MATLAB INTRODUCTION

CONTENT

01 Vector & Matrix Operations

02 Loop

03 Algorithm

04 Input & Output

05 Graph

01

Vector & Matrix Operations

Matrix and Vector (as a special case of matrix)

◆ An n-by-m matrix is a 2D array.

◆ In Matlab, several different ways to represent the same matrix

```
A=[1,2,3      A=[1 2 3        A=[1,2,3;4,5,6]
      4,5,6]        4 5 6]    A=[1 2 3;4 5 6]
```

◆ Column vector: b=[1;2;3]   b=[1 2 3]’   b=[1

```
                                             2
                                             3]
```

◆ The operator ' means “transpose” for real matrices.

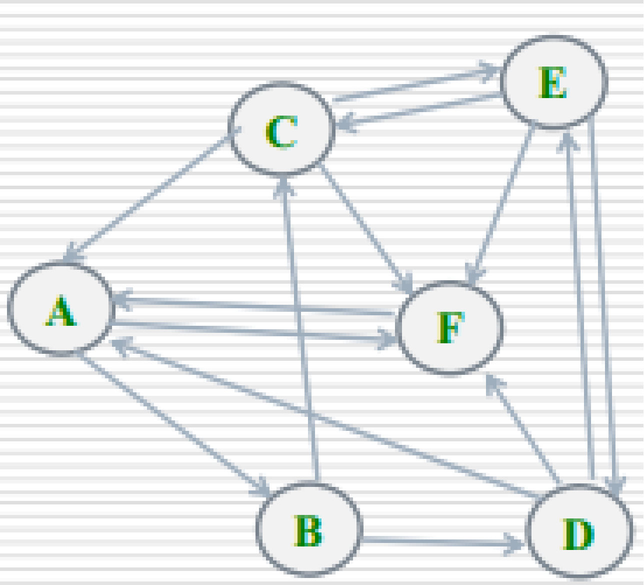
For complex matrices, one has to use ’.

Matrix Indexing

	A	B	C	D	E	F
A	0	0	0.33	0.33	0	1
B	0.5	0	0	0	0	0
C	0	0.5	0	0	0.33	0
D	0	0.5	0	0	0.33	0
E	0	0	0.33	0.33	0	0
F	0.5	0	0.33	0.33	0.33	0

- M(i,j)--element of row i, column j
  Eg. M(3,5)=0.33
- M(i, :)-- all elements in row i
  Eg. M(5,:)=[0, 0, 0.33, 0.33, 0, 0];
- M(:, j)-- all elements in column j

M(:,2)=[0
0
0.5
0.5
0
0]



Default Column by Column Operations

example

```
A=[3.2 3.5 3.25 4.0 3.0
2.0 3.1 3.8 3.7 3.75
3.83 2.7 2.8 2.6 3.65]
```

```
3.8300 3.5000 3.8000 4.0000 3.7500
2.0000 2.7000 2.8000 2.6000 3.0000
```

- Usually most operations for vectors can be applied to matrices, in a column-by-column way by default.

Example

max(A) will give

min(A) will give

This convention also applies to many other ops.

mean, median, mode, var, std, sum, prod…

- One may specify the dimension along which to operate.

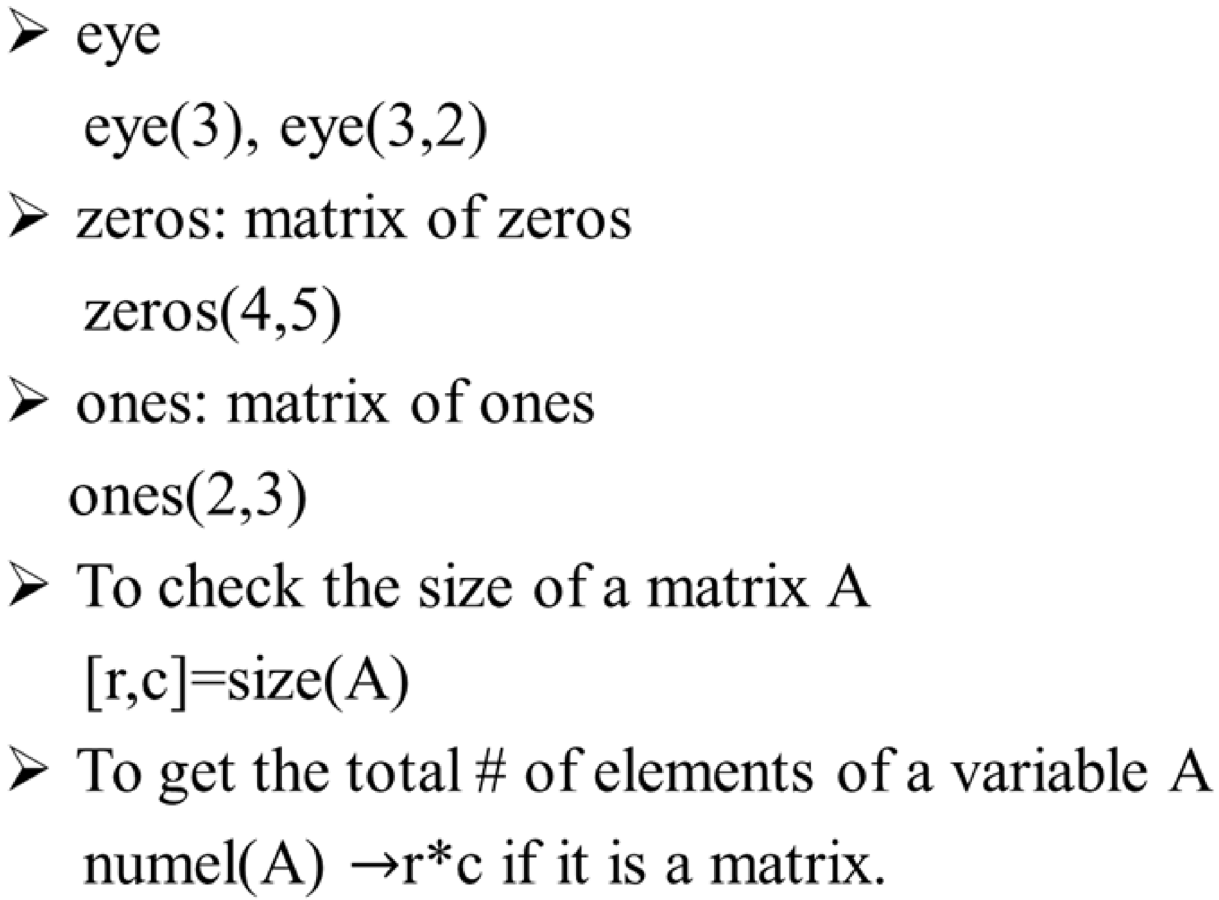
E.g. mean(A,2) give row means.

- Note: max(A,2) gives the max of each entry of A and 2

Q: How do we, say, get the max entry of A?

Answer: max(max(A))

Matrix Construction



- eye

  eye(3), eye(3,2)

- zeros: matrix of zeros

  zeros(4,5)

- ones: matrix of ones

  ones(2,3)

- To check the size of a matrix A

  [r,c]=size(A)

- To get the total # of elements of a variable A

  numel(A) →r*c if it is a matrix.

Vector Construction

- Row vector from m to n

  m:n

- Try 1:0.5:5.1? 6:-2:1?

  something like 1:-0.5:6?

  Empty Matrix

- linspace-- 100 point by default

  x=linspace(0, 2*pi); plot(x,sin(x));

  y=linspace(0,2*pi,10); plot(y, sin(y));

- To check the length of a vector

  length(b)

  length for matrix: the largest dimension

```
b=[60
38
195
165
450
1550]
```

Element-wise operators (对应元素间运算)

- Matrix/array addition and subtraction

  A=[1,2,3, 4,5,6], B=[1,1,1, 2,2,2];

  A-B=? A+B=?

- A*B?

  Error msg

- In general, when we need to do the multiplication, division, and multiplication elementwise,

  we apply .*, ./, .^

02

Loop

Cycle and Condition Judgment

```
for var=some-range %控制循环变量
end
body of the loop %可以多条命令
End
```

- Nested loops（嵌套循环）

Suppose A=[1,3,5;
                2, 4, 6]; [r,c]=size(A);

```
for i=1:r
   for j=1:c
       disp(A(i,j));
    end
 end
```

VS

```
for j=1:c
    for i=1:r
        disp(A(i,j));
     end
end
```

o In principle, you could have any number of nested loops.

while Cycle and Condition Judgment

- Logical values and relational expressions

  Every variable has a type associated with it

  The most common one is ‘double’, which stands for

- double precision

  A very useful type is ‘logical’: represents a logical true or false state

  using the numbers 1 and 0. You could have arrays of logical values.

- For example: A=true; B=false; C=1:5;

  A test for you

Relational Operations

- Relational operators: relate two expressions

  >, >=, <, <=

  == is for equality, and ~= is for inequality

  The type of the result is logical, not double.

- e.g. of Relational operators

  The expression ‘3<1’ gives a 0 (false)

- Element-wise (对应元素间的运算)

  If A=[1,2,3]; B=[1,2, 4]; and C=(A==B) will have

  C=[1,1,0].

Element-wise logical Operators: AND, OR, NOT(对应元素间运算)

- &: ‘AND’. The value of the expression is true (1) if both are true; is false (0) otherwise.
- | : ‘OR’. The value of the expression is true (1) if one or both of the two operands is true.
- ~: ‘NOT’. Applies to single operand; the opposite of the operand.
- Eg., suppose GPA=[3.5, 2.7, 1.5] for 3 students

  ‘GPA>3’: a relational expression, gives [1, 0, 0] in logical type (not double).

  (GPA<2) | (GPA<2) : gives [1, 0, 1] in logical type.

  ~(GPA<3): gives [1, 0, 0]

Column-wise logical functions: all, any（逐列运算）

- If A is a logical vector, say, A=[true,false];
- any(A) returns 1 if any of them is true; all(A) returns 1 only if all are true.
- If A is some numerical vector. any(A) is 1 if there is any non-zero elements; all(A) is 1 if all are non-zero elements.
- If A is an m-by-n matrix, all and any are column-wise

  Eg. A=[1, 2, 0; 0, -3, 0];

  all(A) returns 0 1 0, and any(A) returns 1 1 0

Conditional Control: ‘if’

- ‘if’ is used to select at run-time which block of code to execute.

  Example:

  ```
  if logical_expression
      block A of code
  end
  ```

- ‘else’ and ‘elseif’ statements further conditionalize the ‘if’ statement.

Loop：for vs while

- for loop:

```
for iter_var = range
    some steps
end
```

- while loop:

```
while(condition judgements)
    some steps
end
```

- 可以嵌套(多层循环)，也可以break或者continue

  break: 跳出所在的循环层

  continue：跳过本次循环剩下的语句，执行本层的下一次循环.

