Assignment 2

Due Date: October 16th, 2022

1. Test for Randomness:

Random number generators (RNG) have wide range of applications in the areas of Computer simulation, Cryptography (generation of OTP, Nonce etc), online gambling (lottery, poker and rummy) and finally computer games. However, while we do not have true random number generators most OS and programming languages provide a library function to generate pseudo random number sequences. The pseudo random number generators mimic true random number generators using mathematical functions. A poorly implemented random number generator significantly weakens the cryptographic protocols provided as part of major crypto libraries such as OpenSSL.

While there is no absolute test to verify the randomness of a random number generator, we often rely on specific statistical tests to ensure that the random sequence generated is sufficiently random enough.

In this assignment your goal is to test the randomness of ***2 of the following 3*** random functions.

1. ***rand*** function provided as part of your operating system.
2. ***rand*** function provided as part of python (uses Mersenne Twister).
3. ***rand*** function provided as part of Java language.

Having chosen a ***uniform*** random number generator (from the above), perform chi-square test and KS tests for randomness using the following process.

1.1 chi-square test

Generate a random sequence of at least 5000 ***integers*** using U_{int} (1, 200). Performance significance using chi squared test as explained in the class for 95% and 99% confidence intervals.

Repeat the test ***13 times*** and in each case choosing a different random seed. Basically initialize an array with 13 integers of your choice and use each number in the array as a seed for your random number generator. In case you have trouble choosing 13 integers you could generate these 13 integers randomly from a random number generator of your choice.

The process for chi squared test is given below:

- $\mathbf{H_0}$: The series is from uniform random generator
- $\mathbf{H_\alpha}$: The series is not from uniform random generator
- Perform **95%** significance test i.e. $\boldsymbol{\alpha} = \mathbf{0.05}$ (2 tail test).
- Set γ(df) = **k-1** and find $\chi^2_{\text{critical }[1-\alpha, \gamma]}$ from table.
- Calculate $\chi^2_{\text{statistic}} = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$
- Accept $\mathbf{H_0}$ if $\chi^2_{\text{statistic}} < \chi^2_{\text{critical}}$

1.2 Reference Material from Class Notes

Kolmogorov-Smirnov Test

- Hypothesis testing is identical to Chi-Square Test.

k	Y_k (Increasing Order)	$F_n(k-1) = (k-1)/N$	$F_0(Y_k)$ (cdf) =val1/(b-a))	$F_n(k)=k/N$	$\lvert F_n(k-1)-F_0(Y_k)\rvert$	$\lvert F_0(Y_k)-F_n(k)\rvert$
1	val1					
2	val2					
3	val3					
4	val4					

- Test statistic is max of ($\lvert F_n(k-1)-F_0(Y_k)\rvert$, $\lvert F_0(Y_k)-F_n(Y_k)\rvert$)
- a, b are the lower and upper bounds of U_{rand}
- N is the number of samples.

K-S Test: Example

k	y_k	$F_n(y_{k-1})$	$F_0(y_k)$	$F_n(y_k)$	$\lvert F_n(y_{k-1}) - F_0(y_k)\rvert$	$\lvert F_0(y_k) - F_n(y_k)\rvert$
1	0.26	0.000	0.130	0.125	0.130	0.005
2	0.33	0.125	0.165	0.250	0.040	0.085
3	0.55	0.250	0.275	0.375	0.025	0.100
4	0.77	0.375	0.385	0.500	0.010	0.115
5	1.18	0.500	0.590	0.625	0.090	0.035
6	1.41	0.625	0.705	0.750	0.080	0.045
7	1.46	0.750	0.730	0.875	0.020	**0.145**
8	1.97	0.875	0.985	1.000	0.090	0.015

- $F_n(y_{k-1}) = (k-1)/N$
- $F_n(y_k) = (k)/N$
- $F_o(y_k) = y_k/(b-a)$ [random source produces nos between 0 and 2]
- N→no.of samples
- (a,b) are the lower and upper bound of rand function

1.3 Kolmogorov-Smirnov Test

Generate a random sequence of 30 ***floating*** numbers using $U_{float}[0, 20]$. Performance significance testing using KS test as explained in the class for 95% and 99% confidence interval. As with chi-squared test repeat your hypothesis testing for 13 runs and in each case choosing a different seed.

The process for hypothesis testing is similar to chi-squared test except for the calculation of test statistic. The test statistic for KS is calculated using the following formula.

$$D = \max_{1 \le i \le N} \left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i) \right)$$

Where,

$F(Y_i)$ represents the cdf for $U_{float}[0, 5]$ and is given by the formula $(Y_i/5)$.

Y_i is the random number produced by the rand function.

NOTE:

- Your program should print chi squared/KS test statistic calculated, table value, confidence interval, your conclusion (*Random* or *Not Random*) for each of the 13 runs.
- Provide details such as the version of kernel and OS used in your tests.