Stochastic Signal Processing

Lesson 4: Experimental Report 1

- Syntax

% returns a random scalar drawn from the uniform distribution in the interval (0,1).

X = rand

% returns an n-by-n matrix of uniformly distributed random numbers.

X = rand(n)

%returns an sz1-by-...-by-szN array of random numbers where sz1,...,szN indicate the size of each dimension. For example, rand(3,4) returns a 3-by-4 matrix.

X = rand(sz1,...,szN)

- Syntax

%creates a probability distribution object for the distribution distname, using the default parameter values.

```
pd = makedist(distname)
```

% creates a probability distribution object with one or more distribution parameter values specified by name-value pair arguments.

```
pd = makedist(distname,Name,Value)
```

list = makedist returns a cell array list containing a list of the probability distributions that makedist can create.

- Commonly used distribution

  Binomial distribution

  Poisson distribution

  Uniform distribution

  Normal distribution

  Exponential distribution

  Rayleigh distribution

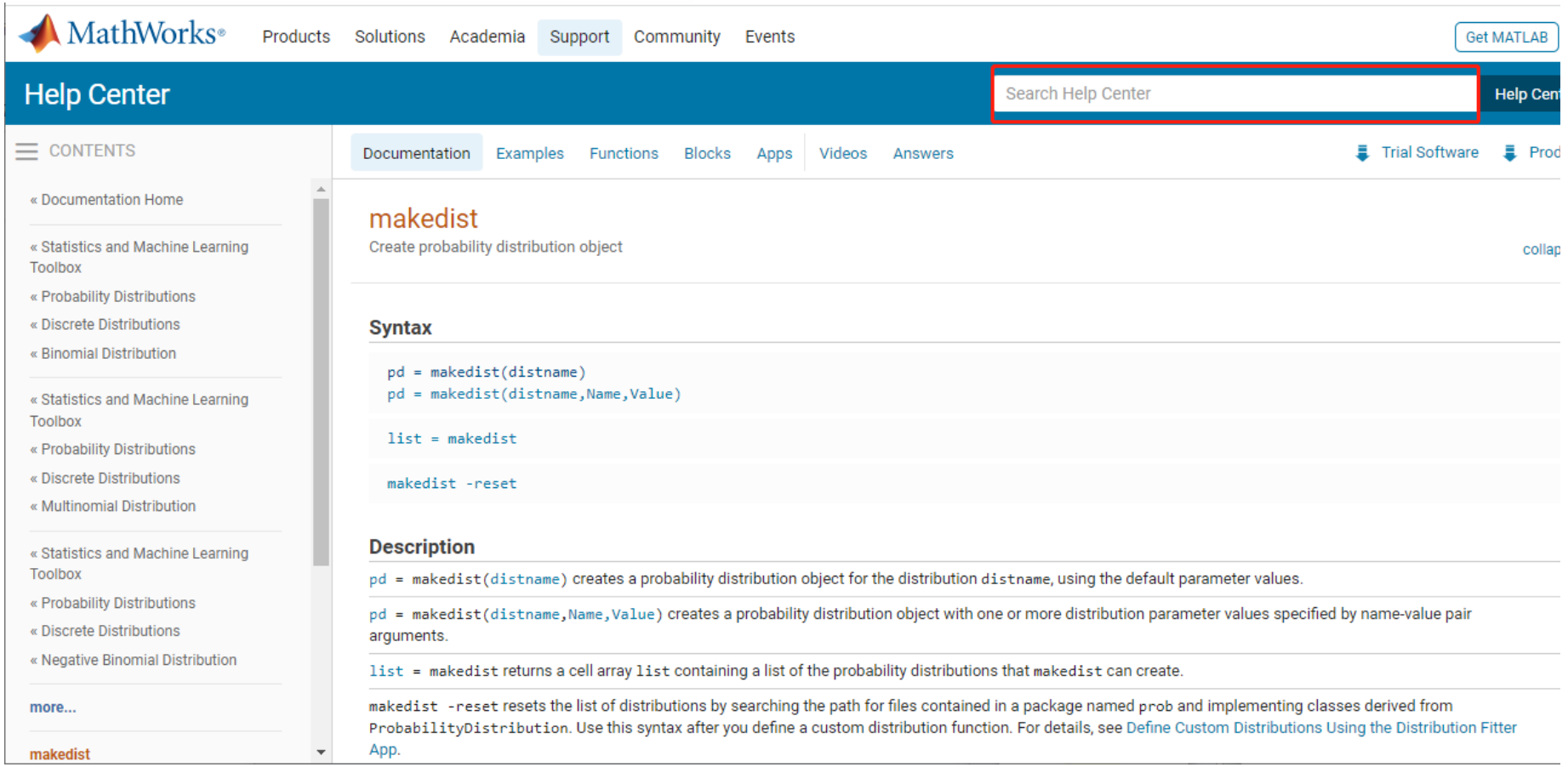
- Example

  pd = makedist('Normal');

  r = random(pd); % evaluate the Normal distribution and generate random numbers

- How to use ‘help’ in Matlab?

```
>>
>>
>>
>> help makedist
makedist - Create probability distribution object

    This MATLAB function creates a probability distribution object for the
    distribution distname, using the default parameter values.

    pd = makedist(distname)
    pd = makedist(distname,Name,Value)
    list = makedist
    makedist -reset

    另请参阅 distributionFitter, fitdist

    makedist 的文档

fx >>
```



- Practice 1

Generate two groups of standard normal distribution numbers X and Y, compute its mean, variance, standard deviation and their 2-by-2 covariance.

```
L = 1000;          % length
x = randn(1,L);    % create a new 1-by-L vector of random numbers

y = randn(1,L);
m = mean(x);       % compute the mean
v = var(x);        % compute the variance
s = std(x);        % compute the standard deviation
c = cov(x,y);      % compute the covariance between two random variables X and Y
```