Example: matrices multiple

```
[m,n]=size(A);
[r,k]=size(B);
if(n~=r)
    disp('Error! Dimensions of A and B don't match!') %显示出错信息
else
C=zeros(m,k);
for ii=1:m
    for jj=1:k
        for t=1:n
            if( (A(ii,t)==0)|| ~B(t,jj) ) %演示各种逻辑判断使用
                continue; %跳到t的下一值继续循环
            else
                C(ii, jj)=C(ii, jj)+A(ii, t)*B(t, jj);
            end
        end
    end
end
end
```

Input variables

Local Variables(局部变量): 变量M, d, R_err 是PageRank内部的局部变量, 在当中的值不影响在函数外的同名变量的值。 若在函数外没有定义的变量，则执行完函数后外面也没有这些变量。

```
>>[R,iter]=PageRank(M,0.85,1e-3);%在函数中d为0.85
>>d
??? Undefined function or variable ’d’.
```

但执行完函数之后，外面并不会因此多了变量d的值

Only the order of the arguments matters, not the names of the arguments:

```
>>d=0.85;R_err=1e-3;
>>[R,iter]=PageRank(M,d,R_err)
>>[R,iter]=PageRank(M,R_err,d).
```

```
function [R, iter] = PageRank(M, d, R_err)
```

Local Variables(局部变量) 总结

- 在函数定义行中指定的输入变量名是在该函数内有效的局部变量。所以，在调用函数的时候，输入变量的名字并不重要，用户可以将其他名字的变量作为输入传送给函数。在调用函数时重要的是输入变量的顺序，而不是名称。

- 函数内部的局部变量在函数调用结束的时候就消失了，除非是在输出变量列表中的变量(且调用时已经指定将赋值给某些变量)。

- 例如PageRank中的last_R在函数调用结束后不会出现在工作空间。(可用空间、面具/角色来比喻).

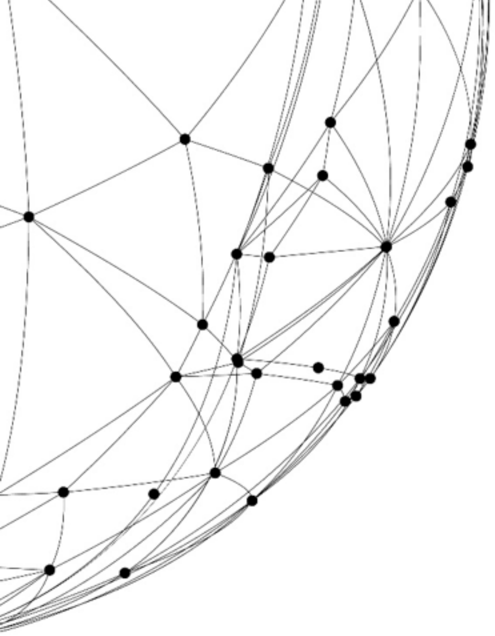
Output variables 输出变量

- 函数可以有多于一个输出变量， 且它们在函数定义行中被置于方括弧内. 当只有一个输出变量时，方括弧可以省略。
- E.g., 以下函数 circle计算一个给定半径为r的圆的面积A和周长 C， 半径r就是输入变量。

```
function [A, C] = circle(r)
    A = pi*r.^2;
    C = 2*pi*r;
```

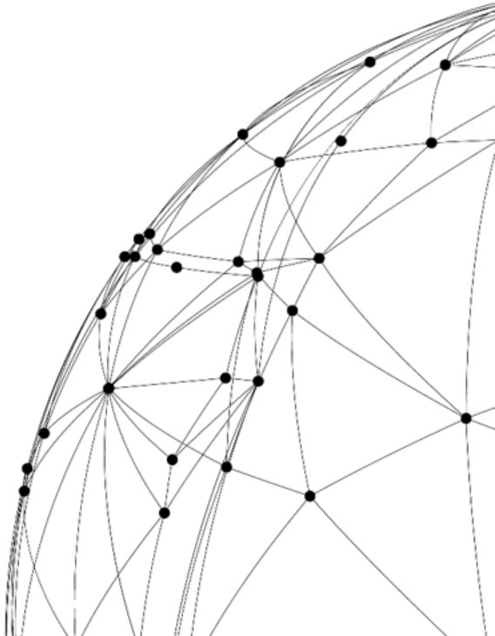
假设半径为4， 则该函数的调用如下：

```
>>[A, C] = circle(4)
A =
50.2655
C =
25.1327
```



03

Algorithm



定义函数式M文件

- function [返回变量列表]=函数名(输入变量列表)

  %注释说明语句段

  函数体

➢ 第一行function关键字表示即将定义一个函数

a. function后定义函数名和输入输出变量， 即输入变量与返回变量；函数被调用时按此格式执行；

b. 注意: 输出变量在方括号[ ]中, 输入变量在 ( )中.

➢ 文件名必须与函数名一一对应

➢ 函数执行完后， 只返回结果， 不保留中间过程；

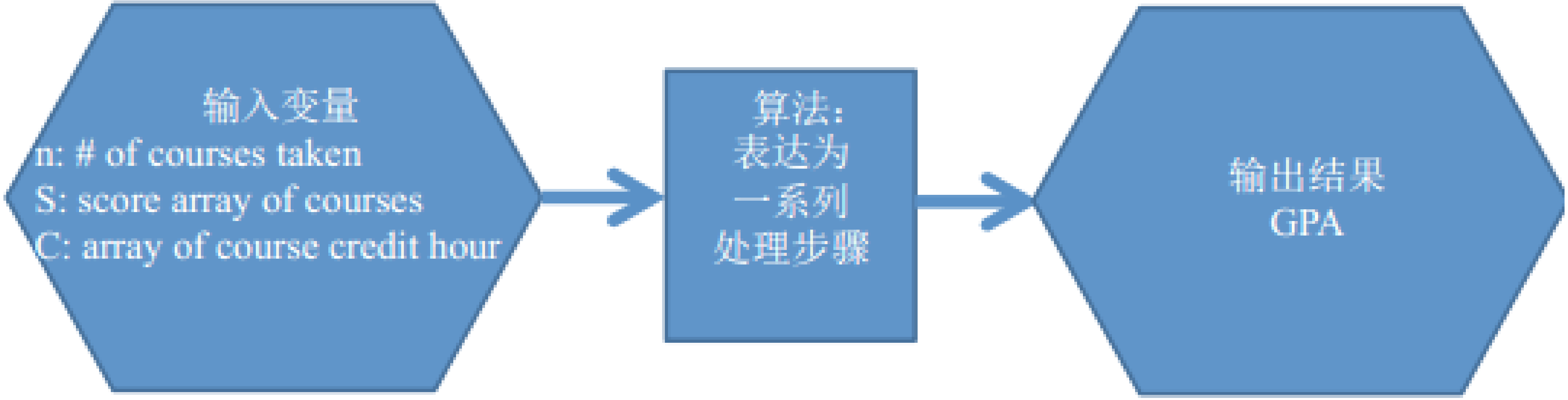
➢ 例如 function [R, iter] = PageRank(M, d, R_err)

函数名为PageRank, 所以文件名应该是PageRank.m

- 在调用函数时重要的是输入变量的顺序， 而不是名称

What is Algorithm

- A set of well-defined computational steps that transform input into output to solve a certain problem.
- Example: compute the GPA of a student, given his/her score of each course and the course credit hour.



输入变量
n: # of courses taken
S: score array of courses
C: array of course credit hour

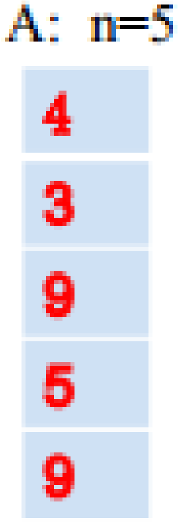
算法：
表达为
一系列
处理步骤

输出结果
GPA

Searching for the max (or the min)

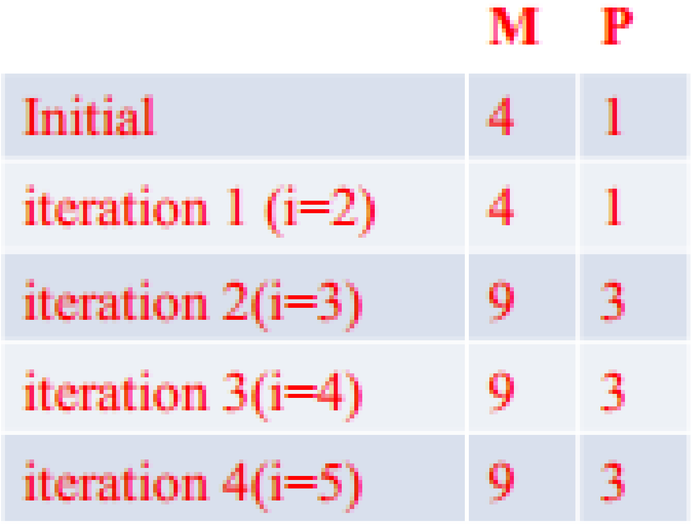
- Given any (numeric) array and its length n
- Write a function to get the maximum of it. MyMax.m
- Input: Array A and length n.
- Output: the max value and its position in A

```
function [M,P]=MyMax(A,n)
M=A(1); P=1;
for i=2:n
if A(i)>M
M=A(i); P=i;
end
end
```



