```
%%-------------------------------------------------------------------------
%%Project-2:: Question - 1
%%To Simulate sampling uniformly on interval -3 to 2.Generate Histogram,sample mean &
%%variance. Also compute bootstrap confidence interval for mean and standard deviance

%%The below function runs a for loop for high, mid and low range of samples
%%It calculates the theo mean & variance and compares with observed mean & variance.
%%It also calculates the bootstrap confidence interval for mean and std_deviation
%%Author                 Date                Revision
%%Rajasekar Raja     01/23/17          Initial Revision
%%-------------------------------------------------------------------------
function [ ] = sampling_ques1(num_of_exe)
  %initialize
  Lower_bound = -3;
  Upper_Bound = 2;
  theo_mean = plus(Lower_bound,Upper_Bound)/2;
  theo_var = power(minus(Upper_Bound,Lower_bound),2)/12;
  disp(['Theoritical Mean = ',num2str(theo_mean),':and Variance = ',num2str↙
(theo_var)]);
  %For loop for high, mid and low number of samples to analyse the behaviour
  trails = [1000000,10000,100];
  for iter = 1:num_of_exe
      for expt = 1:3
          no_of_samples = trails(expt);
          %Rand-unifm distributed random numbers in (0,1).Multiplying it with range
          %& shifting it with lower_bound will result in rand No.s between -3 and 2
          u_samples = (Upper_Bound-Lower_bound).*rand(no_of_samples,1) + Lower_bound;
          subplot(3,1,expt);
          histogram(u_samples);
          disp(['Summary of ',num2str(no_of_samples),' random samples ']);
          disp(['  -Calculated Mean = ',num2str(mean(u_samples))]);
          disp(['  -Calculated Variance = ',num2str(var(u_samples))]);
          disp(['  -Calculated Standard Deviation = ',num2str(std(u_samples))]);
          disp('  -Deviation from theoritical to Calculated as follows');
          disp(['       Mean = ',num2str(minus(theo_mean,mean(u_samples)))]);
          disp(['       Variance = ',num2str(minus(theo_var,var(u_samples)))]);
      end
  end
  u_samples = (Upper_Bound-Lower_bound).*rand(10000,1) + Lower_bound;
  ci_mean = bootci(10000,@mean,u_samples);
  disp('Bootstrap confidence interval for sample mean is ')
  disp(ci_mean);
  ci_std = bootci(10000,@std,u_samples);
  disp('Bootstrap confidence interval for sample Std deviance is ')
  disp(ci_std);
```