University of Leeds

MATH5004M Assignment in Mathematics

The Testing of Random Number Generators

Author:
Paul Burgoine 200469166

Supervisor:
Dr. Jochen Voss

May 3, 2013

Project Overview

This project is focused on testing Random Number Generators. I will outline five different methods and will apply these tests to five Random Number Generators introduced later in this chapter. Each test I perform is done using functions that have been written by myself in the Statistical Package R. Although in some cases an R function already exists for the tests described in this report, I decided to write and use my own functions to illustrate how each method works more clearly.

Initially, I begin by introducing a common example of Pseudo Random Number Generator; the Linear Congruential Generator. This type of Pseudo Random Number Generator will form the basis for most examples contained within this report. In the first chapter of this report, Tests for Distribution, I introduce the Chi-square test and explain its use within Random Number Generator scenario, and then test several Linear Congruential Generators for distribution. I also introduce the Kolmogorov-Smirnov test, and explain two different ways in which this test can be applied to Pseudo Random Number Generators.

Within the next section, Tests for Independence, I explain a Correlation Coefficient known as Spearman's Rank Correlation Coefficient, and provide a proof that both of its forms are equivalent. I also calculate this Correlation Coefficient for several Pseudo Random Number Generators and interpret the Coefficient. I also explain a test that can be applied to Pseudo Random Number Generators to test for Independence known as the Runs Test, and apply two different versions of the test to Pseudo Random Number Generators.

Finally, I look at a method which was used to prove that a famous and well used Pseudo Random Number Generator known as RANDU was in fact a very poor generator of random numbers, known as Invariant Coordinate Selection. I explain how to apply this method to Pseudo Random Number Generators and use it to test several. I also provide a conclusion as to how each of the Pseudo Random Number Generators that have been assessed in this report have performed.

Contents

Random Number Generators

0.1 Introduction

Before we start, it is necessary to explain that this report does not cover anything about True Random Number Generators. True Random Number Generators are use physical phenomenon to generate numbers, which requires sophisticated equipment and often is very time consuming. Pseudo Random Number Generators, the class of generators explored within this report, are not truly random but deterministic in nature, however attempt to mimic behaviour of true randomness. They do this at varying levels of success, and in this report I intend to distinguish between several Pseudo Random Number Generators using various different tests. There exist a great many different tests that can be used to assess the quality of Random Number Generators and it is simply the sheer number of tests that a Pseudo Random Number Generator can pass that determines the quality of a Pseudo Random Number Generator. I will start by defining several abbreviations that will be used throughout this report.

Definition 1. Pseudo Random Number Generator (PRNG) : A mathematical formula that outputs values that appear to be independent and usually distributed continuously and uniformly on the interval $[0, 1]$ [UC08] It is for that reason that they may be used in place of true randomness.

Definition 2. Linear Congruential Generator (LCG) : A specific type of Pseudo Random Number Generator, all of which adhere to a particular formula. The formula is presented in the next section.

0.2 Linear Congruential Generators (LCGs)

The linear congruential generator is a very simple example of a Pseudo Random Number Generator. Its simplicity makes it excellent to showcase how a PRNG works, although its practicality is somewhat lacking due to its ineffectiveness of actually producing a good "random" output and the availability of more sophisticated generators that produce a far more "random" output. A generator that is considered far better than a linear congruential generator is given at the end of this chapter. The linear congruential generator works by taking an initial value, multiplying it by a number and adding another number to it, then reducing modular another value. A general form of the LCG is shown below, as described on pg 9 of [Knu69].

$$X_n = (aX_{n-1} + c) \bmod m$$

Where

$m > 1$ is the modulus

$a \in \{1, 2, \ldots, m-1\}$ is the multiplier

$c \in \{0, 1, \ldots, m-1\}$ is the constant

$X_0 \in \{0, 1, \ldots, m-1\}$ is the initial value or seed

It is the initial value, or seed that gives an LCG (and other PRNGs) its 'randomness'. When not stated by the user, this value is determined by something volatile such as the time or the number of keystrokes since the computer was switched on (if the LCG is on a computer, of course). We first consider some very simple LCGs to understand the formula better.

Example 0.1. Take $m = 8, a = 5, c = 3$, and $X_0 = 7$, thus we construct the linear congruential generator as;

$$X_n = (5X_{n-1} + 3) \bmod 8$$

Which when starting at seed $X_0 = 7$ gives an output of; $6, 1, 0, 3, 2, 5, 4, 7, 6, 1, 0, 3, \ldots$

The output initially looks random, however an LCG has a set period, in this case 8, after which it just loops indefinitely, this is due to the fact that an LCG is actually deterministic and not random. The value X_{n+1} is determined from value X_n. Thus from this we can deduce if we wish to use a linear congruential generator to generate random numbers, then the period must be larger than the number of random numbers n we require. Care must also be taken when choosing the parameter values for a linear congruential generator, as some have period length far less than the maximal value m the modulus.

Example 0.2. Take $m = 12, a = 9, c = 3$, and $X_0 = 0$, thus we construct the linear congruential generator as

$$X_n = (9X_{n-1} + 3) \bmod 12$$

Which when starting at seed $X_0 = 0$ gives an output of; $3, 6, 9, 0, 3, 6, 9, 0, 3, 6, 9, 0, \ldots$

Almost immediately it is apparent that the output does not look random. This LCG has a period of only 4 before repetition so is far less useful than the previous example. For an LCG with a maximum period to be achieved there are several rules that need to be followed. These were followed when the author was constructing the first example, hence a maximal period was achieved. The latter example breaks these rules and thus a maximal period is not attained.

0.2.1 Period Length

Although it may first seem like the period length of an LCG will only depend upon the modulus, this is not the case. Many other variables play a role in the period length, and to obtain a maximal period, several conditions must be satisfied. For a LCG to attain a maximal period the following three conditions must be followed as prescribed on pg 15 of [Knu69]

1. The modulus m, and the constant c are coprime, ie the only integer divisors they share are 1

2. $a - 1$ is divisible by all the prime factors of m. Commonly, this usually means picking m and $a - 1$ as some multiples of 2, as then both will only divide the prime number 2

3. $a - 1$ is some multiple of 4, if m is some multiple of 4

It is now possible to see why the second LCG did not attain a period length of 12. Statement 1 was broken as 12 and 3 are not coprime, and condition 2 was also violated as 8 does not divide all prime factors of 12; 8 divides 2 but not 3. If a maximal period is attained then the seed X_0 has no effect on period length as a maximal period cycles through the full set of possible values the generator can take. In contrast, when these statements are not followed and a maximal period is not attained choice of X_0 may effect the period length, as the LCG may have several different cycles with different values for different X_0.

If a maximum period is attained then the output of the LCG mimics the behaviour of a set of independent and identically distributed random variables uniform on the set $\{0, 1, \ldots, m-1\}$. Typically, LCGs are more useful when uniformly distributed on the interval $[0, 1]$, this way the quality of output is comparable with other LCGs with different set size and other random number generators in general. This is done by dividing the output of a LCG by m. It also means the output can easily be transformed to distributions on any interval or completely different distributions; for example if a set of random numbers from a standard normal distribution are required.

We define the output of a LCG using the following notation

$X_1, X_2, X_3, \ldots, (X_n)$ = The output values of a LCG

$U_1, U_2, U_3, \ldots, (U_n)$ = The output values of a LCG transformed onto the interval [0,1]

0.2.2 LCGs in R

The previous two examples were done by hand and were not that time consuming; however as the period of the LCG becomes larger due to user of the generator requiring say thousands of random values, completing a LCG by hand will not suffice. We therefore require a computer to produce thousands of values, quickly. The R function below has been written so that far larger LCGs can be quickly calculated.

```
LCG <- function(n,m,a,c,X0) {
  X <- c()
  Xn <- X0
  for(i in 1:n){
    Xn <- (a*Xn + c) %% m
    X[i]  <-  Xn
  }
  X <- X/m
  return(X)
}
```

The first line of the R code defines LCG() to be a function that requires inputs n, m, a, c, X_0 which are parameters previously defined at the beginning of this chapter. An empty vector is then labeled as X and Xn is replaced by the value X_0. The LCG algorithm is then ran in a loop for values 1 to n, initially using X_0 and then $X_1, X_2, \ldots$. Each time the algorithm is ran, the value calculated is assigned to the numerical position it was calculated, in the vector X. This loop is then ended at n, and then each value in the vector is divided by m. Finally the set of values are then returned to the user, and the function is ended.

0.2.3 Examples of LCGs

This section provides examples of five Linear Congruential Generators that will be referred to throughout this report. In each section, a test that can be used to test PRNGs will be explained and later applied to the output of each of these LCGs. Then at the end of the report, the results for all of the tests will be analysed and the decisions to 'accept' or 'reject' the LCGs tallied up over all tests to gauge which LCG is deemed the best.

Example 0.3. We define LCG1 to be

$$LCG1 : X_n = (13X_{n-1} + 3) \bmod 1024, \tag{1}$$
$$X_0 = 0, n = 0, \ldots, 400.$$

This is an example of an LCG that does not break any rules for maximal period and thus attains a maximum period. It uses very small numbers and is but a toy example.

Example 0.4. We define LCG2 to be

$$LCG2 : X_n = (13X_{n-1} + 3) \bmod 1025, \tag{2}$$

$$X_0 = 0, n = 0, \ldots, 400.$$

Here maximal period has not beed achieved purposefully, rule 2 is violated from the maximal period attainment conditions in the section on LCGs.

Example 0.5. We define LCG3 to be

$$LCG3 : X_n = (13X_{n-1} + 3) \bmod 1024, \tag{3}$$

$$X_0 = 0, n = 0, \ldots, 2000.$$

This LCG is the exact same LCG as in example 1; even with the same initial seed, however this LCG runs for 2000 values; meaning once it reaches the end of its period (up to 1024) it will simply repeat again from the first value it took. When using a LCG to generate random numbers it is essential that the period length is larger than the number of random values required to avoid this repetition.

Example 0.6. We define LCGNR to be

$$LCGNR : X_n = (1664525X_{n-1} + 1013904223) \bmod 2^{32} \tag{4}$$

$$X_0 = 0, n = 0, \ldots, 2000.$$

This LCG is used in Numerical Recipes, a book on algorithms, and as such one would assume this is a good LCG. There are many other examples of good LCGs, however one one is used here as they all pass statistical tests easily and is the reason they are used in computer packages and programs.

Example 0.7. We define RANDU to be

$$RANDU : X_n = (65539X_{n-1}) \bmod 2^{31} \tag{5}$$

$$X_0 = 1, n = 0, \ldots, 2000.$$

RANDU is a rather famous LCG, and its use was widespread until it a discovery was made which highlighted that its output was actually rather poor and could not be used to simulate randomness. Although it is the case that the seed of a LCG may be user chosen, it is a requirement of RANDU that the seed, $X_0 =$ odd.

0.2.4 Other Pseudo Random Number Generators

There are also countless other PRNGs that differ tremendously from LCGs. One such example is the Mersenne Twister Algorithm, which is far more sophisticated than any LCG. It's output is also considerably more random than that of LCGs, and thus its inclusion here would be rather redundant as would pass all tests without fail. An excellent paper covering the Mersenne Twister Algorithm is by Matsumoto and Nishimura entitled 'Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator' [MN98].

Chapter 1

Tests for Distribution

In this chapter I explain two tests for distribution that can be applied to Pseudo Random Number Generators, the Chi-square test and the Kolmogorov Smirnov test. I then implement these tests to the five LCGs outlined in the introductory chapter, and analyse the results.

1.1 The Chi-Squared Test

1.1.1 Description of the Test

The Chi-square test is one of the best-known statistical hypothesis tests and appears in many different forms; many testing a different hypothesis. A test is described as Chi-square if the resulting test statistic obtained is compared to a Chi squared distribution with ν degrees of freedom; usually using a Chi-squared distribution table or using the Chi-square distribution R function at a chosen significance level. Two main forms of the test are the Chi-square "goodness of fit" test and the Chi-square test for independence. The "goodness of fit" test tests a hypothesis that some observed data comes from a theoretical probability distribution. The independence test tests the hypothesis that paired observations with two categories are independent; data is usually expressed in a contingency table. For testing PRNGs the former Chi-square test will be utilised. The testing method can simply be applied to the observations (the output) and expectations (calculated using probabilities from a proposed density) of a PRNG to yield a result, however the Chi-square "goodness of fit" test also form an integral part of many of the more complex empirical tests that utilise an altered version of the output of a PRNG before applying a statistical test such as a Chi-square test.

When hypothesis testing in relation to "goodness of fit", we are testing a null hypothesis that the sample being tested follows a specified distribution. This null hypothesis usually represents the notion of no significant difference between the proposed distribution and distribution of the sample under scrutiny. An alternative hypothesis is also defined as there being significant difference between the proposed distribution and sample distribution. The significance is user prescribed, and thus one may choose to look at significance at multiple levels. Formally, when considering the output of a PRNG the hypotheses may be defined in the following way

$H_0 : P(X = i) = p_i$ for $i = 1, \ldots, k$.

$H_1 : X$ has a different distribution.

Almost universally, including pg 37 of [Knu69] the test statistic appears in the following form;

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} = \sum_{i=1}^{n} \frac{(O_i - np_i)^2}{np_i}$$

Where

O_i = the observed frequency of event i

$E_i = np_i$ = the expected frequency of event i under the model proposed in the null hypothesis

In our case the expected count $E_i = np_i$, means that the ith expected count is the presumed probability of i occurring multiplied by the total count of events and the observed count O_i is the actual amount of times i occurred. When calculated, this value is then compared to values in a Chi-square distribution table. Chi-square distribution tables use degrees of freedom as a parameter to identify which particular Chi-square distribution will be compared to the value. Degrees of freedom is calculated as the number of parameters minus the number of constraints on the parameters; ie the number of free parameters. In our case there are $k - 1$ degrees of freedom, as there are $p_1, p_2, \ldots, p_k$ parameters which is overall k parameters, and there is one constraint on those parameters that $p_1 + p_2 + \cdots + p_k = 1$.

Assumptions

The Chi-square test requires that the expected group value $np_i > 5$ for all i groups as stated on pg 38 of [Knu69], if this is not the case then groups should be merged so that the new groups satisfy $np_i > 5$. This must be performed in a way that the categories still make sense. For example some of the larger values from a sample could be merged together and represent all values larger than a number. In the case of PRNG's, a computer will be running PRNG algorithms with very high n, the expected values for each group will be high enough that this pre-requisite need not be worried about, and in a case that $np_i < 5$, then the algorithm can be re-run with larger n.

Chi-square "goodness of fit" tests also require independence between identically distributed random variables as stated on pg 1 of [GN96]. Although this assumption is broken when testing deterministic PRNGs, this is not an issue. If a PRNG is sufficient it can be used in place of an actual set of independent and identically distributed random variables due to its behaviour being sufficiently similar.

1.1.2 Derivation of the Chi-square distribution from the test statistic

In this section I aim to show that the test statistic of the χ^2-test is approximately $\chi^2_{(r-1)}$-distributed for large n. To do so, I first take some results from 'A Guide to Chi-Squared Testing' [GN96]. Although a proof is given in this book, I have greatly expanded upon what is given in the book.

Let $\boldsymbol{\nu}$ be a vector of frequencies, $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_r)^T$. Then the distribution of $\boldsymbol{\nu}$ is a multinomial with parameters n and $\mathbf{p}$, where $\mathbf{p}$ is a vector of probabilities, taken from pg 2 of [GN96].

Further, as stated on page 3 [GN96], we have the following properties that

$$\mathbb{E}(\boldsymbol{\nu}) = n\mathbf{p} \in \mathbb{R}^r, \tag{1.1}$$

and

$$\mathrm{Cov}(\boldsymbol{\nu}) = n(D - \mathbf{p}\mathbf{p}^T) = n\Sigma \in \mathbb{R}^{r \times r}, \tag{1.2}$$

where D is a diagonal matrix with the probabilities $p_1, \ldots, p_r$ on the diagonal. As p_i for $i = 1, \ldots, r$ are probabilities such that $p_1 + \cdots + p_r = 1$, it is therefore the case that Σ is not of full rank, and $\mathrm{rank}(\Sigma) = r - 1 \Rightarrow \Sigma^{-1}$ does not exist.

We note that the vector of deviations has form $\boldsymbol{\nu} - n\mathbf{p}$. Then consider the statistic X_n, a modified version of these deviations, such that

$$X_n = \frac{1}{\sqrt{n}} D^{-\frac{1}{2}}(\boldsymbol{\nu} - n\mathbf{p}) \to \mathcal{N}(0, D^{-\frac{1}{2}}\Sigma(D^{-\frac{1}{2}})^T),$$

where $\to$ represents a convergence in distribution, by the Central Limit Theorem.

We may simplify the variance as

$$D^{-\frac{1}{2}}\Sigma(D^{-\frac{1}{2}})^T$$
$$= D^{-\frac{1}{2}}\Sigma D^{-\frac{1}{2}},$$

as D is a diagonal square matrix and hence is symmetric, so $D = D^T$.

$$= D^{-\frac{1}{2}}(D - \mathbf{p}\mathbf{p}^T)D^{-\frac{1}{2}}$$
$$= D^{-\frac{1}{2}}DD^{-\frac{1}{2}} - D^{-\frac{1}{2}}\mathbf{p}\mathbf{p}^T D^{-\frac{1}{2}}$$
$$I - D^{-\frac{1}{2}}\mathbf{p}\mathbf{p}^T D^{-\frac{1}{2}}$$

Hence

$$X_n \to \mathcal{N}(0, I - D^{-\frac{1}{2}}\mathbf{p}\mathbf{p}^T D^{-\frac{1}{2}}),$$

where again, the $\to$ represents a convergence in distribution.

We have shown that the statistic X is normally distributed, with parameters mean 0 and variance $I - D^{-\frac{1}{2}}\mathbf{p}\mathbf{p}^T D^{-\frac{1}{2}}$. From here we now wish to show that $\|X\|^2 \sim \chi^2_{(r-1)}$, and by doing so will have shown that the χ^2 test statistic is approximately $\chi^2_{(r-1)}$ distributed. This is because the the χ^2_k distribution is the squared sum of k independent, normal random variables. And although our vector is of length r, we lose one degree of freedom due to the constraint that $p_1 + \cdots + p_r = 1$, giving us $\chi^2_{(r-1)}$.

Now, we label the variance of X as $A = I - D^{-\frac{1}{2}}\mathbf{p}\mathbf{p}^T D^{-\frac{1}{2}}$, and can further experiment with its form,

$$A = I - D^{-\frac{1}{2}}\mathbf{p}(D^{-\frac{1}{2}}\mathbf{p})^T,$$

using the rule that $(EF)^T = F^T E^T$, and that D is a symmetric matrix. We also recall it is diagonal, with the probabilities $p_1, \ldots, p_r$ on the diagonal hence

$$A = I - \begin{pmatrix} \frac{p_1}{\sqrt{p_1}} \\ \vdots \\ \frac{p_r}{\sqrt{p_r}} \end{pmatrix} \begin{pmatrix} \frac{p_1}{\sqrt{p_1}} \\ \vdots \\ \frac{p_r}{\sqrt{p_r}} \end{pmatrix}^T$$
$$= I - \begin{pmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_r} \end{pmatrix} \begin{pmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_r} \end{pmatrix}^T,$$

which is a form we will make use of later.

Now if choose U such that $U^T U = I$, then

$$\|Ux\|^2$$
$$= (Ux)^T Ux$$
$$= x^T U^T Ux$$
$$= x^T x$$
$$= \|x\|^2$$

Then if we can find U such that

$$\text{Cov}(UX) = U^T AU = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}$$

and U satisfies $U^T U = I$ then we have shown that $\|Ux\|^2 \sim \chi^2_{(r-1)}$.

