SampleUniformSums.m and SampleBinarySums.m

function X = SampleUniformSqSums(nSamples,nSum)

% function X = SampleUniformSqSums(nSamples,nSum)
% % Returns nSamples i.i.d. random variables, each
% % of which is a sum of nSum i.i.d. squared uniform rv's.

% loop through the number of samples
for n = 1:nSamples

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate one sample from
% a sum of squared uniform rv's
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% initialize the sum to zero
Usqsum = 0;
% add up nSum i.i.d. uniform rv's
for k = 1:nSum
    usq = rand^2; % rand^2 is i.i.d. uniform squared
    Usqsum = Usqsum + usq;
end

% record the nth random variable
X(n) = Usqsum;
end

return

Sum of 50 squared uniform r.v.'s. The above figure was made with the following commands:

X = SampleUniformSqSums(50000,50);
hist(X,40) % if the second argument of hist is a number, then this is number of bins
function X = SampleBinarySums(nSamples,nSum)

% function X = SampleBinarySums(nSamples,nSum)
% Returns nSamples i.i.d. random variables, each
% of which is a sum of nSum i.i.d. binary rv's with p(1) = .3 and p(0) = .7.

% loop through the number of samples
for n = 1:nSamples

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate one sample from
% a sum of binary rv's
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% initialize the sum to zero
Usqsum = 0;
% add up nSum i.i.d. binary rv's
for k = 1:nSum
    u = rand; % u is a uniform r.v.
    % convert u to binary with the right probabilities
    if u < .3
        u = 0; % u will be zero with prob .3
    else
        u = 1; % u will be one with prob .7
    end
    % update the sum
    Usqsum = Usqsum + u;
end
% record the nth random variable
X(n) = Usqsum;
end
return

Sum of 50 binary r.v.'s (p(1)=0.7). The above figure was made with the following commands:
X = SampleBinarySums(100000,50); % this takes a little while to run
hist(X,20:50) % if the second argument of hist is a vector, then this is bin centers