SCS2211- Laboratory II

Introduction to Octave

By Dasun Bamunuarachchi/ slides adapted from Randil Pushpananda

dtb@ucsc.cmb.ac.lk

University of Colombo School of Computing

Content

- Linear Equations & Matrices
- Polynomials
- Curve Fitting & roots
- Data Analysis

Systems of Linear Equations

- Given a system of linear equations
  - $x + 2y - 3z = 5$
  - $-3x - y + z = -8$
  - $x - y + z = 0$
- Construct matrices so the system is described by Ax=b
  - » A = [1 2 -3;-3 -1 1;1 -1 1];
  - » b = [5;-8;0];
- And solve with a single line of code!
  - » x = A\b; %x is a 3x1 vector containing the values of x, y, and z

- **The \ will work with square or rectangular systems.**
- Gives least squares solution for rectangular systems. Solution depends on whether the system is over or underdetermined.

Systems of Linear Equations

$$
\begin{aligned}
x - y + z &= 0 \\
-3x - y + z &= -8 \\
x + 2y - 3z &= 5
\end{aligned}
$$

$$
\begin{bmatrix} 1 & 2 & -3 \\ -3 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ -8 \\ 0 \end{bmatrix}
$$

```
>> x = A\b
x =

   2.0000
   3.0000
   1.0000

>> A,b
A =

   1   2  -3
  -3  -1   1
   1  -1   1

b =

   5
  -8
   0
```

Worked Example: Linear Algebra

- Solve the following systems of equations:

$$\begin{aligned} x+4y&=34 \\ -3x+y&=2 \end{aligned}$$

```
» A=[1 4;-3 1];
» b=[34;2];
» x=inv(A)*b;
» x=A\b;
```

$$\begin{aligned} 2x-2y&=4 \\ -x+y&=3 \\ 3x+4y&=2 \end{aligned}$$

```
» A=[2 -2;-1 1;3 4];
» b=[4;3;2];
```

4

```
» x=A\b;
```

More Linear Algebra

- Given a matrix
  - » **A = [1 2 -3;-3 -1 1;1 -1 1];**
- Calculate the transpose (we discussed this before as well)
  - » **A^T = transpose(A)** % alternatively A' can give the transpose of complex conjugate of A (Hermitian conjugate)
- Calculate the rank of the matrix
  - » **r = rank(A);**% size of the largest square sub matrix of A with a non-zero determinant
- Calculate the trace of the matrix
  - » **TrA = trace(A);** % sum of the diagonal elements
- Returns the identity matrix
  - » **I = eye(size(A));** % diagonal matrix with size A OR eye(3)

More Linear Algebra

- Calculate the determinant
  - » **d = det(A);**
    - mat must be square; matrix invertible if det nonzero

- Get the matrix inverse
  - » **E = inv(A);**
    - if an equation is of the form A*x=b with A a square matrix, x=A\b is the same as x=inv(A)*b

- Get the condition number
  - » **c = cond(A);**
    - if condition number is large, when solving A*x=b, small errors in b can lead to large errors in x (optimal c==1)

Matrix Decompositions

- Matrix decomposition or Matrix factorization is a factorization of a matrix into a product of matrices.
- Some of them:
  - » [L,U] = lu(X)
    - LU decomposition
  - » [V,D] = eig(X)
    - Eigenvalue decomposition

  % Other decompositions exist

Polynomials

Polynomials

- Many functions can be well described by a high-order polynomial
- Octave represents a polynomials by a vector of coefficients
  - if vector P describes a polynomial

$$ax^3 + bx^2 + cx + d$$

[P(1) P(2) P(3) P(4)]

- P=[1 0 -2] represents the polynomial x^2-2
- P=[2 0 0 0] represents the polynomial $2x^3$