Under these circumstances, the norm squared of x, denoted as $\|x\|^2$ represents the squared size of the vector x, and if the norm squared of $\|Ux\|^2 = \|x\|^2$ then by considering Ux we do not change the squared norm and hence do not change the value of the test statistic. Again, this means when we consider the covariance of $U^T AU$, this is simply the covariance of x multiplied by the vector U^T before and U after, giving $U^T AU$, which as already proved does not change the squared norm of x, which is the test statistic χ^2 of x.

Recalling that A is symmetric $\Rightarrow$ there exists a set of orthonormal eigenvectors, $q_1, \dots, q_r$, such that $Aq_i = \rho q_i$ for $i = 1, \dots, r$. Due to the eigenvectors being orthonormal, we have that $q_i A q_j = 0$ for all $i \neq j$.This idea is explored in more detail in the section entitled Invariant Coordinate Selection, later in this report.

Then, if we take $U = (q_1, \dots, q_r)$,

$$\Rightarrow U^T AU = \begin{pmatrix} \rho_1 & 0 & \dots & 0 \\ 0 & \rho_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \rho_r \end{pmatrix},$$

and $U^T U = I$.

We now need to consider the eigenvalues of A. We need to verify that $r-1$ eigenvalues are 1 and that one remaining eigenvalue is 0.

Lemma 1.1. Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of A. Then $I - A$ has eigenvalues $1 - \lambda_1, \dots, 1 - \lambda_n$.

Proof.

$$Ax_i = \lambda x_i$$
$$\Leftrightarrow (I - A)x_i = x_i - Ax_i = x_i - \lambda_i x_i = (1 - \lambda_i)x_i$$

□

Since

$$A = I - \begin{pmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_r} \end{pmatrix} \begin{pmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_r} \end{pmatrix}^T$$

We can define the vector $\mathbf{b}$ as

$$\mathbf{b} = \begin{pmatrix} \sqrt{p_1} \\ \vdots \\ \sqrt{p_r} \end{pmatrix}$$

Thus

$$A = I - \mathbf{b}\mathbf{b}^T = I - B,$$

and we may instead choose to find the eigenvalues of B, which we hope to be $0, \dots, 0, 1$, and if they are then by Lemma 2.1, there will be $r-1$ eigenvalues of A that are 1, and one eigenvalue that is 0.

Now if we multiply B and b together, we yield the following

$$B\mathbf{b} = B\begin{pmatrix}\sqrt{p_1}\\ \vdots \\ \sqrt{p_r}\end{pmatrix} = \mathbf{b}\mathbf{b}^T\mathbf{b} = \mathbf{b}\begin{pmatrix}\sqrt{p_1}\\ \vdots \\ \sqrt{p_r}\end{pmatrix}^T\begin{pmatrix}\sqrt{p_1}\\ \vdots \\ \sqrt{p_r}\end{pmatrix} = \mathbf{b}(p_1 + \cdots + p_r) = 1\mathbf{b}.$$

This is a good result; we have shown that $Bb = 1b$, which by the eigenvector/eigenvalue relation means we have one eigenvalue of B equal to 1. Next, we need to show that the rest of the eigenvalues of B are 0. Now if we look at the rows and columns of B in more detail, we will be able to see the rank of B rather clearly. We note B has form

$$B = \begin{pmatrix} p_1 & \sqrt{p_1p_2} & \sqrt{p_1p_3} & \cdots & \sqrt{p_1p_r} \\ \sqrt{p_1p_2} & p_2 & \sqrt{p_2p_3} & \cdots & \sqrt{p_2p_r} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sqrt{p_1p_r} & \sqrt{p_2p_r} & \sqrt{p_3p_r} & \cdots & p_r \end{pmatrix}.$$

The rank of a matrix such as B can be said to be the amount of linearly independent columns of B. It is clear to see that there is only one linearly independent row of B and hence $\text{rank}(B) = 1$. This is because each row is simply a multiple of any other row. For example to get from the first row to the second row we simply need to multiply the first row by $\frac{\sqrt{p_2}}{\sqrt{p_1}}$. A general rule, is to get from row i to row j simply multiply row i by $\frac{\sqrt{p_j}}{\sqrt{p_i}}$. So we now see that $\text{rank}(B) = 1$. Now by the rank nullity theorem which is taken from pg 199 of 'Matrix Analysis and Applied Linear Algebra' by Carl D. Meyer [Mey00] and states that

$$\text{rank}(B) + \text{nullity}(B) = n, \text{ for all } m \times n \text{ matrices.} \tag{1.3}$$

Now as B is a $r \times r$ square matrix, and of $\text{rank}(B) = 1$, then the $\text{nullity}(B) = r - 1$ by the rank nullity theorem and $\Rightarrow$ there is at most 1 non-zero eigenvalue, which we discovered was equal to 1. Hence there must be no more non-zero eigenvalues, so there must be $r - 1$ eigenvalues equal to 0, which correspond to the fact $\text{nullity}(B) = r - 1$.

Then by Lemma 2.1 we know that A has $r - 1$ eigenvalues equal to 1 and one eigenvalue equal to 0. Thus we have shown that $\|x\|^2 \sim \chi^2_{(r-1)}$.

1.1.3 One-sided or Two-sided Test

We need to consider whether, when testing PRNGs, we wish to perform a one-sided or two-sided Chi-square test. Usually a Chi-square test is one-sided in nature, however is this the most appropriate version of Chi-square testing for when testing PRNGs?

One-sided

The likelihood of the Chi-square value is assessed by comparing the value to a Chi-distribution with degrees of freedom ν based on the number of parameters and constraints. A critical value is chosen from the Chi-square distribution and then a critical region is defined to be the area of the distribution to right of the critical value where we would reject the null hypothesis if the test statistic is contained within this area. This is because the Chi-square distribution is a non-symmetric and non-negative distribution and is measuring the deviation of a set of observables from a proposed set of expected values, hence the distribution is bounded on the left side by 0 (which can be attained). As the usual hypothesis is testing whether a sample differs from a prescribed distribution, it is the right tail of the distribution

that represents significant deviation from the distribution, and where the critical region is located. If 5% significance test is required then the critical value is chosen with significance $\alpha = 0.05$, meaning that the right hand tail containing 5% of values will be the rejection region. This is typically how Chi-square tests are performed, as we would consider no deviation from a prescribed distribution as a 'perfect' result. Would we want this 'perfect' result from our PRNGs? This would represent a generator that contains no random noise and would perhaps be too predictable.

Two-sided

A two-sided test rejects values of the Chi-square value if they appear in either tail of the distribution. As the Chi-square distribution is non-negative and measures deviation between observed values and expected values normally values in the left tail of the distribution are not rejected. However, in the case of testing PRNGs this method of rejection is adopted and is also the method Knuth uses in 'The Art of Computer Programming Vol. 2'. This is because if there is too little deviation between the observed values and the expected values for a PRNG then we would have to question its use for generating random numbers as the output values would be too predictable. In the case of two-sided tests if we again require a 95% test then we would choose two critical values $\alpha = 0.025$ and $\alpha = 0.975$. This way, we accept the middle 95% of the distribution and reject test statistics in either tail of the distribution when they are greater than the largest critical value or less than the smaller critical value.

1.1.4 Discrete data

A requirement of the Chi-square "goodness of fit" test is that the sample data takes only a finite number of values, say, k. When data is of this form applying the "goodness of fit" test is trivial. For an example of this we now consider answering Exercise question 3 from Section 3.3.1 of "The Art of Computer Programming Vol 2" by Knuth.

Example 1.1. Question

Some dice are suspected to be loaded. They are rolled 144 times and the following values observed

i	2	3	4	5	6	7	8	9	10	11	12
O_i	2	6	10	16	18	32	20	13	16	9	2

Apply the Chi-square test to these observations assuming it is not known the dice are faulty. Does the test detect the dice are faulty? If not, discuss why not.

Answer

As we assume that the dice are not loaded, we assume that both die are identical and each individual dice throw is an observation taken from a discrete uniform distribution between 1 and 6, meaning $P(j = 1) = P(j = 2) = \cdots = P(j = 6) = \frac{1}{6}$. From this we can construct the null and alternative hypotheses

$H_0 : P(X = i) = p_i$ for $i = 1, \ldots, k$ where values for p_i are shown in the table below.

$H_1 : X$ has a different distribution.

Next we construct the test statistic by calculating the expected values using the probabilities calculated for outcomes $i = 2, \ldots, 12$. The table below assembles the Chi-square statistic step by step.

i	2	3	4	5	6	7	8	9	10	11	12
O_i	2	6	10	16	18	32	20	13	16	9	2
p_i	$\frac{1}{36}$	$\frac{1}{18}$	$\frac{1}{12}$	$\frac{1}{9}$	$\frac{5}{36}$	$\frac{1}{6}$	$\frac{5}{36}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{36}$
E_i	4	8	12	16	20	24	20	16	12	8	4
$\frac{(O_i-E_i)^2}{E_i}$	1	$\frac{1}{2}$	$\frac{1}{3}$	0	$\frac{1}{5}$	$\frac{8}{3}$	0	$\frac{9}{16}$	$\frac{4}{3}$	$\frac{1}{8}$	1

$\chi^2 = \sum_{i=1}^{n} \frac{(O_i-E_i)^2}{E_i} = 7.720833$

This example contains 11 parameters, p_i for $i = 1, \ldots 11$, hence the degrees of freedom is $\nu = 11 - 1 = 10$ as we have the constraint that $p_1 + \cdots + p_1 1 = 1$. When looking at the table for the Chi-square distribution with 10 df we find the value of 7.72 to be not suspicious at all; we would expect a χ^2 value between 3.94 and 18.31 90% of the time. In fact is revealed that the dice were loaded and that one of the dice always returned either 1 or 6 with equal probability. The reason that the Chi-square test does not correctly identify that the sample is not from the null distribution is because the sample size is not large enough and also the the individual probabilities are not largely different to the ones in the assumed distribution. In fact, this example violates an assumption of the Chi-square test that $np_i > 5$ for $i = 2$ and $i = 12$. We would require a minimum of $n = 180$ for the expected value of these two categories to be equal to 5. Although $n = 144$ is quite large, the fact that there is 11 different categories, some with very small probability is the reason this n is not large enough in this example.

1.1.5 Continuous data

As previously mentioned, the Chi-square test works for a finite number of categories so requires discrete observations. Chi-square testing can still be done on continuous data, however observations must be grouped into discrete groups that span chosen intervals, remembering that each category must satisfy the criterion that $np_i > 5$ for all $i \in 1, \ldots, k$ groups. A simple example could be the continuous data of the height of a group of people that could be grouped as, say <140cm, 140cm - 160cm, 160cm - 180cm, 180cm - 200cm and >200cm, as long as each group had an expected number of at least 5. An obvious question arises here, that is just how many groups or "bins" should the data be split into? Obviously we require $np_i > 5$ for all $i \in 1, \ldots, k$ groups, but how then should they be chosen? In a case such as measuring heights, intuition, or what feels natural can be applied to the problem. When the alternative hypothesis is not assigned a specific distribution [GN96] suggests it is best to choose groups $1, \ldots, k$ as all being equally probable intervals such that

$\Pr(X_i \in 1 | H_0 \text{ true}) = \frac{1}{k}$

The output of a LCG is a set of values continuous on the interval $[0, 1]$. The following transformation is made to the output values $U_1, U_2, U_3, \ldots, U_n$ of a PRNG, where values are "binned" together into k groups, all "bins" are equally probable as under H_0. $U_1, U_2, U_3, \ldots, U_n$ are uniformly distributed under the null hypothesis and hence k equal sized intervals between 0 and 1 have equal probability $\frac{1}{k}$ of U_i being in any one interval.

$u_i \in [0, \frac{1}{k}) \to 1$

$u_i \in [\frac{1}{k}, \frac{2}{k}) \to 2$

$\vdots$

$u_i \in [\frac{k-1}{k}, \frac{k}{k}) \to k$

Then the frequency of each category from $1, \ldots, k$ are;

$Y_1, Y_2, Y_3, \ldots, Y_k =$ how often we get outcome $1, 2, 3, \ldots, k$

For a sensible value for k [GN96] recommends the following as a minimum;

$k \leq \min(\frac{1}{\alpha}, \log n)$,

where α is the significance level ($0 < \alpha < 0.5$)

The null and alternative hypothesis for the Chi-square test when applied to PRNGs are defined as follows.

H_0 : The generators output can be considered random; the sample is appears to be distributed uniformly on the interval $[0, 1]$.

H_1 : The generators output cannot be considered random as is either too predictable or values are not uniformly distributed enough.

1.1.6 Chi-square Test in R

The output of a PRNG be assessed for distribution using Chi-square test, however the output typically contains a very large amount of numbers, essentially continuous and uniform on the interval $[0, 1]$. Testing this by hand would again be very impractical hence the Chi-square test has been written as an R function that simply requires three inputs; the output of an PRNG (can be generated for example, using the previous R function in the LCG section as our five chosen LCGs were) and the number of "bins" k, that the user wishes the data be assigned into and finally a given significance a used when comparing to the Chi-square distribution. In terms of testing for a distribution, this function tests the frequency of values in each category $1, \ldots, k$ against how many would be expected in each category; $\frac{n}{k}$. As we are hoping the PRNG is producing values that can be used in place of true random values that are uniform on the interval $[0, 1]$, then when we "bin" these values we would expect each "bin" to theoretically have equal total frequency. The R function wrote is shown and explained below.

```
rngchi <- function(x, k, a) {
  z <- cut(x, 0:k / k, labels=FALSE)
  z[is.na(z)] <- 0
  O <- c()
  for(i in 1:k) O[i] <- sum(z==i)
   E <- c()
   for(i in 1:k) E[i] <- (1/k) * length(x)
       chi.squared <- sum((O - E)^2 / E)
       if(chi.squared>qchisq(a/2, df=k-1) && chi.squared<qchisq(a/2, df=k-1, lower.tail=F)){
return(list("ACCEPT", chi.squared))
}else{
return(list("REJECT", chi.squared))
}
}
```

This function requires an input of a string of numbers to test for distribution; x, and a number k which is how many bins the continuous values x will be split into. It also requires a significance level which is used to find the critical value on the the appropriate Chi-square distribution, which has $k - 1$ degrees of freedom. Firstly the data x is cut into equal sections using the cut() function and is labeled as z. This returns NA for when values are exactly 0 due to the way the cut function works at boundaries hence NA values are converted to 0 on the next line of the function. O and E are defined to be empty vectors, where O is filled in numerical order with the frequency each value from 0 to k occurs, and obviously refers to the observed values. E is then a vector of length k each value taking $\frac{n}{k}$ where n =length of the vector x. Then chi.squared is defined as the usual Chi-square test statistic previously defined. Now if the Chi-squared value is between the upper and lower critical values of the appropriate Chi-square distribution then the function returns the statement "ACCEPT" along with the value of the test statistic, which is determined using an if statement. If this condition is not met then the function outputs "REJECT" along with the test statistic.

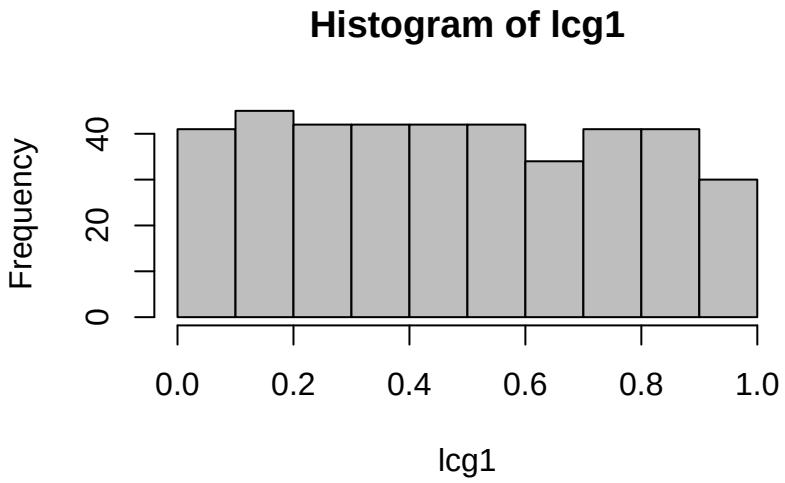

Figure 1.1: This figure shows a histogram of the output of LCG1. We observe a histogram that indicates that the output of LCG1 could come from a Uniform distribution.

1.1.7 Examples of Chi-square Test on LCGs

We now apply the Chi-square function I have written in R to assess the five example LCGs previously defined. A decision is made to split the data for each LCG into $k = 10$ bins for simplicity and due to running large n, there is no worry that the expectation for any group is less than 5. The Chi-square test was run using many different values of k and was results were found to be consistent across sensible choices of k and hence only $k = 10$ is included here. Also by doing this the Chi-square value attained for each LCG is directly comparable; all examples use $\nu = 9$ as the degrees of freedom $(k - 1)$. The critical values at a 5% significance level for a two sided test on a Chi-square distribution with 9 degrees of freedom are 2.70 and 19.02, meaning we would reject the null hypothesis in favour of the alternative hypothesis when we see $\chi^2 < 2.70$ or $\chi^2 > 19.02$, however the function I wrote makes this decision automatically. For all examples that require a significance level, I have chosen to use 0.05 significance to keep results consistent.

Example 1.2. We first apply the Chi-square test to LCG1 from example 0.3. Let us begin our analysis by firstly considering a histogram for LCG1 with $k = 10$ bins.

As seen histogram 1.1 for LCG1 we observe a histogram that, by eye, appears sufficiently random. The histogram has ten bars, which corresponds to our chosen k when performing the Chi-square test. All bars are similar heights, if not the same height, indicating a small amount of random variation. It certainly seems plausible that the values from LCG1 came from a uniform distribution. Then let us apply the Chi-square function. The test statistic and decision were outputted as follows.

χ^2 value	Decision
4.5	Accept

We observe a test statistic value of 4.5 which is within our critical values, and as such the function determines we accept the null hypothesis in this case. There is insufficient reason to doubt these observables came from a distribution Uniform on $[0, 1]$. The value is far closer to the lower critical bound, and if it was below this threshold we would have rejected the null hypothesis due and concluded that there was too little variability or 'random noise' for the LCG to be considered random.

Example 1.3. Next let us consider LCG2 from example 0.4. We begin our initial analysis of this LCG by considering a histogram of the observables split into $k = 10$ bins.

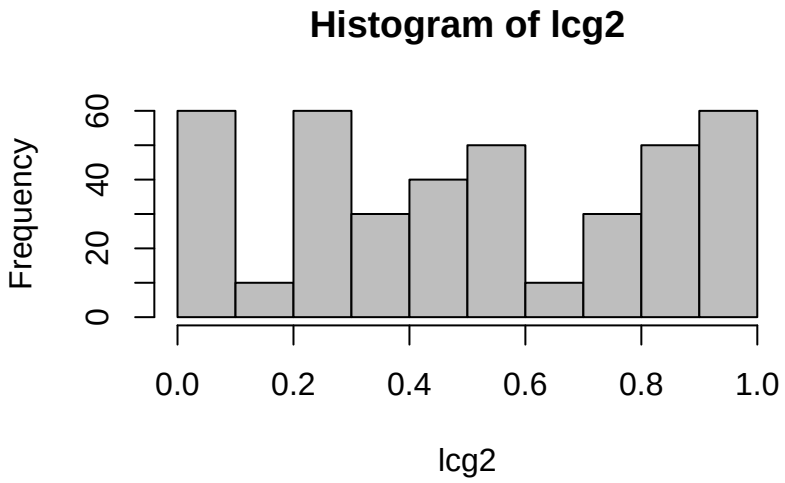

Figure 1.2: This figure shows a histogram of the output of LCG2. We observe a histogram which indicates that the distribution of LCG2 does not appear to be Uniform. Large differences in bars clearly show that the distribution of the observed values is not 'box' shaped as we would expect from Uniformly distributed observations.

