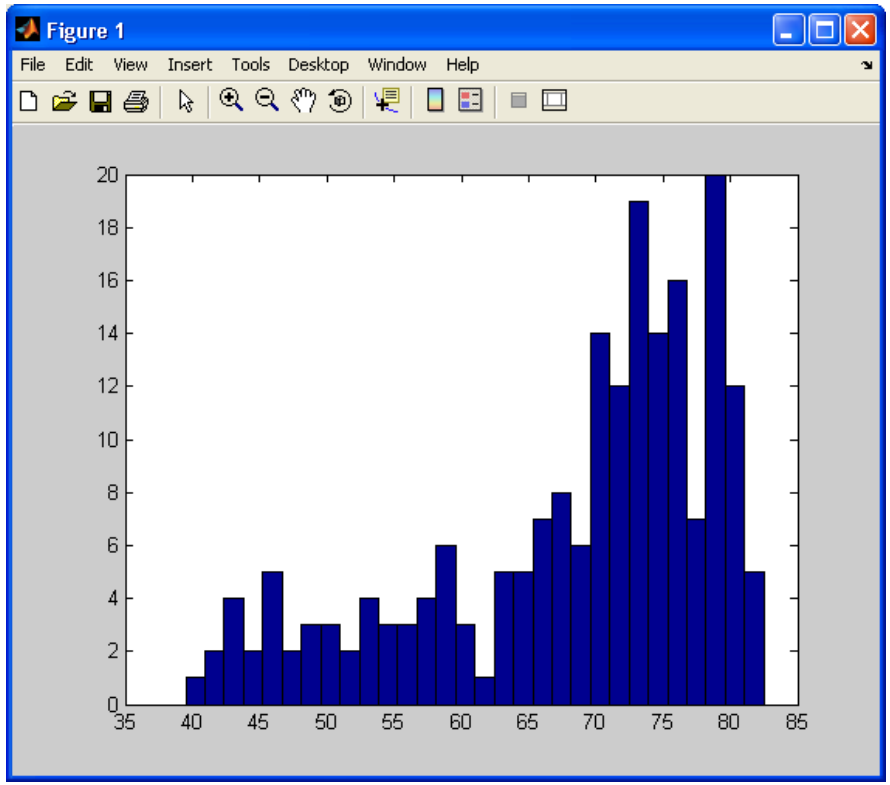
```
>> life = A(:, 4);
>> hist(life)
```



Exercises: Datasets – countries

- Alternatively, a second parameter specifies the number of bars:

```
>> life = A(:, 4);
>> hist(life, 30)
```



Exercise 1

- **Statistics**: Generate a vector of N random numbers with **`randn(N,1)`**
  Calculate the mean and standard deviation. Do the mean and standard deviation converge to certain values, for an increasing *N*?
  Optional: Display the histogram and compare the output of the following two commands
- **`hist(randn(N,1))`**
- **`hist(rand(N,1))`**

Exercise 2

- **Demographics**: From the countries.m dataset, find out why there is such a large difference between the mean and the median population of all countries.

  Hint: Use **`hist(x, n)`**
  Also **`sort()`** can be useful.
- Plus: play with **`subplot()`**

Exercise 3

- **Demographics**: From the countries.m dataset, see which columns have strongest correlation. Can you explain why these columns have stronger correlations?

  Hint: Use `corrcoef()` to find the correlation between columns. Use `imagesc()` to get an immediate visualization of the correlations.

Exercise 4 – optional

- **Zipf's law**: Zipf's law says that the rank, **`x`**, of cities (1: largest, 2: 2nd largest, 3: 3rd largest, ...) and the size, **`y`**, of cities (population) has a power-law relation: $y \sim x^b$
- Test if Zipf's law holds for the cases in the **`cities.m file`**. Try to estimate *b*.
  Hint: Use **`log()`** and **`plot()`** (or **`loglog()`**)
- Plus: use the fitting tool **`cftool()`** to get b

References

- Additional material on plotting on the home page of the course http://www.soms.ethz.ch/matlab
- "Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigene Cultures", *Science* **320** (5880): 1217
- Fechner law