Lecture 3 II

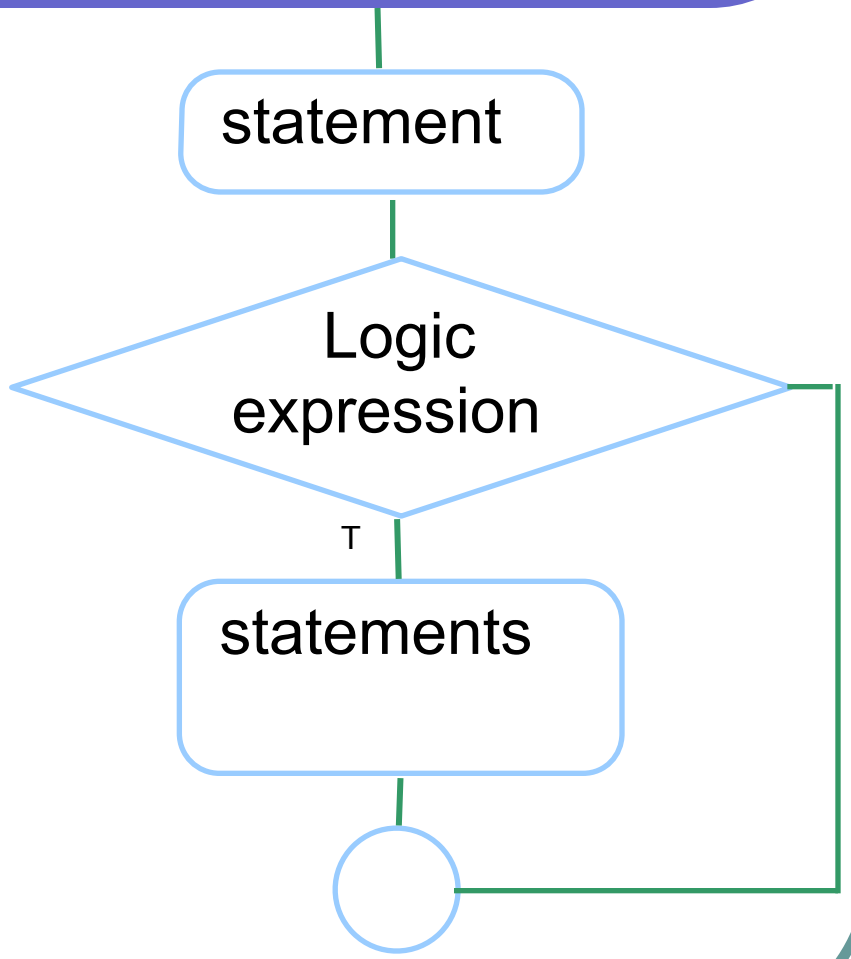
- If, Switch
- Vector codes: Find
- Line fitting

Outline

- Conditional flow chart
- Thresholding
- Piece-wise function
- Expression evaluation
- Grading
- Unit circle indication
- Matlab find

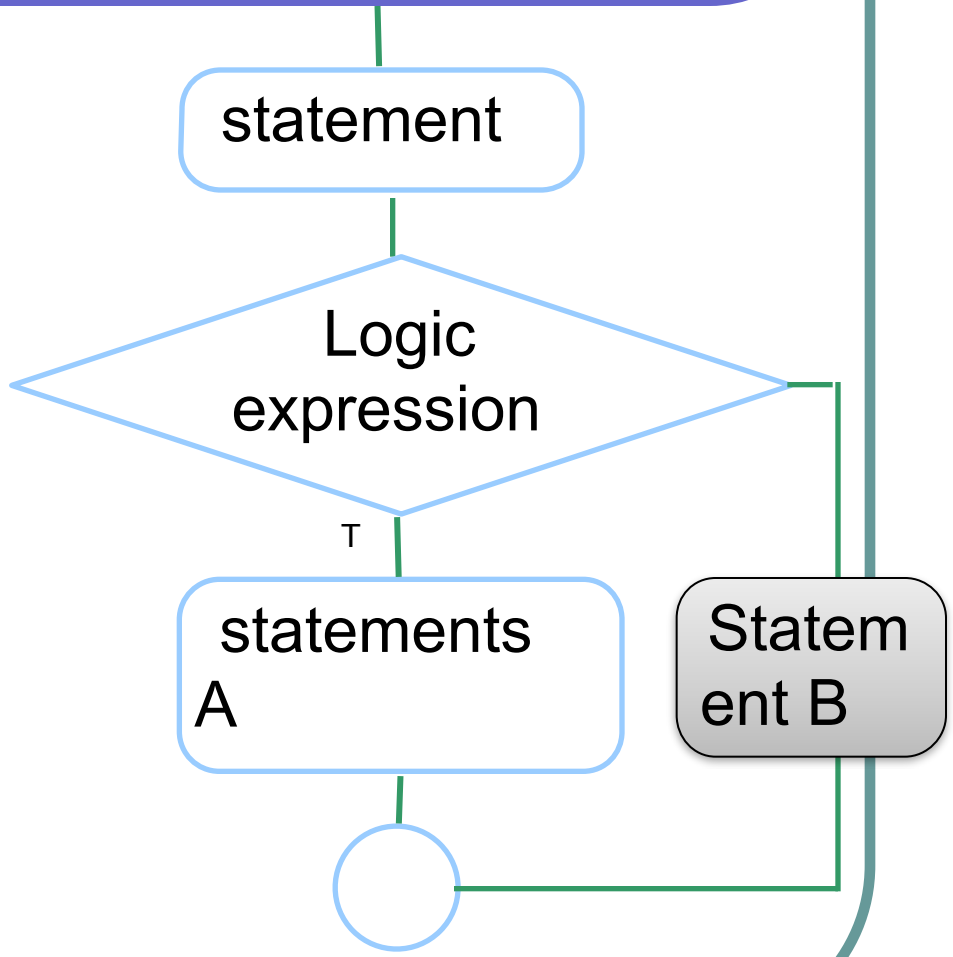
Conditional structure

```
if logic expression
    Statements
end
```



Conditional structure

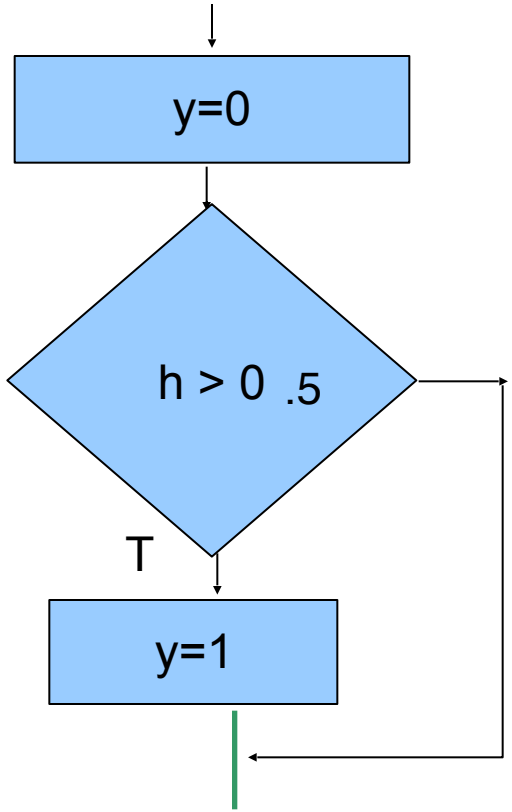
```
if logic expression
    Statements A
else
     Statements B
end
```



Threshold function

$$f(h) = 1, \quad \text{if } h > 0.5$$

$$= 0, \quad \text{otherwise}$$



threshold

```
function y=my_threshold(h)
    y=0;
    if h > 0.5
        y=1;
    end
```

Circle support

$$\Omega = \{ x \mid \| x \| \leq 1, x \in R^2 \}$$

$$f(x) = 1 \quad \text{if } x \in \Omega$$

$$= 0 \quad \text{otherwise}$$

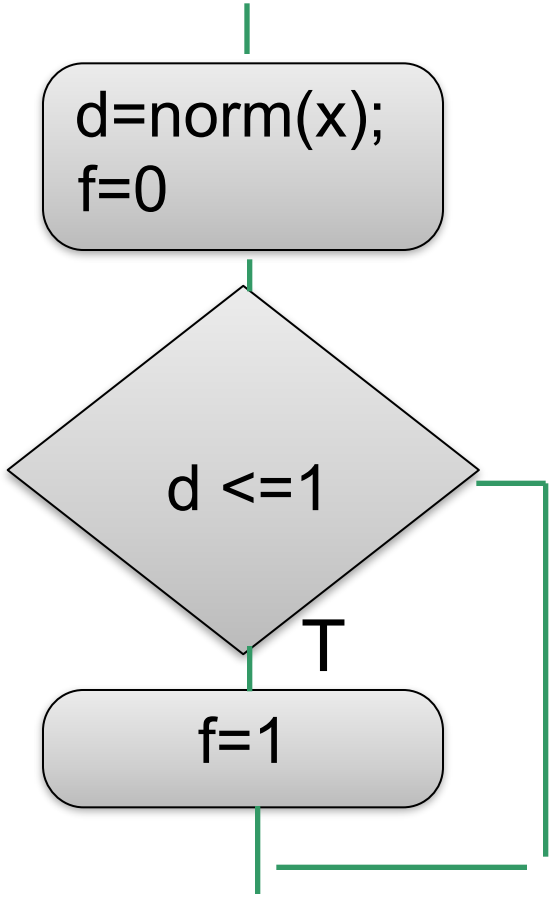
Support indication

function f=i_circle(x)

iPad

MATLAB Mobile

```
>> x=rand(1,2);norm(x)

ans =

    0.9147

>> x=[1 0];norm(x)

ans =

     1
```