Polynomial Operations

- P is a vector of length N+1 describing an N-th order polynomial

- To get the roots of a polynomial

```
» r = roots(P)
```

  - r is a vector of length N

- Can also get the polynomial from the roots

```
» P = poly(r)
```

  - r is a vector length N

Polynomial Operations

- To evaluate a polynomial at a point

```
» y0 = polyval(P,x0)
```

  - x0 is a single value; y0 is the returned single value

- To evaluate a polynomial at many points

```
» y = polyval(P,x)
```

  - x is a vector; y is a vector of the same size

Polynomial Fitting

- Octave makes it very easy to fit polynomials to data

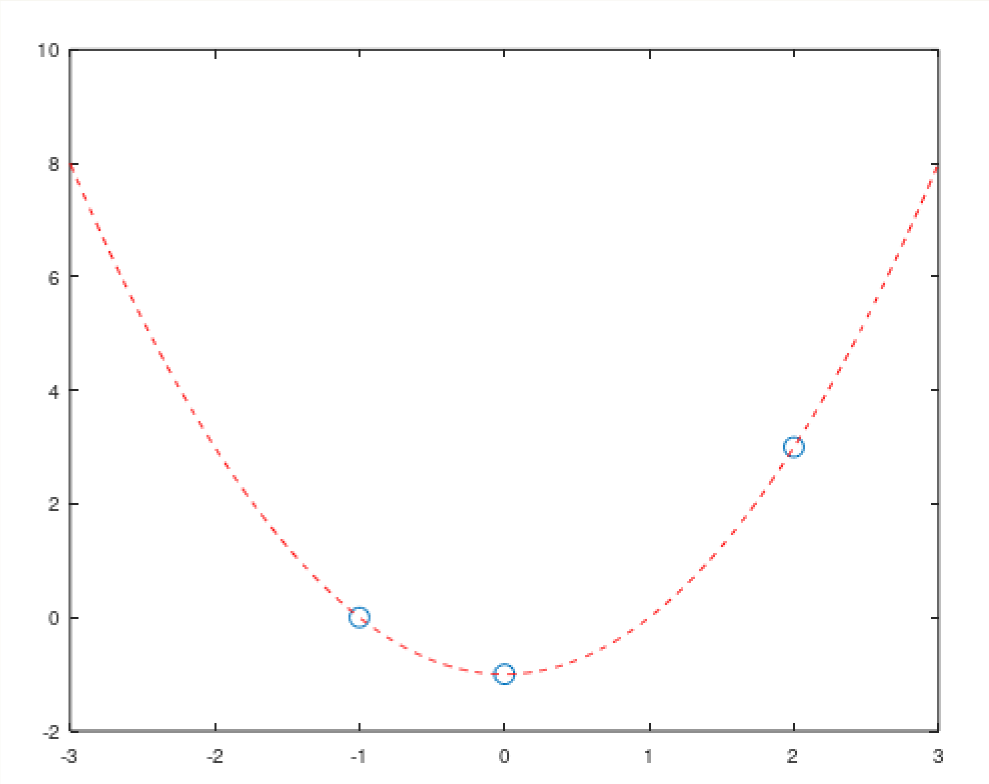
- Given data vectors X=[-1 0 2] and Y=[0 -1 3]

```
» p2=polyfit(X,Y,2);
```

  - finds the best (least-squares sense) second-order polynomial that fits the points (-1,0),(0,-1), and (2,3)

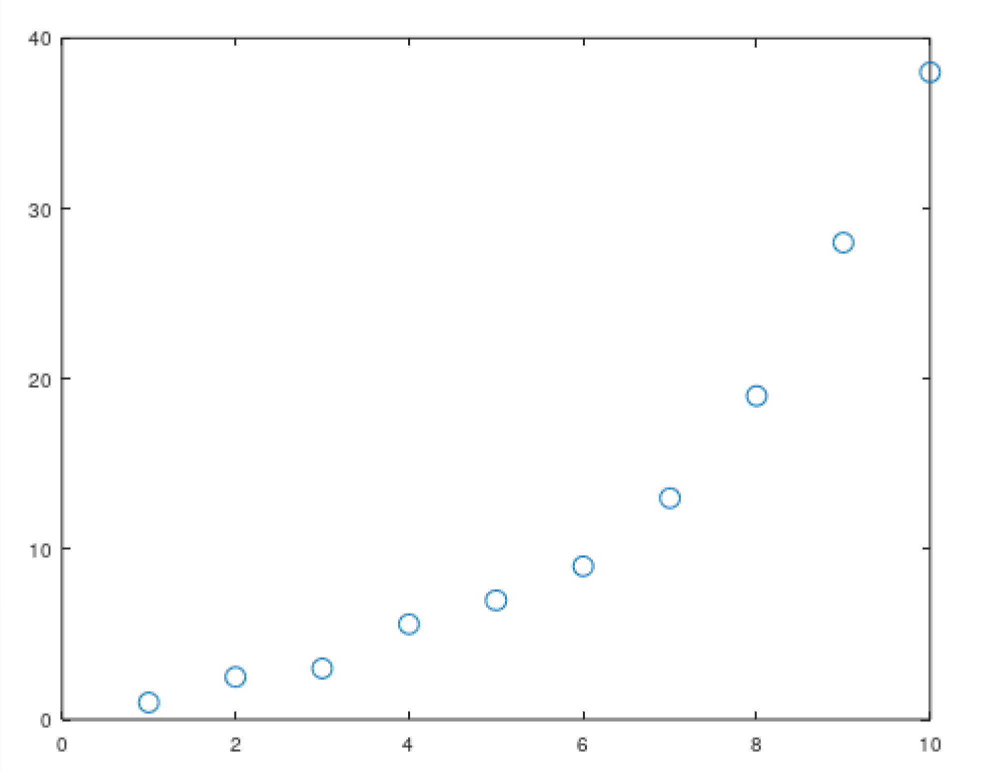
```
» plot(X,Y,'o', 'MarkerSize', 10);
» hold on;
» x = -3:.01:3;
» plot(x,polyval(p2,x), 'r--');
```