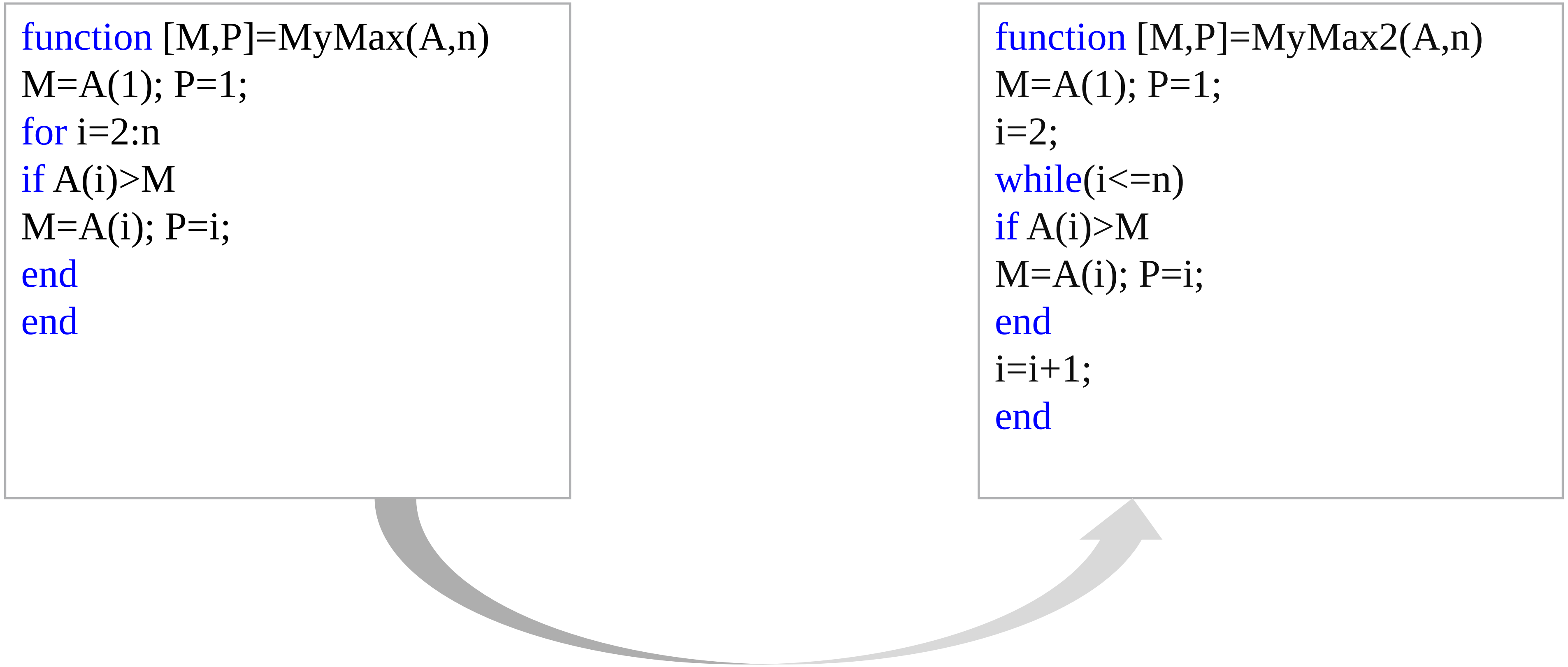
A: n=5

4
3
9
5
9



	M	P
Initial	4	1
iteration 1 (i=2)	4	1
iteration 2(i=3)	9	3
iteration 3(i=4)	9	3
iteration 4(i=5)	9	3

'for' loop vs. 'while' loop

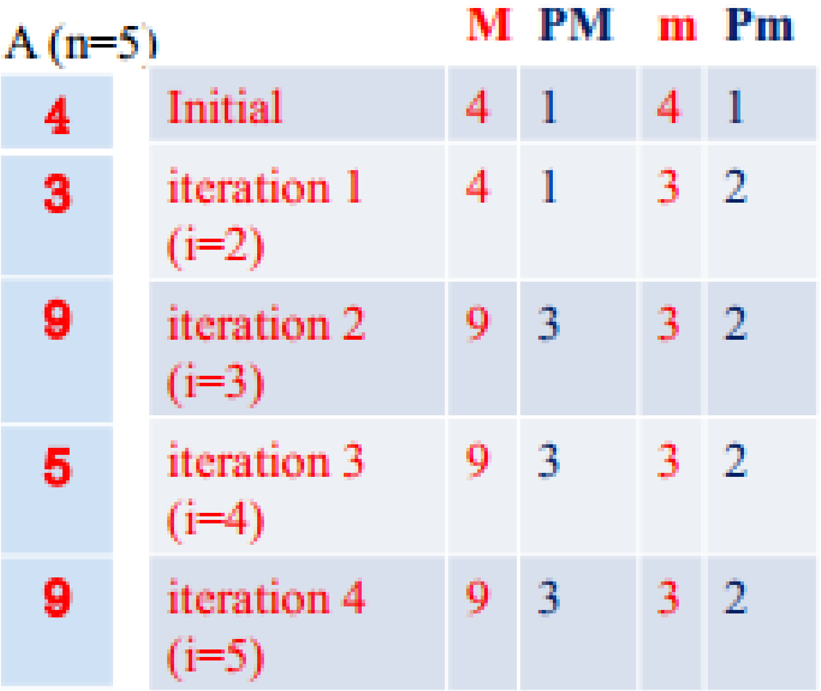


```
function [M,P]=MyMax(A,n)
M=A(1); P=1;
for i=2:n
if A(i)>M
M=A(i); P=i;
end
end
```

```
function [M,P]=MyMax2(A,n)
M=A(1); P=1;
i=2;
while(i<=n)
if A(i)>M
M=A(i); P=i;
end
i=i+1;
end
```

A function of searching

```
function [M, m,PM,Pm]=Max_min(A,n)
%M and m are max and min
M=A(1); PM=1; m=A(1); Pm=1;
for i=2:n
     if (A(i)>M)
         M=A(i); PM=i;
     elseif (A(i)<m)
         m=A(i); Pm=i;
     end
end
```



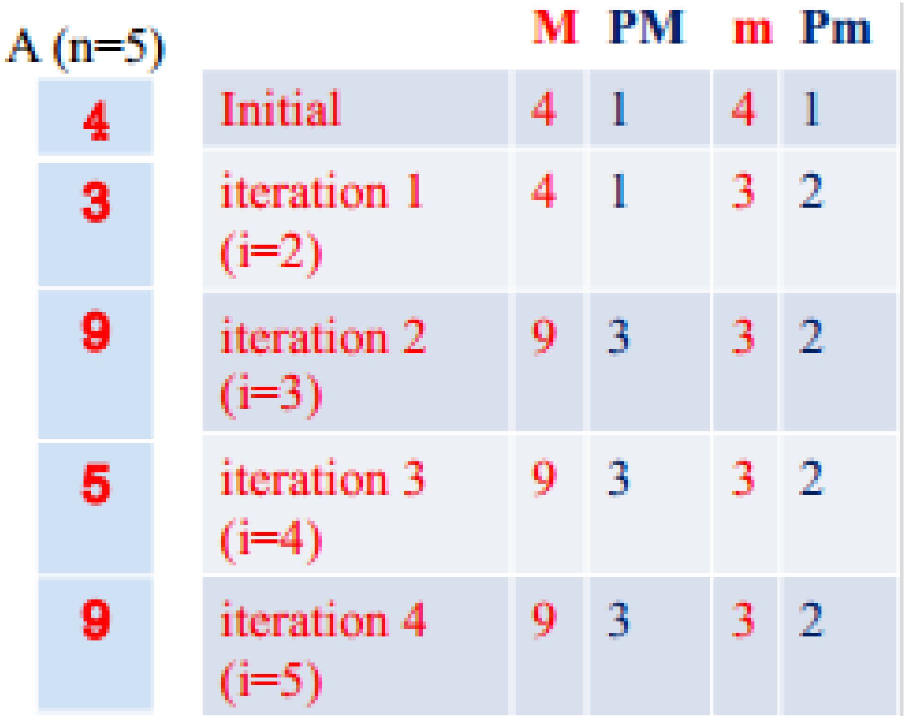
A (n=5)		M	PM	m	Pm
4	Initial	4	1	4	1
3	iteration 1 (i=2)	4	1	3	2
9	iteration 2 (i=3)	9	3	3	2
5	iteration 3 (i=4)	9	3	3	2
9	iteration 4 (i=5)	9	3	3	2

- Why put it this way? Is there any A(i) that doesn’t pass the first if-test but could pass the elseif-test?

‘Loop Invariant’ in Algorithm (Optional)

- Initialization. A property is true prior to the first iteration code.(Right after i=2 here)
- Maintenance. If it is true before an iteration, it remains so before next iteration.
- Termination. When the loop terminates, the ‘invariant’ gives a useful property to show the algorithm’s correctness.

```
function [M, m, PM,Pm]=Max_min(A,n)
%M and m are max and min
%PM and Pm are their positions in A
M=A(1); PM=1; m=A(1); Pm=1;
for i=2:n
     if (A(i)>M)
         M=A(i); PM=i;
     elseif (A(i)<m)
         m=A(i); Pm=i;
      end
end
```



A (n=5)		M	PM	m	Pm
4	Initial	4	1	4	1
3	iteration 1 (i=2)	4	1	3	2
9	iteration 2 (i=3)	9	3	3	2
5	iteration 3 (i=4)	9	3	3	2
9	iteration 4 (i=5)	9	3	3	2

The Sorting Problem

- Input: n-the length of a sequence, and a sequence of numbers {a1, a2, … , an}.
- Output: A reordering {a1’, a2’, … , an’} of the input sequence such that a1’< a2’< … < an’
- Any algorithm you can design for this?

➢ If you have learned this before, shut up and just let others to think about it.

➢ We have a function 'MyMax' last time

➢ Is MyMax function useful in this job?

```
function [M,P]=MyMax(A,n)
M=A(1); P=1;
    for i=2:n
        if A(i)>M
           M=A(i); P=i;
    end
end
```

A sorting algorithm using our MyMax

Description of Algorithm Sort.1:

```
[M, P]=MyMax(A,n)
Exchange the value of A(n) and A(P)
for i=n-1 downto 1
[M, P]=MyMax(A(1:i), i)
Exchange the value of A(i) and A(P)
end the for-loop
```

The (modified) Algorithm Sort.1 in Matlab

```
for i=n:-1:1
   [M, P]=MyMax(A(1:i), i)
   temp=A(i); A(i)=A(P); A(P)=temp;
   %Here, could just: A(P)=A(i); A(i)=M;
end
```

Loop Invariant

- Initialization: Prior to the first iteration, A(i+1) stores the largest value in {A(1), A(2),… A(i+1)}.
- Maintenance: If it is true before an iteration, it remains true that the largest value in {A(1), A(2),… A(i+1)} is stored in A(i+1).
- Termination: The ith largest value is stored in A(i) for all i.