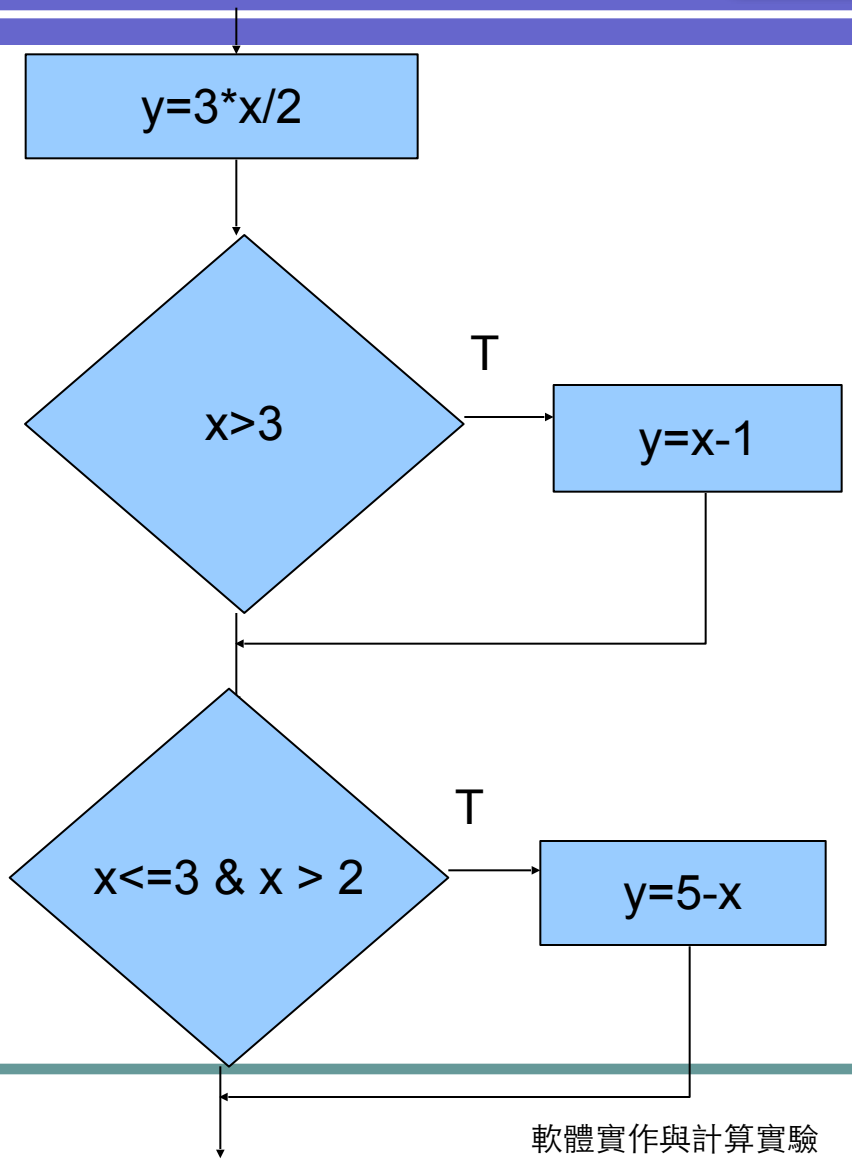
piece-wise function

$$f(x) = x - 1 \quad \text{if } x > 3$$

$$= 5 - x \quad \text{if } 3 \geq x > 2$$

$$= \frac{3x}{2} \quad \text{if } 2 \geq x$$

function y=pwfun(x)



```
function  y=f(x)
   y=3*x/2;
   if x>3
        y=x-1;
   end
   if x <= 3 & x>2
        y=5-x;
   end
```

Expression evaluation

- Evaluate an infix-order expression
- It is required to apply an appropriate operation to two operands according to the operator

```
z=prefix('*',10,15)
```

日期: 2013年3月28日 上午9:36:57 [GMT+08:00]
收件人: <jmwu@mail.ndhu.edu.tw>
標題: **【美商特翠歐公司徵聘人工智慧程式 (Artificial Intelligence),類神經網路設計師 (Neural Networks), 演算法工程師, Quantitative Analyst乙名】**

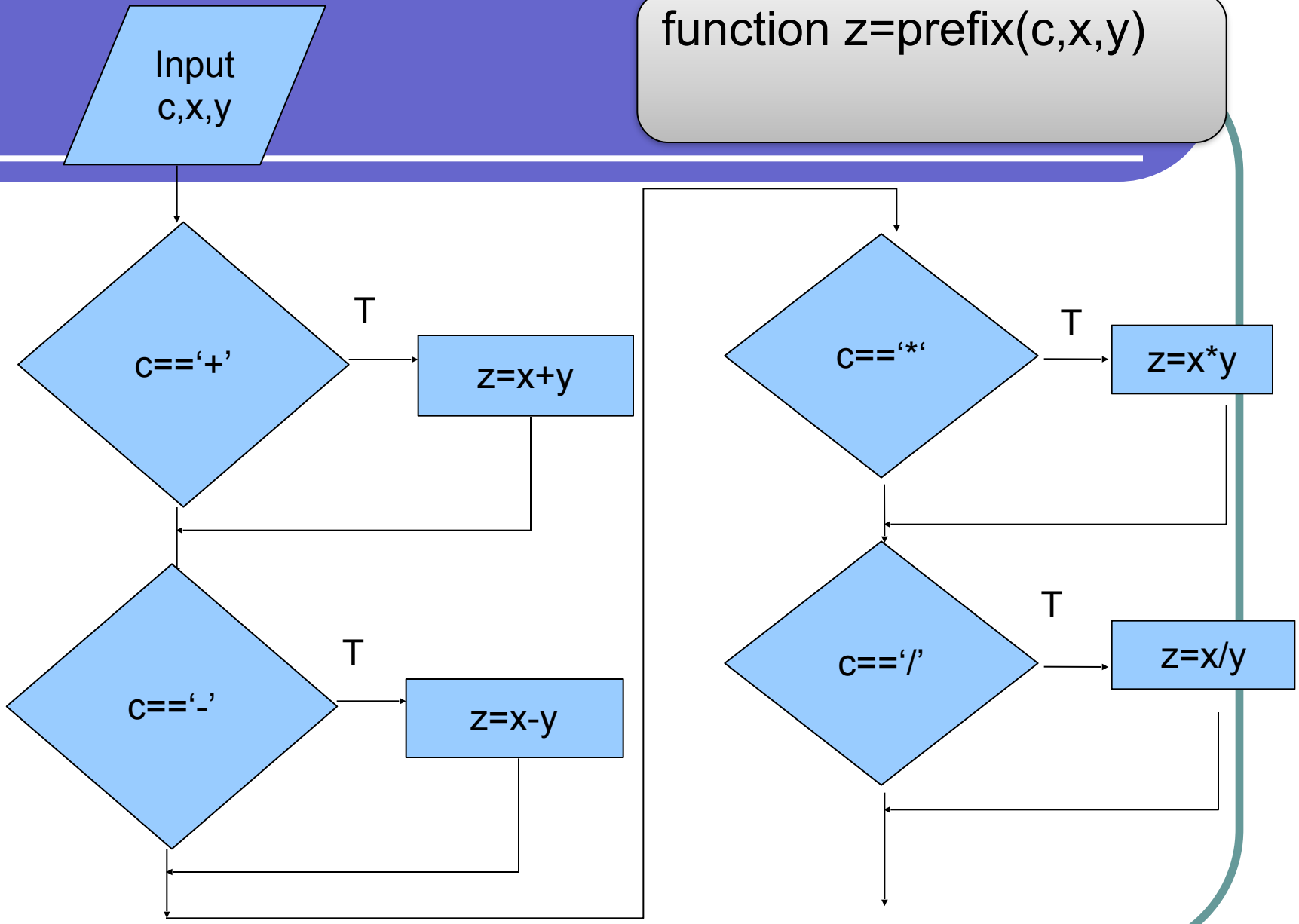
【美商特翠歐公司徵聘人工智慧程式 (Artificial Intelligence),類神經網路設計師 (Neural Networks), 演算法工程師, Quantitative Analyst乙名】
公司網站http://www.tetrio.com/

【起聘日期】
依到職日起聘。

【應徵資格】
-Experience in one or more of these areas: genetic algorithms, neural networks, machine learning, financial analysis
-R , Matlab or other statistical programming language (R is preferred)
-Python, C/C++, or Java (enough fluency to implement basic financial algorithms)
-Maters' degree in mathematics, computer science, engineering, physics or statistics
-Prior experience in finance or trading is preferred, but not required.