The histogram for LCG2, histogram 1.2 looks vastly different to the histogram for LCG1, both of which span $[0, 1]$ and have the same intervals for bars. The bars for LCG2 are highly unequal, some containing only 10 members, and other bars as high as 60. Using only this histogram, it appears obvious that LCG2 is a poor PRNG and the output does not uniformly span the interval $[0, 1]$. However, for a definitive answer, we apply the Chi-square test, the which confirms our prior suspicions.

χ^2 value	Decision
77.5	Reject

The test statistic for LCG2 is extremely large, and well above the upper critical bound on the appropriate Chi-square distribution; it is more than 4 times larger than it in fact. Hence, we conclude that observables from LCG2 are not distributed uniformly on the $[0, 1]$ interval as we strongly reject the null hypothesis in this case.

Example 1.4. Next let us consider LCG3 from example 0.5, recalling that this LCG is the same as LCG1, however runs for a larger n and thus over runs is period length and begins repeating values, which is reflected in the histogram 1.3. We observe the bars of this histogram are almost identical, and there is little overall deviation. This is reflected in the Chi-square test calculated for this LCG.

χ^2 value	Decision
0.0775	Reject

We strongly reject our null hypothesis in this case, due to such a small test statistic value. This indicates a lack of randomness in the output of LCG3, and although the histogram certainly seemed uniform, we deem this 'too perfect'. Obviously if we wish to use a LCG to generate random values from a uniform distribution, we do not want values to be too predictable, as is the case here.

Example 1.5. We now move on to a LCG used in Numerical Recipes, LCGNR from example 0.6. Histogram 1.4 for this LCG looks rather good. There appears to be enough random variation within the values, however all bars are similar heights. We next apply the Chi-square function to this LCG.

χ^2 value	Decision
8.24	Accept

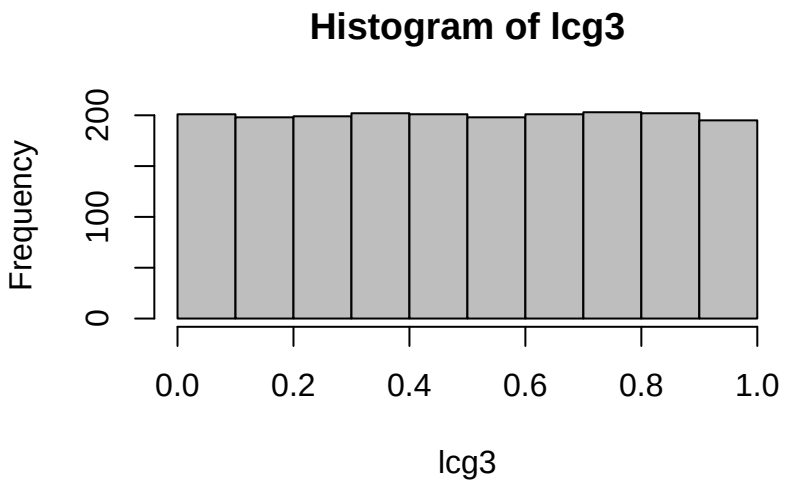

Figure 1.3: This figure shows a histogram of the output of LCG3. We see that the output for LCG3 does appear to be Uniformly distributed as the histogram has the usual 'box' shape we would expect of Uniformly distributed values. All bars on the histogram, however, are almost equal, differing in frequency by only a small number compared to the total number of values. We may suspect a lack of 'random noise' in the output for LCG3, and we will see if this is the case when we apply the Chi-square test to LCG3.

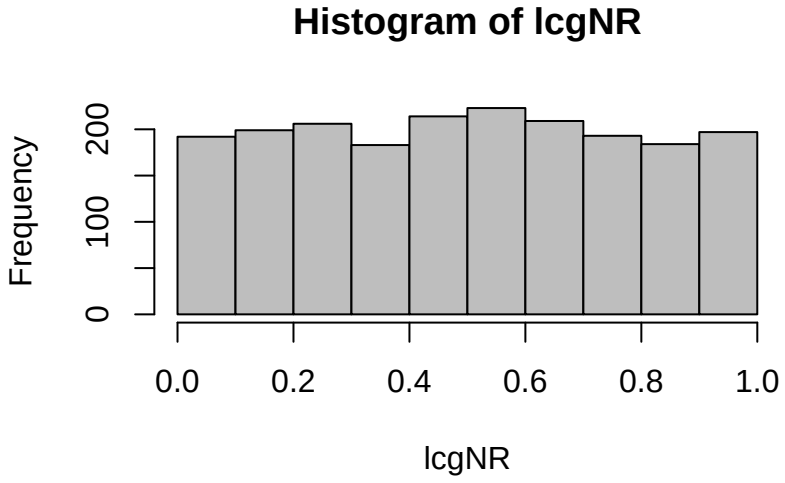

Figure 1.4: This figure shows a histogram of the output of LCGNR. We observe an excellent histogram for LCGNR, in terms of the distribution of values, including a small amount of 'random noise' within the data. It certainly seems plausible that the distribution is Uniform in this instance.

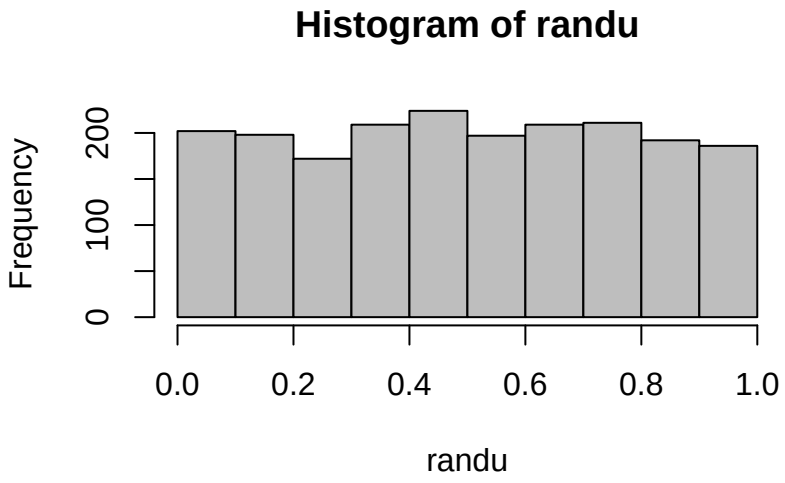

Figure 1.5: This figure shows a histogram of the output of RANDU. We observe an excellent histogram for RANDU, in terms of distribution and a small amount of 'random noise' within the data. It certainly seems plausible that the distribution of RANDU is Uniform.

We observe a test statistic well within critical values, and hence we accept the null hypothesis and conclude that we can use this LCG to generate values from a uniform $[0, 1]$ distribution. This is exactly what we would expect from a well known LCG.

Example 1.6. Finally, we focus our attention to the famous LCG, RANDU, from example 0.7. Although it is now known to be a poor LCG, its use was widespread before this discovery. Thus we might expect it to pass a Chi-square test. Let us first consider histogram 1.5 of RANDU. The histogram looks typical of a good LCG, and is similar in form to the histogram for LCGNR; similar sized bars with small variation between bars. This is also reflected in the Chi-square test statistic, which is very close to the test statistic obtained for LCGNR.

χ^2 value	Decision
9.6	Accept

The Chi-square test determines we accept the null hypothesis for RANDU, and we begin to see why this poor LCG was used by many. We see no evidence to suggest that the output of RANDU is not from a Uniform $[0, 1]$ distribution.

1.2 The Kolmogorov-Smirnov Test

1.2.1 Description of the Test

The Kolmogorov-Smirnov test is a non-parametric test that can also be used as a 'goodness of fit' test to assess the quality of a PRNG. Being non-parametric, the test makes no assumptions about the distribution of the sample being tested. This inherently makes the test rather simple to perform [UC08]. The Kolmogorov-Smirnov test can appears as a 'one-sample' test to determine whether a sample is from a proposed distribution. As is common with many statistical tests there is also a 'two-sample' version of the test, where two different samples are compared to determine if they are from the same distribution. The testing of PRNGs uses former of these two tests. The Kolmogorov-Smirnov test is used to test strictly continuous distributions, unlike the Chi-square which tests discrete data, but can also be applied to continuous data via the "binning" technique.

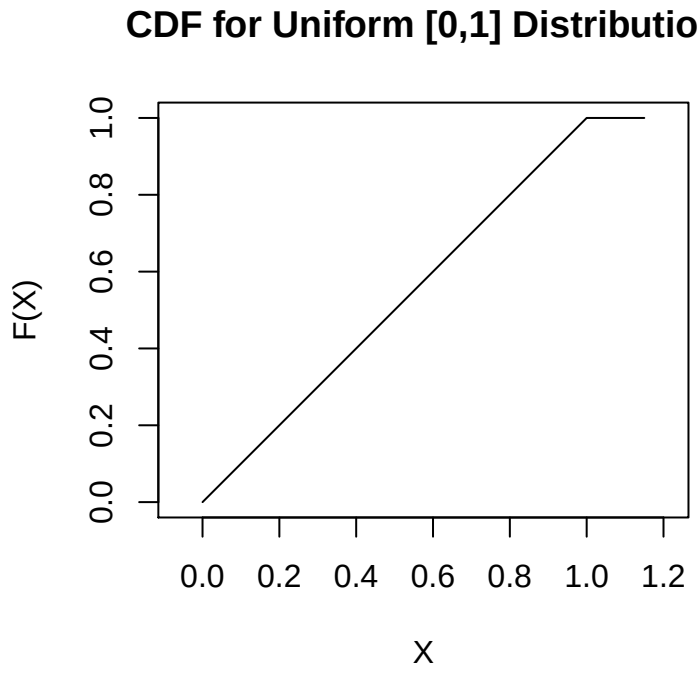

Figure 1.6: This figure shows the Uniform [0,1] CDF

The Kolmogorov-Smirnov test statistic is constructed using the cumulative distribution function $F(x)$, which for random variable X is defined to be

$$F(x) = P(X \leq x).$$

The one-sample Komlogorov-Smirnov test tests the following null hypothesis against the general alternative hypothesis

H_0 : The sample was drawn from a population with continuous distribution $F(x)$

H_1 : The sample was drawn from some other distribution

The Kolmogorov-Smirnov test constructs an observed "empirical" cumulative distribution and contrasts this distribution against a proposed distribution. For testing PRNGs that generate all numbers with equal probability the proposal distribution is the cumulative distribution function, $F(x)$ for a Uniform distribution on the interval $[0, 1]$ and will form the base distribution to which our observed distribution will be compared. It is shown in 1.6.

Given our sample of n observations $U_1, U_2, U_3, \ldots, U_n$, Knuth [Knu69] on pg 43 defines the empirical distribution function $F_n(x)$ as

$$F_n(x) = \frac{\sum_{i=1}^{n} \mathbb{I}[U_i \leq x]}{n},$$

where $\mathbb{I}[]$ is an indicator function meaning,

$$\mathbb{I}[U_i \leq x] = \begin{cases} 1 & U_i \geq x \\ 0 & \text{else.} \end{cases}$$

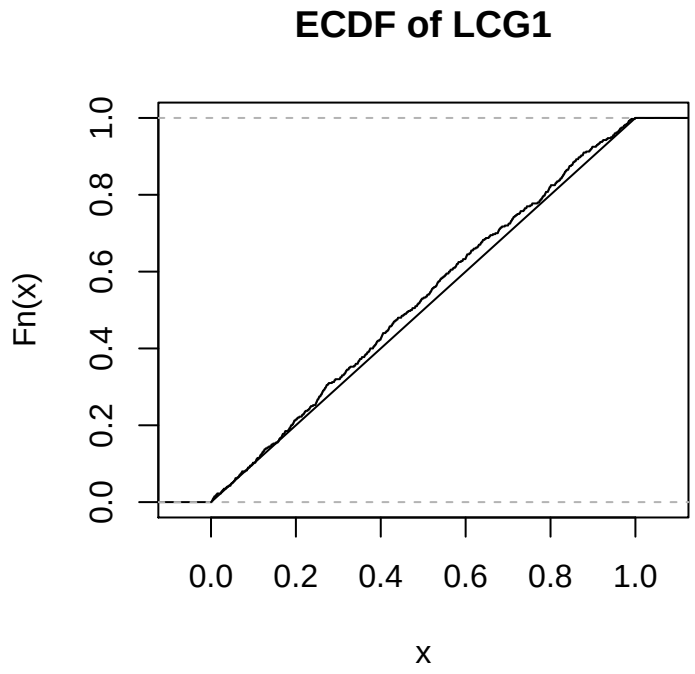

Figure 1.7: This figure shows the Empirical Cumulative Distribution Function of LCG1 against the Cumulative Uniform $[0, 1]$ distribution. We see that the data fits the distribution reasonably well, although the empirical data lags slightly ahead of the Uniform distribution for most of the plot.

The Kolmogorov-Smirnov test statistic involves finding the largest distance between the empirical distribution $F_n(x)$ and the theoretical cumulative distribution function $F(x)$ and then comparing this value to values in a Kolmogorov-Smirnov table. Intuitively, a good PRNG will have an empirical distribution function that is very close to the Uniform $[0, 1]$ cumulative distribution function, meaning the largest distance between these two distributions will be small. Conversely, a bad PRNG will have an empirical distribution function that will deviate significantly from the Uniform CDF and thus, at some prescribed significance level, would mean rejection of the null hypothesis that the sample was drawn from the theoretical distribution, as the largest distance between these two distributions would be too large. From this we would deduce the the distribution of values from the PRNG is not Uniform. We would also reject the null hypothesis if this distance was too small as would represent too little deviation between the two distributions.

Let us consider again consider LCG1 and LCG2. The empirical cumulative distribution is for each generator is plotted against $F(x)$ for Uniform $[0, 1]$ Uniform distribution.

Example 1.7. For LCG1, in figure 1.7 we see very little deviation between $F_n(x)$ and what we would expect; $F(x)$. The distance between the two functions is minimal across the entire interval $[0, 1]$, indicating that LCG1 could indeed be a sample from the uniform distribution. This follows from the Chi-square test on LCG1, which concluded the sample was from a uniform distribution. As this LCG achieved a maximal period, and $n = 400$ ran for less than its period, its output $U_1, U_2, U_3, \ldots, U_n$ are unique values. Each step up on $F_n(x)$ therefore represents a jump up of $\frac{1}{n} = \frac{1}{400}$.

Example 1.8. For LCG2 figure 1.8 displays the empirical cumulative distribution $F_n(x)$ and $F(x)$. $F_n(x)$ deviates from $F(x)$ far more than we saw for LCG1 in example 0.3. We see much larger distances between the two distributions; with $F_n(x)$ deviating both above and below $F(x)$. LCG2 did not achieve a maximal period, hence its output values $U_1, U_2, U_3, \ldots, U_n$ were not unique. This is why the empirical distribution has large jumps up, each jump here is of height $\frac{i}{n} = \frac{i}{400}$ where i is the number of output values that took value x.

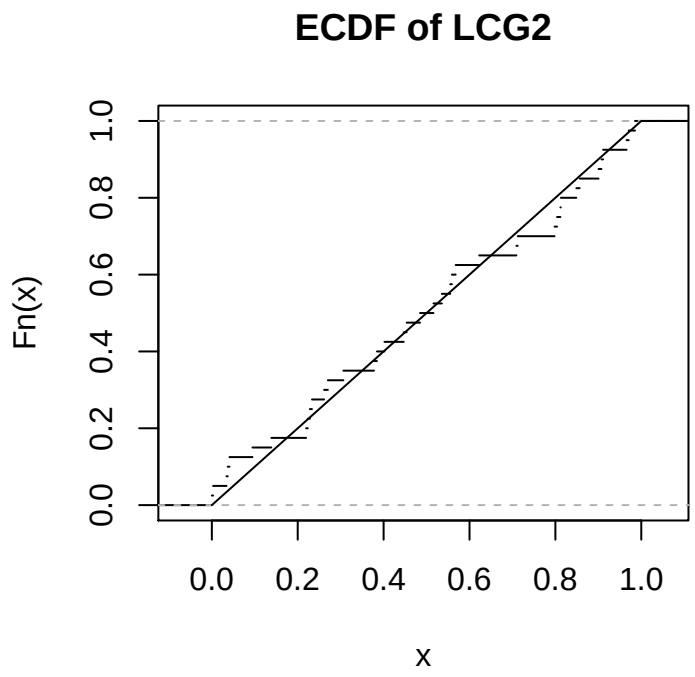

Figure 1.8: This figure shows the Empirical Cumulative Distribution Function of LCG2 against the Cumulative Uniform $[0, 1]$ distribution. We some degree of fit to the data, however the large jumps in the empirical distribution mean a simple straight line Uniform cumulative distribution is not very well fitted to this data.

It is difficult to judge by eye whether or not the deviation between the two distributions is significant enough to dismiss the null hypothesis and conclude that the sample was not drawn from $F(x)$. Thus the Kolmogorov-Smirnov test is used to give a definitive decision regarding $F_n(x)$ and $F(x)$.

Interestingly, we calculate the largest distance between the aforementioned distributions using two different statistics - one when the empirical distribution is above the CDF and when the empirical distribution is below the CDF. Knuth in [Knu69] defines these two distances on p43, and are formulated as follows

$$K_n^+ = \sqrt{n} \max_{-\infty<x<\infty} (F_n(x) - F(x))$$

$$K_n^- = \sqrt{n} \max_{-\infty<x<\infty} (F(x) - F_n(x))$$

Both statistics seek the largest positive distance between the between functions $F_n(x)$ and $F(x)$. K_n^+ represents the largest distance between the two functions when $F_n(x)$ is the larger value (above $F(x)$), due to $F(x)$ being subtracted from $F_n(x)$. To find these distances the data must first be ordered from smallest to largest.

$U_1, \ldots, U_n \to \text{sort}(U_1, \ldots, U_n) \to U_{(1)}, \ldots, U_{(n)}$.

Let us consider these two statistics, in more detail.

1.2.2 K_n^+

K_n^+ considers distances between $F_n(x)$ and $F(x)$ when $F_n(x) > F(x)$. Both functions are constant outside of the interval $[0,1]$. First we consider the Empirical Cumulative Distribution function; $F(x)$. It is true for $F(x)$ that $F(a) > F(b)$ for all $a > b$ when $a, b \in [0,1]$ (because 0 and 1 are the upper and lower bound of $U_{(1)}, \ldots, U_{(n)}$ respectively). Therefore $F(x)$ is strictly increasing on the interval $[0,1]$ for values of $x \in [0,1]$.

$F_n(x)$ is a step function, with jumps of $\frac{i}{n}$ at values of $U_{(1)}, \ldots, U_{(n)}$, where i is the number of $U_{(1)}, \ldots, U_{(n)}$ that share a like value. If all values of $U_{(1)}, \ldots, U_{(n)}$ are unique then there will be n jumps of size $\frac{1}{n}$. When values appear more than once the number of jumps will be the number of unique values and thus at least one jump will be greater than $\frac{1}{n}$. The function is constant outside of the interval $[U_{(1)}, U_{(n)}]$, where $U_{(1)} \geq 0$, and $U_{(n)} \leq 1$. Therefore $F_n(x)$ is an increasing function, but not strictly increasing as for $b > a$, $F_n(b) \geq F_n(a)$ as the function remains constant between observations.

We next consider $F_n(x) - F(x)$ on the interval, where $x \in [U_{(i)}, U_{(i+1)})$. On this interval $F(x)$ is increasing, however $F_n(x)$ remains constant across values of x until the next observation. Therefore when we are searching for the largest distance between $F_n(x)$ and $F(x)$, we need only look at the points $x = U_{(1)}, \ldots, U_{(n)}$ to find it.

