Homework Assignment 5: Applied Probabilistic Models

Aspects of Normal Distribution and Pseudo-random Numbers

5273

1 Introduction

n this work, it is studied the generation of pseudo-random numbers by different algorithms. Also, several experiments have been used to generate normal distributions from those generation methods. The algorithm's sensitivity is also tested changing parameters, and variables that define the different forms of generations are also part of the study.

For the analysis, it is used the R software in its version 4.0.2 [2], and the code used is available on the GitHub repository [3]. This work is run on a MacBook Air with an Intel Core i5 CPU @ 1.8 GHz and 8 GB RAM.

2 Algorithms

It is implemented the Box-Muller transform [1], which from two numbers uniformly distributed, generates normally distributed ones. Those numbers should be independent, but some tests are performed with several dependencies to see this algorithm's behavior. For this procedure, the following code is used, which creates a function that returns a pair of values resulting from applying the corresponding equations.

```r
gaussianNoise <- function (mu, sigma) { #Box-Muller Transform
  u <- runif(2)
  z0 <- sqrt(-2*log(u[1])) * cos(2*pi*u[2])
  z1 <- sqrt(-2*log(u[1])) * sin(2*pi*u[2])
  pair <- c(z0,z1)
  return (sigma * pair + mu) #Return a pair (zo,z1)
}
```

code/a5.R

As can be seen, it takes two uniformly distributed random numbers using the function `runif` from R as a source, and it generates a pair of independent, normally distributed pseudo-random numbers.

On the other hand, it is analyzed the Linear Congruential Generator (LCG) [5], which is an algorithm that generates a sequence of pseudo-randomized numbers. This method is computed with the following code. This algorithm is defined by the seed or initial value (X_0), a modulus (m), a multiplier (a), and an increment (c) .

Table 1: Result of one experiment of the Shapiro test changing values of μ and σ.

	μ	σ	p-value
Test 1	400	20	0.548
Test 2	225	7	0.3549
Test 3	3	0.25	0.0860
Test 4	-23	3	0.5139
Test 5	-100	12	0.2405

```r
linearCongruentialGen <- function (n, seed) {
  a <- 11551
  c <- 27077
  m <- 39709
  x <- seed
  gen_data <- numeric()
  while (length(gen_data)<n){
    x <- (a * x + c) %% m
    gen_data <- c(gen_data,x)
  }
  return (gen_data/(m-1)) #Return a seed generation pseudo-random numbers
}
```

code/a5.R

3 Experiments

Experiments for this work are based on the generation of Normal distributions from different methods. Several variations are studied to test the sensitivity of these algorithms as well. The parameters are the number of repetitions or sample size (n) with a value of 1000, then a mean (μ) of 10, a standard deviation (σ) of 2, and a seed of 27. Those values are arbitrarily selected at first and will be changed for further sensitivity analysis.

3.1 Box-Muller Transform

The first experiment is performed with the Box-Muller transform. It is constructed a generation of pseudo-random numbers using the elements generated by the variable Z_0. The same procedure is executed with the variable Z_1, and both distributions are compared with the generated using the `rnorm` function of R. This comparison is shown in Figure 1.

To perform a sensibility analysis for this algorithm, the parameters μ and σ were changed, and Shapiro Test is performed 30 times for each combination. An example of the results of one experiment is shown in Table 1. As a result of the test, it is important to highlight that it passed in all repetitions for every test, except for Test 3, in which the algorithm had a p-value less than 0.05 in two times.

In addition, it is generated a boxplot of these distributions (see Figure 2), and an analysis of variance (ANOVA) of one way is performed. In the ANOVA test, the null hypothesis (H_0) implies that there is not enough evidence to prove the mean of the group is different from another. And the alternative hypothesis (H_1) that at least, the mean of one group is different. In this case, the p-value is greater than 0.05, so there is no evidence to reject the null hypothesis, and therefore it is concluded that the means are identical.

