**CALIFORNIA STATE UNIVERSITY, LONG BEACH**

**EE 381 - Probability, Statistics, and Stochastic Modeling**

**Projects**

# Project on

# Random Variables and Stochastic Experiments

## 0. Introduction and Background Material

### 0.1. Simulating Coin Toss Experiments

As mentioned in class, there are many ways to model stochastic experiments. The following two programs simulate the toss of a fair coin for N= 100,000 times, and calculate the experimental probability of getting heads (`p_heads`) or tails (`p_heads`). Both programs provide the same results, but they differ in the way the models are coded.

- The first model is programmed using "for loops".
- The second model makes use of the MATLAB vector objects, and it is computationally very efficient.

**MODEL 1 -- INEFFICIENT CODE**

```
clear
N=100000;
heads=0;
tails=0;
for f=1:N
    coin=round(rand);
    if(coin==1)
        heads=heads+1;
    else
        tails=tails+1;
    end
end
p_heads=heads/N
p_tails=tails/N
```

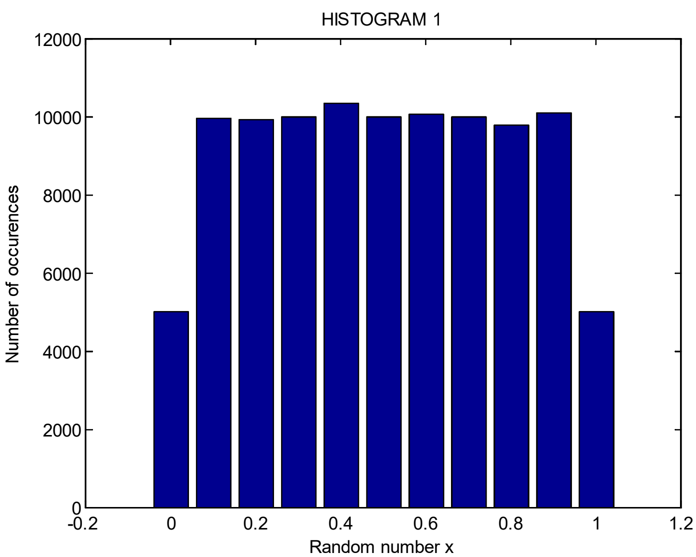
**MODEL 2 -- EFFICIENT CODE**

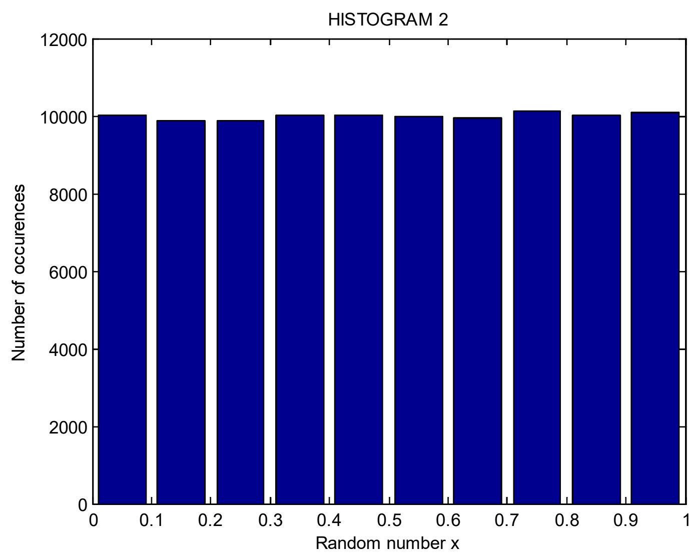
```
clear
N=100000;
heads=sum(round(rand(N,1)));
tails=N-heads;
p_heads=heads/N
p_tails=tails/N
```

### 0.2. Creating Histograms

This experiment generates N=100,000 random numbers. The results of the experiment are plotted in a histogram, using the MATLAB "`hist`" function. Depending on the limits set for the bins "`hist`" function, the histogram plots will be slightly different, as shown in the two plots.

