## Table of Contents

```
% Rebecca Gartenberg & Sophie Jaro
% Stoch Project 1
% Dungeons and Dragons

clear all; close all; clc;
```

# Question 1

```
% Loop to simulate 1e6 ability scores using the regular method and the
 fun
% method
totalRolls = 1e5;
for i = 1:totalRolls
    regAbilityScore(i,:) = regularMethod;
    funAbilityScore(i,:) = funMethod;
end
```

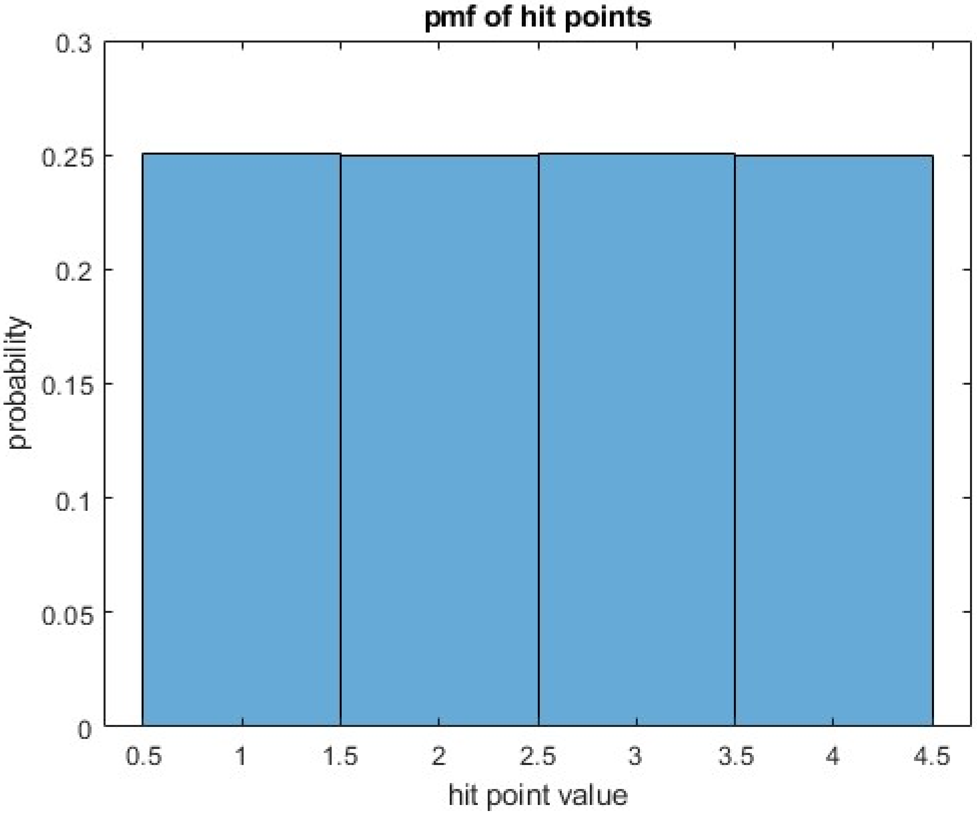
part a Regular method The probability that any one roll of 3 dice generates an ability score of 18 is 1/216.

