Lecture 1: Fundamentals of Matlab

Su, Danyang

What Is Matlab:

- High-level programing language: between C++ and Stata
- Commonly used in numerical analysis, and visualization.
- Data are stored as matrices (for the purpose of this course)
- Learning by doing

Pros:

- Vectorization
- Powerful toolbox (including visualization, and optimization.
- Relatively low cost of learning.
- Flexible and consistent; good for DIY

Cons:

- Vectorization
- Not fool-proof
- Not fast enough
- Not as specialized as Stata

Types of files:

- Scripts: executable scripts
- Functions: called by scripts and other functions
    - Example: 1 + sqrt(2)
- Data: affix .mat, store variables in workspace (save)

Interface:

- Command window: for simple use
- Text editors: create/edit .m file
- Variable editors: display variables in **current** workspace

variables:

- vectors

```
x = zeros(5,2) + 2*ones(5,2);
%matrix size must matches.
y = zeros(4,1);
%always preallocate variables
y(1) = 1; y(2:3) = 4; y(4) = []; y(5) = 2==3;
%what does y look like?
```

- Cells: a = cell(5); a{1} = ‘Name’, a{2} = age;

Basic Operations:

- Scalar operators: (+ - * / ^)
- Matrix operators: (.* ./ * / .^ \)
  - Quiz: How to express $X = (A)^{-1}B$?

If:

- If arguments:

```
if a<b
    a = 1;
elseif a==b
    a = 2;
else
    a = 3;
end
```

Iterations:

- For Loop:

```
for i=1:size(x,1)
    for j=1:size(x,2)
        x(i,j) = (i-1)*size(x,2) + j;
    end
end
```

Iterations:

- While loop:

```
dist = 1; %initial value must not satisfy condition
x0 = 1
while dist>0.01 %don't mess up the sign
    x_new = 0.9*x0 + 0.1*f(x0);
    dist = abs(x_new-x0); %remember abs
    x0 = x_new; %don't mess up order
end
```

Vectorization:

- Try to vectorize whenever possible (mostly replace for loop):

```
y = 0;
for i=1:4
    y = y + x1(i) + x2(i);
end
y = x1*x2';
```

Run Into Problem?:

- use function **help**

  ```
  help sum
  ```

- Google search
- Read documentation
- Ask your friends/me

Some Useful Commands:

- clc: clear display
- clear: clear variables
- clear all: clear more than just variables.
- Ctrl+c: stop ongoing execution
- A(:,1) : ":" stands for all
- A(index,end-1): index is a vector of indexes or logics, "end-1" second to last column
- C = [A B]; C=[A;B]: append matrices
- kron(A,B): Kronecker product of A and B

Some Useful Commands:

- a<=b: a less than or equal to b
- a~=b: a not equal to b
- a & b: and condition
- a | b: or condition
- ~a: not condition
- Comment/uncomment: %
- Section: %%

Some Useful Commands:

- eye(n): create identity matrix
- diag(A): return diagonal of A
- 1:2:8: [1 3 5 7]: create a matrix that starts with 1 with interval 2, up to 8. Note that 8 is not
- isempty(A): whether elements in A are empty?
  - Quiz: how to evaluate whether A is empty matrix?
- isnan(A):whether elements in A are NaN
- ans: display most recently unassigned evaluation

Some Useful Commands:

- log(A): element by element log of A
- exp(A), sqrt(A), abs(A)
- ceil(A): round towards +inf
- floor(A): round toward -inf
- round(A): round toward nearest integer
- norm(A): norm of A
- sum(A,d), min(A), max(A)

Random Numbers:

- rng(1)
- rand(n,m)
- randn(n,m)
- randsample(population,k,true,w)
- mvnrnd(mu,sigma)
- Quiz: how to generate binomial distribution samples without using mvnrnd?