| CODE FOR HISTOGRAM 1 | CODE FOR HISTOGRAM 2 |
|---|---|
| `clear; N=100000;`<br>`x=rand(N,1);`<br>`bins=0.0:.1:1.0;`<br>`[yvalues,xvalues]=hist(x,bins);`<br>`bar(xvalues,yvalues)`<br>`title('HISTOGRAM 1')`<br>`xlabel('Random number x');`<br>`ylabel('Number of occurrences')` | `clear; N=100000;`<br>`x=rand(N,1);`<br>`bins=0.05:.1:0.95;`<br>`[yvalues,xvalues]=hist(x,bins);`<br>`bar(xvalues,yvalues)`<br>`title('HISTOGRAM 2')`<br>`xlabel('Random number x');`<br>`ylabel('Number of occurrences')` |



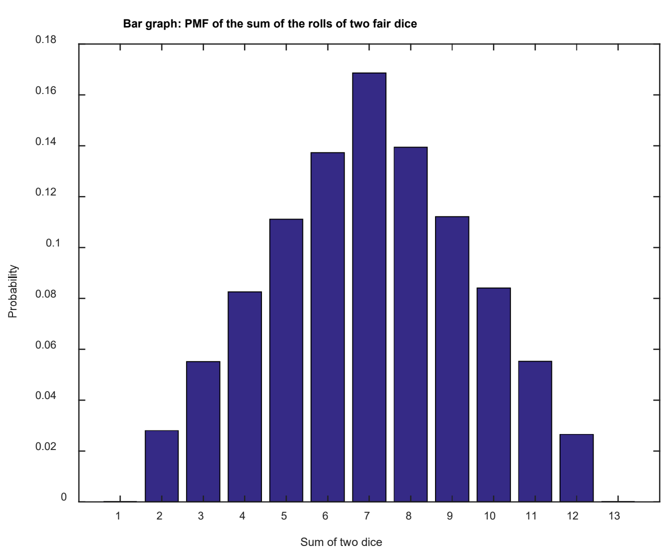



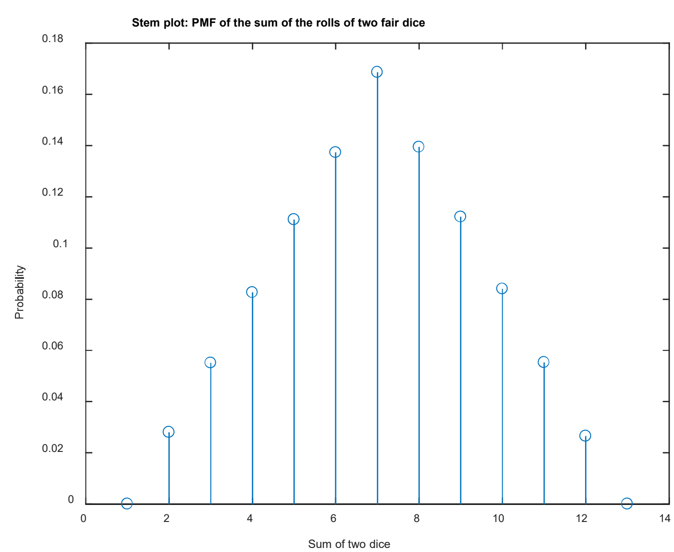
## 0.3. Roll of Two Fair Dice; Probability Mass Function (PMF)

This experiment models the roll of a pair of dice for N=100,000 times. The sum each roll is recorded, and stored in vector "s". The probability of each possible outcome is calculated and plotted in a "Probability Mass Function" (PMF) plot. The PMF plot is plotted in two different forms: as a bar graph or as a stem plot. The stem plot is the standard plot to be used for PMF graphs.

