Results

Mean (10 rolls)	3.1
Standard Deviation	1.9692
Mean (1000 rolls)	3.543
Standard Deviation	1.7208

Table 1

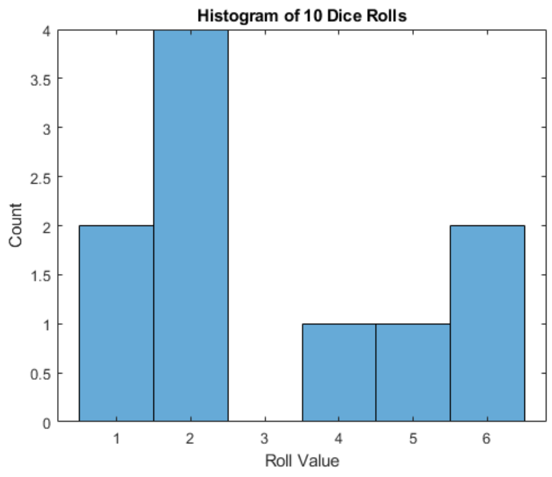

Figure 1: 10 singular dice rolls

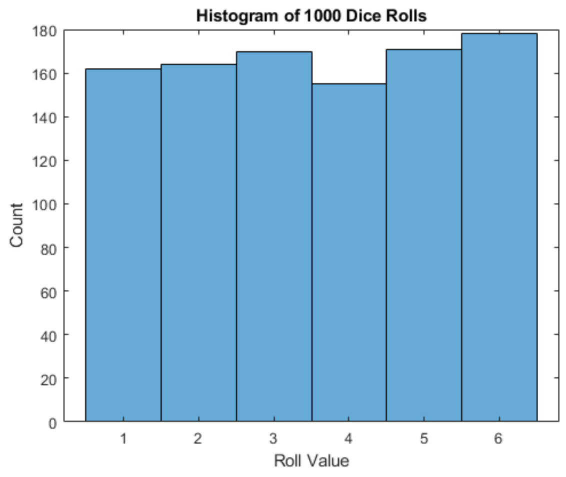

Figure 2: 1000 singular dice rolls

Mean (10 rolls)	6.6
Standard Deviation	2.4129
Mean (1000 rolls)	6.973
Standard Deviation	2.4286

Table 2

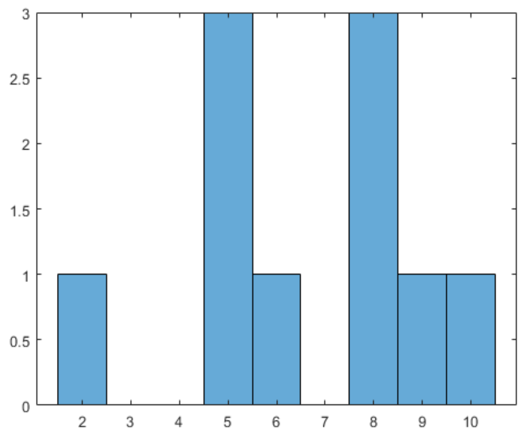

Figure 3: two rolled dice, summed, 10 times

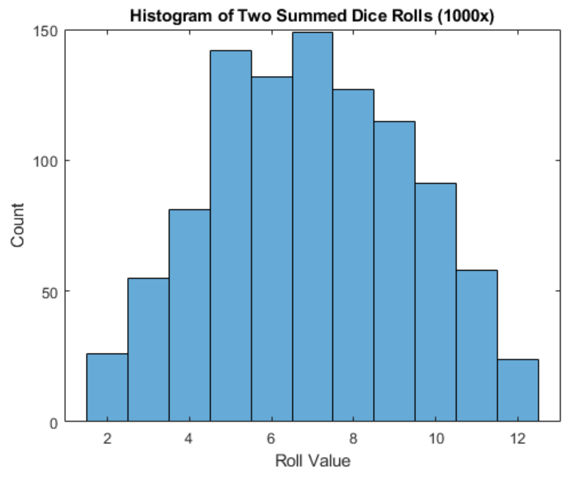

Figure 4: two rolled dice, summed, 1000 times

Discussion

For the one die, a low number of rolls results in a seemingly random probability of 1-6 being rolled. However, as the number of rolls increased by orders of magnitude, the probability approached the expected value of 1/6 (16%). For the two summed dice, the conclusion is roughly the same – when the number of rolls is low, the distribution of sums is random; when the number of rolls is high, the distribution better reflects the probability of two dice summing to a median number (6, 7, 8) than an extreme number (1, 2). This distributed probability is due to 6, 7, and 8 having multiple combinations to sum to them, while the extreme values only have one possible combination.

In the means for both test configuration, we also see an interesting relationship develop. For both configurations, the mean and median are equal – for a singular die the mean and median for all possible values 1-6 is 3.5 and for two summed die, that value is 7. For a low number of rolls, the mean falls below the expected median and mean whereas with magnitudes more rolls, the dataset's mean approaches the expected values.

MATLAB Code (R2020b)

```
% Homework 2 for OCE 2901 Surf Engineering Analysis
% Braidan Duffy
% Due: 01-22-2021

%% Question 1

rolls_10 = roll_die(10);         % Generate array of 10 dice rolls
rolls_1000 = roll_die(1000);     % Generate array of 1000 dice rolls

% Calculate mean for each set of rolls
mean_10 = mean(rolls_10)
mean_1000 = mean(rolls_1000)

% Calculate standard deviation for each set of rolls
stdev_10 = std(rolls_10)
stdev_1000 = std(rolls_1000)

% Generate histograms of results
figure(1)
hist_10 = histogram(rolls_10);
xlabel('Roll Value')
ylabel('Count')
title('Histogram of 10 Dice Rolls')
figure(2)
hist_1000 = histogram(rolls_1000);
xlabel('Roll Value')
ylabel('Count')
title('Histogram of 1000 Dice Rolls')
```

```
%% Question 2

rolls_two_10 = roll_two_dice_sum(10);       % Generate array of 10 dice
rolls
rolls_two_1000 = roll_two_dice_sum(1000);   % Generate array of 1000 dice
rolls

% Calculate mean for each set of rolls
mean_two_10 = mean(rolls_two_10)
mean_two_1000 = mean(rolls_two_1000)

% Calculate standard deviation for each set of rolls
stdev_two_10 = std(rolls_two_10)
stdev_two_1000 = std(rolls_two_1000)

% Generate histograms of results
figure(3)
hist_two_10 = histogram(rolls_two_10);
xlabel('Roll Value')
ylabel('Count')
title('Histogram of Two Summed Dice Rolls (10x)')
figure(4)
hist_two_1000 = histogram(rolls_two_1000);
xlabel('Roll Value')
ylabel('Count')
title('Histogram of Two Summed Dice Rolls (1000x)')


%% Utility

% Rolls a singular dice a specified number of times
% @param itr: number of times dice is rolled
% @return an array of dice rolls
function rolls = roll_die(itr)
    rolls = [];
      for x = 1:itr
        rolls(end+1) = randi(6);
    end
end

% Rolls two dice and sums their values a specified number of times
% @param itr: number of times dice is rolled
% @return an array of two summed dice rolls
function rolls = roll_two_dice_sum(itr)
    rolls = [];
    for x = 1:itr
        rolls(end+1) = randi(6) + randi(6);
    end
end
```