Brandon Castro

816549663

EE 300

April 27, 2016

Probability Project – Discrete Random Variables

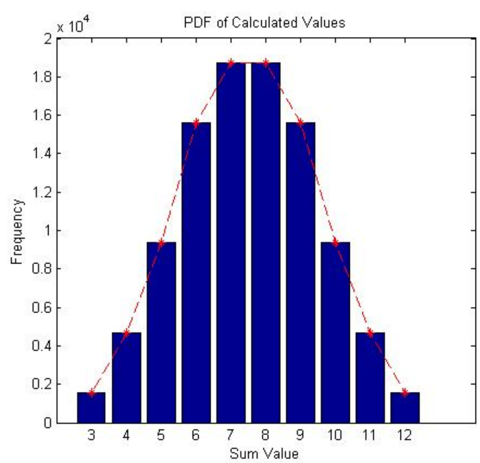
The probability experiment that I chose dealt with throwing three four-sided dice and collecting the total sum of all three. Over the course of thousands of trials, the results of the PDF and CDF graph would prove if calculations and expected values would match the simulated ones. Matlab will be used to simulate the experiment and facilitate with the viewing of the data using plots and histograms.

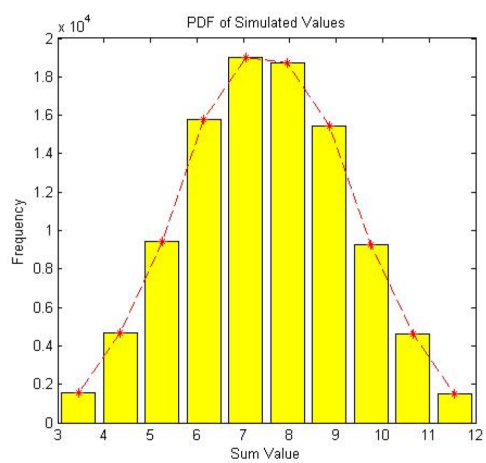
The approximate amount of time I spent on this project was ~3-4 hours. With some previous Matlab experience, most of the code was somewhat similar. However, I still had to read documentation on various Matlab functions from the MathWorks website. Calculating the probability by hand took less than 10 minutes.

To calculate the probability beforehand, the experiment and all its parts had to be analyzed. There are four sides on this dice and three are being thrown at once, with all face values being summed up. The total number of outcomes from three dice is 4x4x4 = $4^3$ = 64. The minimum value the sum could be is 3 (1+1+1) and the maximum is 12 (4+4+4). To calculate the probability of a specific number (for example 5) the total number of possible combinations of the dice to sum 5 must be divided by the total possible outcomes. There are 6 ways to get a sum of 5 from 3 dice (1+1+3, 3+1+1, 1+3+1, 1+2+2, 2+2+1, and 2+1+2), so the probability to get a sum of 5 = P(5) = 6/64 = 0.09375 = 9.375%. Finding all values of possible sums between 3 and 12 resulted in the following data:

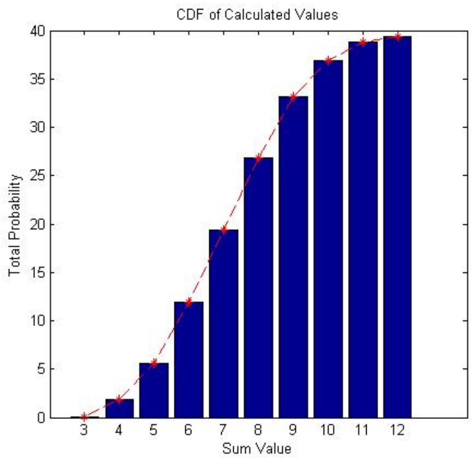
| P(3) | P(4) | P(5) | P(6) | P(7) | P(8) | P(9) | P(10) | P(11) | P(12) |
|---|---|---|---|---|---|---|---|---|---|
| 1/64 | 3/64 | 6/64 | 10/64 | 12/64 | 12/64 | 10/64 | 6/64 | 3/64 | 1/64 |