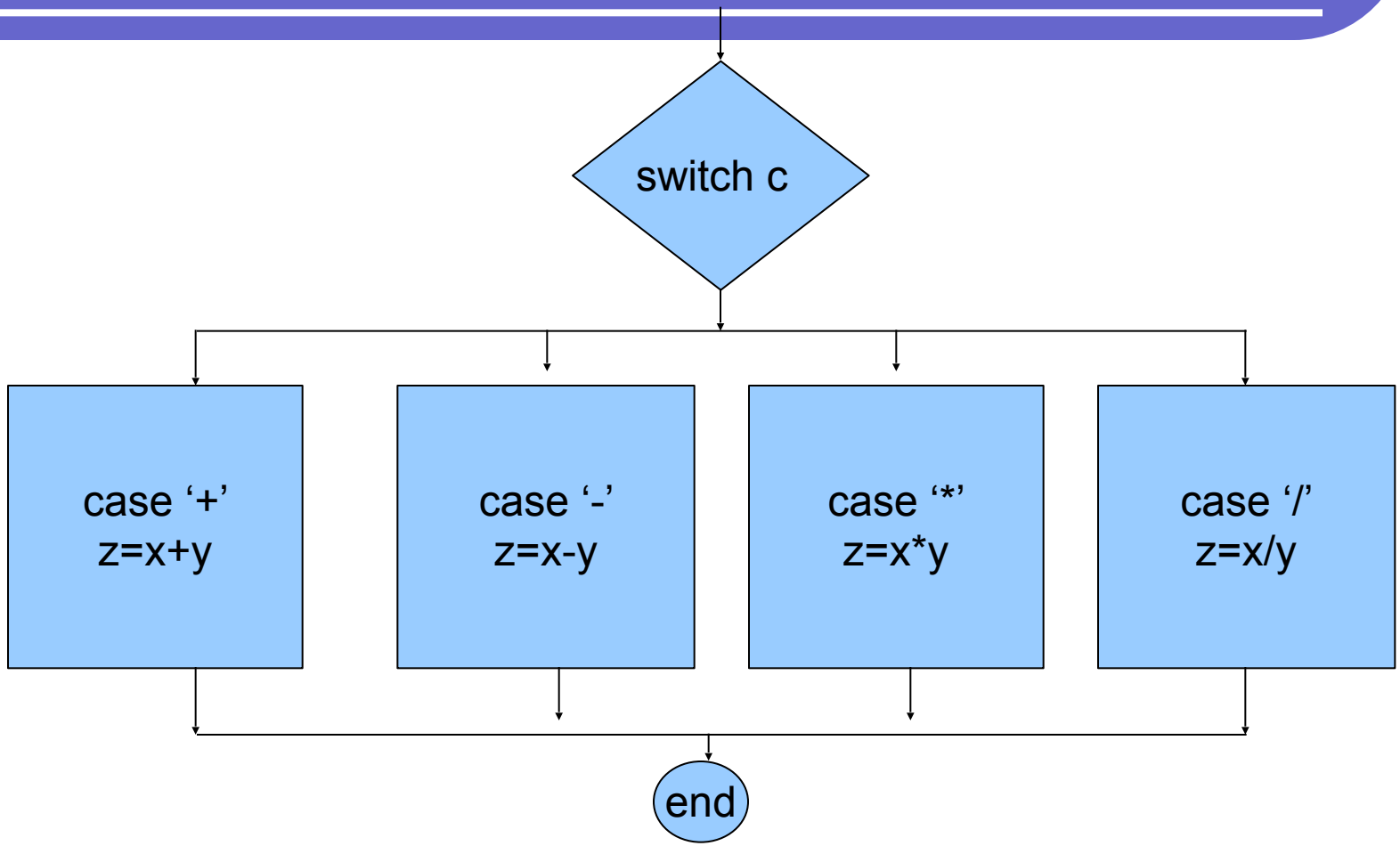
1. 對任以下研究方法有一個或多個經驗者：遺傳演算法，神經網絡，機器學習，或金融分析
2. R、Matlab或其他統計編程語言（R優先）。
3. Python、C / C + +或Java（能流暢運作基本的金融演算法）。
4. 數學、資工、物理、統計相關大學科系或研究所。
5. 有財務或金融交易經驗者為優，但並不是必要條件。
6. 溝通能力強，挑戰我們的想法，為自己的設計提出令人信服的論據。
7. 自發性從頭到尾完成項目，特別是小細節的部分。
8. 注重細節、能快速學習新技術、不斷進步。
9. 隨時追求乾淨、現代、簡潔的設計。
10. 能在沒有現有框架及樣版的情況下建立網頁。

function z=prefix(c,x,y)



```
function z=prefix(c,x,y)
if c==‘+’
    z=x+y;
end
if c==‘-’
    z=x-y;
end
if c==‘*’
    z=x*y;
end
If c==‘/’
    z=x/y;
end
```

switch



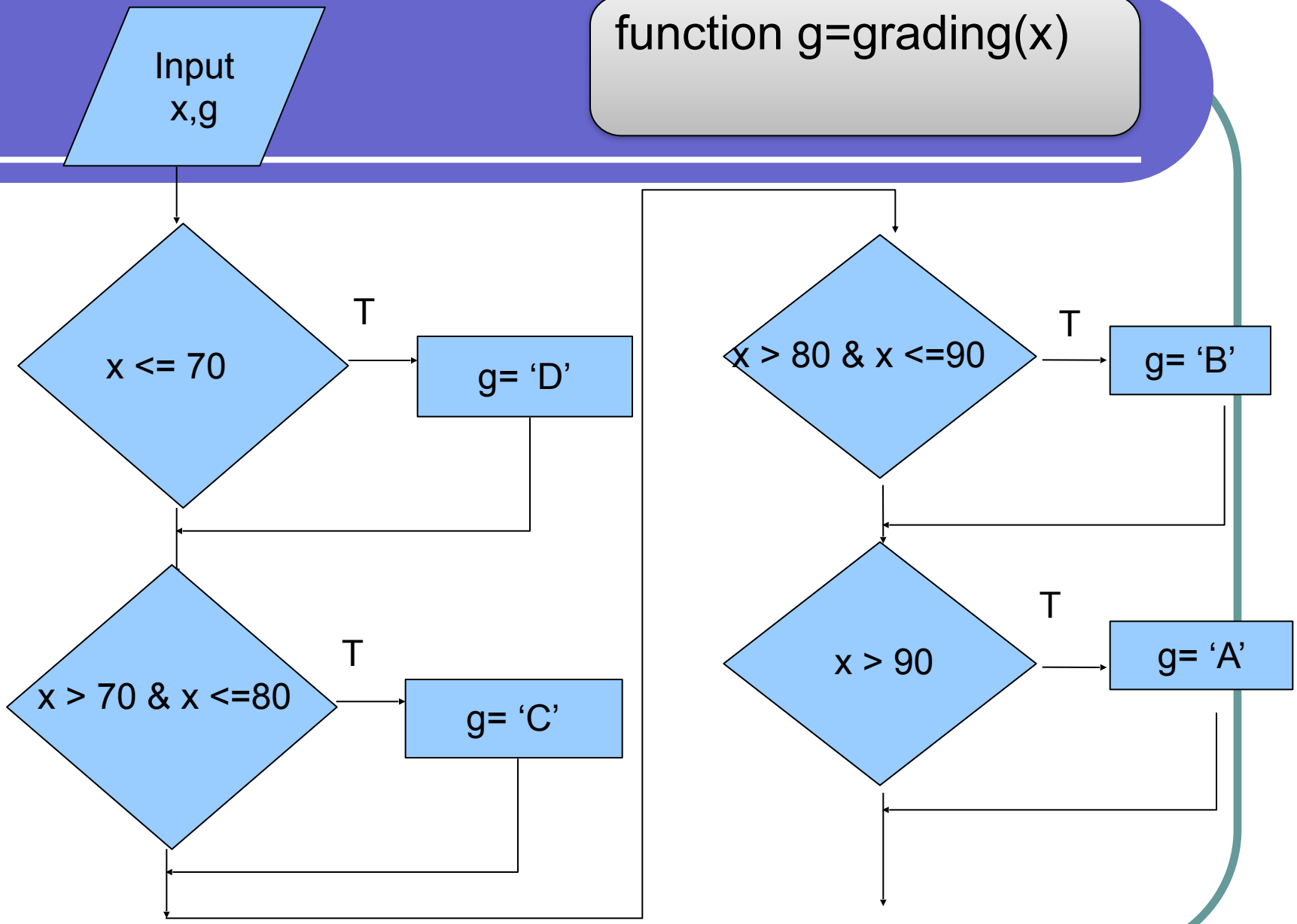
```
function z=prefix(c,x,y)
    switch c
        case '+'
            z=x+y;
        case '-'
            z=x-y;
        case '*'
            z=x*y;
        case '/'
            z=x/y;
    end
```

Grading

function g=grading(x)

- Grade scores
- x : a score
- Grade
  - g='D' if x <= 70
  - g='C' if x > 70 and x <= 80
  - g='B' if x > 80 and x <= 90
  - g='A' if x > 90

function g=grading(x)



Matlab find

- Seek indices of nonzero elements

```
>> x=[1 1 0 0 1 1]

x =

     1     1     0     0     1     1

>> ind=find(x)

ind =

     1     2     5     6
```

Find

- Return indices of nonzero elements in a vector
- ind=find(x)
  - x: a vector
  - ind: indices of non-zero elements in x

find(x>0)

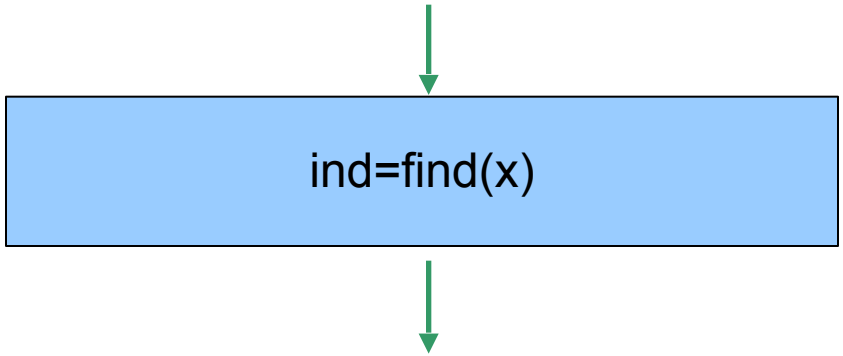
- Find indices of positive elements

```
>> x=[-1 -2 0 1 2 3]

x =

    -1    -2     0     1     2     3

>> find(x>0)

ans =

     4     5     6
```