Thus,

$$\max_{x \in [U_{(i)}, U_{(i+1)})} (F_n(x) - F(x)) = F_n(U_{(i)}) - F(U_{(i)}).$$

We have found the largest distance between the two distributions when $[U_{(i)}, U_{(i+1)})$ which is bounded by the lower limit $U_{(1)}$ and the upper limit $U_{(n)}$, but what happens outside these values, could there be a largerK_n^+ between distributions here?

First we consider the interval to the right of $U_{(n)}$; where $x \in [U_{(n)}, \infty)$. This interval contains the point where $x = 1$, which the largest attainable value $U_{(n)}$ can take. For all $x \geq 1$ both $F_n(x) = 1$ and $F(x) = 1$ and remains constant. The largest value $F_n(x) - F(x)$ can take in this instance is 0. This accounts for $U_{(n)}$ when $U_{(n)} = 1$, however this upper bound is not always attained. This means the interval $[U_{(n)}, \infty)$ may contain values of $x < 1$. When $U_{(n)} < 1$ the function $F_n(U_{(n)})$ attains 1 earlier than $F(x)$. In this case $F(x)$ will keep increasing until $x = 1$ at which point it will then level out and remain constant. In this case The difference between $F_n(U_{(n)})$ and $F(U_{(n)})$ is not trivially 0. Under both circumstances the greatest distance between $F_n(x)$ and $F(x)$ is at point $U_{(n)}$. For $U_{(n)} < 1$ the distance is unique and the largest on the interval $x \in [U_{(n)}, \infty)$ if $U_{(n)} < 1$. When $U_{(n)} = 1$ then the distance is 0 for all $x \in [U_{(n)}, \infty)$. That is, for $U_{(n)} = 1$ then

$$\max_{x \in [U_{(n)}, \infty)} (F_n(x) - F(x)) = F_n(U_{(n)}) - F(U_{(n)}) = F_n(1) - F(1) = 0.$$

If $U_{(n)} < 1$, then

$$\max_{x \in [U_{(n)}, \infty)} (F_n(x) - F(x)) = F_n(U_{(n)}) - F(U_{(n)}) = \varepsilon,$$

where $\varepsilon > 0$ is some arbitrary value.

Finally we consider the values in the interval $x \in (-\infty, U_{(1)})$. Much like the upper bound, the lower bound may or may not be attained. When the lower bound is attained and $U_{(1)} = 0$ both functions are constant and $F_n(x) = 0$ and $F(x) = 0$ for all $x < U_{(1)}$ The maximum distance cannot be found here as the difference between these functions here is 0 for all values in the interval $(-\infty, U_{(1)})$. When the lower bound is not attained, $F_n(x)$ remains constant at 0 and past it until the point $U_{(1)}$. However, $F(x)$ begins increasing at 0 irregardless, meaning that the difference between the functions on the interval

$x \in (-\infty, U_{(1)})$ will be negative as $F(x) > F_n(x)$. Thus the maximum value that can be attained over the interval is 0.

1.2.3 K_n^-

Conversely, K_n^- represents the largest distance between the two function when $F(x)$ is the larger value (above $F_n(x)$). This means on the interval $x \in (U_{(i-1)}, U_{(i)}]$ the largest distance between distributions is found on the right most point of each step for $F_n(x)$, at the point where $x = U_{(i)}$. From this we deduce again that we only need to look at the distances between the functions at points $x = U_{(1)}, \ldots, U_{(n)}$. It is worth noting that when calculating the distances between $F_n(x)$ and $F(x)$ at data point $U_{(i)}$ for $F(x)$, we actually take point $U_{(i-1)}$ for $F_n(x)$, due to the largest distance between distributions is at the right most point of each step. This is shown further in the R function written for K_n^-.

The arguments for $x \in [U_{(n)}, \infty)$ and $x \in (-\infty, U_{(1)})$ are reversed compared to K_n^+. Meaning on the interval $x \in [U_{(n)}, \infty)$ the distance $F(x) - F_n(x)$ is trivially 0 for all x. On the interval to the left of the data points $x \in (-\infty, U_{(1)})$ the largest distance $F(x) - F_n(x)$ can take is at $U_{(1)}$.

1.2.4 The Kolmogorov-Smirnov Test Statistic

Evaluating the Kolmogorov-Smirnov test statistic is not always so simple considering there are many different versions of the test and multiple ways to interpret the results. Many sources only consider distance D, the largest of K_n^+ and K_n^- such as in [Wol73] on pg 226 where D is defined to be

$$D = \max(K_n^+, K_n^-) = \max_{x \in \mathbb{R}} |F_n(x) - F(x)|$$

In [Knu69] however, both K_n^+ and K_n^- are considered. In our case we are interested in a two sided version of the test. Just as we did in the case of the Chi-Square test we choose to reject values returned by the Kolmogorov-Smirnov test that are too small, as they represent not enough randomness in our PRNG. We also reject a test statistic value that is too large as this indicates the observed values do not come from the proposed distribution. The statistics are stated below taken from pg 45 of [Knu69] and formed slightly differently from previously. This is because from the arguments presented in the previous sections regarding K_n^+ and K_n^-, the largest value that these statistics can take are found at one of the ordered points for $x = U_{(1)}, \ldots, U_{(n)}$ and are $U_{(1)}$ only, and thusly we need not consider values of x in all of $(-\infty, \infty)$. Thus these statistics become

$$K_n^+ = \sqrt{n} \max_{(1) \leq i \leq (n)} (\frac{i}{n} - F(x_i))$$

$$K_n^- = \sqrt{n} \max_{(1) \leq i \leq (n)} (F(x_i) - \frac{i-1}{n})$$

For the Kolmogorov-Smirnov test the choice of n is easier to advice than for the Chi-square. It will take a rather large n to deduce whether the random variables $U_{(1)}, \ldots, U_{(n)}$ are sampled from assumed distribution $F(x)$ or some other distribution. However, too large values of n, will reduce the significance of local non random behaviour in the observations, so for the Kolmogorov-Smirnov test, there is reason to not simply pick the largest possible n. [Knu69] suggests a compromise of $n = 1000$.

There are two possible ways of applying the Kolmogorov-Smirnov test, the first simply applying the test to the output of a PRNG and is explained here. We simply calculate the values K_n^+ and K_n^- from the output of an LCG. Doing so will yield us with two values, which can be compared to Kolmogorov-Smirnov tables. If these values are not within the critical regions of the distribution then we will accept

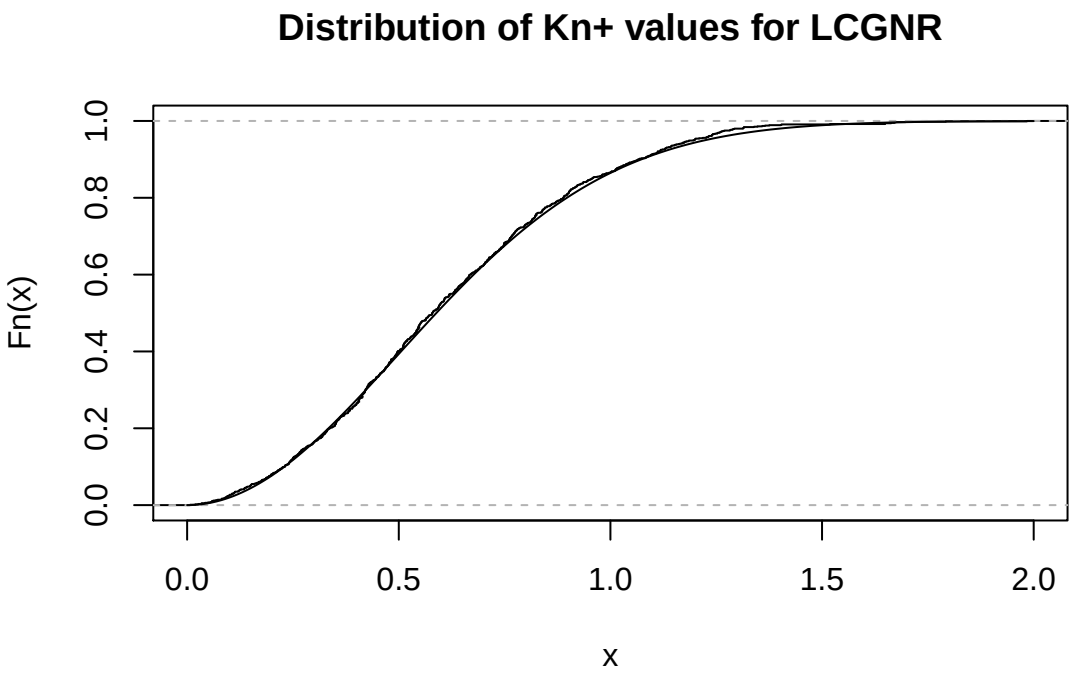

Figure 1.9: This figure shows the distribution of K_n^+ for LCGNR together with the function $F_\infty(x)$. We observe that K_n^+ for LCGNR is extremely well approximated by the function $F_\infty(x)$, seeing very little deviation from the function.

the null hypothesis and conclude that the distribution of the PRNG is sufficiently Uniform on the $[0, 1]$ interval. I outline a function later, which can give the values we require to come to a decision. This is the first method, and another method, which involves calculating many Kolmogorov-Smirnov test statistics is given below.

Knuth [Knu69] suggests performing multiple Kolmogorov-Smirnov tests on an LCG. Further it suggests doing so by performing a large number of Kolmogorov-Smirnov test statistics; $K_n^+(1), \ldots, K_n^+(r)$ and $K_n^-(1), \ldots, K_n^-(r)$ starting at different seeds $1, \ldots, r$ for the same random number generator. The Kolmogorov-Smirnov test can then be applied to these values of $K_n^+(1), \ldots, K_n^+(r)$ or $K_n^-(1), \ldots, K_n^-(r)$. Thus we obtain a K_n^+ and K_n^- for $K_n^+(1), \ldots, K_n^+(r)$ and $K_n^-(1), \ldots, K_n^-(r)$ and for large n, the distribution of these two statistics $F_r(x)$ is closely approximated by the function

$$F_\infty(x) = 1 - e^{-2x^2}, x \geq 0. \tag{1.4}$$

Although this is not proved here, I justify that this is the case using a numerical example and argue that this justification is good enough for our purposes.

The multiplication by a factor of $\sqrt{n}$ in both K_n^+ and K_n^- is there for good reason. We will only consider K_n^+ when showing the importance of the $\sqrt{n}$, as the argument is identical for K_n^- because the two statistics are considered to be distributed the same. To illustrate the relevance of $\sqrt{n}$ in K_n^+ we consider our best LCG, LCGNR 0.6 which is a far superior to our other LCGs. We run LCGNR for $r = 1, \ldots, 1000$ times, each with a length of $n = 2000$ and using starting seeds $x_0 = 1, \ldots, 1000$. K_1000^+ is then evaluated for each of the runs. A plot of these values against the theoretical $F_\infty(x)$ is shown in 1.9.

As figure 1.9 shows, the distribution of K_n^+ is approximated by the function $F_\infty(x)$ for $x \geq 0$. As

K_n^+ includes a factor $\sqrt{n}$, and we see visually through figure 1.9 that the approximation is good, this means the inclusion of the $\sqrt{n}$ is warranted. We would draw the same conclusion when conducting a Kolmogorov-Smirnov, however the null and alternative hypothesis are adapted accordingly.

H_0 : The empirical distribution of test statistics $K_n^+(1), \ldots, K_n^+(r)$ are approximated by the function $F_\infty(x)$ for $x \geq 0$ and for large enough n.

H_1 : The test statistics $K_n^+(1), \ldots, K_n^+(r)$ are distributed differently.

It is better to calculate multiple $K_n^+(1), \ldots K_n^+(r)$ and $K_n^-(1), \ldots K_n^-(r)$ and then to calculate a final K_n^+ and K_n^- as this is greater at detecting local and global non random behaviour than calculating one single Kolmogorov-Smirnov test. We see this in the examples that follow. Before this the critical values must be calculated, which provide the boundaries for which we will accept or reject the null hypothesis. The Kolmogorov-Smirnov table contains some percentage points, however notably the 2.5% and 97.5% points are missing meaning a two sided 95% test cannot be constructed. For large enough n we use the approximation of $F(x)$ to $F_\infty(x)$, and solve for both 0.025 and 0.975 level significance to get boundary estimates. The estimate is applicable for both K_n^+ and K_n^-, and for when performing a single test, or the latter method of calculating K_n^+ and K_n^- from either$K_n^+(1), \ldots K_n^+(r)$ or $K_n^-(1), \ldots K_n^-(r)$.

$P(K_n^+ \leq a) \approx 1 - e^{-2a^2} \stackrel{!}{=} 0.025$

$\Leftrightarrow e^{-2a^2} = 0.975$

$\Leftrightarrow -2a^2 = \log 0.975$

$\Leftrightarrow a = \sqrt{-\frac{1}{2}\log 0.975}$

$\Leftrightarrow a = 0.1125$

Doing this calculation for 0.975, yeilds $a = 1.3581$.

Thus from this we can construct a two sided 95% test statistic where we accept values of K_n^+ and K_n^- when $0.1125 < K_n^+, K_n^- < 1.3581$.

1.2.5 Kolmogorov-Smirnov Test in R

The following functions have been written in R to calculate the Kolmogorov-Smirnov test statistic in the two ways explained in this report, both of which utilising K_n^+ and K_n^-. The first two functions calculate K_n^+ and K_n^- using either the empirical uniform function, or the $F_\infty(x)$ function, which the user chooses, based on the other input.

```
k.plus<-function(lcg, F){
  n<-length(lcg)
  lcg<-sort(lcg)

  d <- c()
  for (i in 1:n) {
   d[i] = i/n - F(lcg[i])
  }
  return(sqrt(n)*max(c(0, d)))
}
```

Where

```
F <- function(x) { return(punif(x)) }
```

if the input values are the raw LCG values and

```
F = function(x) { return(1-exp(-2*x^2)) }
```

if this function is calculating K_n^+ from $K_n^+(1), \ldots K_n^+(r)$ or K_n^- from $K_n^+(1), \ldots K_n^-(r)$.

The other input this function requires is a string of LCG values or a string of already computed K_n^+ or K_n^- values based on the same LCG but using different seeds. Given this input, labeled lcg in the code the function first orders the sample in ascending order using the sort() function. It then defines n to be the length of the input vector (the sample size). F is defined to be the cumulative function for the Uniform distribution or $F_\infty(x)$. Next d then assigned to be an empty vector. A loop is ran which for i taking values of 1 to n, is the difference between i/n and the uniform distribution value for the ith number in the vector lcg. The loop is ended and the maximum is taken of all these values and 0. We concatenate 0 (and specifically on the left side) as accounts for the maximum value on the interval $x \in (-\infty, U_{(1)})$. The maximum value attainable for $x \in [U_{(n)}, \infty)$ is at $U_{(n)}$ and is accounted for in the loop.

```
k.minus<-function(lcg, F){
  n<-length(lcg)
  lcg<-sort(lcg)

  d <- c()
  for (i in 1:n) {
   d[i] = F(lcg[i]) - (i-1)/n
  }
  return(sqrt(n)*max(c(d, 0)))
}
```

The function above calculates K_n^- in almost the same way as the function for K_n^+. There are some subtle changes to the formula aside from the change of $F_n(x) - F(x)$ in the code for K_n^+, to $F(x) - F_n(x)$ for K_n^-. Firstly, in the loop, to account for the largest distance between the functions now being found at the right most point of each step for $F_n(x)$, we now take $(i-1)/n$ away from the value obtained for the ith observation for F. Again, we concatenate 0 to the values before taking a maximum value, this time on the right side of the values to represent value $U_{(n)}$.

```
k.plus.seed <- function(n2,n,m,a,c){
  knp<-c()

  for (i in 1:n2){
   knp[i] = k.plus(LCG(n,m,a,c,i),function(x) { return(punif(x)) })
  }
  return(knp)
}
```

The function above calculates $n2$ K_n^+ statistics for a Linear Congruential Generator with values n, m, a, c and with starting seed i. Here, i takes values from $1, \ldots, n2$ in a loop and each value calculated is stored in a vector labeled knp. This vector is then returned and the function is ended.

```
k.minus.seed <- function(n2,n,m,a,c){
  knm<-c()

  for (i in 1:n2){
```

```
   knm[i] = k.minus(LCG(n,m,a,c,i),function(x) { return(punif(x)) })
  }
  return(knm)
}
```

The R function above calculates $n2$ K_n^- statistics for a Linear Congruential Generator with values n, m, a, c and with starting seed i. It uses the same method as the function written for K_n^+.

The above two functions are suitable for any LCGs that do not make any specifications about seed. RANDU requires an odd seed, hence I have written a slightly different functions for use with RANDU that will ensure only odd seeds are used. These are shown below.

```
k.plus.odd<-function(n2,n,m,a,c){
  knp<-c()

  for (i in seq(1, by = 2, len = n2)){
   knp[i] = k.plus(LCG(n,m,a,c,i))
      knp<-knp[!is.na(knp)]
  }
  return(knp)
}


k.minus.odd<-function(n2,n,m,a,c){
  knm<-c()

  for (i in seq(1, by = 2, len = n2)){
   knm[i] = k.minus(LCG(n,m,a,c,i))
      knm<-knm[!is.na(knm)]
  }
  return(knm)
}
```

1.2.6 Examples of the Kolmogorov-Smirnov Test on LCGs

As already explored, there exist multiple ways to apply the Kolmogorov-Smirnov Test to PRNGs. In this section I will apply two of these methods; firstly by applying the Kolmogorov-Smirnov test to our five LCGs outlined in the first section, with seed stated. Secondly I will apply a Kolmogorov-Smirnov test to these LCGs, however using a string of seeds for each LCG, and comparing these multiple Kolmogorov-Smirnov tests to the function $F_\infty(x)$.

Example 1.9. In this example we consider K_n^+ and K_n^- for LCG1 0.3, LCG2 0.4, LCG3 0.5, LCGNR 0.6 and RANDU 0.7. They are calculated using the 'k.plus' and 'k.minus' functions written in R. These statistics for each LCG are presented in the table below.

LCG	LCG1	LCG2	LCG3	LCGNR	RANDU
n	400	400	2000	2000	2000
K_n^+	0.8398	1.6805	0.1621	0.8253	0.6019
K_n^-	0.0680	1.9610	0.1017	0.7325	0.6622
Decision	Reject	Reject	Reject	Accept	Accept

Lower bound for K-S Test	Upper bound for K-S Test
0.1125	1.3581

For LCG1 we see a completely acceptable value for K_n^+, lying almost in the centre of the boundary values. This means that the observations do not deviate too far above the function $F(x)$ nor is the value too small that there is not enough local random behaviour. K_n^- is far too low, and well below the lower boundary, suggesting not enough randomness in the LCG when the observations are below $F(x)$. Thus we would reject the null hypothesis in this case and conclude this LCG does not have enough local random behaviour. This contrasts the result obtained when a Chi-square test was completed on LCG1 in which we accepted the null hypothesis an thus concluded that LCG1 was a sufficient LCG, however the value obtained was rather close to the lower bound.

For LCG2, the results are concordant with that of the Chi-square test. Both K_n^+ and K_n^- are far too large; they are both above the upper boundary and hence we conclude that the observations from LCG2 are not taken from the distribution $F(x)$.

For LCG3, we see that both K_n^+ and K_n^- are very small values. K_n^- is slightly below the boundary hence we would reject the null hypothesis, and although K_n^+ is in the acceptance region, it is also close to the lower bound. This means there is not enough random "noise" in LCG3 and is not behaving as as random as we would expect.

For LCGNR, our example of a good LCG, we observe excellent values for K_n^+ and K_n^-, being almost in the middle of the upper and lower critical values for the test. Hence we would not reject the null hypothesis, and conclude that these values appear to be sampled from the distribution stated.