In general, to exchange the value of variable a and b: temp=a; a=b; b=temp; Called ‘swap a and b’

Running time of the Algorithm Sort.1

- For MyMax(A,n), the running time cost is about c1*n for some constant c1>0.
- For Sort.1, c2*n steps in assignments, and runs MyMax(A(1:i), i) for i=n down to 1, costing c1*(n+n-1+…+1) =c1*(n+1)*n/2
- Totally Sort.1 takes about c3*n2+c4*n running time cost.
- When n is large enough, c4*n <<c3*n2. Asymptotically, the running time is denoted Θ(n2)

The Algorithm Sort.1

```
function A=Sort1(A,n)
for i=n:-1:1
  [M, P]=MyMax(A(1:i), i)
  %exchange A(P) and A(i)
  A(P)=A(i);
  A(i)=M;
end
```

The function MyMax

```
function [M,P]=MyMax(A,n)
     M=A(1); P=1;
     for i=2:n
         if A(i)>M
            M=A(i); P=i;
     end
end
```

Asymptotic notation: Θ-notation (Big Theta)

- “f(n) is in Θ (g(n))”: g(n) is an asymptotically tight bound for f(n)
- Θ(g(n)) = {f(n)| There exist positive constants c1， c2， so that 0 ≤ c1g(n) ≤ f(n) ≤ c2 g(n) for large enough n}
- “f(n)= Θ(g(n))” means “f(n) is in Θ(g(n))”
- E.g.: Show 3n2 + 6n = Θ (n2)
- f(n)= 3n2 + 6n, g(n)=n2. Let c1=3, c2=4. When n>=6, we have 3n2 <=3n2 + 6n<= 4n2. This proves that f(n) is in Θ(n2).
- Note, f(n) =Θ(g(n)) if and only if g(n) =Θ(f(n))

O-notation (Big O) and Ω-notation (Big Omega)

- “f(n) is in O (g(n))”: g(n) is an asymptotically upper bound for f(n)

➢ O(g(n)) = {f(n)| There is c1>0 that 0 ≤ f(n) ≤ c1 g(n) for large enough n}

- “f(n)= O(g(n))” means “f(n) is in O(g(n))”
- Obviously Θ (g(n)) ⊆ O(g(n)),

➢ Eg. 6n is in O(n)， since 6n is in Θ (n). 6n is also in O(n2), since f(n)=6n<=6n2. But n2 is not a tight bound for 6n

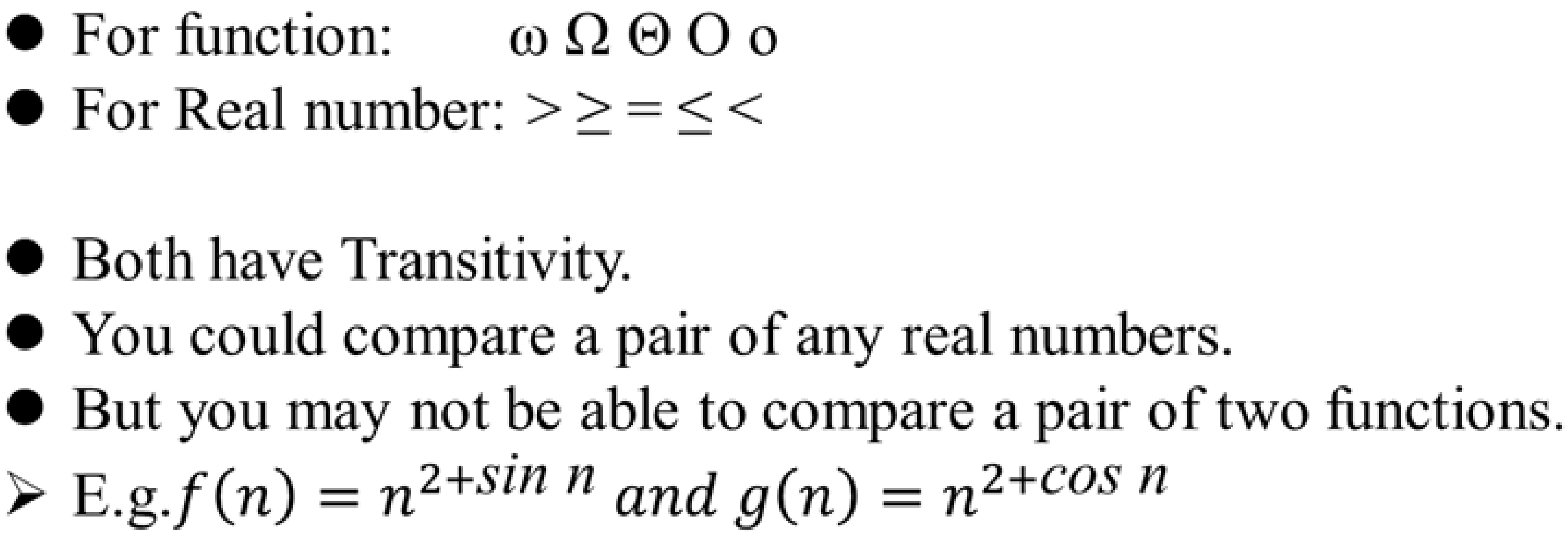
- “f(n) is in Ω (g(n))”: g(n) is an asymptotically lower bound for f(n)

➢ Ω(g(n)) = {f(n)| There is c1>0 that 0 ≤ c1 g(n) ≤ f(n) for large enough n}

o-notation (Little o) and ω -notation (little omega)

- “f(n) is in o (g(n))”: lim f(n)/g(n)=0 when n goes to infinity. (f(n)>=0 and g(n)>0).
- E.g., f(n)=n is in o(n1+p) for any given p>0.
- Similarly “f(n) is in ω (g(n))”: lim f(n)/g(n)=∞ when n goes to infinity. (f(n)>=0 and g(n)>0).

Analogy of Read Number Comparison



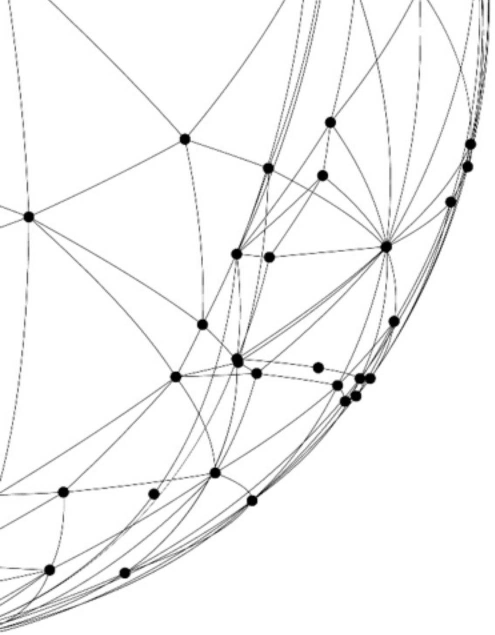
- For function: ω Ω Θ O o
- For Real number: > ≥ = ≤ <

- Both have Transitivity.
- You could compare a pair of any real numbers.
- But you may not be able to compare a pair of two functions.

➢ E.g.$f(n) = n^{2+sin\ n}\ and\ g(n) = n^{2+cos\ n}$

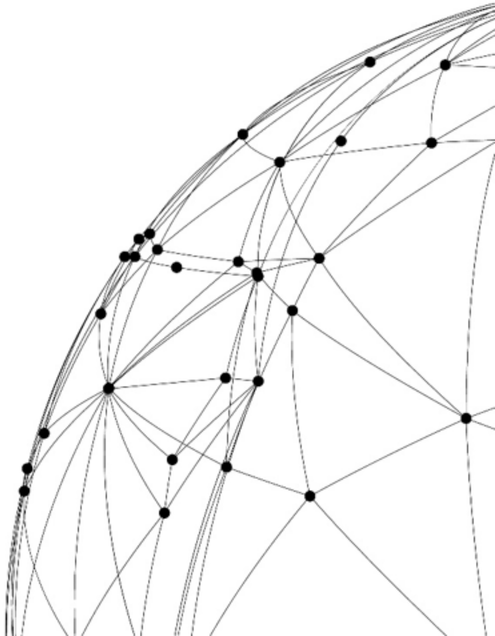
O-notation (Big O) and Ω-notation (Big Omega)

- “f(n) is in O (g(n))”: g(n) is an asymptotically upper bound for f(n)
- ➢ O(g(n)) = {f(n)| There is c1>0 that 0 ≤ f(n) ≤ c1 g(n) for large enough n}
- “f(n)= O(g(n))” means “f(n) is in O(g(n))”
- Obviously Θ (g(n)) ⊆ O(g(n)),
- ➢ Eg. 6n is in O(n)， since 6n is in Θ (n). 6n is also in O(n2), since f(n)=6n<=6n2. But n2 is not a tight bound for 6n
- “f(n) is in Ω (g(n))”: g(n) is an asymptotically lower bound for f(n)
- ➢ Ω(g(n)) = {f(n)| There is c1>0 that 0 ≤ c1 g(n) ≤ f(n) for large enough n}



04

Input & Output



MATLAB的输入输出概述

- 输入输出(I/O)

➢ 数据输入和输出（I/O）

输入： 从键盘、磁盘、网络文件获得数据，加载到工作空间（workspace）。

输出： 从工作空间变量输出到屏幕、文件中。

➢ 上层文件和底层文件I/O操作

上层： 针对不同的数据格式，提供不同的I/O函数，有现成的函数供使用，仅需少量编程。例如读写xls, csv等

底层： 使用文件标识符可访问任何类型的数据文件，更加灵活地完成相对特殊的任务，需要复杂编程

从键盘输入 input 函数（简略介绍）

- result= input(prompt)
- prompt是用户指定的一个提示字符串，该命令在屏幕上显示提示字符串。 等待来自键盘的输入，对输入中的任何表达式进行计算，然后返回结果。
- 例如,请自行尝试以下命令

```
Age=input('How old are you?')
Name=input('What is your name?','s')
class(Age)
class(Name)
```

- 猜测上面的's'代表什么？请查看帮助文档

输出到屏幕： disp 和fprintf

- disp(‘I am the instructor!’)
- disp(pi);
- format long
- format short
- fprintf: 格式化输出

一般用于输出到文件（底层文件操作），但也可以用于输出到屏幕

fprintf(FormatSpec, var1, var2, …,varN): FormatSpec是指定的格式，需要输出的var1…varN是变量名。例如

- fprintf('My Age=%d, and your age=%d', 25, 21);
- 以整数的形式输出25 和 21在对应的位置(若被输出的变量是浮点数， %d会使之先转换成整数),
- fprintf('\n') 换行

字符转换

```
>> fprintf('Pi=%.2E, and e=%.11f\n', pi, e)
Pi=3.14E+000, and e=2.71828182846
```

转换字符	意义
%c	以单个字符输出
%d	以有符号十进制整数输出.
%e	采用指数格式输出，采用小写字母 e，如： 3.1415e+00
%E	采用指数格式输出，采用大写字母 E，如： 3.1415E+00
%f	以定点数的格式输出, 例如fprintf(‘a=%.2f’, 1.2345) 输出1.23
%s	以字符串形式输出对应的变量