Vector code

x =

-1 -2 0 1 2 3



ind =

1 2 4 5 6

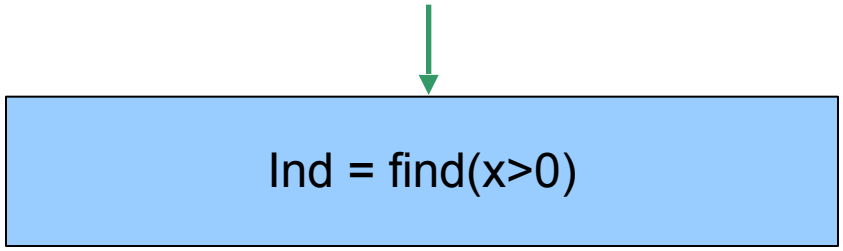
Vector code

- Mathematical Manipulation of all elements or selected partial elements in vectors or matrices at a time

Indices of positive elements

x =

-1 -2 0 1 2 3



Ind =

4 5 6

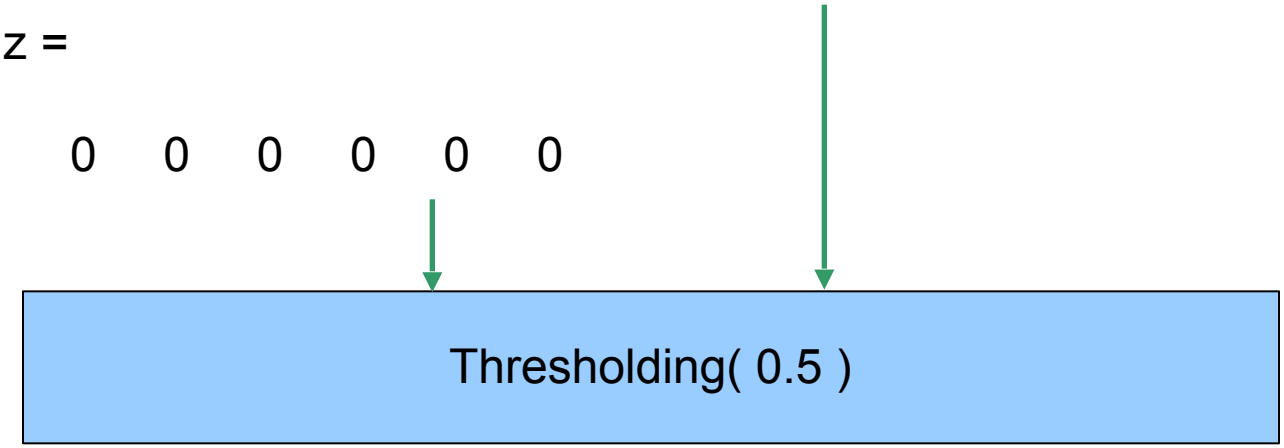
Thresholding

x =

0.9501 0.2311 0.6068 0.4860 0.8913 0.7621

z =

0 0 0 0 0 0



z =

1 0 1 0 1 1

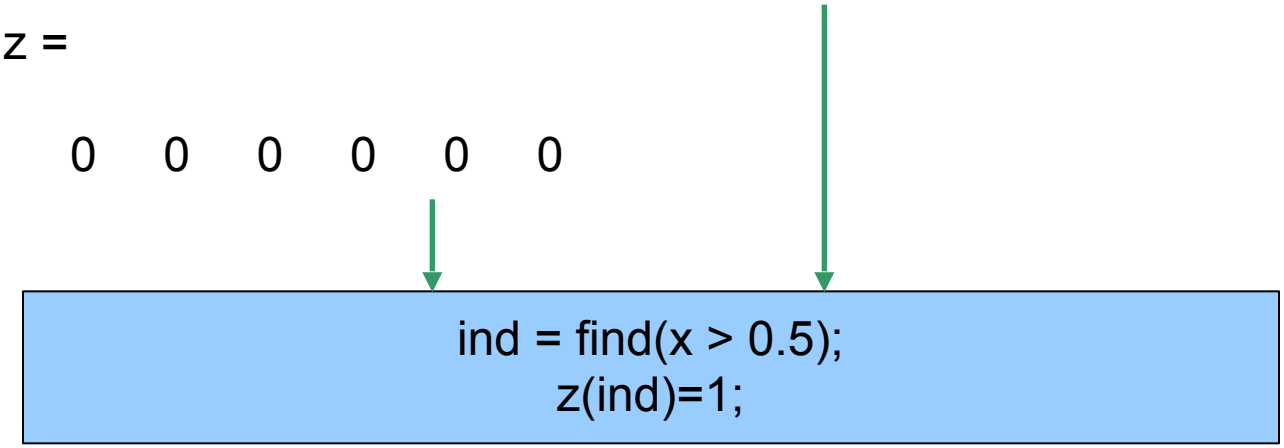
Thresholding by Matlab find

x =

0.9501 0.2311 0.6068 0.4860 0.8913 0.7621

z =

0 0 0 0 0 0



z =

1 0 1 0 1 1

Vector Thresholding

function z=v_threshold(x,t)

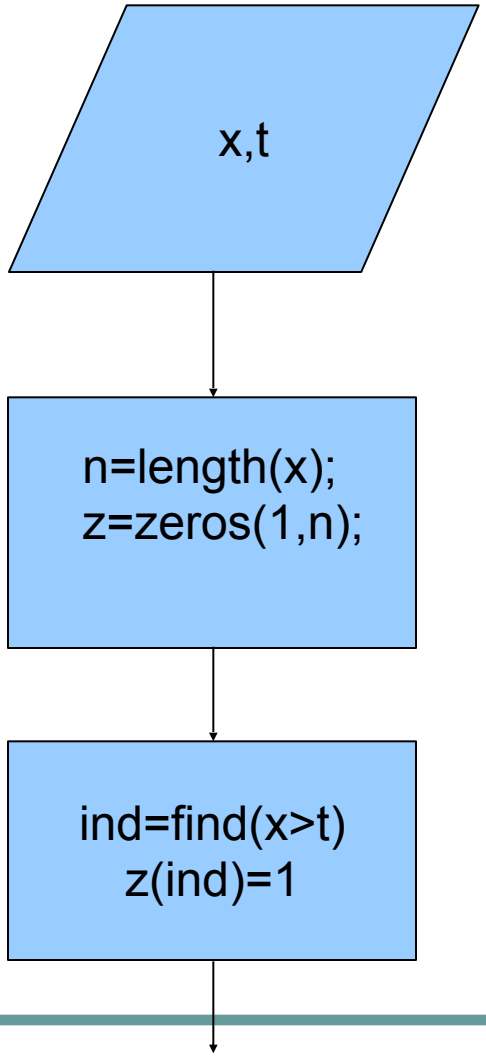
- Input : a vector
- Output: indices of elements that are greater than a given threshold

$$f(x;\theta) = 1 \text{ if } x > \theta$$

$$= 0 \text{ otherwise}$$

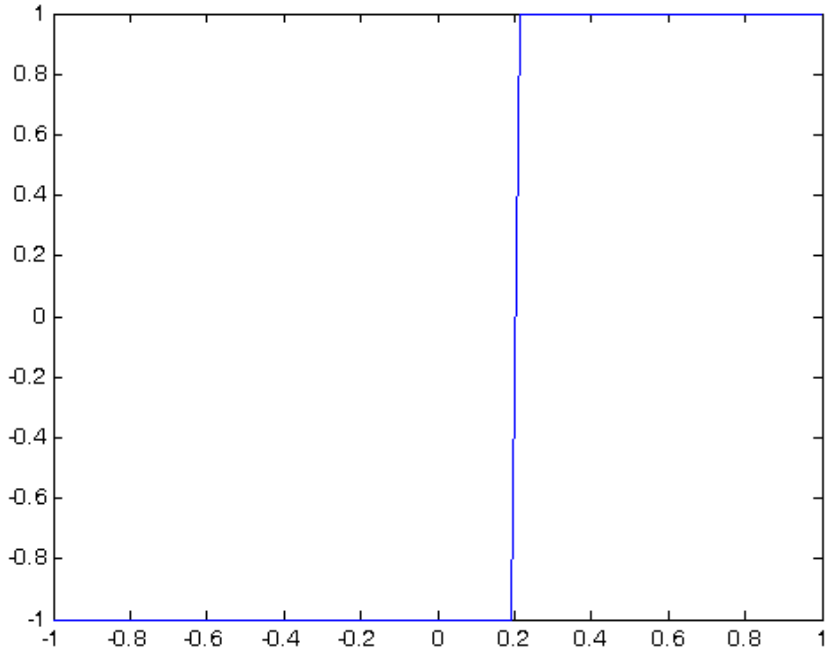
- The function is evaluated in vector form

function z=v_threshold(x,t)



```
function z=v_threshold(x,t)
    n=length(x);
    z=zeros(1,n);
    ind=find(x>t);
    z(ind)=1;

return
```

```
>> x=linspace(-1,1);
>> y=v_threshold(x,0.2);
>> plot(x,y)
```