Polynomial Fitting



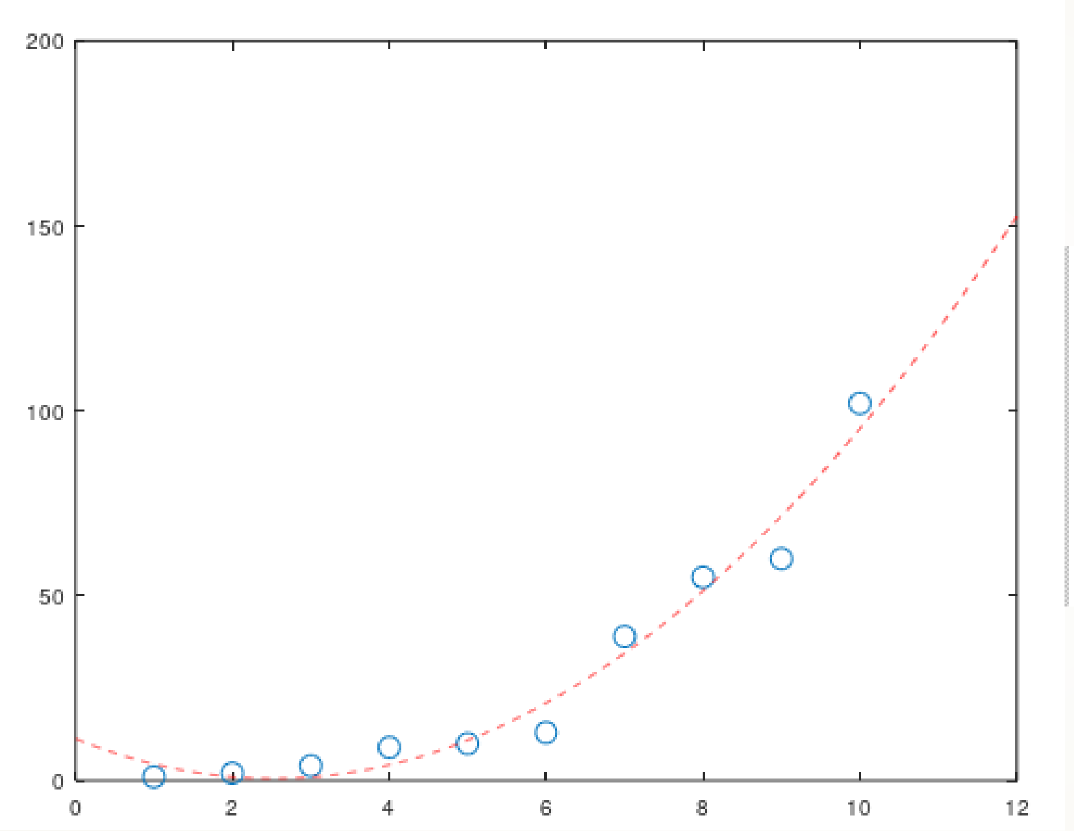
Points: X=[-1 0 2], Y=[0 -1 3]