转义符： 转换字符用于指定输出的符号，可以选择的内容如表所示。

转义符	功能
\n	换行符
\t	tab
\\	\，反斜线
\" 或 " （两个单引号）	'，单引号
%%	%，百分号

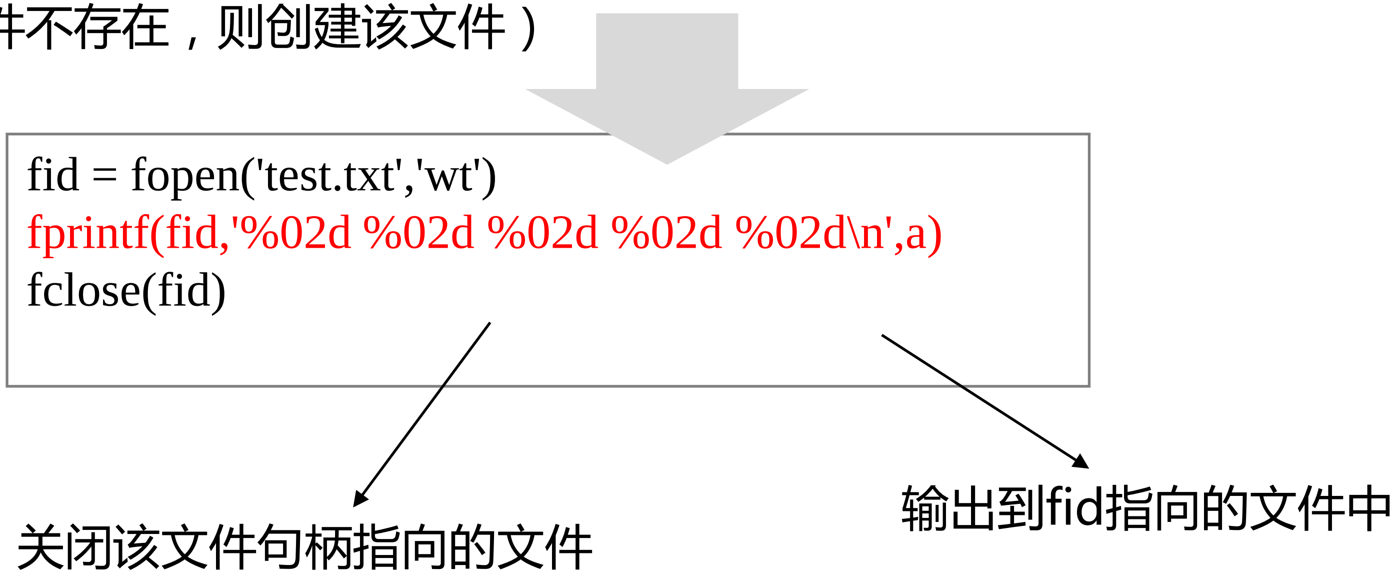
```
>> fprintf('It''s Tuesday today.\n')
It's Tuesday today.
```

fprintf 练习：

```
a = [1:10;11:20];
fprintf('%d\n',a)
fprintf('%2d\n',a)
fprintf('%02d\n',a)
fprintf('%02d %02d\n',a)
fprintf('%02d %02d %02d %02d\n',a)
```

先想一想各条fprintf语句在屏幕上的输出是什么？
然后再看matlab的输出结果，与你想的一样吗？

底层文件操作：在当前工作目录以文本的方式打开一个叫'test.txt'的文件，并将其句柄赋值给fid（若文件不存在，则创建该文件）



```
fid = fopen('test.txt','wt')
fprintf(fid,'%02d %02d %02d %02d %02d\n',a)
fclose(fid)
```

关闭该文件句柄指向的文件

输出到fid指向的文件中

mat文件格式

- ● mat: MATLAB特殊的数据文件格式
- ➢ 这种文件是一种二进制格式文件，扩展名为 .mat
- ➢ 跨平台的数据交互(不同平台下的MATLAB 在载入文件时进行必要的转换)。
- ● 例： Pearson的父子身高数据文件Pearson.mat, 含有矩阵FS
- ➢ load Pearson.mat
- ➢ 第二和第三列为1078 对父子的身高(Father, Son)
- ➢ 可以看看图plot(FS(:,2), FS(:,3), '.')
- ➢ What if we use plot(FS(:,2), FS(:,3)) ?

保存变量

- save 将当前工作空间中的变量保存到指定的数据文件中

save	将当前工作空间所有变量保存在当前目录下文件matlab.mat内
save filename	将当前工作空间中所有的变量保存到指定的文件中
save filename var1 var2......	将当前工作空间中的变量var1、 var2等保存到指定文件中
save –option1 fname var1 var2	将当前工作空间中的变量var1、 var2等保存到指定文件中, 当中涉及某些选项(option)
save('filename',......)	save指令的函数格式用法

- 如果 ‘filename’已经存在，会发生覆盖。若需要避免，可以加上 ‘-append’选项。例如, 若要添加变量t到原有文件，可以:

```
save -append -mat Pearson.mat t
```

- 在Octave下：若读取或保存Matlab文件(.mat)，请加上 -mat 选项
- 若需使用文本形式,可以指定选项 '-ascii

读取mat文件变量

- load 将数据文件的数据导入到MATLAB的工作空间

load	将‘matlab.mat’文件中所有的变量加载到当前的工作空间（前提是该文件存在）
load fname	将指定文件中所有的变量加载到当前的工作空间
load fname var1 var2	将指定文件中的指定变量加载到当前工作空间
load –option1 fname var1 var2	将指定文件中的指定变量加载到当前工作空间, 当中涉及某些选项(option)
load(‘filename’, ‘var1’, ‘var2’......)	load指令的函数格式

Note：使用load指令加载数据文件时，数据文件只要保存在MATLAB的搜索路径上即可，同时若不指明数据文件的扩展名，则数据文件默认按照MAT文件格式加载。

- 在Octave下：若读取或保存Matlab文件(.mat)，请加上 -mat 选项
- 若需使用文本形式,可以指定选项 '-ascii'

例: save和load指令示例

```
clear all
x1 = 2;
x2 = 3;
x3 = 4;
y1=0;
save '-mat' 'xdata.mat' x1 x2
save '-mat' '-append' 'xdata.mat' y1
```

执行load 命令后，请查看当前工作目录，确认是否生成了相应的文件。

```
clear all
load 'xdata.mat'
```

执行load 命令后，请查看内存中是否有相应的变量

电子表格的读写

- csvread:读取以逗号作为间隔符的文本文件
- csvwrite:保存数据到文本文件，逗号作为间隔符
- dlmread:按照指定的间隔符读取文本文件的数据
- dlmwrite:按照指定间隔符将数据写入文本文件
- 注意：这是针对矩阵（或数组）的，所以数据的类型应全部相同。例如，不能字符与数值类型混杂在一起使用这些表格

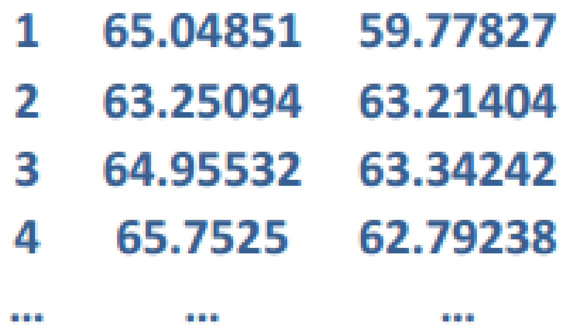
从cvs文件读取数据：矩阵型数据文件

- .csv 文件：以逗号作为间隔符的文本文件
- 当数据文件包含的全是数值，可以用csvread命令读取数据，其返回的是矩阵
- 例如Pearson的父子身高数据FS_Matrix.csv
- FS= csvread('FS_Matrix.csv');
- plot(FS(:,2),FS(:,3),'r.'); %以红点的形式将第二列与
- 第三列数据作散点图

csvread的一般用法

- csvread读取的文件只能包含数值，并且数值之间以逗号分隔。
- M = csvread(‘filename’)
- ➢ 返回矩阵M的行数与文件行数相同，列数为原文件中列数的最大值，对元素不足的行以0补足
- M = csvread(‘filename’, row, col) %从略
- ➢ 从filename 的第row行第col列开始读取数据。
  注意： 此时的行、列从 0 开始。
- M = csvread(‘filename’, row, col, range) %从略
- ➢ 从filename 的第row行第col列开始为读取区域，读取的
- 数据由数组 range 指定
- ➢ range 的格式为： [R1 C1 R2 C2]，其中 R1、 C1 为读取区域左上角的行和列，R2、 C2为读取区域右下角的行和列

svread的一般用法举例



1	65.04851	59.77827
2	63.25094	63.21404
3	64.95532	63.34242
4	65.7525	62.79238
…	…	…

● Pearson的父子身高数据文件（图为前4行）

● M = csvread('FS_Matrix.csv')

➢ 矩阵M存下所有文件FS_Matrix.csv的数据

● M = csvread'FS_Matrix.csv',1, 1)

➢ M存下了从第 ‘1’ 行第 ‘1’ 列到最后一行第 ‘1’ 列的所有数据。

➢ 第 ‘1’ 行（列）实际是第二行（列），因为从0行0列开始标号。 M中前3行：

63.25094 63.21404
64.95532 63.34242
65.7525 62.792438

● M = csvread('FS_Matrix.csv',1, 1, [1,1, 2, 2])

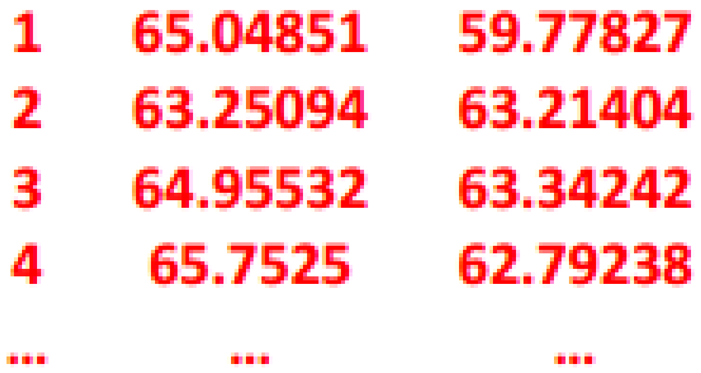
➢ 标号方式同上，第 ‘1’ 行第 ‘1’列到第 ‘2’ ‘1’ 行 ‘第 ‘2’ 列的所有数据。 M为

63.25094 63.21404
64.95532 63.34242

➢ 注意：诸如csvread('FS_Matrix.csv', 0, 0, [1,1, 2, 2]) 的用法会警告。可以尝试

csvwrite的一般用法

- csvwrite(filename,M)
  将矩阵 M 写入 filename 中
- csvwrite(filename,M,row,col)
  将矩阵 M 写入以指定行和列偏移开始的 filename 中。行和列参数从零开始， 因此 row=0 和 col=0 指定文件中的第一个值。



1	65.04851	59.77827
2	63.25094	63.21404
3	64.95532	63.34242
4	65.7525	62.79238
...	...	...

dlmread和dlmwrite

- 类似于csvread和csvwrite，只是分隔符可以不是逗号
- dlmread

  M = dlmread('filename')

  M = dlmread('filename', delimiter)

  M = dlmread('filename', delimiter, R, C)

  M = dlmread('filename', delimiter, range)
- dlmwrite

  dlmwrite('filename', M)，

  dlmwrite(‘filename’, M, delimiter) (如果需要 tab 键做分隔，可以
- 用 ‘\t’指定delimiter；

  dlmwrite('filename', M, '分隔符名字', R, C)，指定写入数据的起始位置；

  dlmwrite('filename', M, '-append')，如果 filename 指定的文件存在，
- 在文件后面写入数据，不指定时则覆盖原文件

例： load Pearson.mat

dlmwrite('TestFile.txt',FS, '\t')

readtable (Available since MATLAB R2013b)