Sampling from a square

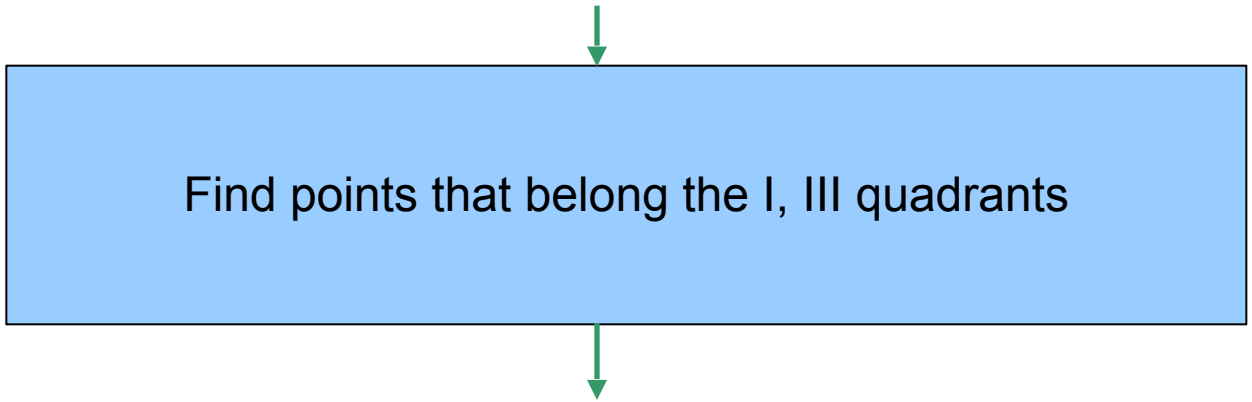
- Create uniform random points within [-1 1]x[-1 1]

```
>> X=rand(2,5)*2-1

X =

    0.8709   -0.1795   -0.8842    0.6263   -0.7222
    0.8338    0.7873   -0.2943   -0.9803   -0.5945
```

find

X =

0.8709	-0.1795	-0.8842	0.6263	-0.7222
0.8338	0.7873	-0.2943	-0.9803	-0.5945



0.8709	-0.8842	-0.7222
0.8338	-0.2943	-0.5945

Find points that belong the I, III quadrants

X =

0.8709 -0.1795 -0.8842 0.6263 -0.7222
0.8338 0.7873 -0.2943 -0.9803 -0.5945

z= x(1,:).*x(2,:)

z =

0.7262 -0.1413 0.2602 -0.6140 0.4293

Ind = find(z > 0)

Ind =

1 3 5

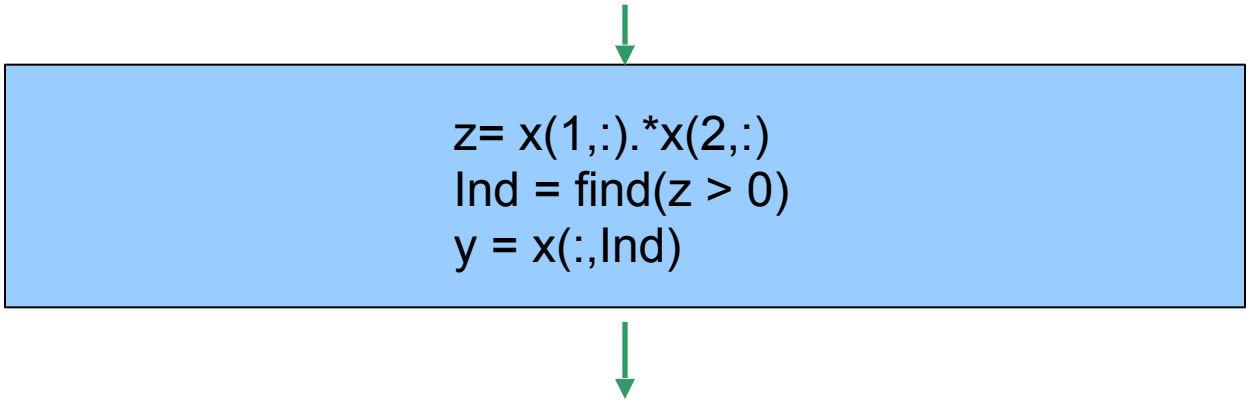
y = x(:,Ind)

0.8709 -0.8842 -0.7222
0.8338 -0.2943 -0.5945

Find points that belong the I, III quadrants

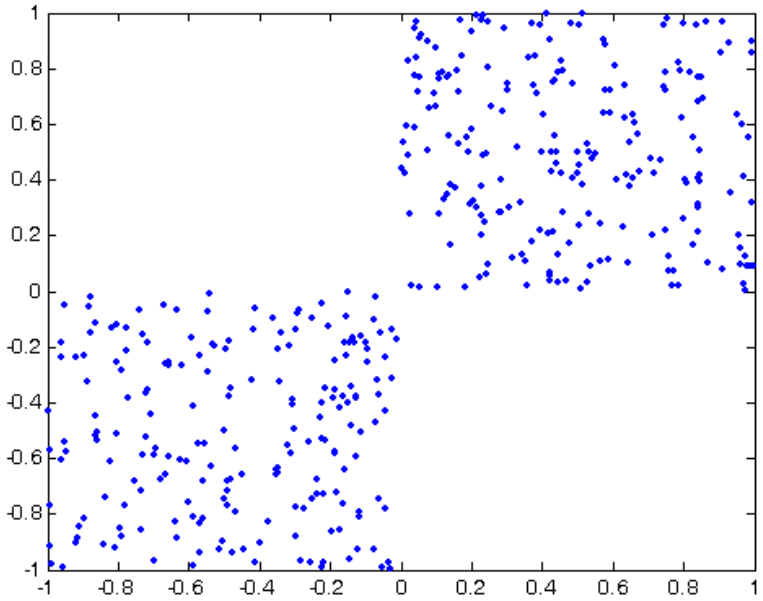
X =

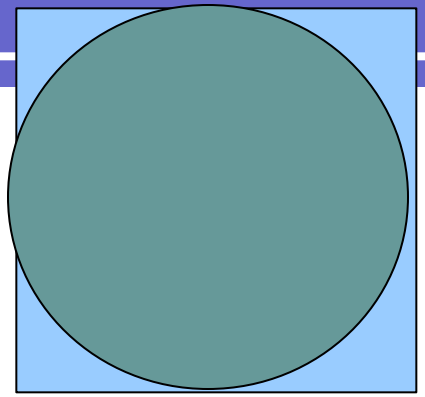
0.8709	-0.1795	-0.8842	0.6263	-0.7222
0.8338	0.7873	-0.2943	-0.9803	-0.5945



0.8709	-0.8842	-0.7222
0.8338	-0.2943	-0.5945

```
X=rand(2,800)*2-1;
Z=X(1,:).*X(2,:);
ind=find(Z>0);
Y=X(:,ind);
plot(Y(1,:),Y(2,:),'.');
```