Finally, for our poor LCG, RANDU, we also see excellent values for K_n^+ and K_n^-, again both are nowhere near the critical regions and hence we would not suspect RANDU is a poor LCG. We would accept the null hypothesis that these values were sampled from the stated distribution.

Example 1.10. In this second example we consider K_n^+ and K_n^- again for our five LCGs however we do not calculate these values using the observations, but instead calculate $K_n^+(1), \ldots, K_n^+(1000)$ and $K_n^-(1), \ldots, K_n^-(1000)$ and then calculate K_n^+ and K_n^- for both of these distributions. The distribution of $K_n^+(1), \ldots, K_n^+(1000)$ and $K_n^-(1), \ldots, K_n^-(1000)$ are the same hence we can calculate K_n^+ and K_n^- for both distributions. The number of observations has been set to $n = 1000$ for all LCGs as suggested to be a reasonable value for n by [Knu69]. This is done by using the 'k.plus.seed' and 'k.minus.seed' functions written for R, and then using the 'k.plus' and 'k.minus' functions written in R that calculate K_n^+ and K_n^- respectively, comparing to the distribution $F_\infty(x)$. The values are shown in the table below.

LCG	LCG1	LCG2	LCG3	LCGNR	RANDU
n	400	400	2000	2000	2000
K_n^+ of A^+	7.5233	0	27.4644	0.6662	0.0783
K_n^- of A^+	1.2926	22.4463	0.1204	0.7797	1.8568
K_n^+ of A^-	5.9687	0	28.0075	0.6650	1.9843
K_n^- of A^-	1.6096	20.9263	0	0.4346	0.0228
Decision	Reject	Reject	Reject	Accept	Reject

Where

$A^+ = K_n^+(1), \ldots, K_n^+(1000)$

and

$A^- = K_n^-(1), \ldots, K_n^-(1000).$

Lower bound for K-S Test	Upper bound for K-S Test
0.1125	1.3581

We see that for LCG1 the convergence to $F_\infty(x)$ is not sufficient enough for K_{1000}^+ and K_{1000}^-. The departure for both these plots is above the curve, and as such K_n^+ is the value responsible for the failure of the Kolmogorov-Smirnov Test for LCG1.

For LCG2 we witness no convergence to $F_\infty(x)$ from our empirical distribution. We strongly reject the null hypothesis due to incredibly high values for K_n^- for both K_{1000}^+ and K_{1000}^-. In both plots our empirical distribution lags severely behind our proposed distribution $F_\infty(x)$ under the null hypothesis.

For LCG3, we witness the opposite case from what we saw for LCG2. That is that for both K_{1000}^+ and K_{1000}^- the empirical distribution reaches 1 well before $F_\infty(x)$ does. We reach the same conclusion however, and reject the null hypothesis as K_n^+ for both K_{1000}^+ and K_{1000}^- is far too large.

For LCGNR, our empirical distributions closely mimics the empirical distribution for $F_\infty(x)$. We see only small deviations between the two distributions and as such all our test statistic values are found well within the acceptance region between the critical values calculated. As such we accept the null hypothesis for LCGNR and conclude that the distribution is that stated in the null hypothesis.

Finally, looking at the empirical plots for RANDU, we see that the empirical distribution for RANDU to be far closer to the empirical distribution for $F_\infty(x)$, than for LCG1, LCG2 and LCG3, however is not as good as for LCGNR. In fact, for K_{1000}^+, the empirical distribution for RANDU lags slightly behind $F_\infty(x)$ for the entire plot, not once overtaking. The opposite is the case for K_{1000}^-; the empirical distribution for RANDU lags slightly ahead of $F_\infty(x)$ for the entire plot. It is for these reasons that all 4 test statistics for RANDU fail. For K_{1000}^+, K_n^+ is too small and K_n^- is too large. For K_{1000}^-, K_n^+ is too large and K_n^- is too small.

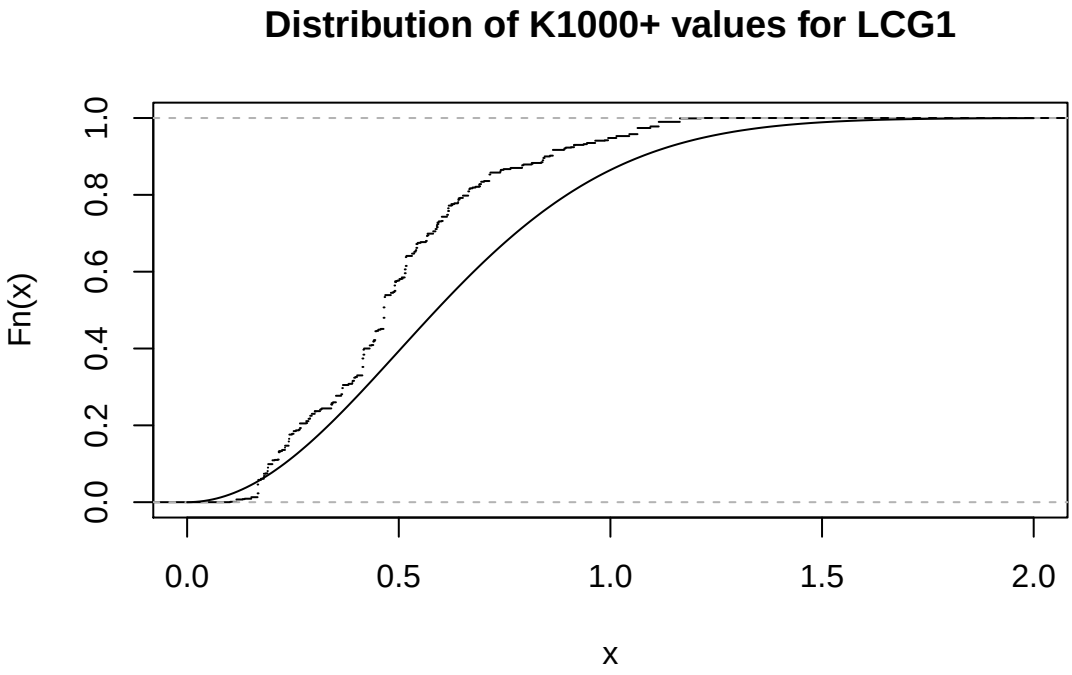

Figure 1.10: This figure shows the distribution of K^+_{1000} for LCG1. We see a rather poor approximation here; it is clear that the empirical observed values are not sufficiently akin to $F_\infty(x)$. We predict a large K^+_n will be obtained when we apply a Kolmogorov-Smirnov test here, due to the empirical observed function appearing far above the cumulative $F_\infty(x)$ function.

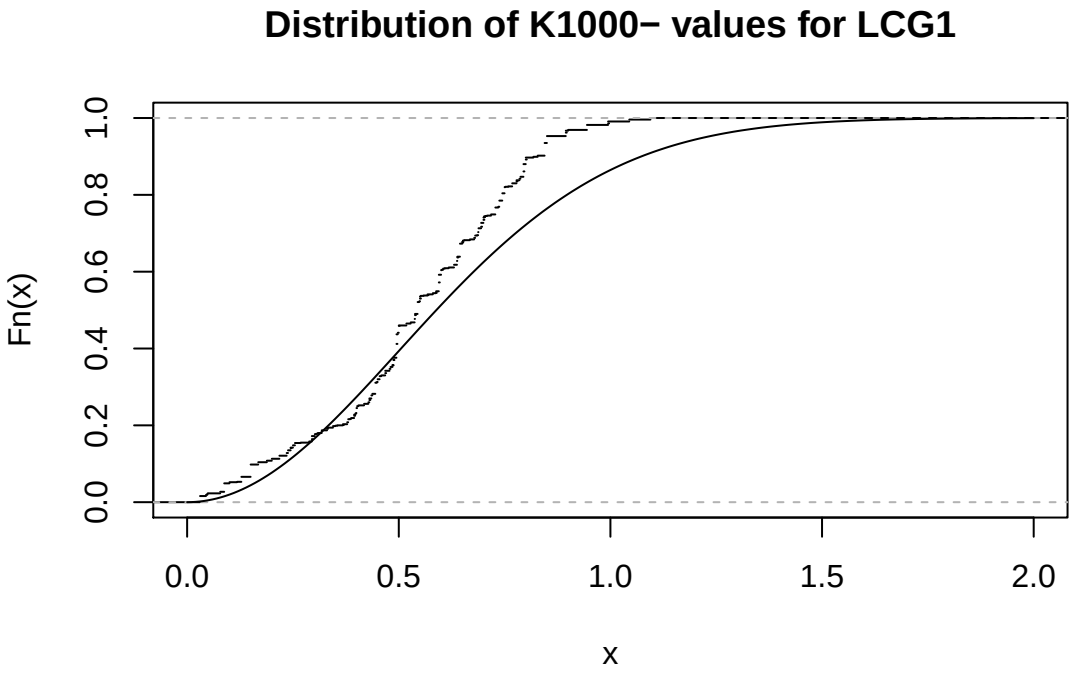

Figure 1.11: This figure shows the distribution of K^{-}_{1000} for LCG1. Again we observe that the output of LCG1 is poor, as is not well approximated by $F_{\infty}(x)$. The empirical cumulative function quickly rises in the middle of the plot, at a rate far exceeding $F_{\infty}(x)$. We predict a large value for K^{+}_{n} again, due to the locations of both functions (the observed function lying above our proposed distribution).

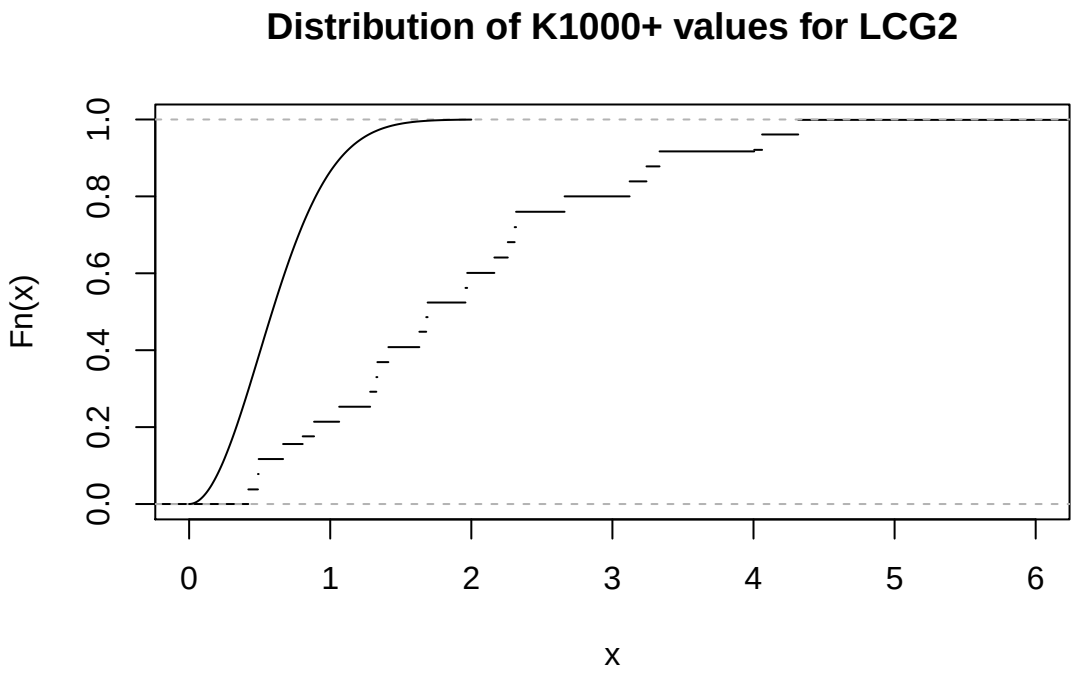

Figure 1.12: This figure shows the distribution of K^+_{1000} for LCG2. The distribution of observed values is not approximated at all by $F_\infty(x)$. We observe completely different functions; $F_\infty(x)$ approaches and reaches 1 far before the cumulative function for the data does. We predict that K^-_n will be incredibly large for LCG2 in this case, and it is apparent that K^+_n will be 0 here as $F_\infty(x)$ is always above the distribution for the data. This indicates LCG2 will fail the Kolmogorov-Smirnov test in this instance without question.

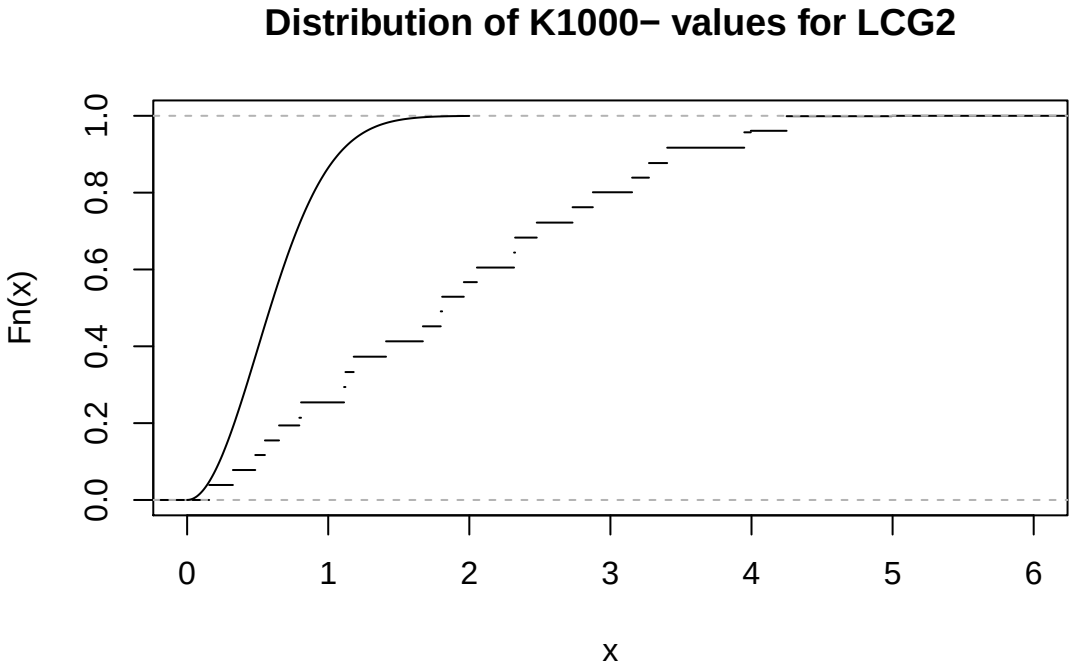

Figure 1.13: This figure shows the distribution of K_{1000}^{-} for LCG2. We see the same results as in the previous plot; an incredibly poor fit that indicates that the data is not approximated by $F_\infty(x)$ well. We predict that K_n^- will be incredibly large for LCG2 in this case, and it is apparent that K_n^+ will be 0 here as $F_\infty(x)$ is always above the distribution for the data. This indicates that LCG2 will fail the Kolmogorov-Smirnov test without question.

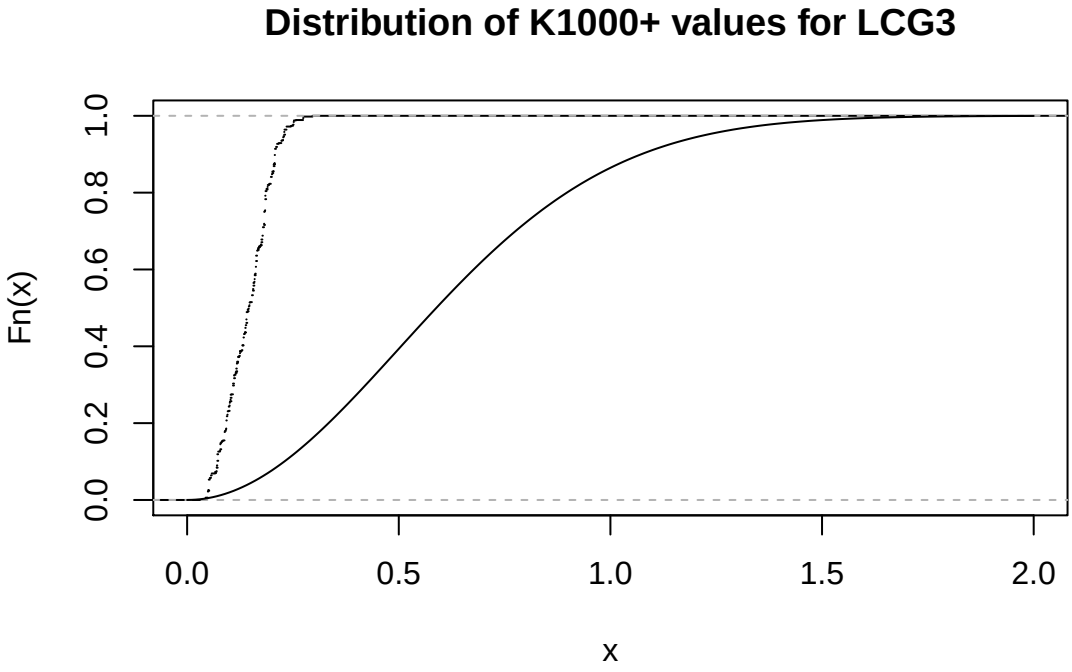

Figure 1.14: This figure shows the distribution of K^+_{1000} for LCG3. In this case we see that the observed cumulative distribution approaches and reaches 1 far before $F_\infty(x)$ does. It is clear that $F_\infty(x)$ does not approximate LCG3 well, and we can deduce that the output of LCG3 is flawed. It is clear that K^+_n will be large in this case and K^-_n will be either very small or 0 based on the behaviour of both distributions at the beginning of the plot which is difficult to see by eye.