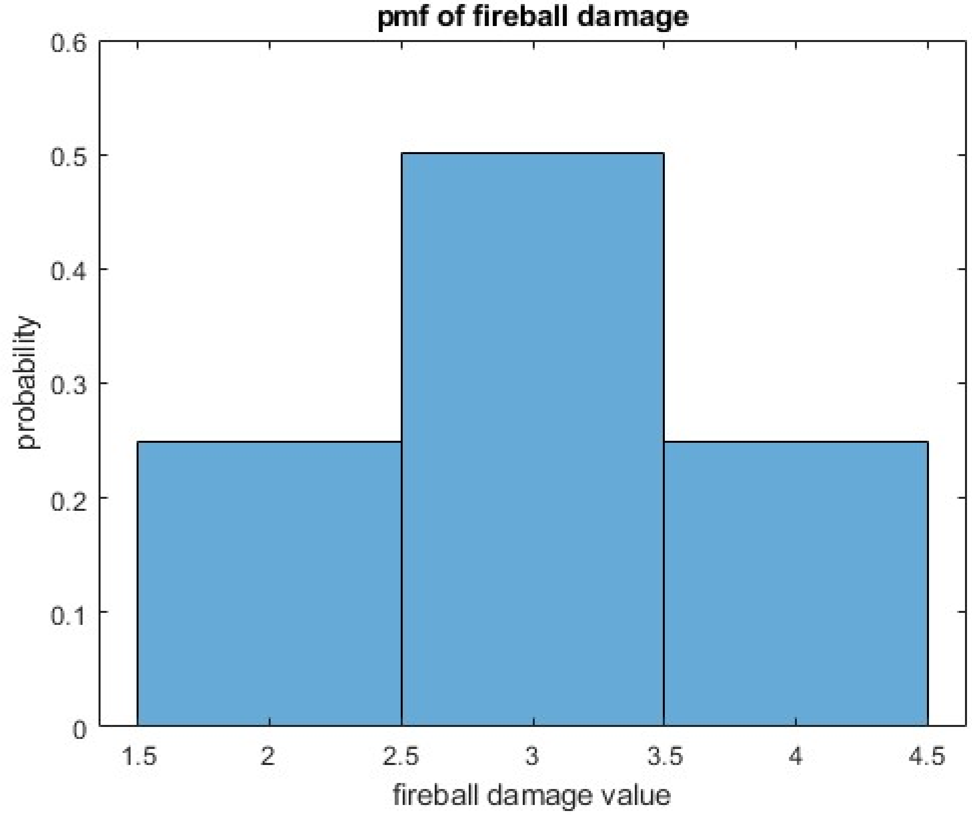
```
oneColumnReg = regAbilityScore(:,1); % This is the rolls for 1 of the
 6 ability scores
eventEighteenReg = length(oneColumnReg(find(oneColumnReg == 18))); %
 This is the number of rolls that equal 18
probEventEighteenReg = eventEighteenReg/totalRolls; % 0.0047
 (approaches 1/216 = 0.0046)

% part b
% Fun method
% The probability of getting an 18 using the fun method is 3 times the
 probabilty of using the regular method.
oneColumnFun = funAbilityScore(:,1); % This is the rolls using the fun
 method for 1 of the 6 ability scores
eventEighteenFun = length(oneColumnFun(find(oneColumnFun == 18))); %
 This is the number of rolls that equal 18
probEventEighteenFun = eventEighteenFun/totalRolls; % 0.0140
 (approaches 3/216 = 0.0139)

% part c
% Perfect character
sumSixScores = sum(funAbilityScore, 2); % The values of all 6 ability
 scores were added to get a complete character score
FF = (sumSixScores == 108); % The highest possible character score is
 18x6 = 108
eventFF = length(find(FF == 1)); % This finds the number of perfect
 characters simulated
```

```
probEventFF = eventFF/totalRolls; % The probabilty of getting a
 perfect character is (1/216).^6. The simulated probability is 0.

% estimating the perfect character by counting total eighteens
 generated per ability
eighteens = (funAbilityScore == 18); % Creating logical array checking
 if ability scores are equal to 18
totalEighteens = sum(eighteens, 1); % Total number of eighteens per
 ability
probEighteens = totalEighteens/totalRolls; % probability of getting an
 eighteen per ability
probAllEighteens = prod(probEighteens'); % probability of getting 6
 eighteens is 6.7832e-12 approaches (1/216).^6 = 9.8464e-15

% part d
% Average character
averageArray = (funAbilityScore == 9); % Creating logical array
 checking if ability scores are equal to 9
sumNines = sum(averageArray, 2); % Find number of nines for each
 character
allNines = length(find(sumNines == 6)); % Finds number of characters
 with all nines
probEventSK = allNines/totalRolls; % The probabilty of getting an
 average character is (25/216).^6. The simulated probability is 0.

% estimating the perfect character by counting total nines generated
 per ability
nines = (funAbilityScore == 9); % Creating logical array checking if
 ability scores are equal to 9
totalNines = sum(nines, 1); % Total number of nines per ability
probNines = totalNines/totalRolls; % probability of getting a nine per
 ability
probAllNines = prod(probNines'); % probability of getting 6 nines is
 1.9574e-09 approaches (25/216).^6 = 2.4039e-06
```