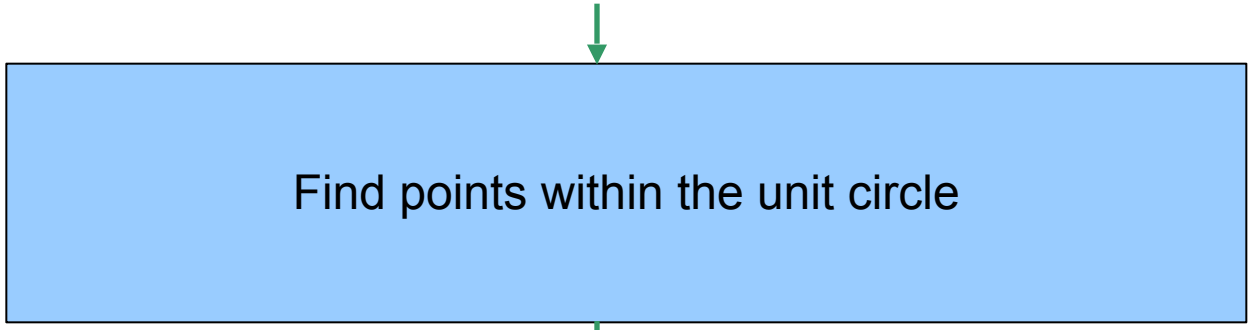


- Find points within the unit circle

find

X =

0.8709 -0.1795 -0.8842 0.6263 -0.7222
0.8338 0.7873 -0.2943 -0.9803 -0.5945



ans =

-0.1795 -0.8842 -0.7222
0.7873 -0.2943 -0.5945

X =

0.8709 -0.1795 -0.8842 0.6263 -0.7222
0.8338 0.7873 -0.2943 -0.9803 -0.5945

```
RD=(X(1,:).^2+X(2,:).^2);
```

RD =

1.4537 0.6521 0.8684 1.3532 0.8750

```
Ind = find(RD < 1)
```

ind =

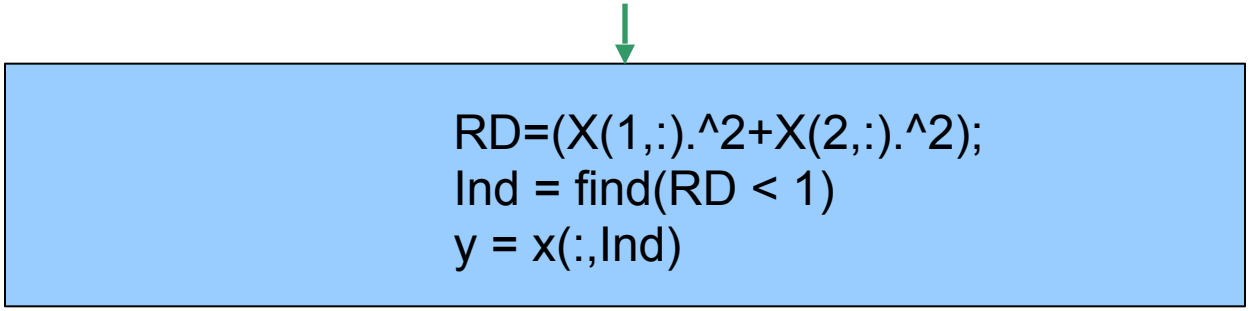
2 3 5

```
y = x(:,Ind)
```

-0.1795 -0.8842 -0.7222
0.7873 -0.2943 -0.5945

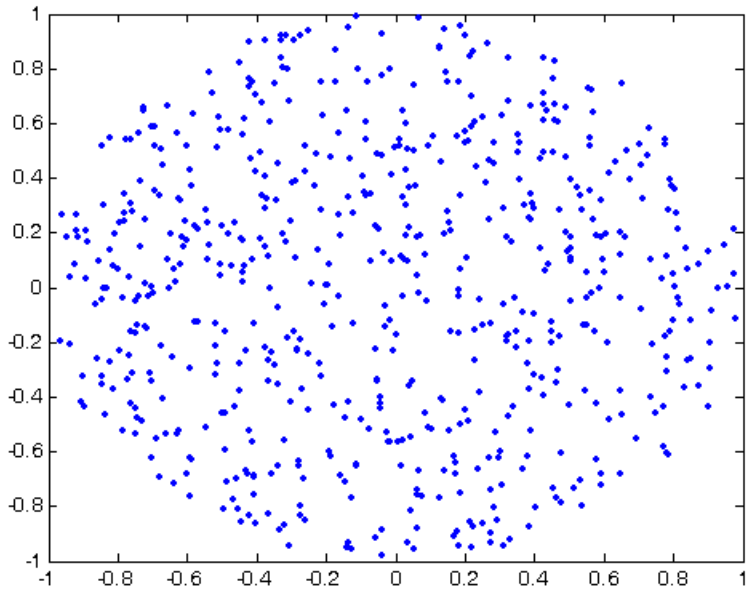
X =

0.8709 -0.1795 -0.8842 0.6263 -0.7222
0.8338 0.7873 -0.2943 -0.9803 -0.5945



-0.1795 -0.8842 -0.7222
0.7873 -0.2943 -0.5945

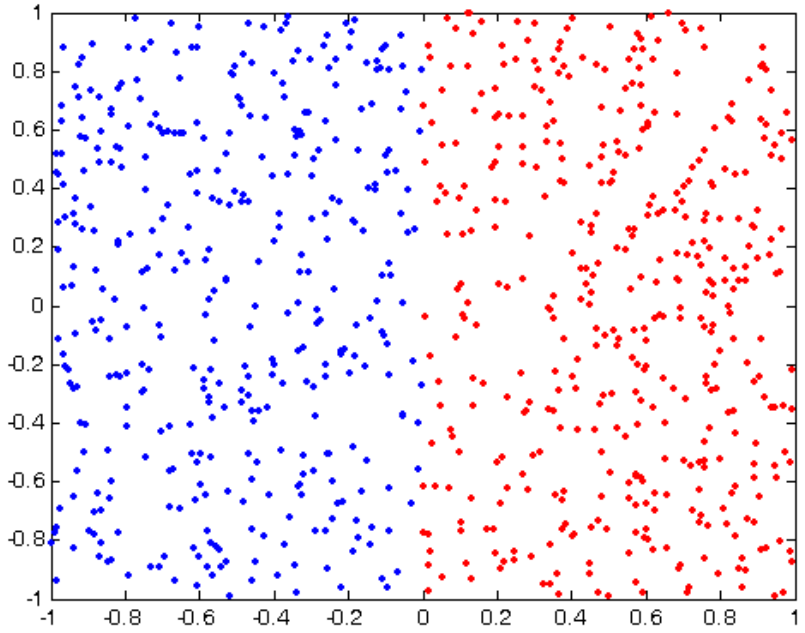
```
X=rand(2,800)*2-1;
D=sqrt(sum(X.^2));
ind=find(D<=1);
Y=X(:,ind);
plot(Y(1,:),Y(2,:),'.');
```

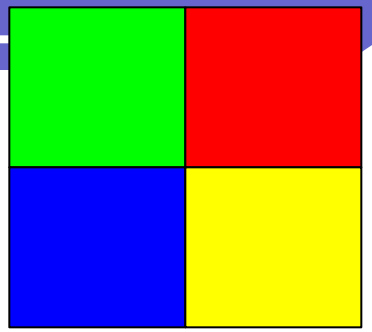


Demo_find3

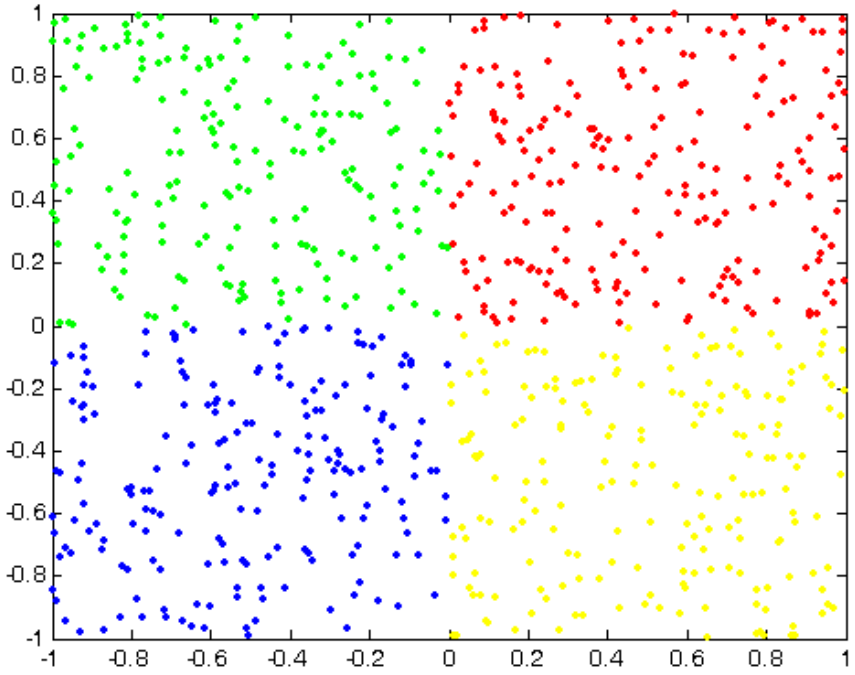
- Generate a uniform sample from [-1 1]x[-1 1]
- Plot those within the 1st and 4th quadrants in red points and remains in blue points

```
X=rand(2,800)*2-1;
ind=find(X(1,:)>0); XP=X(:,ind);
ind=find(X(1,:)<=0); XN=X(:,ind);
plot(XP(1,:),XP(2,:),'r.');hold on;
plot(XN(1,:),XN(2,:),'b.');
```





- Generate a uniform sample from [-1 1]x[-1 1]
- Plot points
  of different quadrants in different colors


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Dot product of two vectors

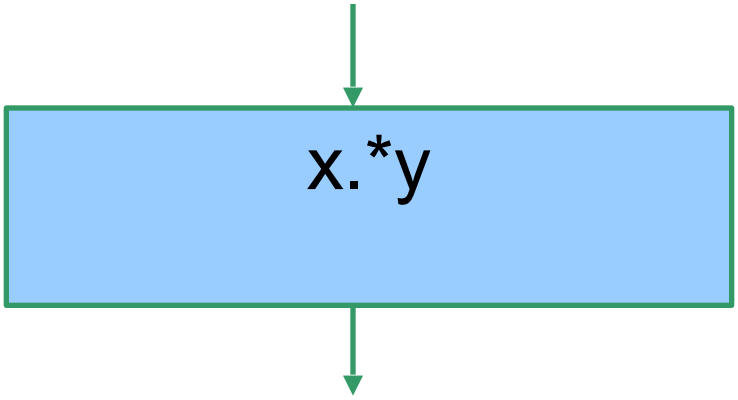
$$x = [x_1, x_2, \cdots, x_n]$$

$$y = [y_1, y_2, \cdots, y_n]$$

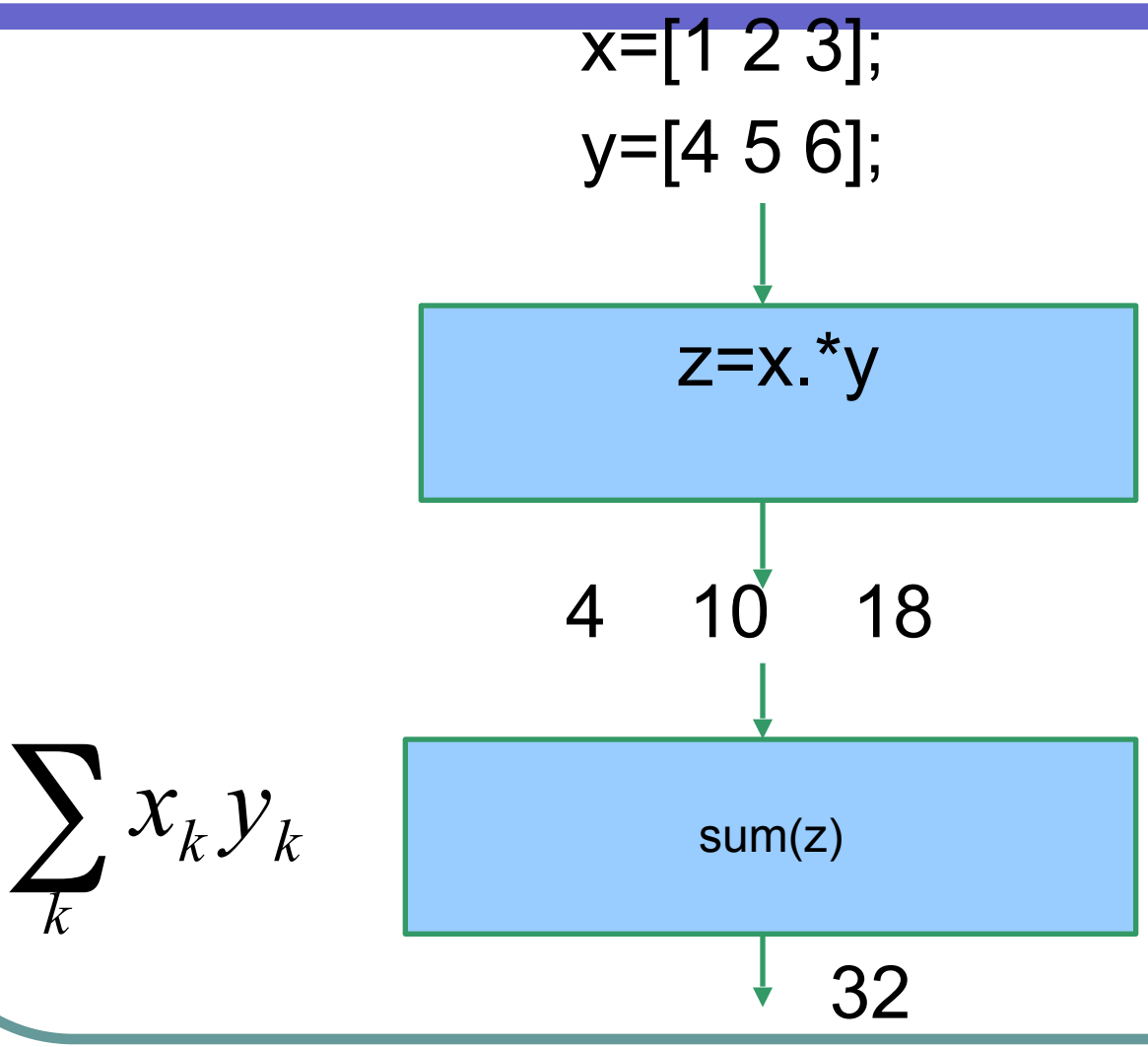
$$x.*y$$

```
x=[1 2 3];y=[4 5 6];
>> x.*y

ans =

     4    10    18
```

