Statistical Computing and Simulation

Spring 2025

Assignment 1, Due March 11/2025

1. (a) We can generate random numbers ranging over (0,1) by using the functions "exp" and "floor", such as *x1=exp(x0)* & *x1=x1-floor(x1)*. Write a computer program using these two functions generate 10,000 random numbers and use the Kolmogorov-Smirnov Goodness-of-fit test to see if the random numbers that you create are uniformly distributed. (Note: You must notify the initial seed number used, and you may adapt 0.05 as the α value.)

   (b) Consider the combination of 3 multiplicative congruential generators, i.e.,

$$u_i = \frac{x_i}{30269} + \frac{y_i}{30307} + \frac{z_i}{30323} \pmod 1$$

   with $x_i = 171 x_{i-1} \pmod{30269}$, $y_i = 172 y_{i-1} \pmod{30307}$, $z_i = 170 z_{i-1} \pmod{30323}$.

   Compare the results in (a) and (b), and discuss your findings.

2. (a) Fibonacci numbers, defined as $X_{n+1} = X_n + X_{n-m}\ (mod\ \ 1)$, is another way of generating random numbers. The usual setting is letting $m = 1$ and see if (X_n) is a sequence of random numbers from *U(0,1)*. However,$x_n < x_{n+1} < x_{n-1}$and$x_{n-1} < x_{n+1} < x_n$never appear under this setting. In general, the performances of Fibonacci numbers would be close to "random" as *m* increases. Write a program to generate Fibonacci numbers and test if they are "good" random numbers given varies choices of *m*. (Note: You could simulate 10,000 random numbers, and use goodness-of-tests & independence tests to evaluate Fibonacci numbers.)

   (b) In class, we often use simulation tools in R, e.g., "sample" or "ceiling(runif)," to generate random numbers from 1 to *k*, where *k* is a natural number. Using graphical tools (such as histogram) and statistical tests to check which one is a better tool in producing uniform numbers between 1 and *k*. (Hint: You may check if the size of *k* matters by, for example, assigning k a small or big value.)

3. There are several ways for checking the goodness-of-fit for empirical data. In specific, there are a lot of normality tests available in R. Generate a random

sample of size 10, 50, and 100 from N(0,1) and t-distribution (with degrees 10 and 20) in R. You may treat testing random numbers from t-distribution as the power. For a level of significance $\alpha = 0.05$ test, choose at least four normality tests in R ("nortest" module) to check if this sample is from N(0,1). Tests used can include the Kolmogorov-Smirnov test and the Cramer-von Mises test. Note that you need to compare the differences among the tests you choose.

4. Write your own R programs to perform Gap test, Permutation test, and run test. Then use this program to test if the uniform random numbers generated from Minitab (or SAS, SPSS, Excel) and R are independent.
5. (a) Use the search engine to download the first one million digits of pi (for example, https://www.piday.org/million/), and check via graphic tools if the numbers violate the assumption of random numbers.

   (b) Apply the appropriate tools to test if the random numbers from (a) satisfy the assumption of random numbers.
6. $(\sum_{i=1}^{12} U_i - 6)$ can be used to approximate N(0,1) distribution, where U_i's are random sample from U(0,1).
   (a) Based on $\alpha = 0.05$, compare the results of the Chi-square test and the Kolmogorov-Smirnov test, and see if there are any differences.
   (b) Design two tests of independence (which are not the same as you saw in class) and apply them on the random sample that you generate.