Polynomial Fitting



Y = [1 2 4 9 10 13 39 55 60 102];
X = [1 2 3 4 5 6 7 8 9 10]
plot(X,Y,'o','MarkerSize', 10);

Polynomial Fitting




```
Y = [1 2 4 9 10 13 39 55 60 102];
X = [1 2 3 4 5 6 7 8 9 10];

p2=polyfit(X,Y,2);% find the polynomial p2
plot(X,Y,'o', 'MarkerSize', 10);
hold on;
x = 0:.01:12;
plot(x,polyval(p2,x), 'r--');
```

Exercise: Polynomial Fitting

- Evaluate $y = x^2$ for x=-4:0.1:4.

```
» x=-4:0.1:4;
» y=x.^2;
```

- Add random noise to these samples. Use **randn**. Plot the noisy signal with **.** markers

```
» y=y+randn(size(y));
» plot(x,y,'.');
```

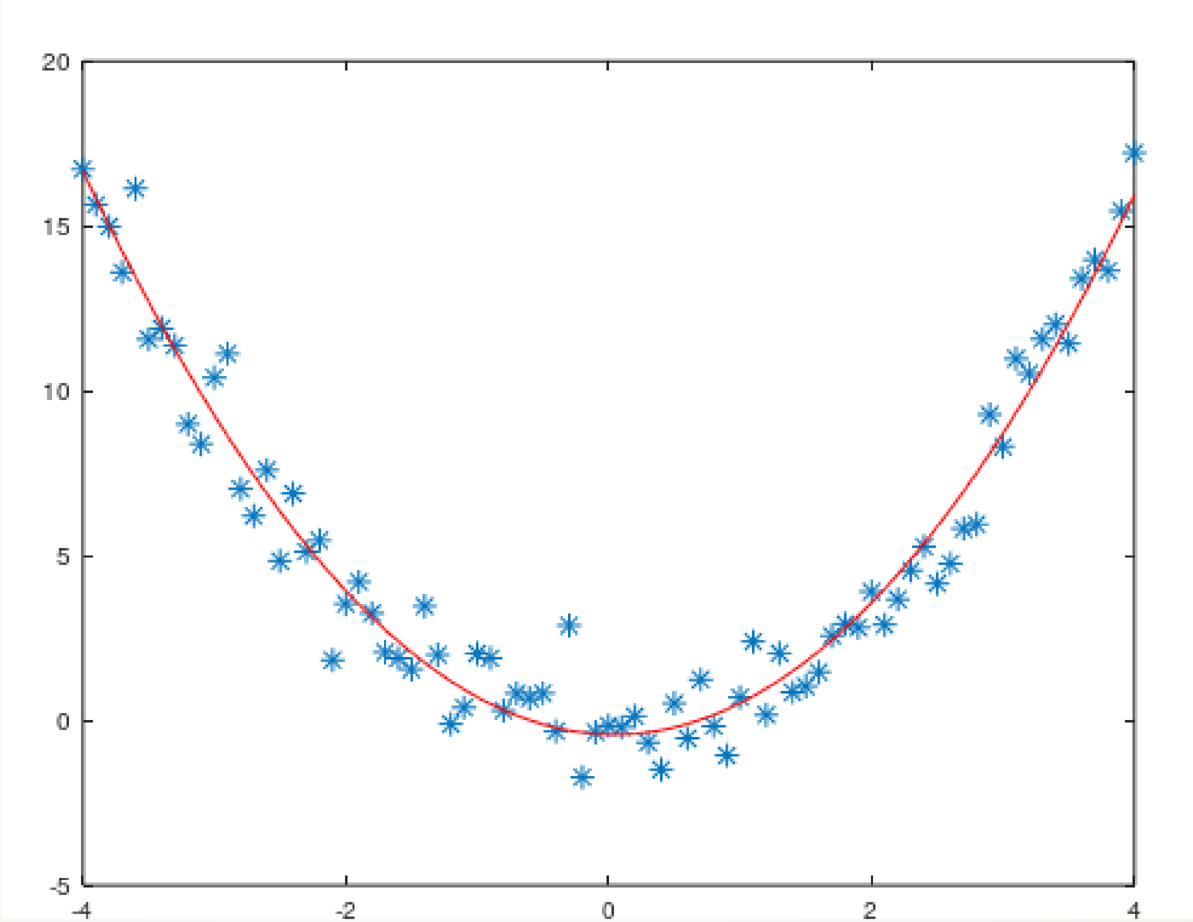
- Fit a 2nd degree polynomial to the noisy data

```
» p=polyfit(x,y,2);
```