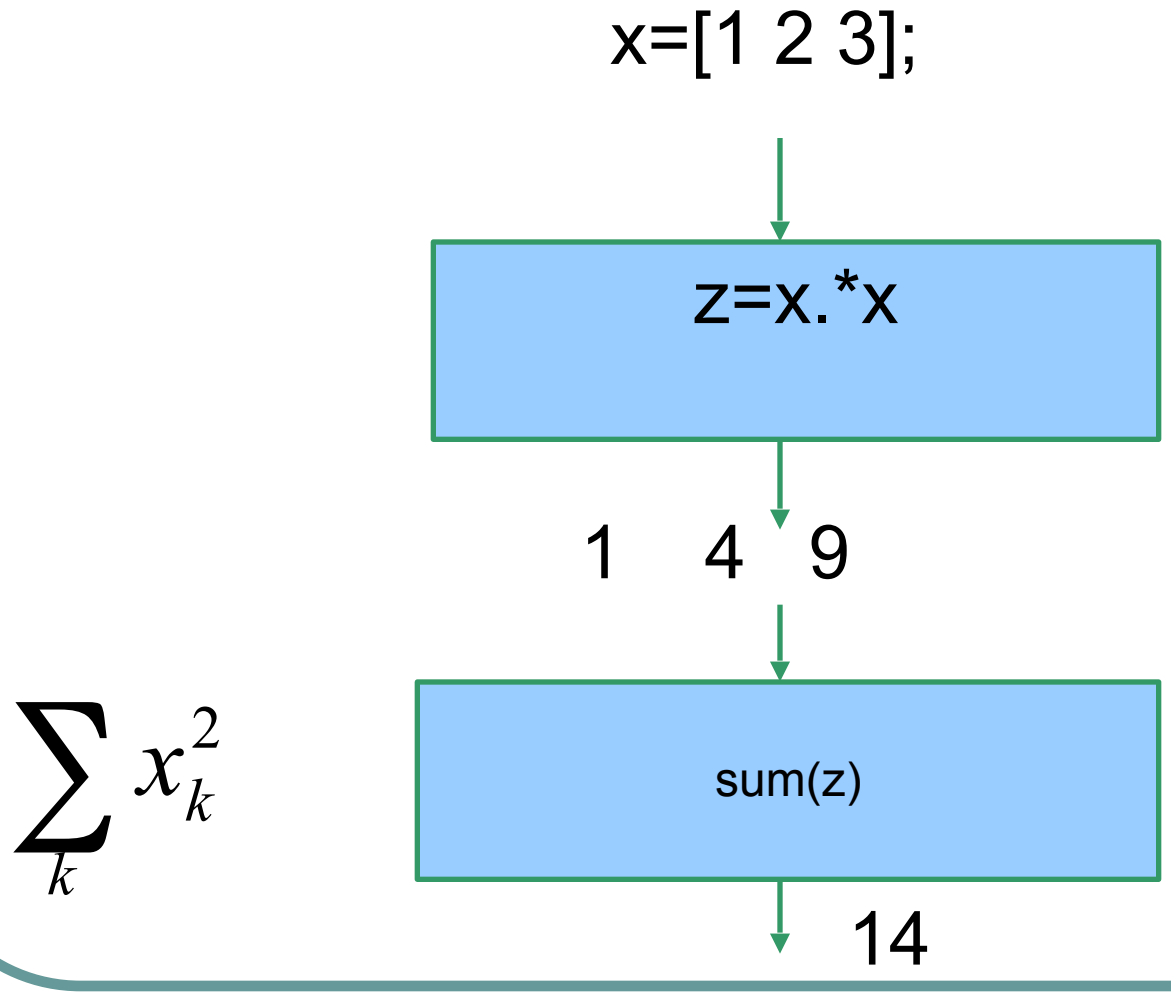
```
x=[1 2 3];
y=[4 5 6];
```




```
4    10    18
```



$$\sum_k x_k y_k$$

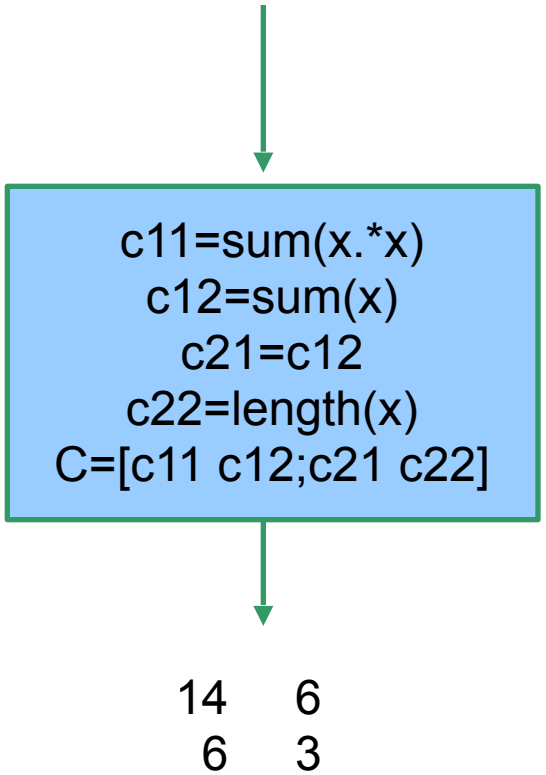


$$\sum_k x_k^2$$

Matrix form

$$C = \begin{bmatrix} \sum_k x_k^2 & \sum_k x_k \\ \sum_k x_k & \mathrm{N} \end{bmatrix}$$

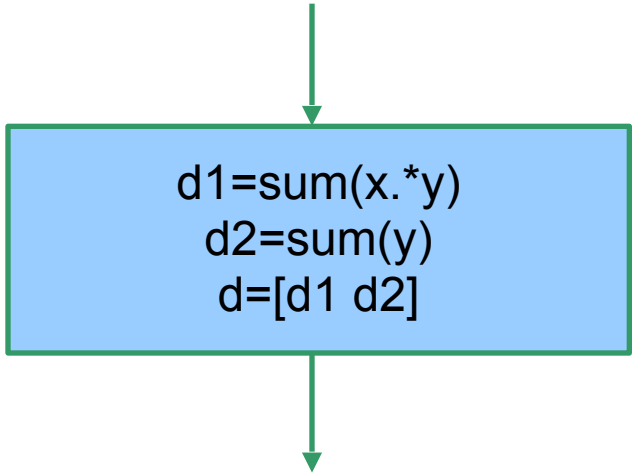
x=[1 2 3];



Form matrix

$$d = \begin{bmatrix} \sum_k x_k y_k \\ \sum_k y_k \end{bmatrix}$$

```
x=[1 2 3];
y=[4 5 6];
```




```
32    15
```