`data.txt`

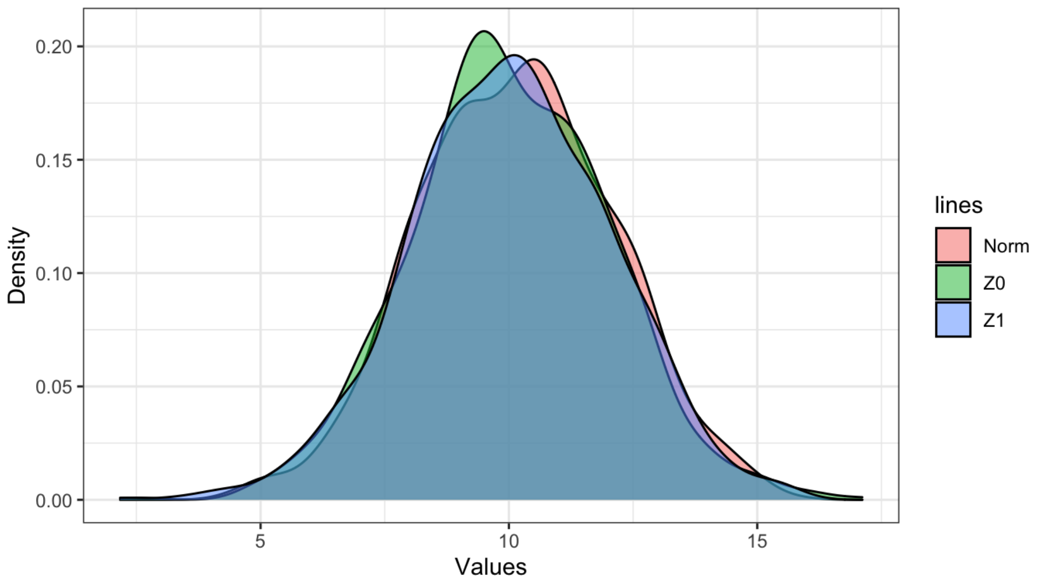

Figure 1: Density Plot of Normal Distributions

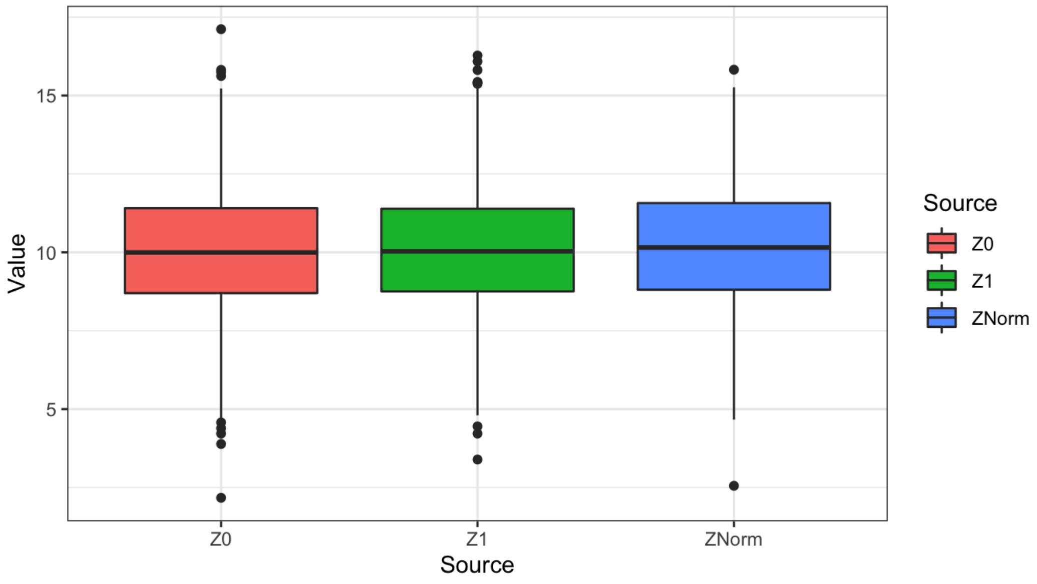

Figure 2: Box Plot of Normal Distributions

Table 2: Results of Shapiro Test with different relations.

Dependency	p-value
$u_2 > u_1$	2.667e-13
$u_2 = u_1$-`runif(1)`	0.3877
$u_2 =$ `runif(1,1,10)`-$2u_1$	0.6331
$u_2 =$ (`runif(1)`-u_1)$/u_1$	0.0186
$u_2 = 2u_1$	4.506e-16
$u_2 = u_1$*`runif(1)`	2.351e-13

```
            Df Sum Sq Mean Sq F value Pr>F
Source       2      8   3.875   0.984  0.374
Residuals 2997  11802   3.938
```

3.1.1 Testing Independency of the uniform pseudo-random generated numbers

A sensitivity analysis changing different relations between the uniformed pseudo-random numbers u_1 and u_2 from the Box-Muller transform is performed. Changes in which the algorithm is subject are defined by how the uniform pseudo-random numbers are generated. Then the transform is performed and repeated 1000 times, and a Shapiro-Test is performed to test the normality of these different generations. These changes are detailed in Table 2. In those cases, the algorithm only kept generating normal distributions if the relation is the difference between u_1 and u_2, and when it is generated a number u_2 between 1 and 10 (it can be considered as a bigger number), and it is subtracted with a multiple of u_1.

3.2 Linear Congruential Generator

Experiments with the LCG are performed based on the uniformed pseudo-random sequence generate with this algorithm. It is tested the distribution of the data with the Chi-squared test provided by the `uniform.test()` function over the histogram of the data. For this test, the p-value is greater than 0.05, so there is no evidence to reject the null hypothesis, and the data might be uniformly distributed.

data.txt

```

        Chi-squared test for given probabilities

data:  hist.output$counts
X-squared = 14.48, df = 9, p-value = 0.1062
```

3.2.1 Using LCG in the Box-Muller Transform

After concluding that data generated are uniformly distributed, it could be used to test these values in the Box-Muller transform as the values u_1 and u_2. The results of the histogram of this experiment are shown in Figure 3. It can be seen with a higher probability the normality of the data using the Box-Muller transform with those values. Shapiro Test throws a p-value of 0.4087, so it can be concluded