# Question 2

```
% Part a
% This section finds the average number of hit points each troll has
 and the
% average amount of damage the fireball spell does.

trolls = 1e6; % number of trolls
HPvec = randi(4, 1, trolls); % vector of hit points for a given number
 of trolls
avgHP = sum(HPvec)/trolls; % average number of hit points per troll
% Average hit point approaches 2.5

fireballs = 1e6; % number of fireball simulations
FBvec = randi(2, 1, fireballs) + randi(2, 1, fireballs); % vector of
 fireball values
avgFB = sum(FBvec)/fireballs; % average value of a fireball
% Average fireball damage approaches 3
% Fireballs can only do damage of 2, 3 or 4
eventFour = length(find(FBvec == 4)); % This finds the number of times
 fireball damage is 4
pFour = eventFour/fireballs; % Probability that fireball damage is 4
% Probability approaches 0.25

% Part b
% Below are the probability mass functions for the amount of damage
 the
% fireball does and the amount of hit points trolls have.

%pmf for HP
% P[HP = 1] = 0.25
% P[HP = 2] = 0.25
% P[HP = 3] = 0.25
% P[HP = 4] = 0.25

%pmf FB
% p[FB = 2] = 0.25
% p[FB = 3] = 0.50
% p[FB = 4] = 0.25


figure
histogram(HPvec, 'Normalization', 'pdf');
title('pmf of hit points')
xlabel('hit point value')
ylabel('probability')

figure
histogram(FBvec, 'Normalization','pdf');
title('pmf of fireball damage')
xlabel('fireball damage value')
ylabel('probability')
```


pmf of hit points
probability
0
0.05
0.1
0.15
0.2
0.25
0.3
0.5
1
1.5
2
2.5
3
3.5
4
4.5
hit point value


pmf of fireball damage
probability
0
0.1
0.2
0.3
0.4
0.5
0.6
1.5
2
2.5
3
3.5
4
4.5
fireball damage value

Part c The purpose of this section is to estimate the probability that Keene slays 6 trolls at a time.

```
trolls = 6;
trials = 1e6;
k = 1;
remainingHP = [];
for i = 1:trials
    HPvec = randi(4,1,trolls); % creates a vector with hit points for
 6 trolls
    FB = randi(2,1,1) + randi(2,1,1); % creates a fireball to hit the
 6 trolls
    deathVec = HPvec - FB; % the fireball damage is subtracted from
 the hit points
    deaths(i) = length(find(deathVec <= 0)); % the number of deaths is
 counted per row of 6 trolls
    % counting number of times 5 trolls were killed
    if(deaths(i) == 5)
        remainingHP(k) = deathVec(find(deathVec > 0));
        k = k+1;
    end
end

allDead = length(find(deaths == 6)); % the number of trolls all 6
 trolls were killed
pAllDead = allDead/trials; % probability that all trolls are killed
% The probability the fireball kills all 6 trolls is 0.3427

% part d
avgHP = sum(remainingHP)/length(remainingHP);
% Expected amount of hit points remaining is 1.0576.

% Part e
damage = 0;
trials = 1e6;
totalDamage = [];

for i = 1:trials
    damage = 0;
    sword = randi(6,1,1) + randi(6,1,1); % damage pooints generated by
 sword
    hammer = randi(4,1,1); % damage points generated by hammer

    twentyDie1 = randi(20,1,1); % rolling 20 sided die once

    % checking if sword is used
    if(twentyDie1 >= 11)
        damage = damage + sword;
        twentyDie2 = randi(20,1,1);
        % checking if hammer is used
        if(twentyDie2 >= 11)
            damage = damage + hammer;
        end
        totalDamage(i) = damage;
    end
```

```
end

avgDamage = sum(totalDamage)/trials; % the expected damage is 4.1322
 which approaches 4.1250
```

# Functions

```
function abilityScore = regularMethod()
    % The regular method rolls 3 six sided dice and adds the values
 together
    % to obtain an ability score. This function uses the regular
 method to
    % create 6 ability scores.
    rollOne = randi(6, 1, 6);
    rollTwo = randi(6, 1, 6);
    rollThree = randi(6, 1, 6);
    abilityScore = rollOne + rollTwo + rollThree;
end

function abilityScore = funMethod()
    % The fun method rolls 3 six sided dice and adds the values
 together.
    % This process is repeated 3 times and the maximum score is kept
 as an
    % ability score. This function uses the fun method to create 6
 ability
    % scores.
    funRollOne = regularMethod;
    funRollTwo = regularMethod;
    funRollThree = regularMethod;
    maxOneTwo = max(funRollOne, funRollTwo);
    abilityScore = max(maxOneTwo, funRollThree);
end
```

*Published with MATLAB® R2019a*