Dot product of two matrices

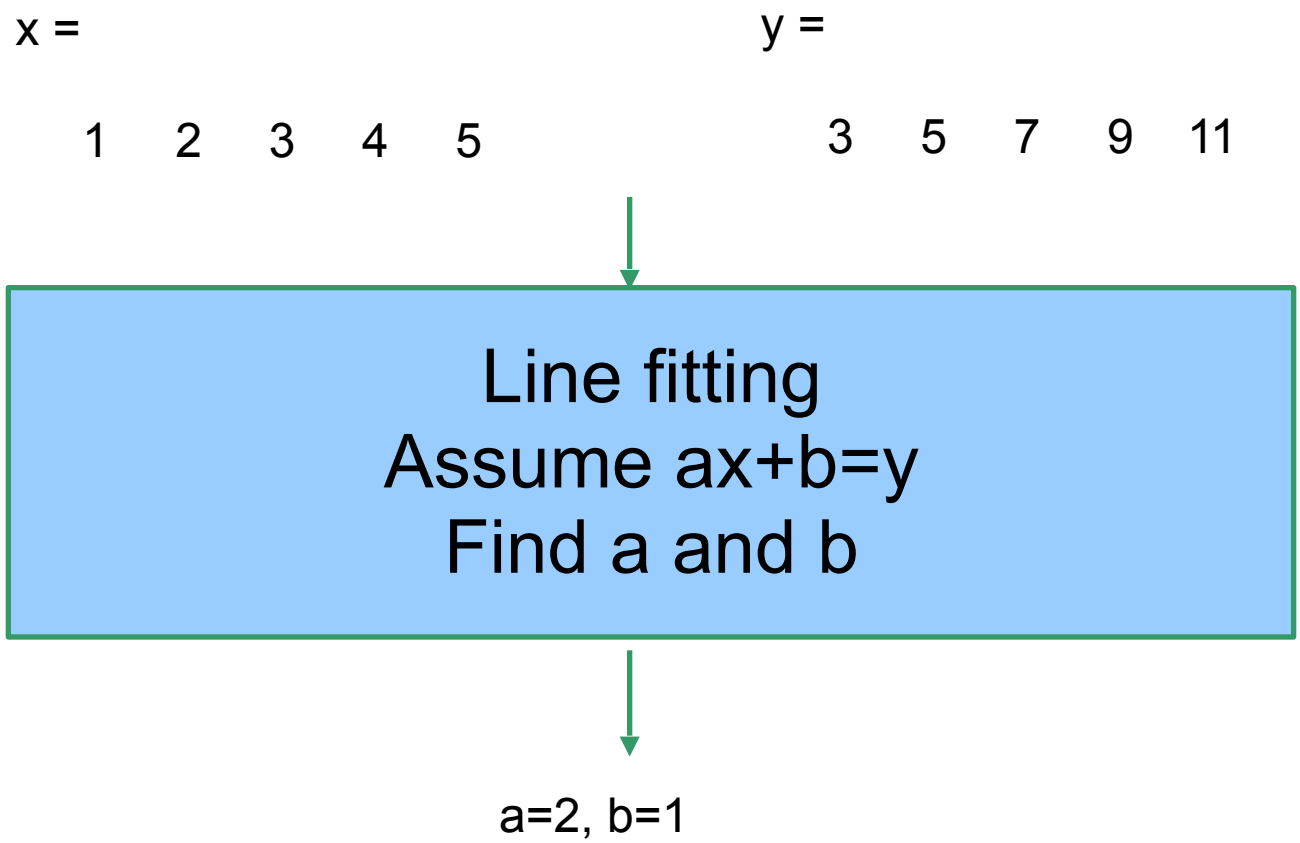
$$A = \lfloor a_{ij} \rfloor$$

$$B = [b_{ij}]$$

$$A.*B$$

```
>> A=reshape(1:9,3,3)

A =

     1     4     7
     2     5     8
     3     6     9

>> A.*A

ans =

     1    16    49
     4    25    64
     9    36    81
```

Line fitting



Errors

$$\mathrm{E}(a,b) = \sum_i (ax_i + b - y_i)^2$$

$$\mathrm{E} \geq 0$$

$$E(a = 2, b = 1) = 0$$

Find a and b by minimizing E

Derivation of normal equations

$$\frac{\partial E}{\partial a} = 0$$

$$\frac{\partial E}{\partial b} = 0$$

Normal equations for line fitting

$$\frac{\partial E}{\partial a} = -2\sum_i (y_i - (ax_i + b))x_i = 0$$

$$\sum_i x_i^2 a + \sum_i x_i b = \sum_i x_i y_i$$

$$\frac{\partial E}{\partial b} = -\sum_i (y_i - (ax_i + b)) = 0$$

$$\sum_i x_i a + Nb = \sum_i y_i = 0$$

Matrix form

$$C\begin{bmatrix} a \\ b \end{bmatrix} = d$$

$$C = \begin{bmatrix} \sum_k x_k^2 & \sum_k x_k \\ \sum_k x_k & \mathrm{N} \end{bmatrix}$$

$$d = \begin{bmatrix} \sum_k x_k y_k \\ \sum_k y_k \end{bmatrix}$$

Solve

$$\begin{bmatrix} a \\ b \end{bmatrix} = C^{-1} d$$

Matrix C

- C(1,1)=sum(x.^2)
- C(2,1)=C(1,2);
- C(2,2)=length(x)

$$C = \begin{bmatrix} \sum_k x_k^2 & \sum_k x_k \\ \sum_k x_k & \mathrm{N} \end{bmatrix}$$

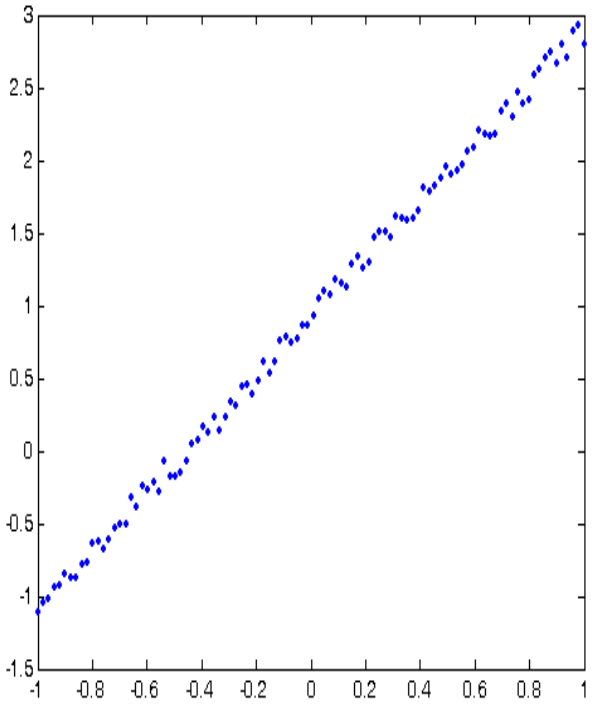
Vector d

- d=[sum(x.*y) sum(y)]
- inv(C)*d'

$$d = \begin{bmatrix} \sum_k x_k y_k \\ \sum_k y_k \end{bmatrix}$$

Noisy sample from a line

```
>> x=linspace(-1,1,100);
>> y=2*x+1+rand(1,100)*0.2-.01;
>> plot(x,y,'.');
```

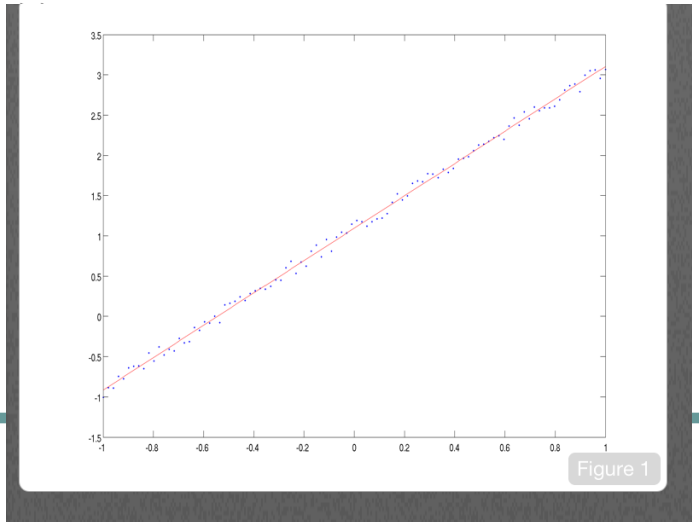


中華電信 3G

MATLAB Mobile

```
>> x=linspace(-1,1,100);
y=2*x+1+rand(1,100)*0.2-.01;
 plot(x,y,'.');
>> C(1,1)=sum(x.^2)
C(1,2)=sum(x);
C(2,1)=C(1,2);
C(2,2)=length(x)

C =

   34.0067


C =

   34.0067    0.0000
    0.0000  100.0000

>> d=[ sum(x.*y) sum(y)]

d =

   68.3086  109.3843
```

```
>> inv(C)*d'

ans =

    2.0087
    1.0938

>> ans=inv(C)*d'

ans =

    2.0087
    1.0938

>> a=ans(1);b=ans(2);
>> yhat=a*x+b;
>> hold on;plot(x,yhat,'r')
```

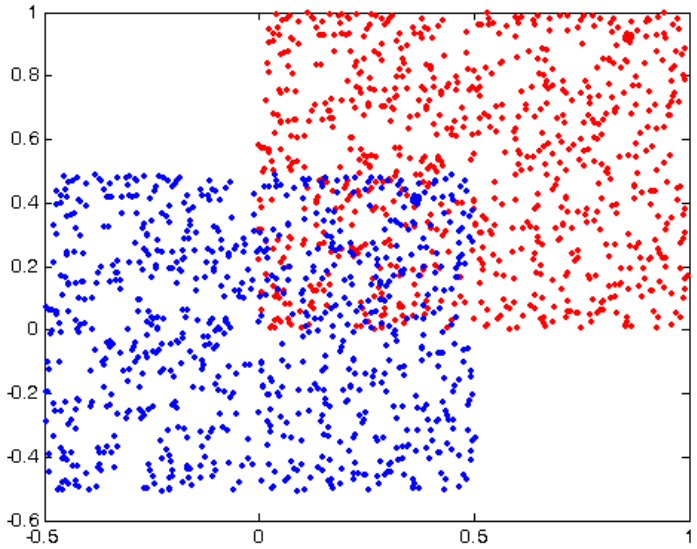


De-mean

- Create random points
- Shift them to have zero mean

demean

```
x=rand(2,800);
plot(x(1,:),x(2,:),’r.’); hold on;
mean_x=mean(x');
x=x-mean_x'*ones(1,800);
plot(x(1,:),x(2,:),’b.’);
```



Evaluation of vector inputs

$$f(x) = x - 1 \quad \text{if } x > 3$$

$$= 5 - x \quad \text{if } 3 \geq x > 2$$

$$= \frac{3x}{2} \quad \text{if } 2 \geq x$$

pw_linear

```
function y=pw_linear(x)
    y=x;
    ind=find(x>3);
    y(ind)=x(ind)-1;
    ind=find(x<=3 & x>2);
    y(ind)=5-x(ind);
    ind=find(x<=2);
    y(ind)=3*x(ind)/2;
return
```

```
>> x=linspace(0,4);
>> y=pw_linear(x);
>> plot(x,y)
```