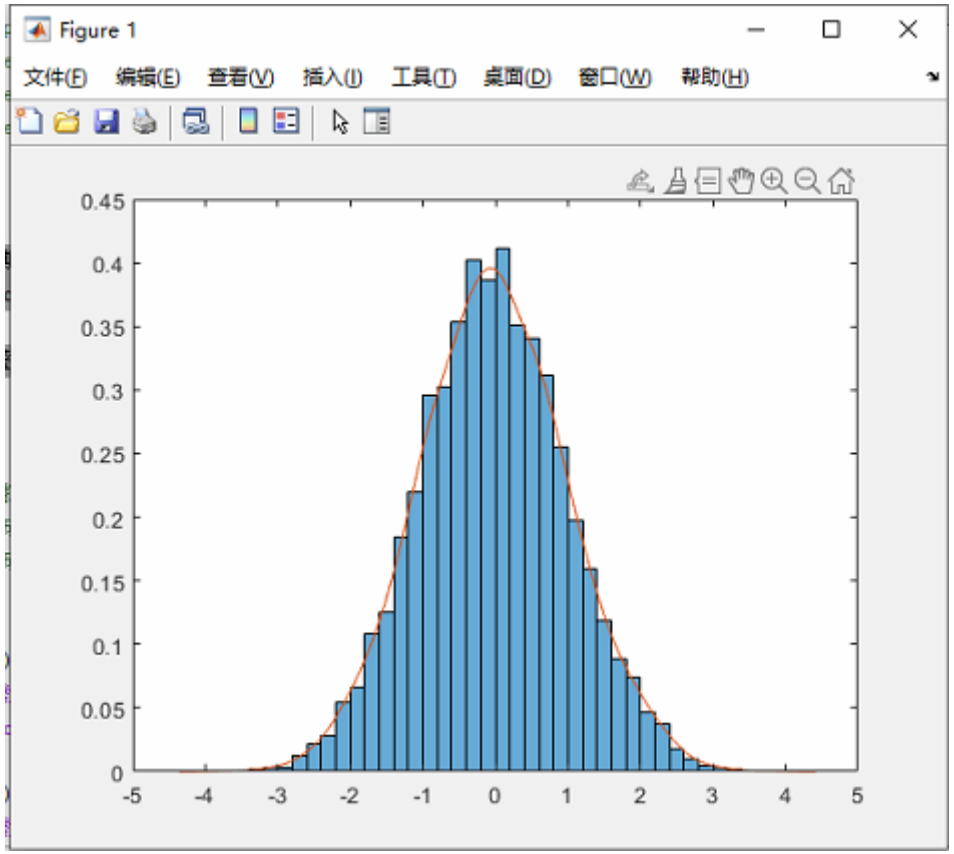
- Practice 2

Generate a 1-by-5000 vector of standard normal distribution , Plot its histogram and probability density function(pdf).（tips: use histogram and ksdensity）

```
N = 5000; % length
x = randn(1,N); % create a new 1-by-L vector of random numbers
histogram(x,'normalization','pdf'); % creates a histogram plot of X
hold on
ksdensity(x); % Kernel smoothing function estimate
```

- Practice 2

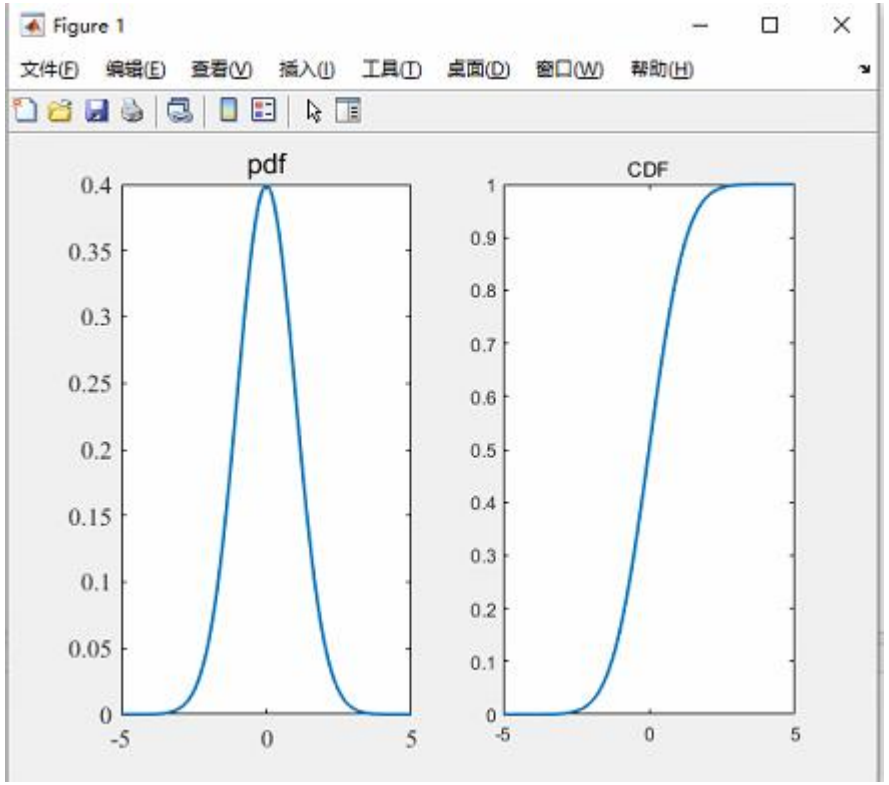
Generate a 1-by-5000 vector of standard normal distribution , Plot its histogram and probability density function(pdf). （tips: use histogram and ksdensity）



- Practice 3

Plot the probability density function and distribution function of the normal distribution. (tips: pdf , cdf)

```
pd = makedist('Normal');    % Create a Normal distribution object
x = -5:0.01:5;              % range
y = pdf(pd,x);              % the probability density function
z = cdf(pd,x);              % the cumulative distribution function
subplot(1,2,1)
plot(x,y,'linewidth',1.5);
title('\fontname{}pdf');
subplot(1,2,2)
plot(x,z,'linewidth',1.5);
title('\fontname{}CDF');
```