- Plot the fitted polynomial on the same plot, using the same x values and a red line

```
» hold on;
» plot(x,polyval(p,x),'r')
```

Polynomial Fitting : noise to data



Some useful methods on functions

Root Finding for Nonlinear Functions

- Many real-world problems require us to solve f(x)=0
- Can use **fzero** to calculate roots for an arbitrary function

- **fzero** needs a function passed to it.

```
function [y] = myfunc1(x)
  y = cos(x).^2 - (x+3).^2 + 5;
endfunction
```

- Make a separate function file

```
» x = fzero('myfunc1',1)
» x = fzero(@myfunc1,1)
```

  - 1 - specifies a point close to where you think the root is
  - `x = fzero('myfunc1',[-5 5]) % OR an interval that changes the sign`

Minimizing a Function

- **fminbnd**: minimizing a function over a bounded interval
  - » `x=fminbnd('myfunc1',-1,2);`
    - myfunc1 takes a scalar input and returns a scalar output
    - myfunc1(x) will be the minimum of myfunc1 for -1≤x ≤ 2

```
>> x=fminbnd('myfunc1',-1,2);
>>
>> x
x =   2.0000
>> x=fminbnd('myfunc1',-1,1);
>> x
x =   0.99995
>>
```

Anonymous Functions

- You do not have to make a separate function file
  - » `x=fzero(@myfun,1)`
    - What if myfun is really simple?

- Instead, you can make an anonymous function
  - » `x=fzero(@(x)(cos(x).^2+x.^2-1),1);`

    input   function to evaluate

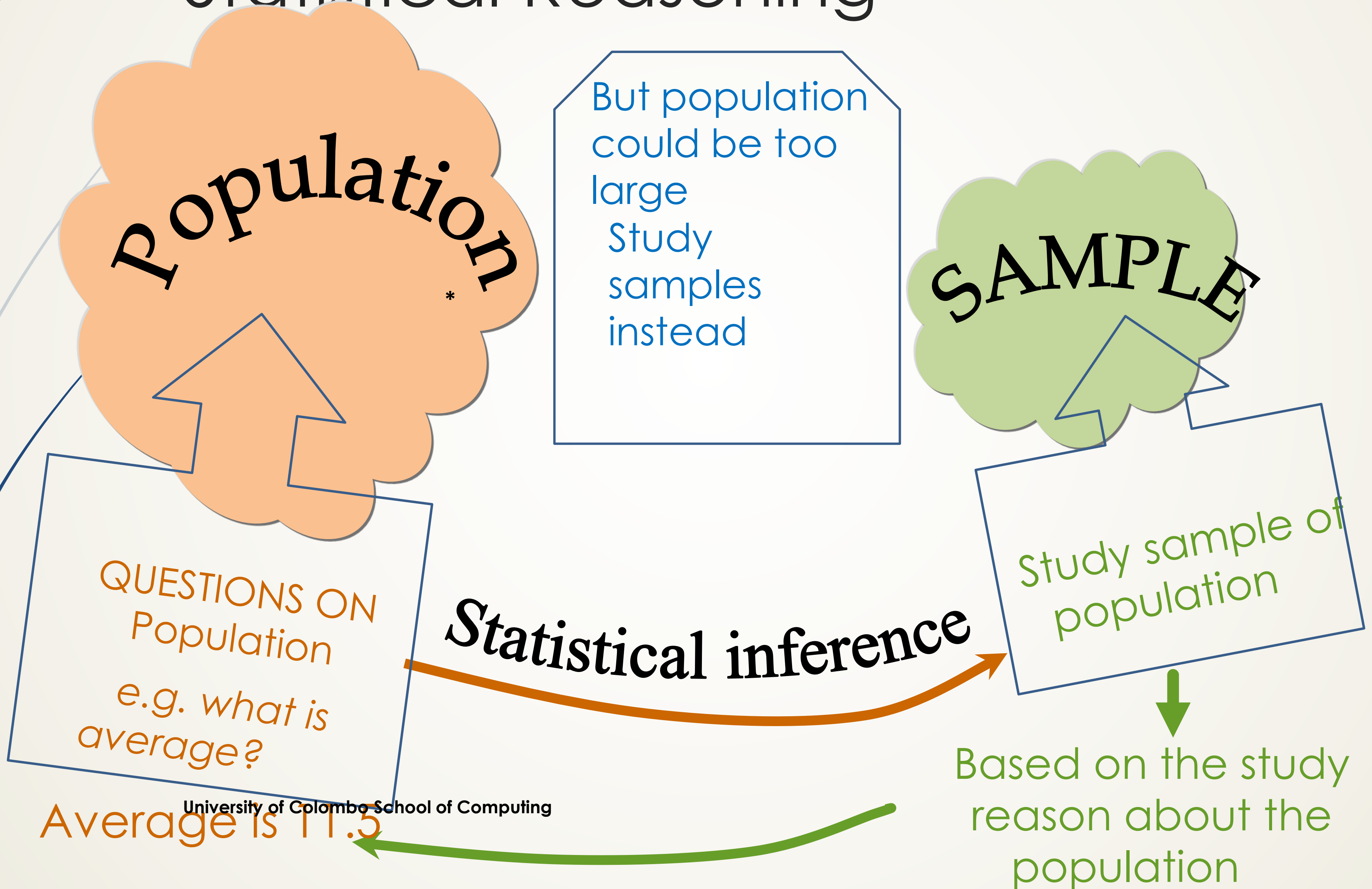
  - » `x=fminbnd(@(x) (cos(x).^2+x.^2-1),-1,2);`