- T=readtable(‘filename’)
  - 从文件创建table类型变量T，filename中每列对应与一个变量。从filename的第一行读取相应变量名称。默认的方式：如果一整列都是数值型，则对于一个double型变量（列向量）；如果一列中有任何非数值型数据，则整列被认为是字符串cell
  - MATLAB通过filename的后缀名辨认文件格式
- New data-type: table
  - Similar to dataframe in R

readtable (Available since MATLAB R2013b)

- 例子Stock_FX.csv：从87年到2006年的一些股票价格数据
- 需要分析FORD和GM两股票的收益率有什么关系
- S=readtable(‘filename’);
- 前三行组成一个表格： S1=S(1:3,:)
- 看两股票价格关系？

➢ plot(S(:,3), S(:,5),'.')? 试试看？

S是一个表格，带有表头

访问table类型的方式

- 例如， GM和Ford的股票价格分别在第3和第5行（变量），可用以下其中之一
  Ford=S{:,5}; GM=S{:,’GM_AC’};
  GM=S.GM_AC; Ford=S.F_AC;
  GM=S.(3); Ford= S.(5);
- 现在可作图看股票价格关系(其中rows为数字, vars可为数字或者字符串)

句法	所得结果	行	变量（vars）
S(rows,vars)	table	一行或多行 (可用冒号)	一个或多个变量 (列)组成的table
S{rows,vars}	提取所得数据	一行或多行 (可用冒号)	一个或多个变量(列)
S.var **S.(varindex)**	提取所得数据	所有行	一个变量(列)
S.var(rows)	提取所得数据	一行或多行 (可用冒号)	一个变量(列)

直接生成table

- T=table({'Isabella'; 'Alex'}, ['M'; 'F'], [35;33], [true;false], 'VariableNames',{'Name','Gender','Age', 'PhD'})
- T 中的每个变量包含 2 行。

  Name 为字符串元胞数组

  Gender为字符数组

  Age 为double数组，

  PhD 为逻辑数组。
- summary: 输出表中各变量的数据类型及其他信息。
- H = height(T) 返回表 T 中的行数
- W= width(T) 返回表 T 中的列数

writetable

- writetable(T,filename)
- 将表 T 写入由 filename 指定名称和扩展名的文件。
- 例：
- T=table({'Isabella'; 'Alex'}, ['M'; 'F'], [35;33], [true;false],
- 'VariableNames',{'Name','Gender','Age‘, ‘PhD’})
- writetable(T,'testwrite.csv');

- 关于table、 readtable和writetable的更多用法，请参见MATLAB的help文档。

05

Graph

plot(x,y)

- The simplest form: plot(x, y), where x and y are
- vectors of the same length
- opens a graphics window and draws an x-y plot of the
- elements of y versus the elements of x.
- Eg. x = -3:.01:3 ; y=exp(-x.^2); plot(x, y)
- “plot” actually connects consecutive points induced by
- the partition with line segments.
- try this: x=-3:3, y=exp(-x.^2); plot(x, y)

Multiple Figures

- You may have noticed that, plotting the 2nd graph eliminate the 1st one
- Want to have different figure?

  graphic handle

  figure(1); plot(x1,y1); figure(2); plot(x2,y2);
- If multiple figures are opened, figure(number) could specify which figure want to work on.
- If you don’t know which is the current, “gcf”

Combine plotting with function handle

- Write a function to plot a 'function' over some interval [a,b]

```
function MyPlot(fun, a, b, n, fig_num)
    if(isempty(n))
        n=100;
end
X=linspace(a,b,n); Y=fun(X);
if(~isempty(fig_num))
  figure(fig_num)
end
plot(X,Y);
```

Plotting a Parametric curve

Sometimes, we want to plot a parametric curve (x(t),y(t))
E.g. We want to plot (cos(t), sin(t)), t from 0 to 2*pi.
(What is that?)

- t = 0:0.01:2*pi ; x = cos(t) ; y = sin(t) ; plot(x, y)

Titles and labels?

- title(‘This is the curve of (cos(t), sin(t))’);
- In general, title(st) will create the title, where st is a string.
- For labels on x-axis and y-axis
  - xlabel('x: cos(t)'); ylabel('y: sin(t)')
- Wanna place some text somewhere in the graph, say,mark the center of the circle?
  - “text(x0, y0, ‘string’)”. E.g. text(0,0, 'Center')

The previous circle looks weird?

- By default, MATLAB scales the axes automatically
- We could change that.

  “axis equal”: same scale and tic marks on axes

  “axis square”: makes the current axes region square--axes of same size (not necessarily same scale)
- axis off: removes the axes
- axis on: restores the axes

More about legend

- legend('stringl', 'string2') and so on places identifier with different strings to match the curves (plots)
- To put it to the desired place

➢ either: legend('string1', 'string 2', 'Location', [left, bottom, width, height])

➢ or: legend('string1', 'string 2', 'Location', 'North')

➢ The one next to 'Location' could be 'North', 'South', 'East', 'West', 'NorthEast', 'NorthWest', etc. or 'EastOutside', 'WestOutside', etc. Could also be 'Best', 'BestOutside'

Marker Specifiers

'+': Plus: sign

'o': Circle

'*' : Asterisk

'. ' : Point

'x' : Cross

'square' or 's': Square

'diamond' or 'd': Diamond

'^': Upward-pointing triangle

'v': Downward-pointing triangle

'>': Right-pointing triangle

'<': Left-pointing triangle

'pentagram' or 'p‘: Five-pointed star (pentagram)

'hexagram‘: Six-pointed star (hexagram)

'none': No marker (default) 仅是将点相连

Example: plot(MyDateNum, F_AC, 'p')

Line types

- Line Style Specifiers:

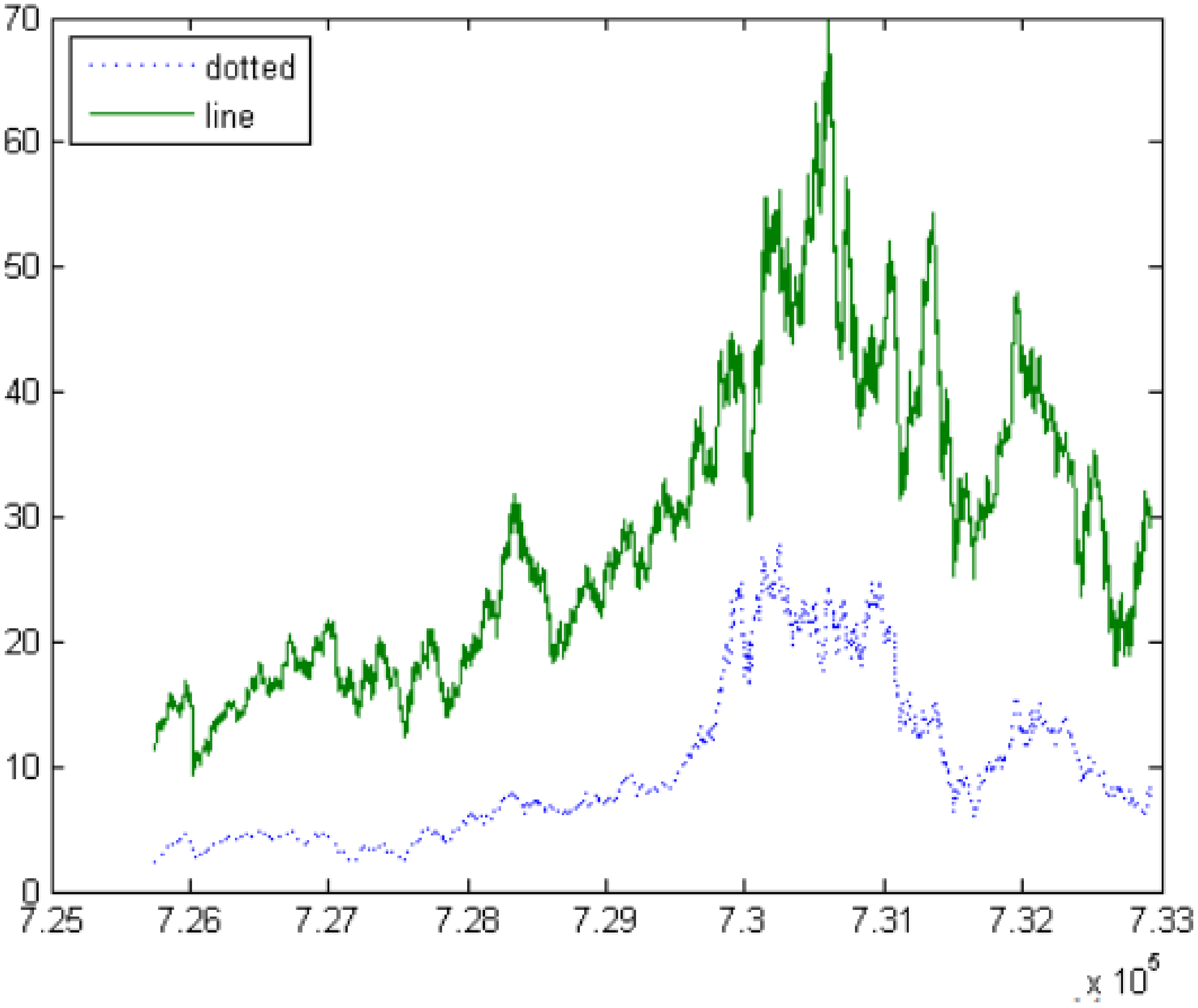
'-': Solid line (default)