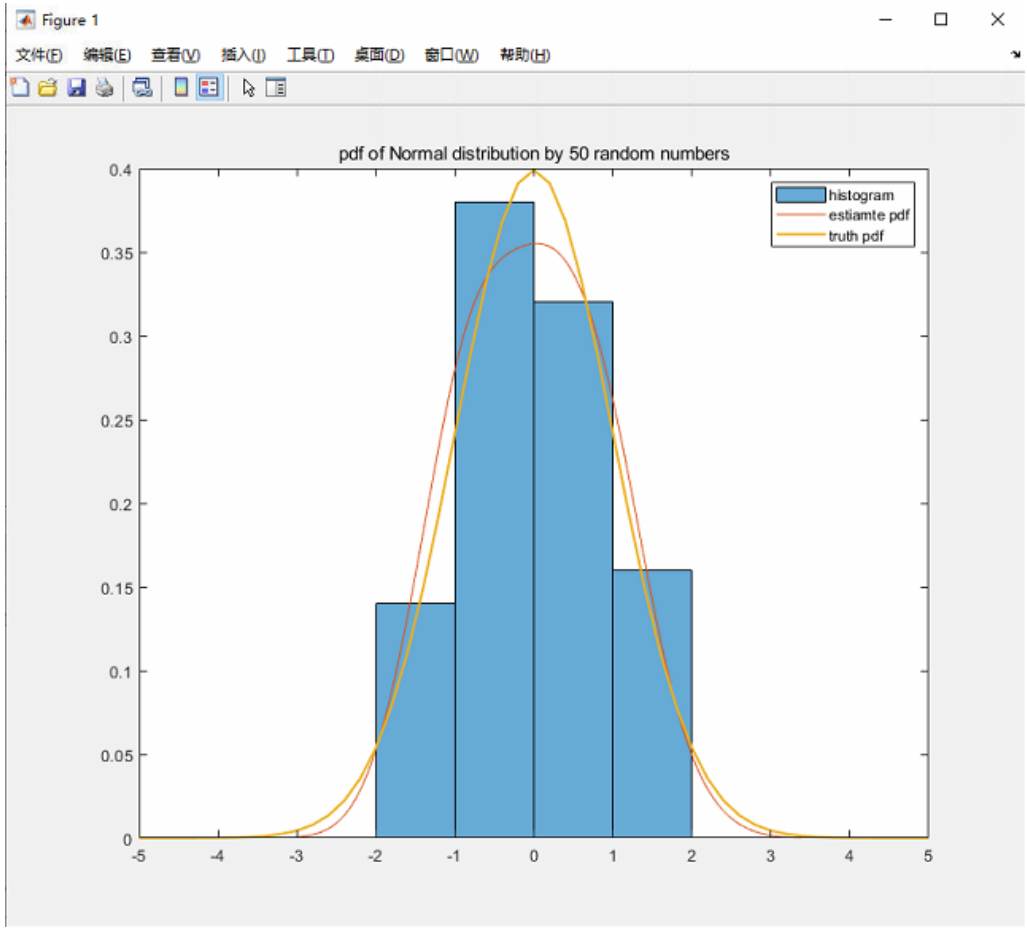
- Common pdf (Probability density function) and CDF (Cumulative distribution function)

	Pdf	Cdf
Binomial distribution	binopdf	binocdf
Poisson distribution	poisspdf	poisscdf
Uniform distribution	unidpdf	unidcdf
Normal distribution	normpdf	normcdf
Exponential distribution	exppdf	expcdf
Rayleigh distribution	raylpdf	raylcdf

- Practice 4

Plot the truth pdf, estimated pdf, and histogram of a Normal distribution by 50 and 5000 random numbers on the same graph.

```
% pdf of Normal distribution by 50 random numbers
x1 = randn(1,50);
histogram(x1,'normalization','pdf');
hold on
ksdensity(x1);      % estimate pdf
hold on
x = -5:0.2:5;
y =normpdf(x);    % Normal distribution pdf;
plot(x,y,'linewidth',1.5);
title('pdf of Normal distribution by 50 random numbers');
```

