Statistical Computing and Simulation

Spring 2023

Assignment 1, Due March 10/2023

1. (a) Use "*scan*" to read the data file "*Graduate_Earnings*" from web page ***The Data and Story Library*** (https://dasl.datadescription.com/) that contains a mixture of character and numeric data. Using the "data.frame" to input the data and then output the data to a text file and an Excel file. (Hint: You can use the function "is.na" to handle the "NA" observations.)

   (b) Draw a boxplot to describe the variables of "Earning" and "Price" for public universities and private universities.

   (c) It is believed that the universities with higher tuition (or "Price") have higher salaries ("Earning"). Check whether this result!

   (d) Draw a scatter plot each for public universities and private universities, using "Price" as x-axis and "Earning" as y-axis. Also, use the function "identify" to mark unusual observations.

2. The greatest common divisor of two numbers can be computed via: (Verify!)

   ```
   gcd = function(a,b)
   {   if (b==0) a else gcd(b,a%%b)   }
   ```

   Use a similar idea of the function "*gcd*" to create a function "*lcm*" for computing the least common multiplier of two numbers.
   (Bonus: Modify these functions to more than two numbers.)

3. (a) Use the commands "rep" and "seq" to create the vector:

   0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4

   (b) Similar to (a), create the following vector:

   1 2 3 4 5 2 3 4 5 6 3 4 5 6 7 4 5 6 7 8 5 6 7 8 9

   (c) Use "rep" and "seq" to create the following vector:

   red, yellow, blue, yellow, blue, green
   blue, green, magenta, green, magenta, cyan

4. (a) Write a computer program using the Mid-Square Method using 6 digits to generate 10,000 random numbers ranging over [0, 999999]. 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. Also, you may find warning messages for conducting the Goodness-of-fit test, and comment on the Goodness-of-fit test.)

(b) Similar to the above, but consider $X_{i+1} = 69{,}069 X_i \pmod{2^{32}}$, i.e., the generator used by Vax before 1993. Use both the χ^2 and Kolmogorov-Smirnov Goodness-of-fit tests to check if the data are from U(0,1) distribution. Compare the result with those in (a) and (b), and discuss your findings.

5. (a) 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 and big value.)

   (b) Hand calculators often use $U_{n+1} = (\pi + U_n)^5 \pmod{1}$ to generate random numbers between 0 and 1. Compare the result with those in #5, and discuss your finding based on the comparison.

   (c) In addition to $U_{n+1} = (\pi + U_n)^5 \pmod{1}$, we can also use $\phi = \frac{1+\sqrt{5}}{2}$ (the golden ratio) or other irrational numbers to replace the value of π, to generate random numbers between 0 and 1. Using graphical tools (such as histogram) and statistical tests to check if π or ϕ has a better performance in producing uniform numbers between 0 and 1.

6. Fibonacci numbers, defined as $X_{n+1} = X_n + X_{n-m} \pmod{1}$, is another way of generating random numbers. The usual setting is letting $m = 1$ and see if $(X_n)'s$ are 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.)