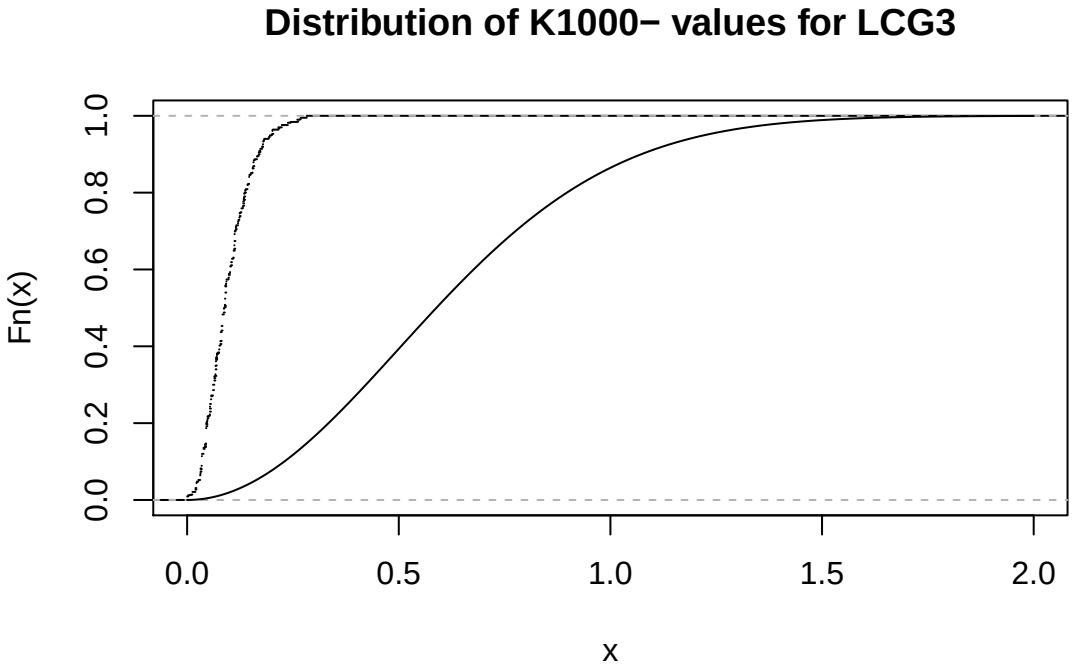

Figure 1.15: This figure shows the distribution of K^{-}_{1000} for LCG3. In this case we see that the observed cumulative distribution approaches and reaches 1 far before $F_{\infty}(x)$ does. It is clear that $F_{\infty}(x)$ does not approximate LCG3 well, and we can deduce that the output of LCG3 is flawed. It is clear that K^{+}_{n} will be large in this case and K^{-}_{n} will be either very small or 0 based on the behaviour of both distributions at the beginning of the plot which is difficult to see by eye

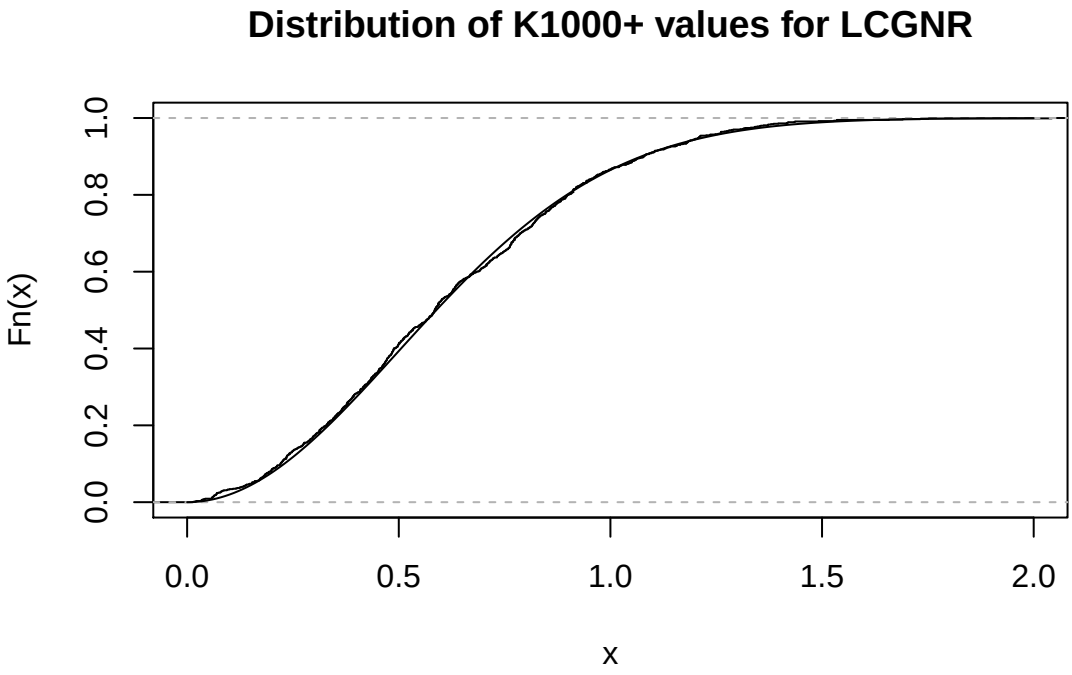

Figure 1.16: This figure shows the distribution of K^+_{1000} for LCGNR. Unlike previous plots, we see that $F_\infty(x)$ approximates the distribution of values for LCGNR very well. We predict that both K^+_n and K^-_n will be acceptable in this case and we see no reason to reject that K^+_{1000} for LCGNR is not distributed like $F_\infty(x)$.

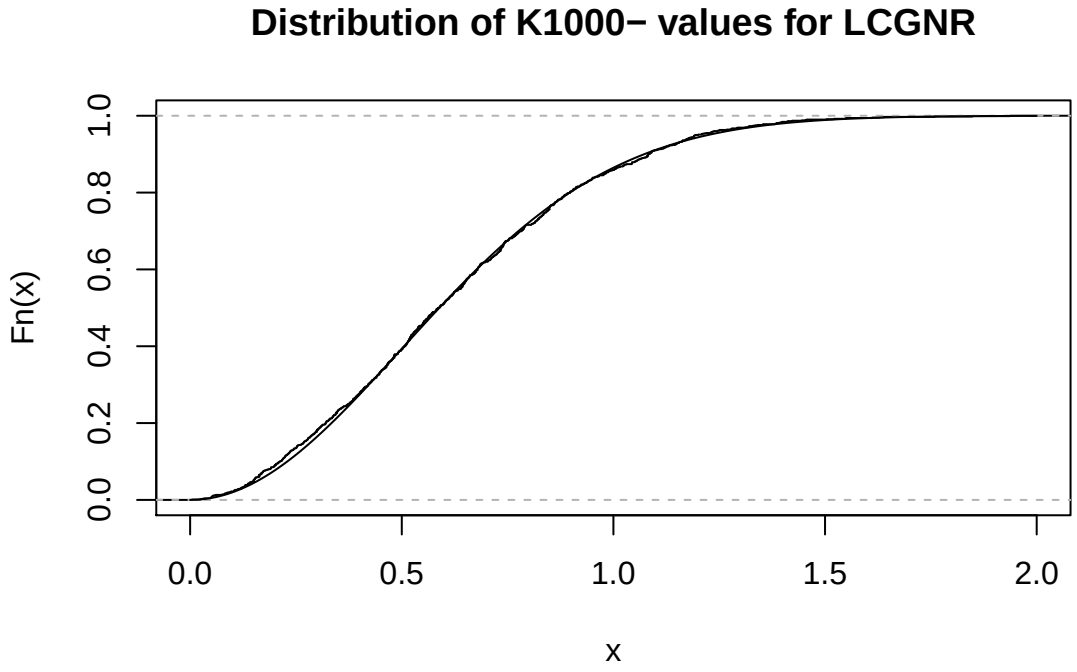

Figure 1.17: This figure shows the distribution of K_{1000}^{-} for LCGNR. We see that $F_\infty(x)$ approximates the distribution of values for LCGNR very well. We predict that both K_n^+ and K_n^- will be acceptable in this case and we see no reason to reject that K_{1000}^{+} for LCGNR is not distributed like $F_\infty(x)$.

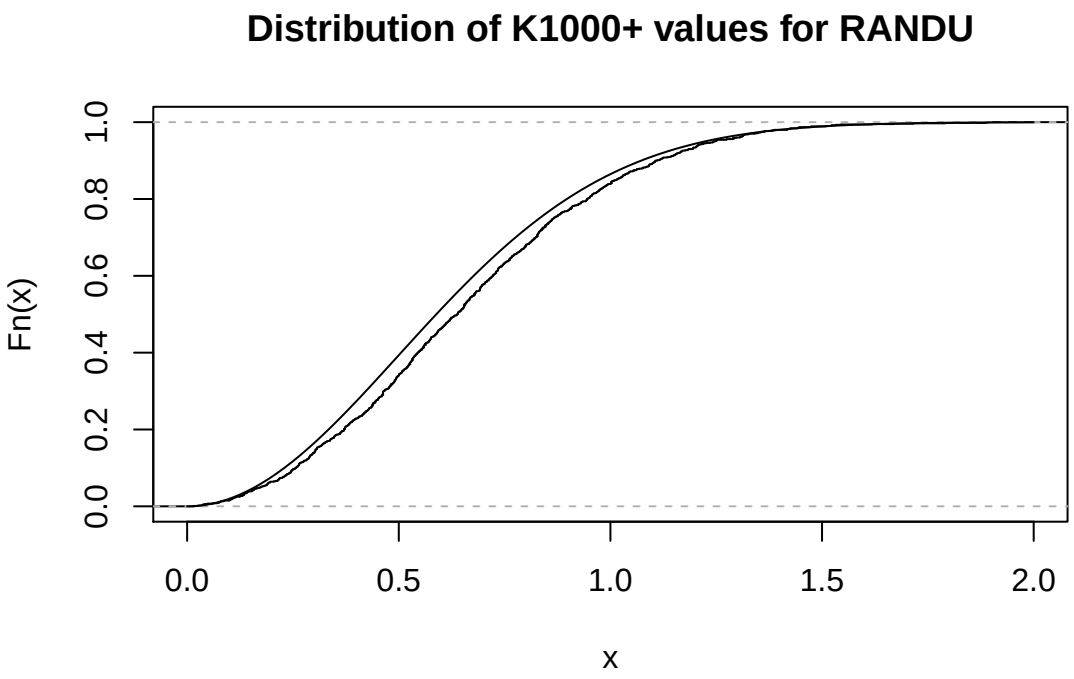

Figure 1.18: This figure shows the distribution of K_{1000}^{+} for RANDU. We see a decent approximation of the data by $F_{\infty}(x)$, however there appears to be a flaw meaning we will reject the approximation in this case. The cumulative distribution for our data lags slightly behind $F_{\infty}(x)$ for all of the plot, and is significant enough that we can strongly conclude that the approximation is not sufficed here. We would conclude that the output of RANDU is not sufficiently distributed correctly. We expect a value of K_{n}^{+} to be perhaps too small, and K_{n}^{-} may just lie in the upper critical region.

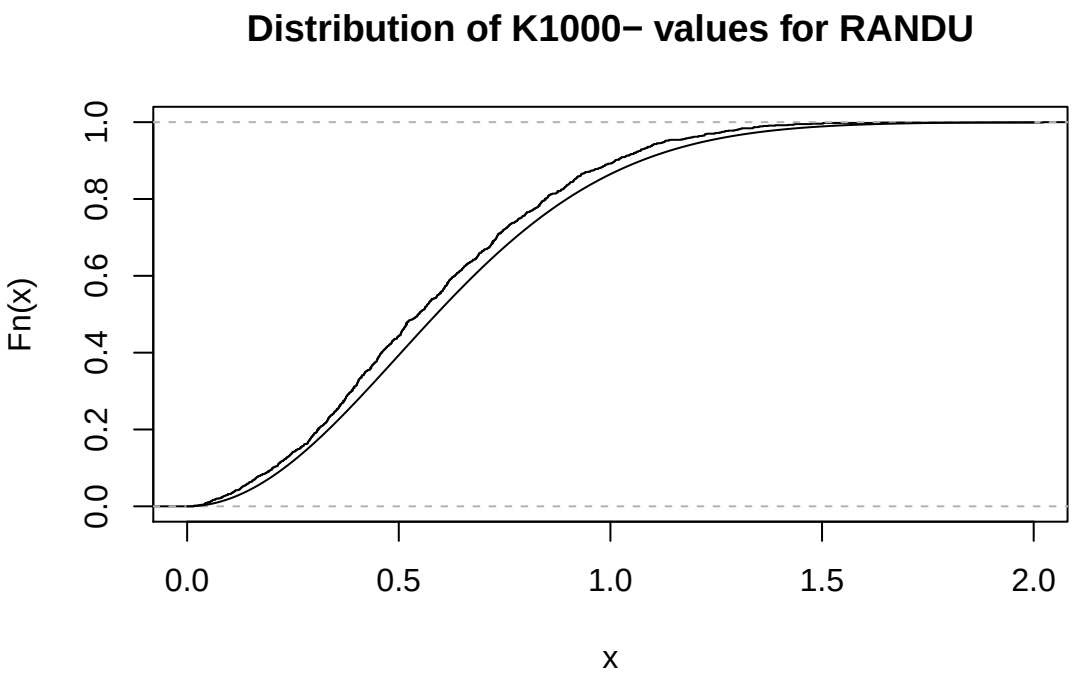

Figure 1.19: This figure shows the distribution of K^-_{1000} for RANDU. We see a plot that is the opposite of the previous plot for RANDU. In this case, the cumulative empirical distribution lags slightly ahead of the distribution $F_\infty(x)$, and although at first does not seem incredibly poor fitting like in the case for LCG1, LCG2 and LCG3, it is a large enough difference to cause serious concern. We would conclude that the output of RANDU is not sufficiently distributed correctly. We expect a value of K^-_n to be perhaps too small, and K^+_n may just lie in the upper critical region.

Chapter 2

Tests for Independence

In this section I introduce two tests for independence that can be applied to Pseudo Random Number Generators known as Spearman's Rank Correlation Coefficient and the Runs Test. I again apply these tests to the five LCGs detailed in the introductory chapter. Spearman's Rank Correlation Coefficient is only a coefficient and not a test in itself, hence I introduce a transformation that can be applied to the coefficient that makes in comparable to a normal distribution.

2.1 Spearman's Rank Correlation Coefficient

2.1.1 Description of the Test

Spearman's Rank Correlation Coefficient or Spearman's Rho is a coefficient obtained that is used used as a test of dependence. It is a non-parametric test and the coefficient obtained - expressed as ρ, takes a value in the interval $\rho \in [-1, 1]$. The test involves determining dependence between two variables; say X and Y. Considering a PRNG simply spews out a list of n numbers with no differentiation between them we define the all odd iterations from the algorithm; $U_1, U_3, U_5, \ldots, U_{n-1}$ to be from distribution X and all even numbers; $U_2, U_4, U_6, \ldots, U_n$ to be from distribution Y.

As a result of this it is therefore sensible to run the PRNG for an even number of iterations. Values from both X and Y are ranked according to their size based on others in the respective distribution and the test statistic uses the difference between ranks from X and Y to determine dependence. The null and alternative hypothesis of the test are defined to be

H_0 : The distributions X and Y are not correlated.

H_1 : The distributions X and Y are significantly correlated.

The first step to take once the output of a PRNG has been sorted into two different sets X and Y, is to assign numeric integer value to each of the X_i based on its value. The smallest X_i should take value 1, the next smallest value 2 and so on. If two X_i share the same value, then simply give one of the X_i the correct numeric position, for example k, and then assign the like value, although the same $k + 1$. The same should be done for values in Y.

Next, ranks can then be assigned independently to each X_i and Y_i. If there are no tied ranks then the integers assigned to values in the last step remain unchanged. We rename the values of X_i as x_i and Y_i as y_i in this step. In the case of tied ranks (where multiple values are equal) we count the position they are in, sum the position numbers and then divide by how many values are tied. For example if positions $1, 2$ and 3 correspond to equal values we then sum these positions $1 + 2 + 3 = 6$ and divide by how many

were tied - 3 values; giving us $\frac{6}{3} = 2$. The next value would take value 4 (assuming no tie ranks) and not 3.

The test statistic ρ is then defined to be

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \tag{2.1}$$

The test is contrasting the ranks between ith pairs; x_i and y_i. In the case of our PRNGs this test is therefore based on the correlation between pairs; $(x_1, y_1), \ldots, (x_n, y_n)$ which are the ranks of consecutive pairs $(U_1, U_2), \ldots (U_{n-1}, U_n)$.

2.1.2 An alternative Form of ρ

Another form of ρ also exists; one that requires no tied ranks. This section derives this alternate form from the one previously stated.

Lemma 2.1. Assume that there are no tied ranks, *i.e.* that $x_i \neq x_j$ whenever $i \neq j$. Then the Spearman rank correlation coefficient ρ from (2.1) can be written as

$$\rho = 1 - \frac{6 \sum_i {d_i}^2}{n(n^2 - 1)},$$

where $d_i = x_i - y_i$ for $i = 1, 2, \ldots, n$.

Proof. This proof was not taken from any other material, and was calculated in full by the author of this project, using the results from analysis below.

Under the condition that there are no tied ranks we can use the following result from analysis. The first result, shown below, can be proved very simply by induction and is a standard result that can be found in any Analysis textbook. As such it is omitted from this report.

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

This is true under the situation where X and Y both have an equal amount of values, and there are no tied ranks, hence $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ are some permutation of the numbers $1, \ldots, n$ and hence the summations are equal.

Another result from analysis, albeit lesser known, is that

$$\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i^2 = \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$$

This uses the same reasoning as the previous result. Proof of this result is worked through below, using induction.

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

The condition obviously holds for the case of 0. We now see for the case of $n = 1$ that

$$\sum_{i=1}^{1} i^2 = 1^2 = 1 = \frac{6}{6} = \frac{1 \cdot 2 \cdot 3}{6} = \frac{n(n+1)(2n+1)}{6}.$$

Hence the result holds for the case of $n = 1$. The inductive step follows; here assume that for k

$$\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}.$$

We now consider the case of $k + 1$, giving

$$\sum_{i=1}^{k+1} i^2 = \frac{(k+1)(k+2)(2(k+1)+1)}{6}.$$

Then using the assumption

$$\begin{aligned}
\sum_{i=1}^{k+1} i^2 &= \sum_{i=1}^{k} i^2 + (k+1)^2 = \frac{k(k+1)(2k+1)}{6} + (k+1)^2 \\
&= \frac{k(k+1)(2k+1)}{6} + \frac{6(k+1)^2}{6} \\
&= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6}.
\end{aligned}$$

Next we take out $(k + 1)$ as a factor from the numerator and continue manipulation,

$$\begin{aligned}
&\frac{k(k+1)(2k+1) + 6(k+1)^2}{6} \\
&= \frac{(k+1)(k(2k+1) + 6(k+1))}{6} \\
&= \frac{(k+1)(2k^2 + k + 6k + 6)}{6} \\
&= \frac{(k+1)(2k^2 + 7k + 6)}{6} \\
&= \frac{(k+1)((2k+3)(k+2))}{6} \\
&= \frac{(k+1)(k+2)(2k+3)}{6} \\
&= \frac{(k+1)(k+2)(2(k+1)+1)}{6}.
\end{aligned}$$

Hence the proof is complete and we have used k to prove for $(k+1)$.

Finally, using the first result we find a way to rewrite $\bar{x}$ and $\bar{y}$

$$\bar{x} = \bar{y} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{\sum_{i=1}^{n} i}{n} = \frac{n(n+1)}{2n} = \frac{(n+1)}{2}.$$

With these tools we can now begin the proof, starting at the general form of ρ.