'--': Dashed line

':' : Dotted line

'-.': Dash-dot line

'none': No line



Color Specifiers

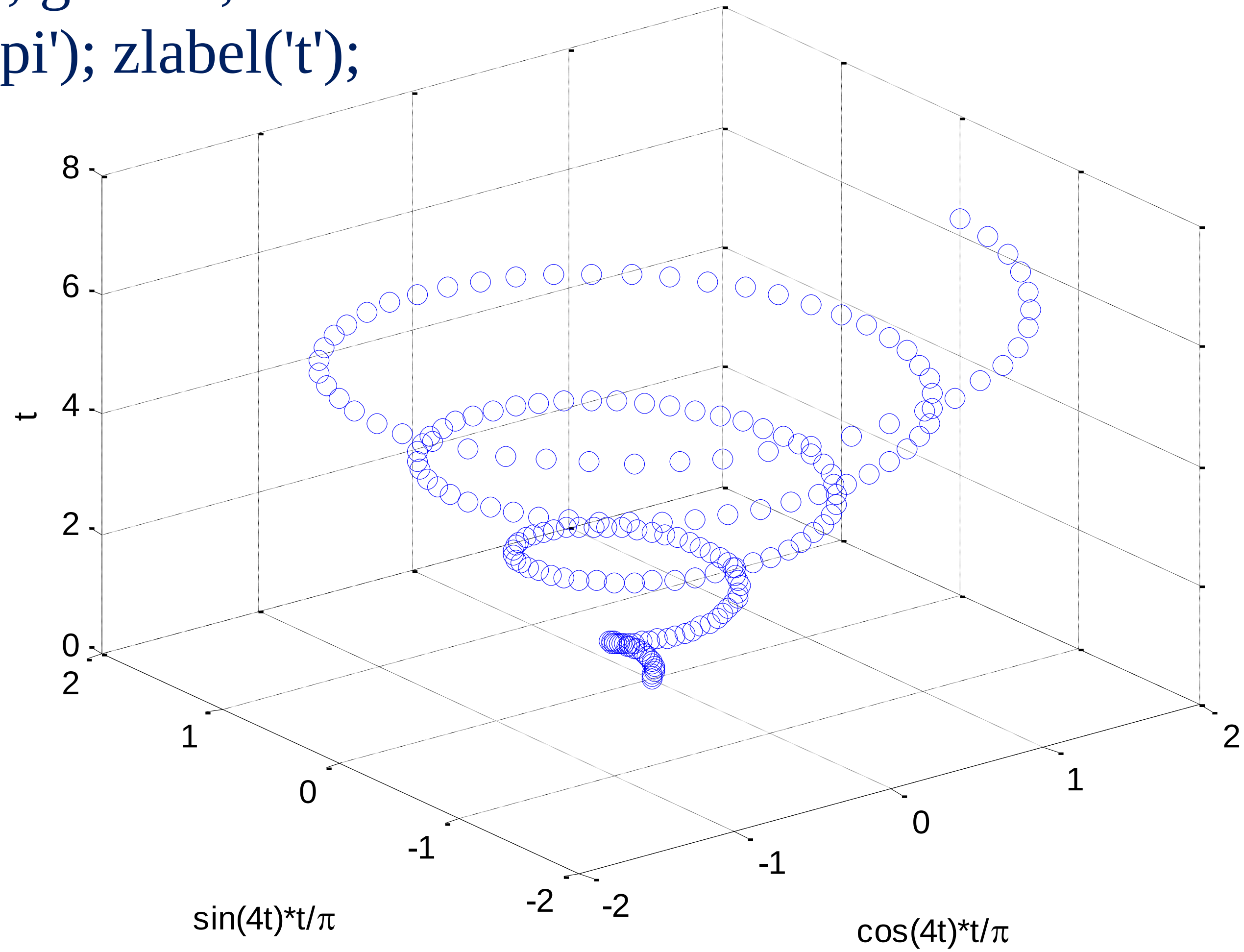
- Most used:
  ‘r’: Red
  ‘g’: Green
  ‘b’: Blue
  ‘y’: Yellow
  ‘k’: Black
- Other:
  ‘c’: Cyan
  ‘m’: Magenta
- Example: t=0:0.01:2*pi; plot(cos(t), sin(t), 'm')

plot3: Plotting 3D points/curves

- Similar to plot: plot3(x, y, z)
- Example: t=linspace(0, 2*pi, 200);

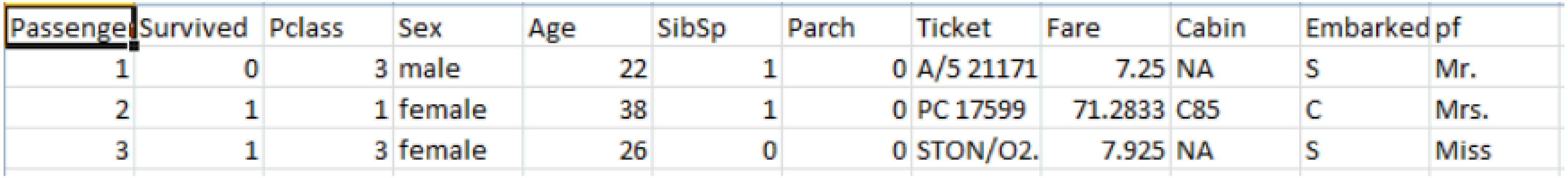
➢ plot3(cos(4*t).*t/pi, sin(4*t).*t/pi, t, 'o'); grid on;

➢ xlabel('cos(4t)*t/\pi'); ylabel('sin(4t)*t/\pi'); zlabel('t');



Explanation of the Titanic Data Set

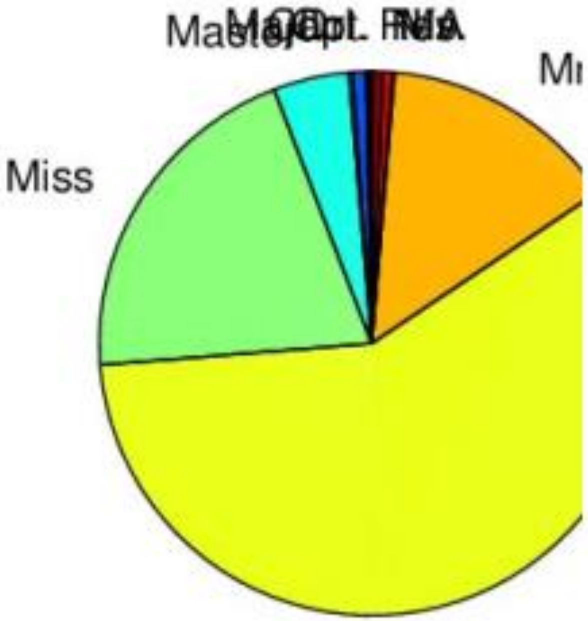
Variable:	Explanation
Survived:	Survival (0 = No; 1 = Yes)
Pclass:	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
Sex:	Gender
Age:	age
SibSp:	Number of Siblings/Spouses Aboard
Parch:	Number of Parents/Children Aboard
Ticket:	Ticket Number
Fare:	Passenger Fare
Cabin:	Cabin Information.
Embarked:	Port of Embarkation(C = Cherbourg; Q = Queenstown; S = Southampton)
pf:	Prefix (Title)

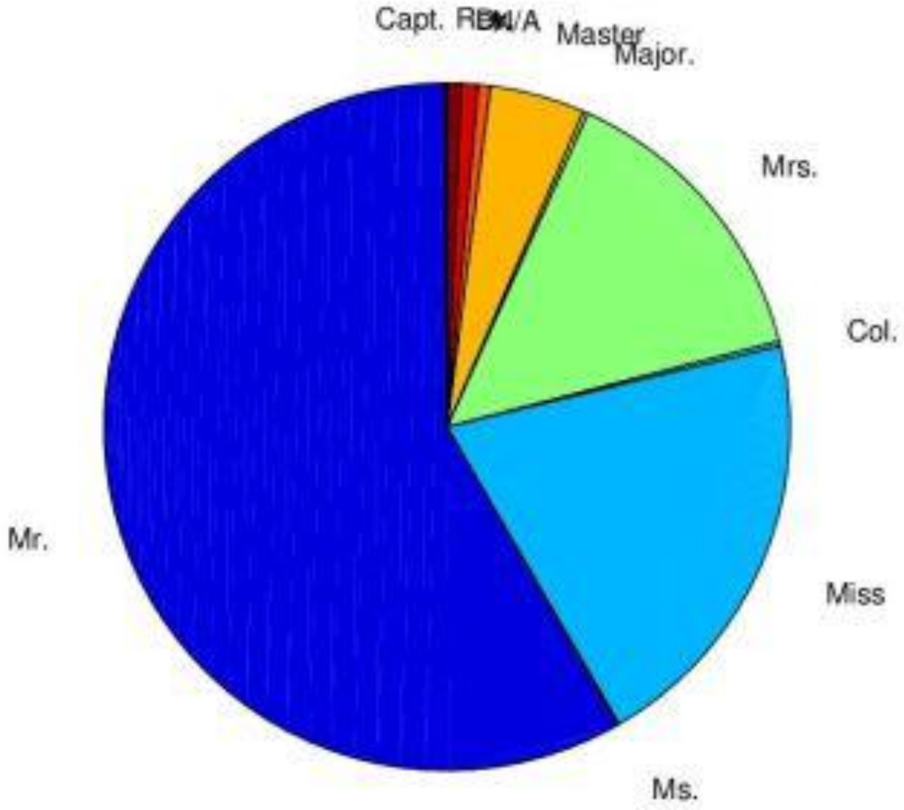


Passenge	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	pf
1	0	3	male	22	1	0	A/5 21171	7.25	NA	S	Mr.
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C	Mrs.
3	1	3	female	26	0	0	STON/O2.	7.925	NA	S	Miss

Pie Chart: Nominal (Categorical) Data

- Want to check the 'Titles' or 'Prefix'?
  [strs,counts,percents]=FreqDist(data.pf)
  %Self Written, provides Frequency Distribution
- subplot(1,2,1);pie(counts,strs);
- subplot(1,2,2); pie3(counts,strs);
- Ugly? Re-Arrange the labels (Alternating)





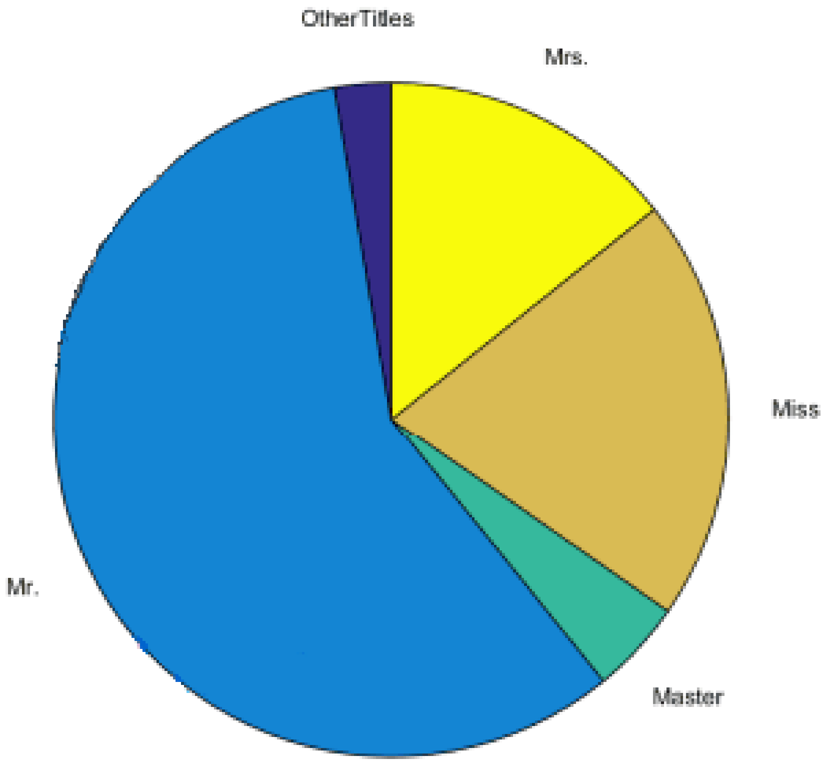
Color Specifiers

Four titles are fairly common:
Master (40), Mr. (518),Miss (182), Mrs. (129)

```
[sc,ind]=sort(counts,'descend'); strs(ind(1:4))
sc(ind(1:4))
```

Title Explanation

Master	**for boys and young men who are usually unmarried.**
Miss	for girls and young women who are usually unmarried.
Mr	for men.
Mrs	for married women.