- Can also store the function handle
  - » `func = @myfunc1; % save it`
  - » `func(1:10); % call it!`

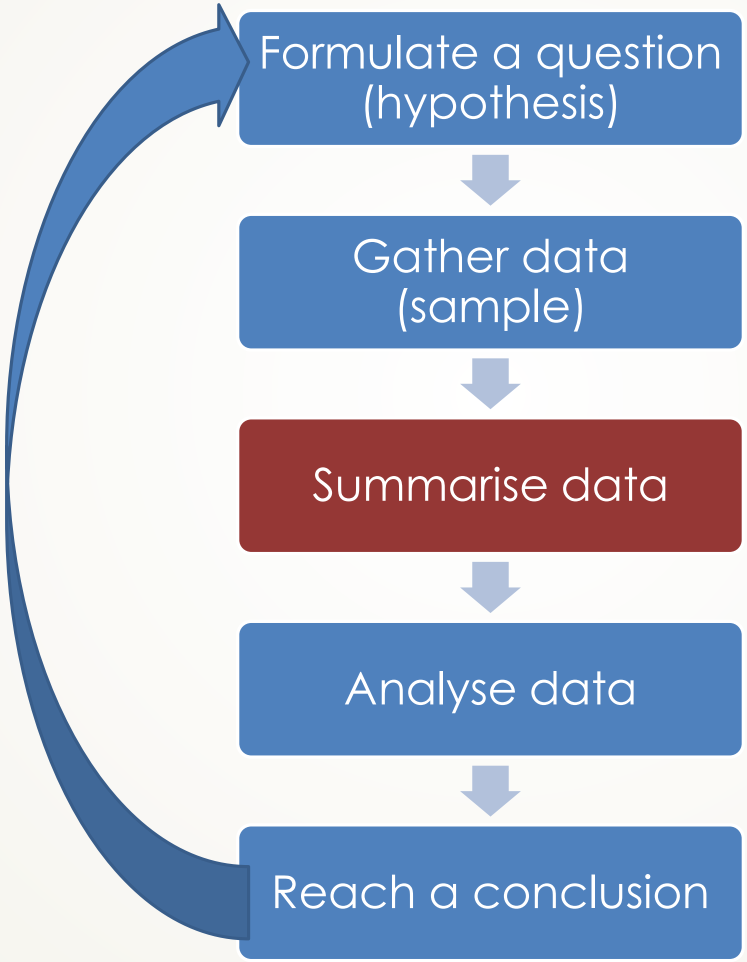
```
>> x=fminbnd(fhandle,-1,2)
x =   2.0000
>>
>> x=fminbnd(@myfunc1,-1,2)
x =   2.0000
>>
```

Data Analysis

Statistical Reasoning



Summarised Steps



How do we summarise a sample of data?