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

For clarity, the numerator and denominator shall be expanded separately. For the numerator of ρ we find

$$\begin{aligned}
&\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \\
&= \sum_{i=1}^{n}(x_i y_i - x_i\bar{y} - y_i\bar{x} + \bar{x}\bar{y}) \\
&= \sum_{i=1}^{n} x_i y_i - \bar{y}\sum_{i=1}^{n} x_i - \bar{x}\sum_{i=1}^{n} y_i + n\bar{x}\bar{y} \\
&= \sum_{i=1}^{n} x_i y_i - \frac{\bar{y}n(n+1)}{2} - \frac{\bar{x}n(n+1)}{2} + n\bar{x}\bar{y} \\
&= \sum_{i=1}^{n} x_i y_i - \frac{(n+1)}{2}\frac{n(n+1)}{2} - \frac{(n+1)}{2}\frac{n(n+1)}{2} + n(\frac{(n+1)}{2})^2 \\
&= \sum_{i=1}^{n} x_i y_i - \frac{(n+1)^2 n}{4} - \frac{(n+1)^2 n}{4} + \frac{(n+1)^2 n}{4} \\
&= \sum_{i=1}^{n} x_i y_i - \frac{(n+1)^2 n}{4}.
\end{aligned}$$

Next we work with the denominator of ρ.

$$\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}$$

$$=\sqrt{\sum_{i=1}^{n}(x_i^2-2\bar{x}x_i+\bar{x}^2)\sum_{i=1}^{n}(y_i^2-2\bar{y}y_i+\bar{y}^2)}$$

$$=\sqrt{(\frac{n(n+1)(2n+1)}{6}-2\bar{x}(\frac{n(n+1)}{2})+n\bar{x}^2)(\frac{n(n+1)(2n+1)}{6}-2\bar{y}(\frac{n(n+1)}{2})+n\bar{y}^2)}$$

$$=\sqrt{\frac{n(n+1)(2n+1)}{6}-2\frac{(n+1)}{2}(\frac{n(n+1)}{2})+n(\frac{(n+1)}{2})^2}$$
$$\times\sqrt{\frac{n(n+1)(2n+1)}{6}-2\frac{(n+1)}{2}(\frac{n(n+1)}{2})+n(\frac{(n+1)}{2})^2}$$

$$=\sqrt{(\frac{n(n+1)(2n+1)}{6}-2\frac{(n+1)}{2}(\frac{n(n+1)}{2})+n(\frac{(n+1)}{2})^2)^2}$$

$$=\frac{n(n+1)(2n+1)}{6}-\frac{n(n+1)^2}{2}+\frac{n(n+1)^2}{4}$$

$$=\frac{n(n+1)(2n+1)}{6}-\frac{n(n+1)^2}{4}$$

We then reconstruct ρ using these expanded forms to get an idea of where this proof is progressing.

$$\rho=\frac{\sum_{i=1}^{n}x_iy_i-\frac{(n+1)^2n}{4}}{\frac{n(n+1)(2n+1)}{6}-\frac{n(n+1)^2}{4}}$$
$$=\frac{12\sum_{i=1}^{n}x_iy_i-3(n+1)^2n}{2n(n+1)(2n+1)-3n(n+1)^2}$$

In the previous step all terms are multiplied by a factor of 12. Once again we turn our attention to the denominator of ρ.

$$2n(n+1)(2n+1)-3n(n+1)^2$$
$$=(2n^2+2n)(2n+1)-3n(n^2+2n+1)$$
$$=4n^3+6n^2+2n-3n^3-6n^2-3n$$
$$=n^3-n$$
$$=n(n^2-1)$$

The denominator is now in the required form. Hence we return to the numerator.

$$
\begin{aligned}
&12\sum_{i=1}^{n} x_i y_i - 3(n+1)^2 n \\
&= 12\sum_{i=1}^{n} x_i y_i - 3n(n^2+2n+1) \\
&= 12\sum_{i=1}^{n} x_i y_i - 3n^3 - 6n^2 - 3n \\
&= 12\sum_{i=1}^{n} x_i y_i + (n^3-n) - 4n^3 - 6n^2 - 2n
\end{aligned}
$$

In the last step a factor of $(n^3 - n) = n(n^2 - 1)$ was separated from the other n terms to make the 1 we require later. We next factorise the remaining n terms.

$$
\begin{aligned}
&12\sum_{i=1}^{n} x_i y_i + (n^3-n) - 4n^3 - 6n^2 - 2n \\
&= 12\sum_{i=1}^{n} x_i y_i + (n^3-n) - (2n^2+2n)(2n+1) \\
&= 12\sum_{i=1}^{n} x_i y_i + n(n^2-1) - 2n(n+1)(2n+1) \\
&= 12\sum_{i=1}^{n} x_i y_i + n(n^2-1) - n(n+1)(2n+1) - n(n+1)(2n+1) \\
&= n(n^2-1) + 12\sum_{i=1}^{n} x_i y_i - 6\sum_{i=1}^{n} x_i^2 - 6\sum_{i=1}^{n} y_i^2
\end{aligned}
$$

The last step was achieved by inserting the results from the previously stated formulas for $\sum_{i=1}^{n} x_i^2$ and $\sum_{i=1}^{n} y_i^2$). Finally we can factorise this as follows.

$$
\begin{aligned}
&n(n^2-1) + 12\sum_{i=1}^{n} x_i y_i - 6\sum_{i=1}^{n} x_i^2 - 6\sum_{i=1}^{n} y_i^2 \\
&= n(n^2-1) - 6\sum_{i=1}^{n} (x_i^2 + y_i^2 - 2x_i y_i) \\
&= n(n^2-1) - 6\sum_{i=1}^{n} (x_i - y_i)^2
\end{aligned}
$$

We piece the numerator and denominator back together to finish this proof.

$$\begin{aligned}\rho &= \frac{n(n^2-1) - 6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2-1)} \\ &= \frac{n(n^2-1)}{n(n^2-1)} - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2-1)} \\ &= 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n(n^2-1)} \\ &= 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}\end{aligned}$$

(as we write $x_i - y_i = d_i$).

Thus the proof is complete and we see under the condition that there are no tied ranks, the two forms of ρ are equivalent. □

2.1.3 Interpreting the Coefficient

Like other correlation coefficients such as the Pearson correlation coefficient, our coefficient ρ lies somewhere on the interval $[-1, 1]$. However the Pearson correlation coefficient is only sensitive to the linear relationship between variables. A correlation of $+1$ or -1 would indicate a perfect linear relationship between two variables and a value of 0 would indicate no linear relationship.

The Spearman's Rank Correlation Coefficient measures how well a monotonic function describes the relationship between variables, so is able to detect dependencies that the Pearson correlation coefficient would not recognise. Further if a value of $\rho = +1$ was obtained then a monotonic function could perfectly describe the relationship between the variables, and similarly if a value of $\rho = -1$ was obtained. A value of $\rho = 0$ would indicate no monotonic relationship or no correlation between variables.

However the range of possible values are found on a scale with some degree of interpretation necessary and as such a problem arises. For example what about for values or ρ such as 0.3, -0.5 or 0.7? How do these values relate to the null and alternative hypotheses previously stated? We need to be able to give a definite answer as to whether there is significant dependence between the variables (consecutive pairs in a PRNG sequence) or not.

We can use the Fisher Transformation to make Spearman's coefficient value comparable to the normal distribution. Such transformation applied to Spearman's Rank Correlation Coefficient is explored in detail in 'Tests For Rank Correlation Coefficients: II' [FP61]. The transformation is defined there on pg 29 as follows

$$F(r) = \frac{1}{2}\ln\frac{1+r}{1-r} = \tanh^{-1}(r).$$

In this transformation r is a correlation coefficient; a measure of correlation on the interval $[-1, 1]$. It is also applicable to other correlation coefficients such as the Pearson Correlation Coefficient.

The transformation is distributed normally with the parameters

$$F(r) \sim \mathcal{N}\left(\frac{1}{2}\ln\frac{1+\rho}{1-\rho}, \frac{1.06}{n-3}\right).$$

Then following from this $\sqrt{\frac{n-3}{1.06}}F(r)$ will follow the standard normal distribution. This can be verified

rather simply, as shown below. We first consider the expectation, remembering under the null hypothesis that $\rho = 0$.

$$\begin{aligned}\mathbb{E}(F(r)) &= \frac{1}{2}\ln\frac{1+\rho}{1-\rho}\\ &= \frac{1}{2}\ln 1\\ &= \frac{1}{2}0\\ &= 0\end{aligned}$$

Then we can verify that the expectation of $\sqrt{\frac{n-3}{1.06}}F(r)$ will also be 0.

$$\begin{aligned}&\mathbb{E}(\sqrt{\frac{n-3}{1.06}}F(r))\\ &= \sqrt{\frac{n-3}{1.06}}\mathbb{E}(F(r))\\ &= \sqrt{\frac{n-3}{1.06}}0\\ &= 0\end{aligned}$$

Next we confirm that $\sqrt{\frac{n-3}{1.06}})^2\text{var}(F(r)) = 0$, using the fact that $\text{var}(F(r)) = \frac{1.06}{n-3}$.

$$\begin{aligned}&\text{var }(\sqrt{\frac{n-3}{1.06}}F(r))\\ &= (\sqrt{\frac{n-3}{1.06}})^2\text{ var }(F(r))\\ &= \frac{n-3}{1.06}\text{ var }(F(r))\\ &= \frac{n-3}{1.06}\frac{1.06}{n-3}\\ &= 1\end{aligned}$$

Thus we have confirmed that the transformed version of $F(r)$, under the null hypothesis ($\rho = 0$) has distribution

$$\sqrt{\frac{n-3}{1.06}}F(r) \sim \mathcal{N}(0,1).$$

For PRNGs, where we may be taking very large samples of numbers (usually in the hundreds or thousands) there is no worry if sample size is enough, even considering sample size is halved as we must split the PRNG sequence splitting into both X and Y to calculate a value for ρ. In 'Tests For Rank Correlation Coefficients: II' [FP61] the transformation is tested for $n = 10, 30$ and 50. The largest change in variance occurred in the case for $n = 10$ with less change observed for $n = 30$ and 50. Even in the case of $n = 10$ the change was minimal and thus we need not worry about sample size as PRNG are typically used to generate large samples of numbers.

So using the transformation previously defined, we can now use a standard normal table or equivalent R command to make a definitive decision as to whether we can detect dependence between consecutive pairs of output values from a PRNG.

2.1.4 Tests for Rank Correlation Coefficients

In this section I concentrate on Spearman's Correlation Coefficient and the application of Fisher's Transformation to this coefficient. Here I will closely follow the work by Pearson, Fieller and Hartley in 'Tests for Rank Correlation Coefficients. I' [FPH57]. The paper considers three coefficients; Spearman's coefficient, Kendall's coefficient and the Fisher-Yates coefficient. Although all are of merit, we are only concerned with Spearman's coefficient. Tied ranks are not of concern in the paper as stated on pg 470 [FPH57]. The papers aims coincide with our aims; we are looking for a transformation that can be applied to Spearman's Rank Correlation Coefficient that will make the variance of the distribution independent of ρ. Under this transformation, comparison with other populations and distributions become viable techniques.

Following from Fieller et al [FPH57] we rename ρ to be r_s as will be considering ρ to be the population value correlation coefficient, and r_s to be the sample value correlation coefficient. We first consider some already known results from prior papers about the distribution of test statistic r_s. The Expectation of r_s is taken from Moran [Mor48] and has form

$$\mathbb{E}(r_s) = \frac{6}{(n+1)\pi}\{\sin^{-1}\rho + (n-2)\sin^{-1}\frac{1}{2}\rho\}.$$

Taken from Kendall, and David, Kendall & Stuart, an approximation for the variance of r_s has form

$$\text{var}(r_s) = \frac{1}{n}(1 - 1.563465\rho^2 + 0.304743\rho^4 + 0.155286\rho^6 + 0.061552\rho^8 + 0.022099\rho^{10} + ...).$$

Stated on page 471 of [FPH57], this large sample approximation for the variance of r_s is not very accurate for small n, such as $n = 10$. However, is of little concern for our use of this, as the PRNGs we are testing have output values in the hundreds or thousands.

Looking at these statistics, in particular the variance of r_s we see that it, and thusly the standard deviation of r_s are therefore dependent on ρ. This means different values of ρ will yield different variance (and standard deviation). Therefore the shape of the distribution changes with ρ, a result parallel to the Pearson Product Moment Correlation Coefficient. Therefore using rank coefficients as a test for independence is difficult. This is because of the problems that arise in comparing correlation in different populations. This would not be possible considering values of ρ may be different. For example how can we compare coefficient values between different PRNGs to get an idea of how independent output values for each are, when the distributions are different shapes.

Pearson, Fieller and Hartley first consider the ranks of generated samples from a bivariate normal with correlation ρ, on pg 472 [FPH57]. Extensive sampling experiments coinciding with analytical approximation, it has been shown that the following transformation is approximately normal, with a variance independent of ρ

$$z_s = F(r_s) = \frac{1}{2}\ln\frac{1+r_s}{1-r_s} = \tanh^{-1}(r_s)$$

for n not too large, and has variance approximately

$$\text{var}(z_s) \sim \frac{1.06}{n-3}.$$

Due to the the var(r_s) being an approximation, the expectation of z_s is poor for small n. Due to the nature of Spearman's correlation coefficient using rankings, this transformation can be applied to a variety of different distributions of paired variables X and Y. This means for our case of PRNGs, generating samples from a Uniform distribution, the transformation is valid. Under the transformation, the transformation of the correlation ρ, z_s can be considered normal with variance independent of ρ and based only on n. Also, z_s is an unbiased estimator of some function of the correlation coefficient ρ. Now z_s can be used to determine any underlying relationship between samples of a population. We can compare with a specified normal distribution to give a definitive result at a given significance.

In the paper by Pearson, Hartley and Fieller [FPH57], the experimental distribution of Spearman's Rank Correlation Coefficient was explored using 25000 sets of correlated normal deviates. These were then split into samples of varying size. There were 2500 samples contained $n = 10$, 833 samples with $n = 30$ and finally 500 with $n = 50$. Each were done on the same 25000 sets of correlated deviates, however it was sampled independently between for each different value of n.

Although worked through, Fieller et al [FPH57] compared the theoretical mean with that observed experimentally and found results to be satisfactory and omitted due to the fact the expectation taken from [Ken49] and [KDS51] was exact and not an approximation.

The variance of r_s, was an approximation and thus the empirical results are displayed in more detail by Fieller el al in their paper [FPH57]. Though the empirical exploration they performed, a 'purely empirical' adjustment is obtained to the variance of r_s taking form

$$\text{var}(r_s) = \frac{1}{n-1}(1 - 1.563465\rho^2 + 0.304743\rho^4 + 0.155286\rho^6 + 0.061552\rho^8 + 0.022099\rho^{10} + 0.019785\rho^{12}).$$

This adjustment is given as the equation for variance of r_s given by Kendall in [Ken49] and [KDS51] did not give the right answers for $\rho = 0$ and $\rho = 1$, where respectively are $\frac{1}{n-1}$ and 0. The adjustment takes care of this discrepancy. Tabulated in [FPH57] is data obtained from the experiment. They show, for varying values of $\rho = 0.1, 0.2, \ldots, 0.9$ the adjusted theoretical value for var(z_s), and the observed experimental values. This is done for the three sample sizes of n previously mentioned; $n = 10$, $n = 20$ and $n = 50$. For the case of $n = 10$ and to a lesser extent $n = 30$, the theoretical variance of z_s is consistently smaller than calculated from the observed samples. For $n = 50$ this is not the case, and considering for PRNGs we are taking our n to be far larger than 50, this is a good result.

The approximation to normal is found to be best at $\rho = 0$, and for values of $|\rho| \to 1$ the distribution behaves less normal and becomes skewed. The extra term added to the adjusted variance of z_s becomes increasingly less important for larger n, and difference between the theoretical variances and empirical variances are found to be insignificant except for the case of $\rho = 0.9$. Again, this is another good result, as under our null hypothesis ($\rho = 0$), ρ is small.

2.1.5 Spearman's Rank Correlation Coefficient in R

To test PRNGs in R, I have written several functions that can be utilised to do this. They are displayed below.

```
sp.rank<-function(lcg){
   n<-length(lcg)
   X<-c()
   Y<-c()
```

```
for(i in 1:(n/2)){
X[i]<-lcg[2*i-1]
}
for(j in 1:(n/2)){
Y[j]<-lcg[2*j]
}
   d<-rank(X)-rank(Y)
   m=n/2
   rho<-(1 - (6*sum(d^2))/(m*((m^2)-1)))
   return(rho)
}
```

The function above calculates ρ using the alternate form of ρ we derived previously. This form assumes no tied ranks, hence would not be the safest choice to test PRNGs unless values were checked for duplicates. This function only requires the output of a PRNG or LCG as an input. n is defined to be the length of the input vector; how many values of the PRNG there are. X and Y are two empty vectors. A loop is then ran where the odd elements of the input vector lcg become the elements of X and the even elements of lcg become the elements of Y. the difference d is then the rank of X minus the rank of Y, using the R function rank(). We then take $m = \frac{n}{2}$ due to each population having size $\frac{n}{2}$. We then take rho to have form shown previously, using what we have calculated so far, and return this value to finish the function.

```
sp.rank2<-function(lcg){
 n<-length(lcg)
   X<-c()
   Y<-c()
for(i in 1:(n/2)){
X[i]<-lcg[2*i-1]
}
for(j in 1:(n/2)){
Y[j]<-lcg[2*j]
}
   x<-rank(X)
   y<-rank(Y)

rho <- sum((x-mean(x))*(y-mean(y)))/sqrt(sum((x-mean(x))^2)*sum((y-mean(y))^2))
return(rho)
}
```

The function above calculates ρ using the broader form which does not assume no tied ranks. It is written in much the same way as the previous function, however just calculates ρ differently at the end, as expected.

```
sp.nml<-function(lcg,a){
    n<-length(lcg/2)
    rho<-sp.rank2(lcg)
    Z<-sqrt((n-3)/1.06)*atanh(rho)
   if(Z>qnorm(a/2) && Z<qnorm(a/2,lower.tail=F)){
   return(list("ACCEPT", Z))
}else{
return(list("REJECT", Z))
}
}
```

The function above is the most helpful, and uses sp.rank2() to give a definite answer as to whether there is significant correlation detected. This function requires lcg as an input, like the previous two functions

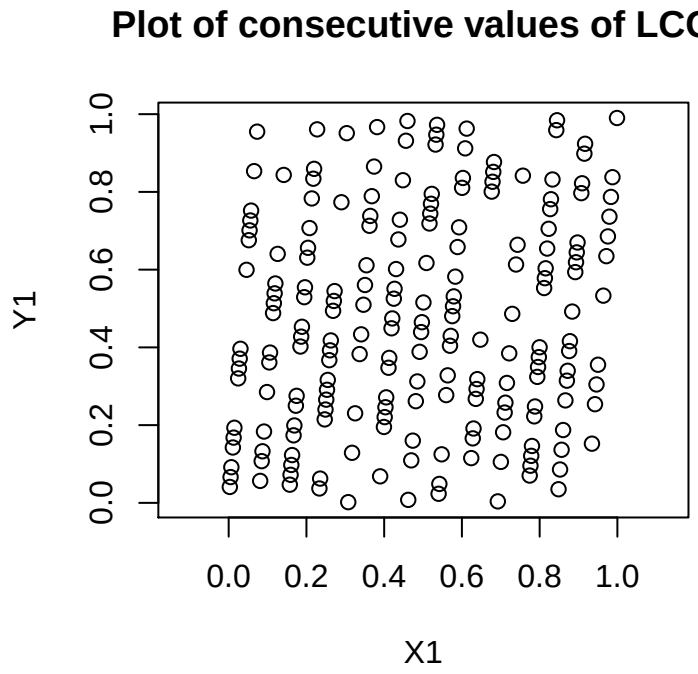

Figure 2.1: This figure shows the consecutive pairs from LCG1

and a which is a significance level. n is the length of the lcg divided by two due to splitting the values into two populations. *rho* is then calculated using previous function sp.rank2(). Z is then defined to be the Fisher Transformation applied to ρ then multiplied by the factor which will make it comparable to the standard normal distribution. Then if this value Z is between the upper and lower critical values of the standard normal distribution using $a/2$ then "ACCEPT" is returned along with the value Z, and if it is outside these boundaries then "REJECT" is returned to user along with value Z.

2.1.6 Examples of Spearman's Rank Correlation Coefficient on LCGs

Using the R code from the previous chapter, we can now apply Spearman's Rank Correlation Coefficient to some example PRNGs. We again visit our five LCGs and calculate Spearman's Rank Correlation Coefficient for each of them.

Example 2.1. We return again to LCG1 0.3 from previous examples. As previously stated, Spearman's Coefficient applied to PRNGs is about testing the correlation between consecutive numbers in an output. As such a plot distributions X and Y might be helpful in detecting any correlation before we use Spearman's coefficient. The plot of consecutive pairs (x_i, y_i) where x_i are the odd number output values and y_i are the even number output values of LCG1 and is shown below.

The plot clearly shows a non random pattern is present between the pairs. All points lie on one of thirteen different lines. From this alone we can see that perhaps LCG1 is a poor PRNG and consecutive pairs are very correlated. We can confirm this by using Spearman's Correlation Coefficient R function written previously. The function required a significance level as an input, and for consistency that will be at the 0.05 level (remembering that the test is two sided). At this level we will accept ρ under the condition that $-1.96 < \rho < 1.96$ otherwise we would reject the null hypothesis that there is no correlation between consecutive values. Below is the test statistic derived from R and the corresponding decision.