**SUM OF THE ROLLS OF TWO FAIR DICE**

```
clear
N=100000;
d1=ceil(6*rand(N,1)); d2=ceil(6*rand(N,1));
s=d1+d2;
bins=1:13;
[yvalues,xvalues]=hist(s,bins);
%
figure(1);  bar(xvalues,yvalues/N)
title('Bar graph: PMF of the sum of the rolls of two fair dice')
xlabel('Sum of two dice')
ylabel('Probability')
%
figure(2); stem(xvalues,yvalues/N)
title('Stem plot: PMF of the sum of the rolls of two fair dice')
xlabel('Sum of two dice')
ylabel('Probability')
```



## 0.4. Generating an unfair multi-sided die

This example models the roll of a 3-sided die, with non-uniform probabilities. The die has three sides $[1,2,3]$ with probabilities: $[p_1, p_2, p_3] = [0.3,\ 0.6,\ 0.1]$.

The experiment simulates the roll of the die for N=100,000 times, and the outcome of the N rolls is plotted as a stem plot. The stem plot verifies that the three sides of the die follow the required probabilities.

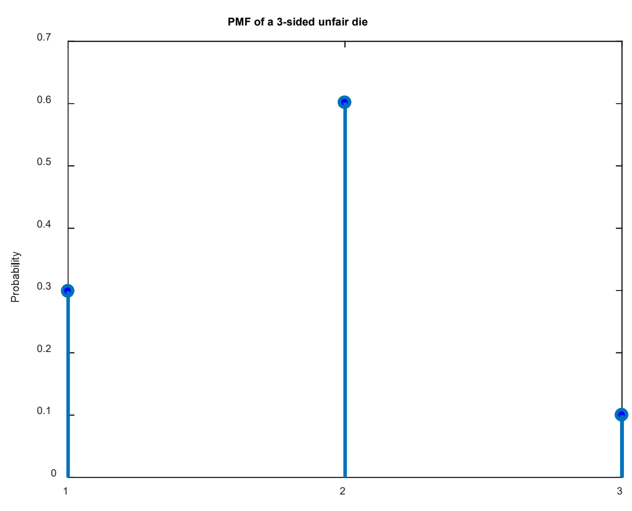
**ROLLING AN UNFAIR 3-SIDED DIE**

```
clear
N=100000;
s=zeros(N,1);
%
p=[0.3, 0.6, 0.1];
cp=cumsum(p);
for k=1:N
r=rand();
if r<=cp(1), y=1;
elseif r<=cp(2), y=2;
elseif r<=cp(3), y=3;
end;
s(k)=y;
end;
%
```

This simulates a single roll of the die

**PLOTTING THE PMF**

```
bins=1:3;
[yvalues,xvalues]=hist(s,bins);
close all
figure(1);
stem(xvalues,yvalues/N, 'LineWidth', 2, 'MarkerFaceColor', 'blue')
xticks([1,2,3])
title('PMF of a 3-sided unfair die')
xlabel('Face of the die')
ylabel('Probability')
```



## 1. Function for a n-sided die

Write a MATLAB function that simulates a single roll of a n-sided die. The inputs and outputs of the function are:

**Inputs:**

- The probabilities for each side, given as a vector $p = [p_1, p_2, \cdots p_n]$

**Outputs:**

- The number on the face of the die after a single roll, i.e. one number from the set of integers $\{1, 2, \cdots n\}$

**Note:** The sum $p_1 + p_2 + \cdots p_n$ must be equal to 1.0, otherwise the probability values are incorrect.

**Save** the function as: `nsided_die(p)` **Test the function** with a 6-sided die, where the probabilities are given by the vector $p$ =[0.20, 0.05, 0.10, 0.25, 0.30, 0.10]. To test the function, roll the die for N=1000 times and plot the outcome as a stem plot.

**SUBMIT a report** that follows the guidelines as described in the syllabus.

- The section on RESULTS must include The PMF in the form of a **stem plot**
- The code must be provided in the appendix

## 2. Number of rolls to needed to get 6 or 9 with two dice

Consider the following experiment:

- You roll a pair of fair dice and calculate the sum of the faces. You are interested in the number of rolls it takes until you get a sum of "6 or 9". The first time you get a "6 or 9" the experiment is considered a "success". You record the number of rolls and you stop the experiment.
- You repeat the experiment N=100,000 times. Each time you keep track of the number of rolls it takes to have "success".
- After the N experiments are completed you should generate a **probability mass function** graph of the number of rolls it takes for "success".