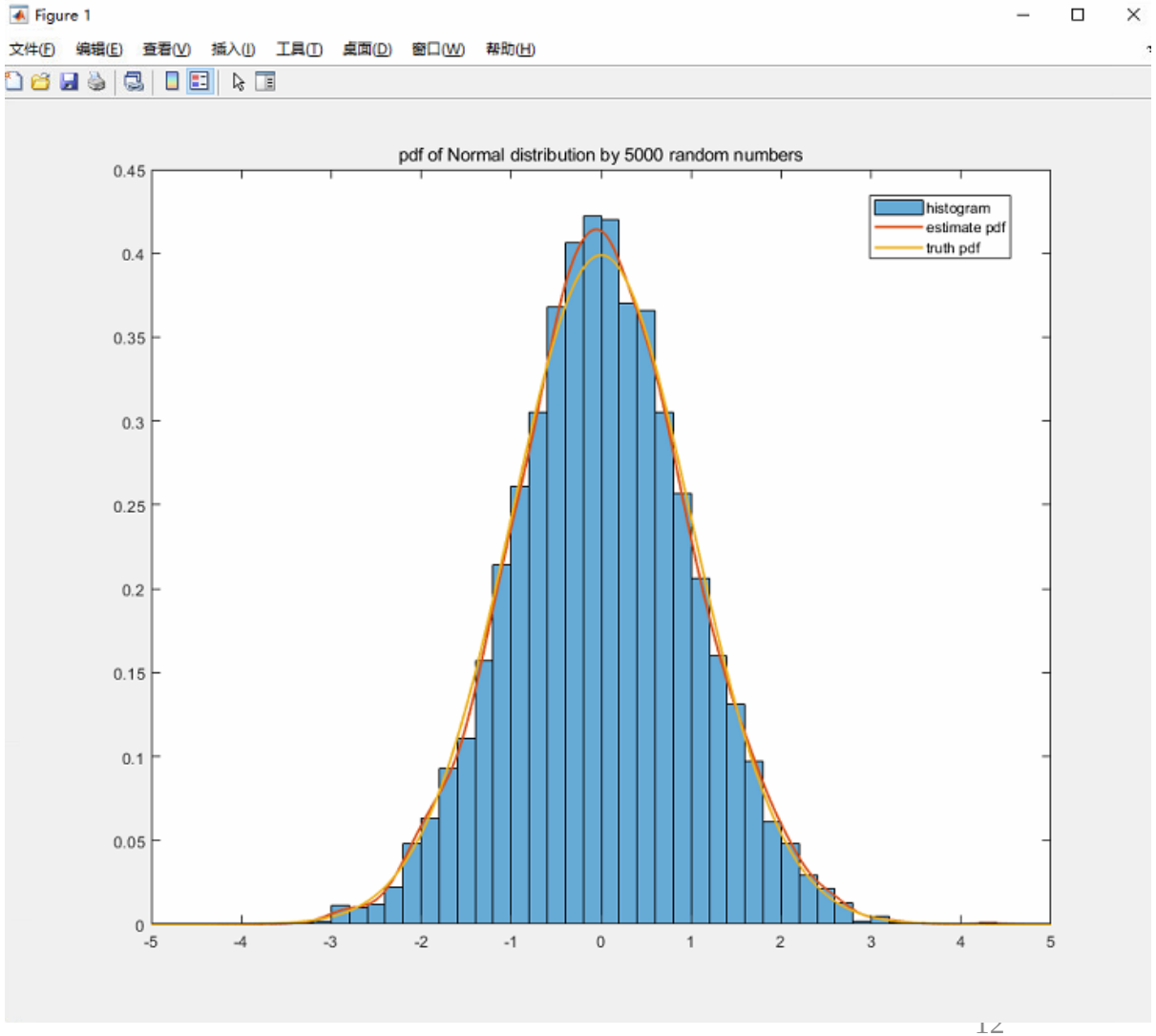


When only 50 random numbers are used to shown the histogram, it might seems not that 'Normal distributed'