The following sample of results were obtained from an experiment. What do they tell us? Is it good? Where should we start?

2.34	4.34	5.43	6.54	7.87	7.65	9.87	6.54	6.54	3.21	4.67	9.87	9.76	3.24	6.54	7.65	8.56	3.23	4.32
7.87	7.65	9.87	4.67	9.87	9.76	9.87	6.54	9.87	9.76	3.24	6.54	4.34	5.43	4.34	5.43	7.65	9.87	6.54
32.4	7.87	7.65	9.87	6.54	4.78	4.67	9.87	9.76	4.34	5.43	7.65	9.87	4.34	5.43	4.34	5.43	6.54	2.12
7.87	7.65	9.87	8.56	3.23	4.32	7.87	7.65	9.87	9.87	6.54	9.87	9.76	3.24	6.54	7.65	9.87	6.54	4.35
4.67	9.87	9.76	4.34	5.43	6.54	4.67	9.87	9.76	9.87	9.76	4.67	9.87	9.76	6.54	9.87	9.76	3.24	6.54
9.87	9.76	3.24	6.54	9.87	9.76	8.56	3.23	9.87	9.76	3.24	6.54	6.54	9.87	9.76	9.87	9.76	3.24	6.54
9.87	9.76	8.56	3.23	4.67	9.87	9.76	4.67	9.87	9.76	7.65	9.87	7.87	7.65	9.87	6.54	3.21	5.45	6.56
6.54	9.87	9.76	9.87	9.76	9.87	6.54	9.87	9.76	6.54	9.87	9.76	8.56	3.23	4.34	5.43	4.67	9.87	9.76
7.87	7.65	9.87	6.54	9.87	9.76	8.56	3.23	9.76	9.87	6.54	9.87	9.76	4.32	7.87	7.65	9.87	6.54	5.55
4.34	5.43	6.54	2.34	4.34	5.43	7.65	8.56	3.23	4.32	9.87	6.54	9.87	9.76	8.56	3.23	4.32	6.54	2.12
7.65	8.56	3.23	4.32	9.87	6.54	9.87	9.76	2.34	4.34	5.43	5.43	6.54	2.34	4.34	5.43	9.87	9.76	1.00

We need to summarise these numbers

How to summarise 1

some questions we may need to answer:

1. How much of it is there?

- Is there enough?
- Is there too much?

2. What sort of data do we have?

- Need to describe the data. Is it continuous or discrete?
- What type of data is it (quantitative or qualitative)
- Is it the right sort of data to answer our initial question?

3. What is a typical value?

- Where is the middle?

How to summarise 2

some more questions we need to answer:

4. How much does the sample vary?

- Does all the data look the same?
- Is it spread out and dispersed?
- How do I measure this?
- What does this mean anyway for my data?

5. What does my data look like?

- How is the data distributed?
- What shape is my data?

6. Will a picture help?

- What chart is appropriate to summarise my data?

What sort of data do we have?

What is the **level of measurement**?

- The amount of information the data convey
- Permissible mathematical and statistical operations on them
- Four major types defined:
  - Nominal
  - Ordinal
  - Interval
  - Ratio

Nominal data

Categorical scale

- A scale in 'name only'
- No numerical value, no order
- Simplest level of measurement
- For example, if we want to categorize male and female respondents, we could use a nominal scale of 1 for male, and 2 for female, but 1 and 2 in this case do not represent any order or distance. They are simply used as labels.
- Other examples: hair colour, choice of newspaper, mode of travel to uni, favourite film

Ordinal data

- Categories again but can order them
- Interval between categories has no meaning
- Can use numbers to indicate place in order
- Example: For example, we can assign rank 1, which is higher than rank 2, and 2 is higher than 3 etc **student grades**

Interval data

- Real numbers
- Can order them like ordinal data
- Can perform meaningful mathematics on them
- Can talk about meaningful intervals between values
- No natural zero (what does 0 degrees C mean, Computer date ?)
- Arbitrary zero
- Examples: Temperature, Time

Ratio data

- Has all properties of interval data...
- Has a natural absolute zero
- Examples: age, height, speed