**SUBMIT a report** that follows the guidelines as described in the syllabus.

- The section on RESULTS must include the PMF in the form of a **stem plot**
- The code must be provided in the appendix

## 3. Getting 500 heads when tossing 1000 coins

Consider the following experiment:

- o You toss 1000 fair coins and record the number of "heads". This is considered a single experiment. If you get exactly 500 heads, the experiment is considered a "success".
- o You repeat the experiment N=100,000 times. After the N experiments are completed count the total successes, and calculate the probability of success, i.e. the probability of getting exactly 50,000 heads.

**SUBMIT a report** that follows the guidelines as described in the syllabus.

- ➢ The section on RESULTS must include the calculated answer. *Use the table below for your answer*. Note: You will need to replicate the table in your Word file, in order to provide the answer in your report. Points will be taken off if you do not use the table.
- ➢ The code must be provided in the appendix

| | |
|---|---|
| Probability of 500 heads in tossing 1000 fair coins | |
| **Ans.** | ***p*** **=** |

## 4. Getting 4 of a kind

Consider the following experiment:

- o You draw 5 cards from a deck of 52 cards. This is considered a single experiment. If you get 4 0f a kind the experiment is considered a "success".
- o You repeat the experiment N=1,000,000 times, keeping track of the "successes".
- o After the N experiments are completed count the total successes, and calculate the probability of getting 4 of a kind.

**SUBMIT a report** that follows the guidelines as described in the syllabus.

- ➢ The section on RESULTS must include the calculated answer. *Use the table below for your answer*. Note: You will need to replicate the table in your Word file, in order to provide the answer in your report. Points will be taken off if you do not use the table.
- ➢ The code must be provided in the appendix

| | |
|---|---|
| Probability of 4 of a kind | |
| **Ans.** | ***p*** **=** |

## 5. The Password Hacking Problem

Your computer system uses a 4-digit passcode for login. It is easy to calculate that the total number of passwords which can be produced is $n = 10^4$.

- A hacker creates a list of $m = 10^4$ random 4-digit passcode, as candidates for matching the passcode. Note that it is possible that some of the m = $10^4$ numbers may be duplicates.
- You are given your own 4-digit passcode and you are going to check if the hacker's list contains at least one number that matches your passcode. This process of checking is considered one experiment. If a number in the list matches your passcode, the experiment is considered a success. Repeat the experiment for $N$ 1000 times and find the probability that **at least one of the** numbers in the hacker's list will match your passcode.
- If the hacker creates a list of $m = 10^6$ random 4-digit, find the probability that **at least one of the numbers** in the hacker's list will match the passcode.
- Find the **approximate number** $(m)$ of the numbers that must be contained in the hacker's list so that the probability of at least one number matching the passcode is $p = 0.5$. You should do this by trial and error: assume a value for $(m)$ and calculate the corresponding probability as you did in the previous part. The requested answer will be value of $(m)$ that makes this probability **approximately equal** to $p = 0.5$.

**SUBMIT a report** that follows the guidelines as described in the syllabus.

- The section on RESULTS must include the calculated answer. *Use the table below for your answer.* Note: You will need to replicate the table in your Word file, in order to provide the answer in your report. Points will be taken off if you do not use the table.
- The code must be provided in the appendix

.

| | |
|---|---|
| $m =$<br>Prob. that at least one of the numbers matches the passcode | $10^4$<br>***p* =** |
| $m =$<br>Prob. that at least one of the numbers matches the passcode | $10^5$<br>***p* =** |
| $p = 0.5$<br>Approximate number of numbers in the list | ***m* =** |

## 6. References


[1] "*Introduction to Probability*," by H. Pishro-Nik. Available online at: https://www.probabilitycourse.com