- Practice 4

Plot the truth pdf, estimated pdf, and histogram of a Normal distribution by 50 and 5000 random numbers on the same figure.

```
% pdf of Normal distribution by 5000 random numbers
pd = makedist('Normal');
x1 = random(pd,1,5000);
x = -5:0.002:5;              % range
[f,x]=ksdensity(x1,x);  % estimate pdf
histogram(x1,'normalization','pdf');
hold on
plot(x,f,'linewidth',1.5);
hold on
y = pdf(pd,x);
plot(x,y,'linewidth',1.5);
title('pdf of Normal distribution by 5000 random numbers');
```



The Experimental Report 1

The experimental Report 1 contains 3+1 parts:

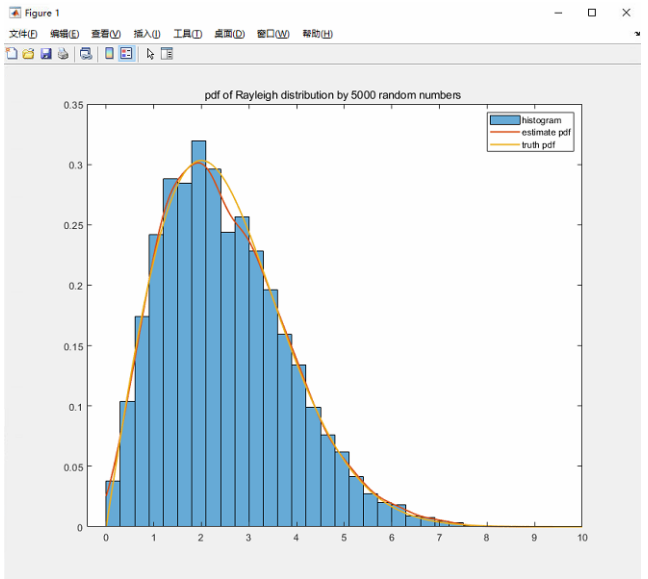
- Basic 1 (30 points) : submitted the program in weeks 1 and 2 and it is done.
  - You should submit your original codes submitted in weeks 1 and 2 again, without any changes
- Basic 2 (40 points) : a direct change of some program and figuring, will introduce next week.
- Advance (30 points) : Correctly use the Bayesian rule to determined the strategy 1 for game 1 and use appropriate system to perform the testing of strategy 1.
- Extra (10 points) :Correctly design the game 2, and correctly use the Bayesian rule to determined the strategy 2 for game 2, and use appropriate system to perform the testing of strategy 2.

- Experimental Report 1: Part 2 (40 points)

Refer to practice 4 and plot Rayleigh distribution， Poisson distribution， Uniform distribution and Exponential distribution. The figure should contains the truth pdf, estimated pdf, and histogram.

Requirement

1. Your code must be runnable (no error, warning accepted), otherwise, 0 point.
2. Explain the characteristics of each function and the different between 50 and 5000 random numbers for pdf.
3. Totally 8 figures , each figure 5 points.



The Experimental Report 1

Note: program already given in Lesson 3

- Advance (30 points) : the new game with Bayes' theorem

  You will now joint this game 1:

  - You will trade with one counterparty; your counterparty's action can be trust or betray
  - You have two strategies: trust and call police, and the return table is:

		A: Your counterparty	
		trust	betray
B: You	trust	A: +10; B: +10	A: +10; B: -10
	call police	A: +10; B: -10	A: -10; B: +10

  - Before trading, you will be given the following information:
    - Your counterparty's probability of betray follows uniform distribution in (0, 1)
    - You will be given the counterparty's previous 10 actions towards other persons, which is an 10*1 vector of {trust, betray}
    - The system and the default strategy will be given, but you are required to design your own strategy, and explain why you design your strategy like this
  - You will trade with this counterparty 100 times, and show your total return
  - You should estimate your counterparty's probability of betray, but, never directly look at your counterparty's actual probability of betray, otherwise, 0 point.