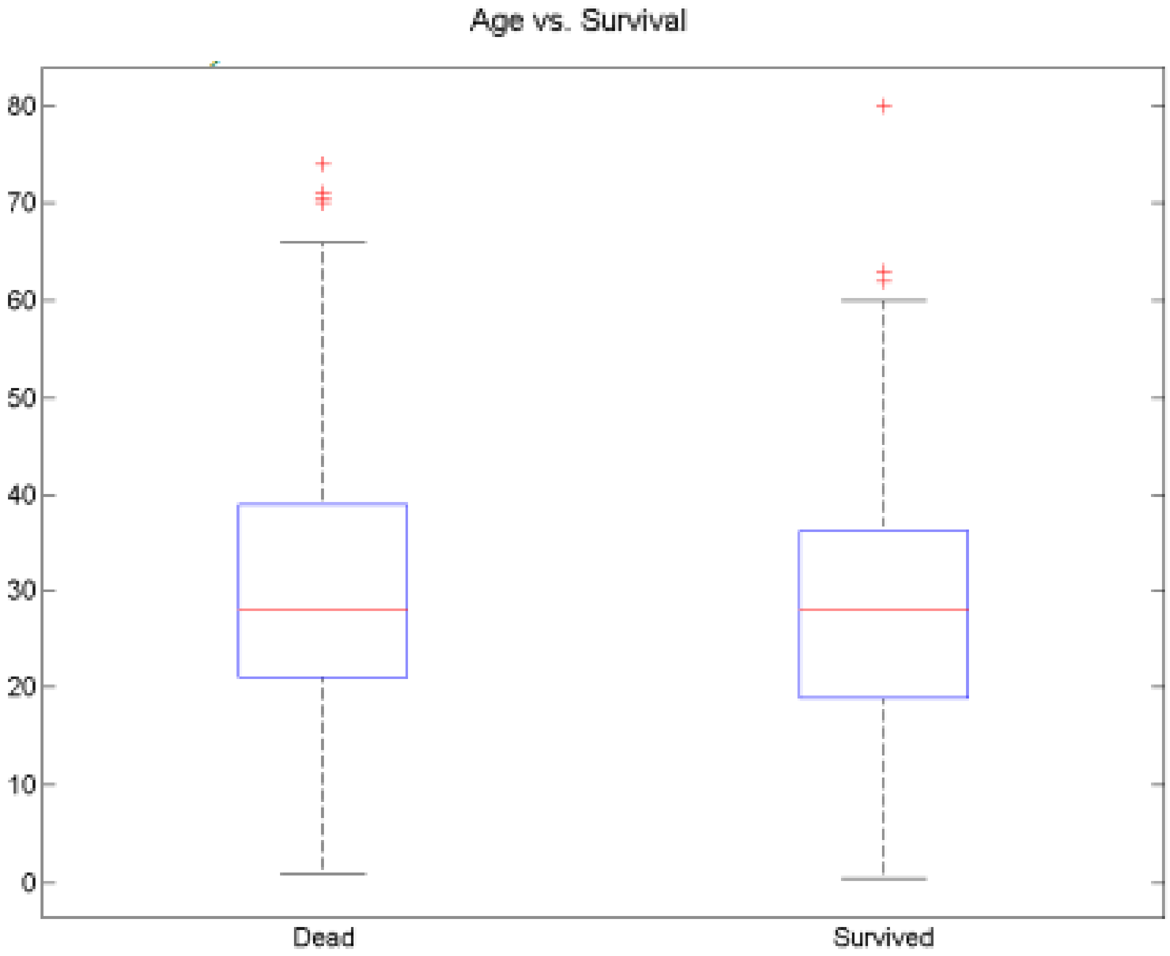


Re-Grouping: Common Titles vs. Other Titles

Box Plot: Quantitative Data

- Example 1: Age vs. Survival Status

```
boxplot(Age,Survived,'labels',{'Dead','Survived'})
title('Age vs. Survival')
```



Data Analysis: Dealing with Missing Values

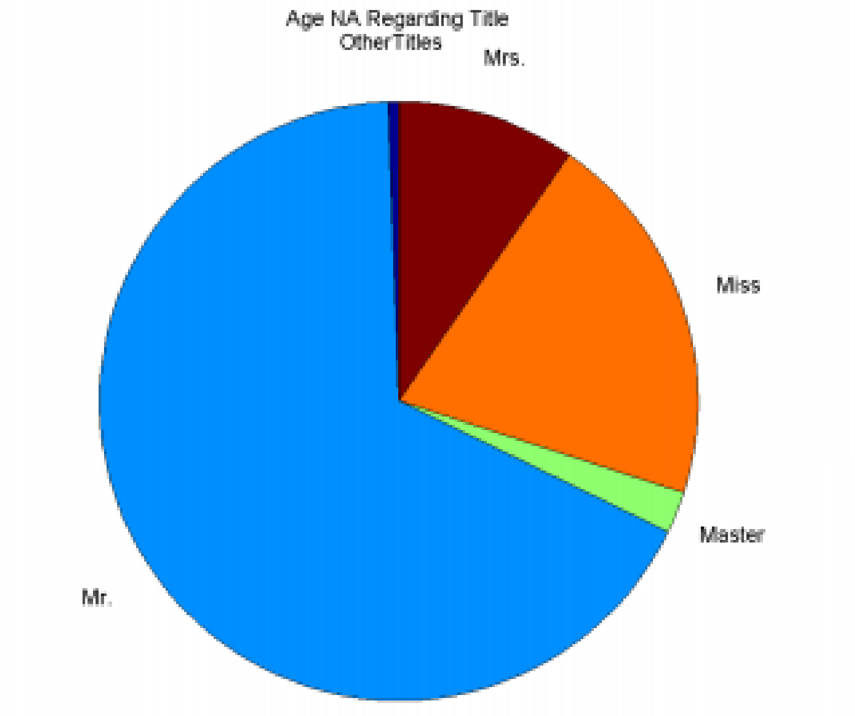
- Age has 177 missing values...
- Before making prediction with some model, clean it up...
- Possible way:

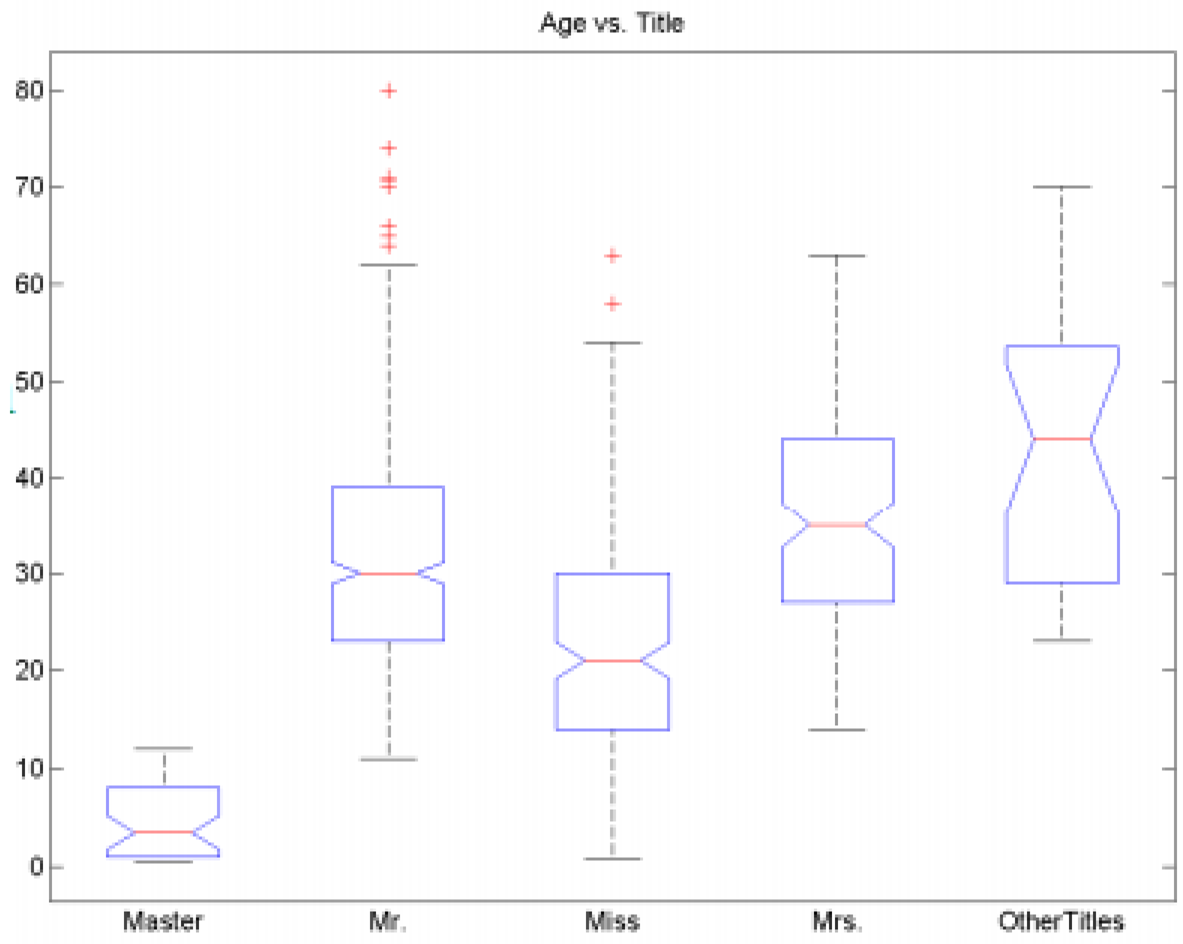
  Delete the observations

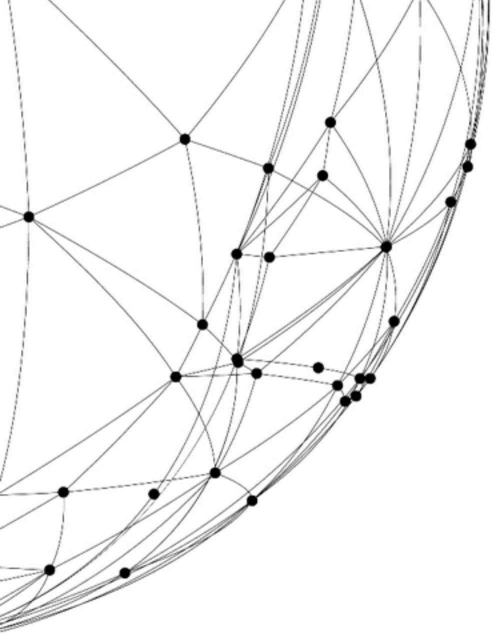
  Delete the variable

  Imputation with Mean/Median

  Some advanced method







THANK YOU VERY MUCH