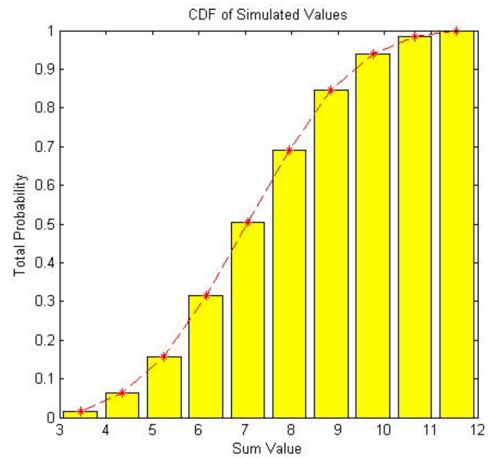
The Matlab simulation proved to be a success. Using an RNG seed of 1738 to repeat the produced results, I simulated the dice rolls over the course of 100,000 trials. To do so I used the randi() function to populate an array of 3 columns by 100,000 rows, one column for each die and a random number between 1 to 4 in each element. To calculate predicted outcomes for each sum value I created another array that multiplied the number of trials by the probability of each number (as seen in the data table above), so if I wanted the expected P(5) it would be 100,000 * 6/64. With these two arrays (calculated and experimental), I proceeded to calculate and graph the probability distribution of each and procured the following graphs:



From analyzing the graphs it was clear that the results were extremely similar. To further prove the simulation with the calculations I derived the cumulative distribution functions for both of the data results and came up with these graphs:




Analyzing the data proved that our knowledge of probability and cumulative distribution functions apply to actual simulations as well. There was an approximate percent error of ~1.28% between the calculated and simulated values. The data correctly matched that of the expected. Both graphs greatly resembled the normal uniform distribution function, meaning that results from random discrete variables can also be approximated with the normal distribution curve, as well as the standard deviation and values of sigma.

In conclusion, the experiment was a greater success than expected. With enough trial runs, the simulated data eventually evens out to that of a normal distribution curve and greatly resembles the calculated outcomes. Both probability and cumulative distribution functions of the simulation came out cleanly and smoothly enough to come to the deduction that calculating PDF and CDF's beforehand can be used to approximate predictions of real results.

**Any external sources used were from MathWorks'* ([http://www.mathworks.com/help/matlab/index.html](http://www.mathworks.com/help/matlab/index.html)) *website and was utilized purely for documentation purposes**

MATLAB Source Code:
--------------------------------------------------------------------------------

```
% Simulates 3 four-sided dice and sums the three values
% Calculates and plots probability/cumulative distribution functions
% of each possible sum
clear all;
close all;

% Using seed and x number of trials
rng(1738);
x = 100000;

% Populate array with x trials of 3 rolls ea
test = randi([1 4],[x 3]);

% Sum of each roll
sums = sum(test, 2);

% Calc PDF of sum value = 3 to 12
pdfx = zeros(10,1);
pdfx(1) = x*(1/64);
pdfx(2) = x*(3/64);
pdfx(3) = x*(6/64);
pdfx(4) = x*(10/64);
pdfx(5) = x*(12/64);
pdfx(6) = x*(12/64);
pdfx(7) = x*(10/64);
pdfx(8) = x*(6/64);
pdfx(9) = x*(3/64);
pdfx(10) = x*(1/64);

cdfx = zeros(10,1);
cdfx(1) = pdfx(1)/x;
for i=2:10
    cdfx(i) = cdfx(i-1) + (pdfx(i)/2500);
end

%% Plotting Calc. PDF Graph %%
figure
subplot(1,2, 1);
bar(pdfx)
hold on
plot(pdfx, '--*r')
hold off
title('PDF of Calculated Values');
xlabel('Sum Value');
ylabel('Frequency');
set(gca,'XTickLabel',{3:12})
%% Plotting Exp. PDF Graph %%

% Tested out histogram, but not all version of Matlab have it
%figure
%histogram(sums)
%title('Simulated PDF of Sum Values')
%xlabel('Sum Value');
```

```
%ylabel('Frequency');

[countsA, binsA] = hist(sums);
cdfA = cumsum(countsA) / sum(countsA);

subplot(1,2, 2);
bar(binsA, countsA,'y');
hold on
plot(binsA, countsA,'--*r')
hold off
title('PDF of Simulated Values')
xlabel('Sum Value');
ylabel('Frequency');
%% Plotting Calc. CDF Graph
figure
subplot(1,2, 1);
bar(cdfx)
hold on
plot(cdfx,'--*r')
hold off
title('CDF of Calculated Values')
xlabel('Sum Value');
ylabel('Total Probability');
set(gca,'XTickLabel',{3:12})
%% Plotting Exp. CDF Graph
subplot(1,2, 2);
bar(binsA, cdfA,'y');
hold on
plot(binsA, cdfA,'--*r');
hold off
title('CDF of Simulated Values');
xlabel('Sum Value');
ylabel('Total Probability');
```