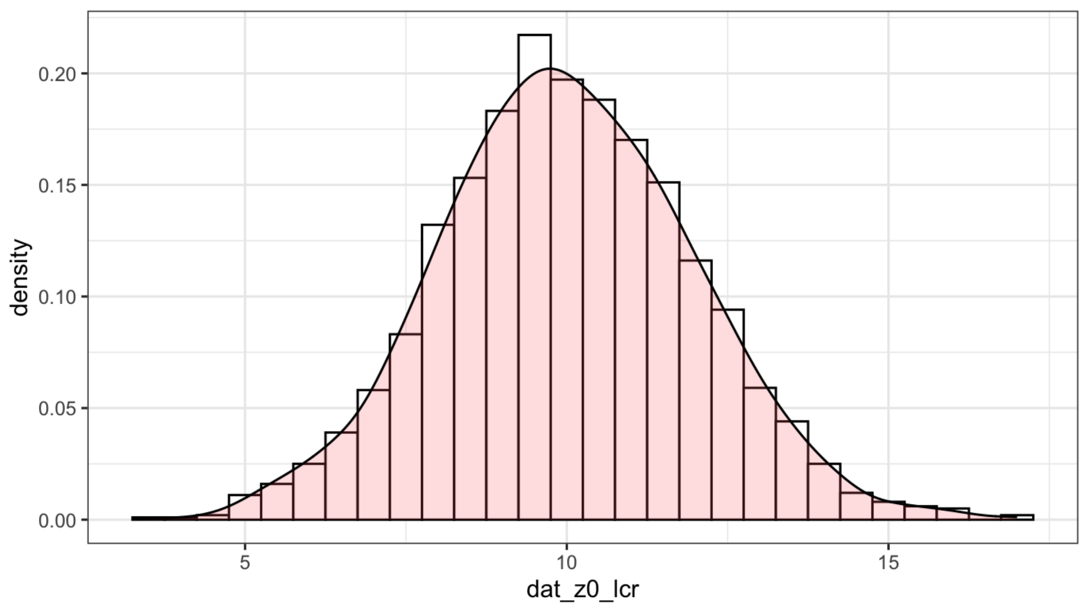

Figure 3: Histogram of the Box-Muller Transform using the LCG

Table 3: Results of the Shapiro Test changing the values a, c, and m.

	a	c	m	p-value
Test 1	3067	4751	7919	0.2793
Test 2	1993	2293	3001	0.2514
Test 3	1049	1459	1931	0.9003
Test 4	463	701	1033	0.0091
Test 5	229	347	461	2.2e-16

data may proceed from a normal distribution.

Other experiments are performed, for example, changing the values of a, c, and m, resulting in the loss of normality when those values decrease with the same seed. This tests are shown in Table 3.

Table 4 also show experiments of different values of a, c, and m, but this time making them as non-prime values. Results show that with these numbers, it is susceptible to applying them into the gaussian algorithm because p-values show in multiple tests that the generated sequence is not normally distributed.

3.3 A nonlinear Generator

Non-linear generators can be defined for example by using a nonlinear transition function f, or a nonlinear output function g, or by combining two or more linear random linear generation, etc. In Knuth [4] it is analyzed a generator based on a quadratic recurrence of the form:

$$x_i = (ax_{i-1}^2 + bx_{i-1} + c) \bmod m$$

For this generator the author gave conditions for a maximal period of m. If $m = 2^e$, then the period is maximal if and only if a is even, $(b - a - 1) \bmod 4 = 0$, and c is odd. The code used for this

Table 4: Results of the Shapiro test with non prime values of a, c, and m.

	a	c	m	p-value
Test 1	11000	27070	39720	0.06912
Test 2	11343	20505	35100	2.2e-16
Test 3	15250	35400	65104	4.966e-05
Test 4	5640	10240	15800	1.829e-09
Test 5	2505	5005	15145	4.13e-16

method is shown below and is adapted to return values in the interval (0,1). It is also tested with the `uniform.test` function or R, with $p - value$ of 0.1459, evidencing uniformity in data.

```
nonlinearGen <- function (n, seed) {
  a <- 20  #an even number
  b <- 41
  c <- 15 #an odd number
  m <- 2^exp(1)
  x <- seed
  gen_data <- numeric()
  while (length(gen_data)<n){
    x <- (a * x^2 + b * x + c) %% m
    gen_data <- c(gen_data,x)
  }
  return (gen_data/(m-1))
}
```

code/knuth.R

References

[1] George EP Box. A note on the generation of random normal deviates. *Ann. Math. Stat.*, 29: 610–611, 1958.

[2] The R Foundation. The R Project for Statistical Computing. `https://www.r-project.org/`, 2020.

[3] Oscar Hernandez. Probability in R. `https://github.com/oscaralejandro1907/probability-in-R/blob/master/assignment1/t1.R`, 2020.

[4] Donald E Knuth. *Art of computer programming, volume 2: Seminumerical algorithms.* Addison-Wesley Professional, 2014.

[5] Pierre l'Ecuyer. History of uniform random number generation. In *Winter Simulation Conference (WSC)*, pages 202–230. IEEE, 2017.