The Experimental Report 1

- Advance (30 points) : the new game with Bayes' theorem

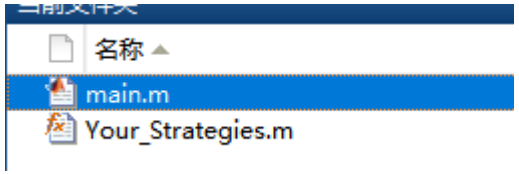
  Requirement for the testing of your strategy 1 for this game 1: (Score points)

  1. You should first submit the whole system with your strategy, and your system must be runnable (no error, warning accepted), otherwise, 0 point.
     - You can change your system for better testing or explanation of your strategy, but your counterparty's probability of betray must be a r.v with uniform distribution in (0, 1) in every independent run, which is, you should not set it as a constant, otherwise, 0 point.
  2. Explain the reason of your strategy, must related to probability. (Hint: see example 7, lesson 2)
  3. From a statistical point of view, how to design a system (which is, modify the default system) to evaluate your strategy? For example, you can evaluate your strategy with 1000 independent runs. Explain your modification of the default system, and explain your evaluation result of your strategy.

  30 points

The Experimental Report 1

- The system and the default strategy




```
clear;
clc;
warning off
% the above two commands clear all the previous record in the Memory
% counterparty action: 0 trust, others betray
% your action: 0 trust, others call police

N_trades = 100;                              % trade N_trades times
Return_total = 0;                            % the retrun
counterparty_betray_prob = rand(1);          % randomly initial the p
% note that it will last for the whole game
counterparty_previous_action_list = rand(10,1);
counterparty_previous_action = double(counterparty_betray_prob > counterparty_previous_act

for n_trade = 1 : N_trades                   % looping
    n_trade                                  % just to show the trade no for quick check
    Your_Strategy = Your_Strategies(counterparty_previous_action);
    % this time, you can pass anything you want into the 'Your_Strategies',
    % except the 'counterparty_betray_prob'
    % you can change the whole system as you wish
    counterparty_action = double(counterparty_betray_prob > rand(1));
    if Your_Strategy==0
        if counterparty_action==0
            Return_current = 10;   % both trust, add 10 points
        else
            Return_current = -10;  % self trust, counterparty betray, -10 points
        end
    else
        if counterparty_action==0
            Return_current = -10;   % self call police, counterparty trust, -10 points
```

```
function Your_Strategy = Your_Strategies(counterparty_previous_action)
    % this is only a default strategy, it is not good
    Your_Strategy = double(0.5 > rand(1));
    % as the mean of the betray rate of your counterparty is 0.5, 50% trust
    % and 50% call police
end
```

The Experimental Report 1

- Extra(10 points) : the new game with Bayes' theorem

  You will now joint this game 2:

  - You will trade with one counterparty; your counterparty's action can be trust or betray
  - You have two strategies: trust and call police, and the return table is:

		A: Your counterparty	
		trust	betray
B: You	trust	A: +10; B: +10	A: +10; B: -10
	call police	A: +10; B: -10	A: -10; B: +10

  - Before trading, you will be given the following information:
    - Your counterparty's probability of betray follows uniform distribution in [0.4, 0.7]
    - You have 100 friends, and they had already trade with this counterparty 100 times independently, your friends will tell you how many times of 'betray' our of 100 this counterparty did in their trading
  - You will trade with this counterparty 1 time only, and show your return
  - You should estimate your counterparty's probability of betray, but, never directly look at your counterparty's actual probability of betray, otherwise, 0 point.

The Experimental Report 1

- Extra(10 points) : the new game with Bayes' theorem

  Requirement for the testing of your strategy 2 for this game 2: (Score points)

  1. You should first submit the whole system with your strategy, and your system must be runnable (no error, warning accepted), otherwise, 0 point.
     - You should design your testing system all by yourself, your counterparty's probability of betray must be a r.v with uniform distribution in [0.4, 0.7] in every independent run, which is, you should not set it as a constant, otherwise, 0 point.
  2. Explain the reason of your strategy, must related to probability. (Hint: see example 8, lesson 3)
  3. From a statistical point of view, how to design a system (which is, modify the default system) to evaluate your strategy? For example, you can evaluate your strategy with 10000 independent runs, record the total return, or record the action as 'success or fail'(but you should give the definition of success and fail). Explain your modification of the default system, and explain your evaluation result of your strategy.

  Extra 10 points