Levels of measurement summary

	Data Type			
Properties	Nominal	Ordinal	Interval	Ratio
Order	✗	✓	✓	✓
Equal Intervals	✗	✗	✓	✓
Absolute Zero	✗	✗	✗	✓
Typical usage	•Locations •Types of computer •Favourite colour	•Social class •Attitudes •Brand preference	•Temperature (C or F) •Time	•Age •Cost •Height •Speed

Importing Data to Octave

- Octave is a great environment for processing data. If you have a text file (*data.txt*) with some data:

jane	joe	jimmy	mary
10	12	17	9
9	7	4	3
0	8	7	4

- To import data from files on your hard drive, use

  importdata('file_Name')

  importdata('file_Name','delimiter')

  importdata('file_Name','delimiter',header_count)

Importing Data to Octave

- To access data >> A.data
- To access element >> A.data(1,2)
- To access headers >> A.colheaders
- To a header >> A.colheaders(1,4)

```
>> A = importdata('data.txt',' ',1)
A =

  scalar structure containing the fields:

    data =

        10    12    17     9
         9     7     4     3
         0     8     7     4

    textdata =
    {
      [1,1] = Jane      Joe      Jimmy      Mary
    }

    colheaders =
    {
      [1,1] = Jane
      [1,2] = Joe
      [1,3] = Jimmy
      [1,4] = Mary
    }
```

Mean, Variance, Standard Deviation

- Mean $$\bar{s} = \frac{1}{N}\sum_{i=i}^{N} s_i \, ,$$ >> mean(s)
- Variance $$\text{var}(s) = \frac{1}{N-1}\sum_{i=i}^{N} (s_i - \bar{s})^2 .$$ >> var(s)
- Standard Deviation $$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} (x_i - \bar{x})^2},$$ >> std(s)
- Skewness $$\text{skew}(s) = \frac{1}{N}\sum_{i=1}^{N} \left(\frac{s_i - \bar{s}}{\sigma}\right)^3 .$$ skewness(s)

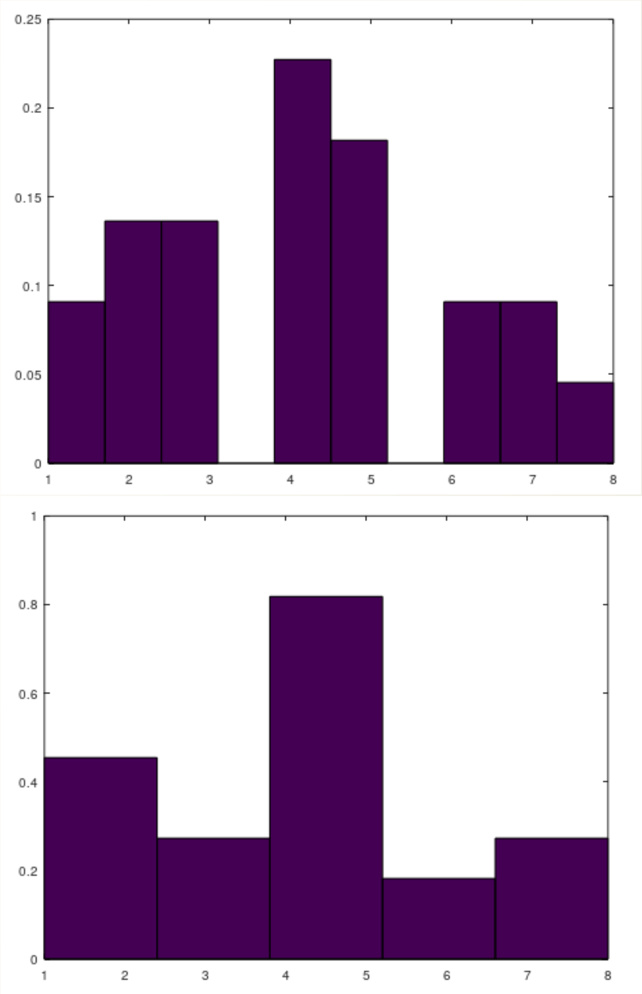
Simple Descriptive Statistics

- Histograms:
  - a graphical display of data using bars of different heights.
  - It is similar to a Bar Chart, but a histogram groups numbers into ranges.
  - We need to decide what ranges to use

>> hist(s,10,1) % s - dataset i.e. vector

>> hist(s,5,2) % 5 number of bins

% 2 – normalize data to this value



Thank You!