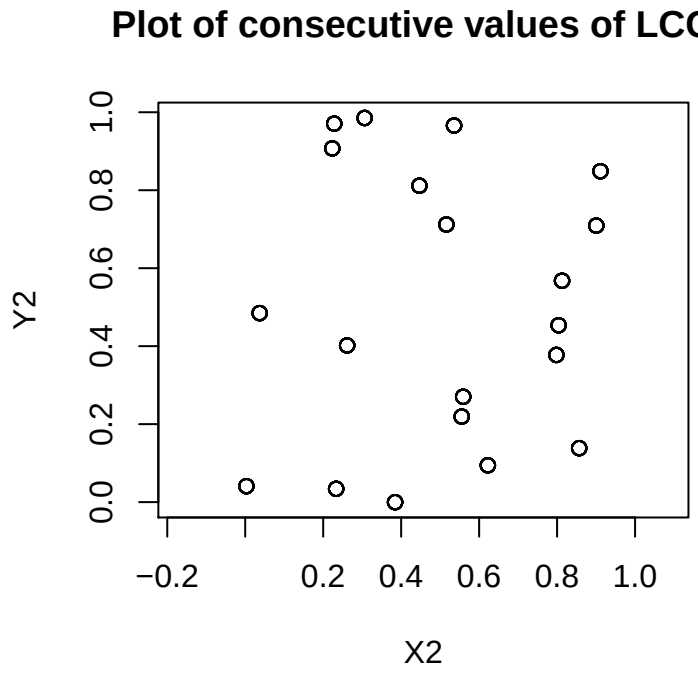

Figure 2.2: This figure shows the consecutive pairs from LCG2

ρ	Decision
3.1724	Reject

The value of 3.172422 is way above the upper boundary, so we strongly reject the null hypothesis and conclude that there is some correlation between consecutive values, in this case.

Example 2.2. Next, we consider LCG2 0.4, which has so far performed poorly in the tests we have applied to it. Let us consider a plot of consecutive pairs, which may reveal any correlation between pairs of values.

The plot for LCG2 is rather interesting. There is certainly not 200 points, as there should be. This is because as the LCG has a small period length, and due to repetitions points are repeated several times, which isn't shown well on the plot. As a result of this, the limited amount of points that are present do appear uncorrelated. This is also reflected in Spearman's Correlation Coefficient, shown below in the table.

ρ	Decision
-0.2328	Accept

As there are so few values, the resulting value for ρ is small, and within range hence we cannot reject the null hypothesis in this case. Better judgment, however, tells us that due to repetition of so many identical points hints that although consecutive points are not seemingly correlated, there is definitely some correlation present, that other tests so far have detected.

Example 2.3. We now move on to LCG3 0.5, a plot of consecutive paired values is shown below.

As LCG3 is simply LCG1 ran for a larger amount of values than its period, the plot is what we would expect. The lines shown in the plot for LCG1 contain even more points. The value of ρ is very large, much like for LCG1 and thus we again reject the null hypothesis and conclude there is significant correlation between consecutive values.

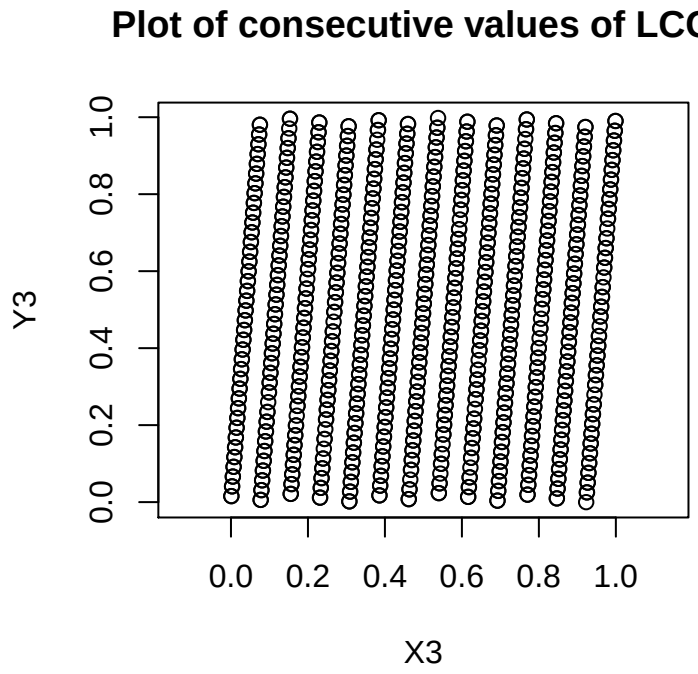

Figure 2.3: This figure shows the consecutive pairs from LCG3

ρ	Decision
3.31264	Reject

Example 2.4. We next move on to LCGNR 0.6, and consider a plot similar to those in our previous examples.

The plot is typical of what we would expect from a good random number generator; the appears to be no such correlation present as there is no noticeable pattern to the points. We then seek a definitive answer by calculating Spearman's Rho.

ρ	Decision
-0.9808	Accept

A value of -0.9808 is well within the acceptance region, and thus we deduce that this test provides no evidence to reject our null hypothesis. Further we can say with confidence that there appears to be no correlation between pairs of values for LCGNR.

Example 2.5. Finally, we consider our last LCG; RANDU 0.7. A plot of consecutive pairs for RANDU is considered before we compute Spearman's Rho.

Assessing this plot by eye, values for RANDU appear sufficiently random. No regular pattern appears to be prevalent in the plot and there appears to be no correlation between pairs. I would personally conclude that values of RANDU appear to be sufficiently uncorrelated. We now consider Spearman's Rho to see if the test agrees with our preliminary analysis of RANDU.

ρ	Decision
-2.2275	Reject

Our initial impression of RANDU appearing sufficiently random is not backed up by Spearman's Rank Correlation Coefficient in this case. A large negative value of -2.2275 means we reject the null hypothesis and conclude that consecutive pairs of RANDU are significantly correlated. It appears to be the case

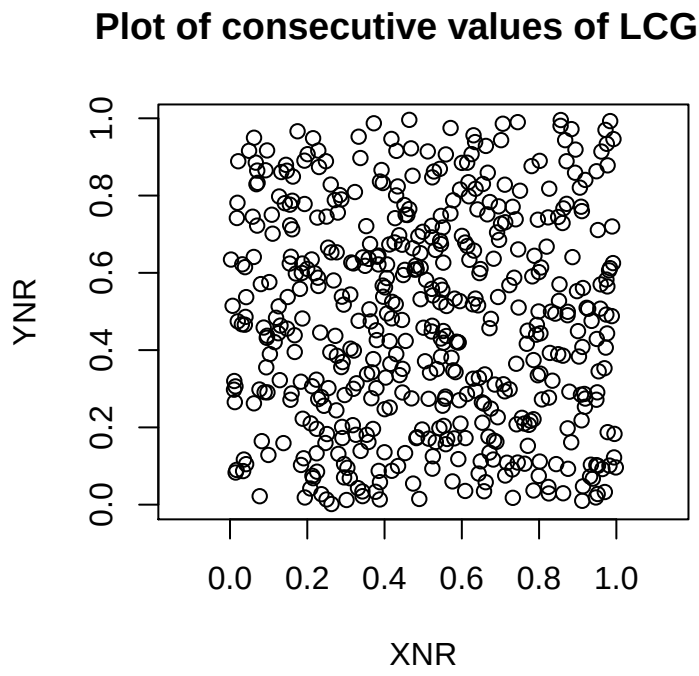

Figure 2.4: This figure shows the consecutive pairs from LCGNR

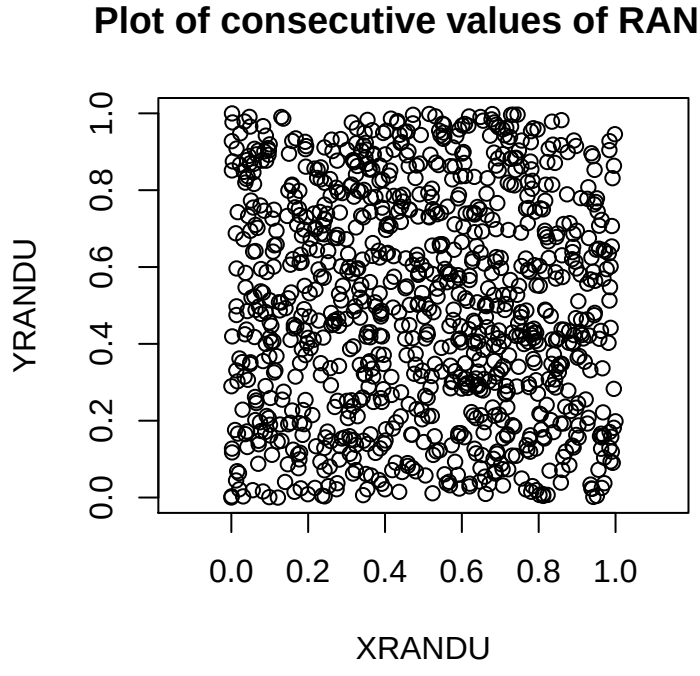

Figure 2.5: This figure shows the consecutive pairs from RANDU

that this test is detecting a level of correlation (or lack of independence) within the generator RANDU that we are as of yet unable to spot through a plot of paired values. Later in this report, through use of 'Invariant Coordinate Selection where consecutive triples of values are treated as coordinates in three dimensions, and transformed using an appropriate method, we realise the extent at which RANDU is flawed as a PRNG.

Note

Although this test seemingly failed to identify LCG2 as a poor PRNG this does not mean that LCG2 is a good PRNG, nor does it mean this Spearman's Rank Correlation Coefficient is bad tool at assessing the quality of PRNGs. We must remember to look at the bigger picture here; no single test is sufficient at categorising all PRNGs as good or bad and this explains why there are a multitude of tests used to filter out bad PRNGs. Further, the more tests that a PRNG passes, the more confidence we have that it is good at generating 'random' samples. Due to different tests assessing PRNGs in different ways and the simple fact that PRNGs are not truly random, we can expect that even good PRNGs may fail certain tests.

2.2 The Runs Test

2.2.1 Description of the Test

The Run test is another test for independence, which can be applied to PRNGs. To make sense of the Run Test, we must first define a few terms used when explaining the Run Test. On pg 271 of 'Distribution-Free Statistical Tests' by James V. Bradley [Bra68], a 'Run' is defined to be an unbroken sequence of increasing or decreasing observations. We take the number of observations to be n, and sampled from a continuously distributed variable. A run of observations monotonically increasing in size is defined to be a 'run up', and a run of observations monotonically decreasing in size is defined to be a 'run down'. These n observations can be replaced by a mixture of $n-1$ $+$ and $-$ signs, where a $+$ represents a run up, and a $-$ represents a run down. We now use the median of the observations to argue that the probability of a run up or run down is not constant for all values. If the series of observations is random then it is unlikely that if an observation is some distance from the median, that the next value will be even further away from the median. There are then two different ways to form a test statistic; either using the total number of runs or alternatively looking at the length of the longest run of $+$'s or $-$'s. Both these versions of the Run Test are use the same rationale and assumptions, and although test the same null hypothesis they have a different alternative hypothesis. We define r to be the total number of runs up and runs down, which is done by counting the runs of $+$ and $-$ signs.

2.2.2 Rationale and Assumptions

The rationale for the Runs Test is that if a process that outputs n unequal numbers is random, then any of $n!$ possible arrangements of the n numbers has equal probability of being the observed sequence as any other ordering. However if there is some nonrandom element to the process then certain sequences will be more likely than others. This means that the observed sequence is also more likely to be one of the more probable sequences. As a consequence of this, the observed number of runs should fall at either tail of the distribution of total number of runs. Then from this we deduce that we may use the total number of runs, r, to test a null hypothesis that the numbers are selected randomly. In the case of testing PRNGs we again choose a two sided test where reject randomness if there are either too few or too many runs. Although this reasoning is related to the number of runs, it also applies to if the longest run up or run down is considered too long.

There are relatively few assumptions that must be satisfied when using the Run Test. These are that the sampling must be random or testing this must form the basis for the test. Also the n observations must be ordered in a way that no observations are sharing a position with another observation i.e. there

is a distinct 'one after another' order. All of the n observations has a unique value; each observation is a different from another.

2.2.3 Number of Runs Up and Down as a Test of Randomness

The probabilities for the total number of runs in an arrangement were obtained using a recursion formula by D. André in 'Sur le nobre de permutations de n éléments qui présentent S séquences' [And83]. It follows from here that we take $P(i|j)$ as the priori probability that in a linear sequence of j numbers where no number is repeated i is the total number of runs up and runs down. From this then we have that

$$P(r|n) = \frac{rP(r|n-1) + 2P(r-1|n-1) + (n-r)P(r-2|n-1)}{n}, \tag{2.2}$$

where r and n have been previously defined.A proof of this can be found on pg 272 of 'Distribution-Free Statistical Tests' [Bra68], but is omitted here.

The null hypothesis and alternative hypothesis for a two sided Run test for the number of runs up and runs down is as follows.

H_0 : Every outcome of the possible $n!$ different outcomes is equally likely to become the observed outcome. If this is the case then we conclude the generator is sufficiently random.

H_1 : The run has either too few fluctuations or too many fluctuations for it to be considered to have come from a random source.

Then to test a data set, we must convert the sequence of n numbers into a sequence of $n-1$ $+$'s and $-$'s and take note of the number of runs up and runs down. When we wish to make a decision with regards to the null hypothesis there are tables that exist that give the probability of the total number of runs up and runs down and can be used as to decide how likely the observed number of runs up and runs down was. This is a rather inconvenient method however, especially if the sequence of numbers, n, is large. Due to the asymptotic normal behaviour of r, we may instead choose to use a normal approximation if the number of observations $n > 25$ to test the total number of runs. Sufficient n is guaranteed for our purposes, as we are testing PRNGs and these typically churn out thousands of observations. The mean and variance of the total number of runs are defined easily due to r being asymptotically normal. As stated on pg 279 by Bradley [Bra68], we define the mean and variance to be

$$\text{mean}(r) = \frac{(2n-1)}{3}$$

and

$$\text{var}(r) = \frac{(16n-29)}{90}.$$

Then using the mean and variance above our test statistic, Z is defined to be

$$\frac{r - \lfloor \frac{(2n-1)}{3} \rfloor}{\sqrt{\frac{(16n-29)}{90}}},$$

and we then refer to the standard normal table to make a decision to either accept or reject randomness. It is usual to subtract $\frac{1}{2}$ from the absolute value of the numerator to correct for continuity when moving from a discrete to continuous environment.

2.2.4 Length of Longest Run as a Test of Randomness

The length of the longest run is also a valid test statistic used to assess the randomness of a group of numbers. Instead, we consider the expected number of runs of consecutive +'s and consecutive −'s, that are greater than or equal to S from a random linear sequence of n different numbers. This expectation has form

$$\mathbb{E}(r_{\pm,\geq S}) = \frac{2 + 2(n-S)(S+1)}{(S+2)!},$$

where $\pm$ subscript represents the fact we do not specify whether the longest run of like difference signs is of +'s or −'s; in this case it can be either. Trivially, we can also retrieve expectations for if the longest run is $\geq S$ and specify beforehand that we want it to be +'s or −'s. To do so we simple divide the previous expectation by two, as both these expectations are identical and the sum of them add to our previous expectation. These have form

$$\mathbb{E}(r_{+,\geq S}) = \mathbb{E}(r_{-,\geq S}) = \frac{1 + 1(n-S)(S+1)}{(S+2)!},$$

where we have specified the + or − beforehand. If we choose $S \geq \frac{n}{2}$, then the expectations we have defined are also the probability of obtaining a complete sequence containing a longest run whose length is $\geq S$. That is

$$P(r_{\pm,\geq S} = 1) = \mathbb{E}(r_{\pm,\geq S})$$
$$P(r_{+,\geq S} = 1) = \mathbb{E}(r_{+,\geq S})$$
$$P(r_{-,\geq S} = 1) = \mathbb{E}(r_{-,\geq S}).$$

Proof of these expectations can be found on pg 275 of [Bra68]. The reason that the probabilities are equal to these expectations can be argued rather easily. I will show it for the case where we do not specify the sign of the longest run, and the same method can be applied to the case where a specification is made.

Lemma 2.2. $P(r_{\pm,\geq S} = 1) = \mathbb{E}(r_{\pm,\geq S})$, for $S \geq \frac{n}{2}$.

Proof. Our run of +'s and −'s, r, can either contain a longest run $\leq S$, or it can contain a longest run $> S$. As we have chosen that $S \geq \frac{n}{2}$, then it cannot contain 2 or more runs $> S$. Thus expectation of this event when $S \geq \frac{n}{2}$ is

$$\mathbb{E}(r_{\pm,\geq S}) = 0 \times P(r_{\pm,\geq S} = 0) + 1 \times P(r_{\pm,\geq S} = 1) = P(r_{\pm,\geq S} = 1).$$

□

This means when $S < \frac{n}{2}$, the above probabilities are only approximations and not strictly equal, however on pg 281 of [Bra68] it is stated that the approximations are good when the probability is ≤ 0.05. Thus we can use these approximations to test our PRNGs.

When considering the longest run, we test a slightly modified alternative hypothesis to the previous method. The null hypothesis and alternative hypothesis for the length of the longest run are as follows.

H_0 : Every outcome of the possible $n!$ different outcomes is equally likely to become the observed outcome. If this is the case then we conclude the generator is sufficiently random.

H_1 : The longest run, or more precisely the longest run up or run down is too long for the sequence to be considered to have come from a random source.

Much like the total number of runs test, the exist tables that contain the probabilities that a run of n numbers will include a run of like sign, of length at least S. But again, this is rather inconvenient for our purposes, and instead it is better to utilise the above formulas to calculate probabilities and then come to a decision; if the probability obtained from the expectation is < 0.05 we will conclude this is rather too unlikely to occur under the null hypothesis and thus will reject independence in this instance.

2.2.5 Runs Test in R

I have written three functions in R that will perform the Runs Test in the two different ways shown above. The first two functions relate to the total runs method, and the third function uses the longest run for Runs Test. The first function, which calculates r, the total runs is shown below.

```
total.runs<-function(lcg){
    runs <- c()
    r <- c()
for (i in 1:length(lcg)-1){
runs[i] <- ifelse( lcg[i] >= lcg[i+1], -1, 1)
}
for (j in 2:length(runs)){
 r[j] <- ifelse( runs[j] != runs[j-1], 1, 0)
 }
    r[is.na(r)] <- 1
return(sum(r))
}
```

This function only requires an input of a string of values from an LCG. Two empty vectors are labeled as *runs* and r. A loop is then ran for i from 1 to the length of the lcg -1, and the ith place in vector *runs* is filled with -1 if the ith position of the lcg is less than or equal to the value of the observation in spot $i+1$ of lcg. Otherwise slot i of *runs* becomes a 1. There are one less of these than the original string of values, and are either a 1 or -1 depending on if the string is increasing or decreasing, from one value to another. Another loop is than ran for j, where the values of r become 1 for every time consecutive values of *runs* are different, and become 0 if they are the same, this is ran from entry 2 of *runs* until the last entry. This is counting up the number of times we see a change in run. Finally we replace the NA in entry 1 of r with 1 as is not included in the above loop, and then add up the values of r using the sum() function.

```
total.runs.nml<-function(lcg,a){
    r <- total.runs(lcg)
    n <- length(lcg)
    Z <- (r - ((2*n - 1)/3))/sqrt((16*n - 29)/90)
if(Z>qnorm(a/2) && Z<qnorm(a/2,lower.tail=F)){
return(list(r,"ACCEPT", Z))
}else{
return(list(r,"REJECT", Z))
}
}
```

This function then calculates a decision to reject or accept the null hypothesis of randomness by comparing to the normal distribution, using an extra input a as a level of significance. It uses the above function I wrote to calculate r, and labels the length of the lcg input as n. The function is rather simple, and calculates a Z value by taking the mean and dividing by the square root, which have forms defined in the previous section. If Z is within the critical regions calculated using the user given significance then the function returns the word "REJECT", along with the value of r and Z, otherwise it will return "ACCEPT" with the value of r and Z.

```
longest.run<-function(lcg,a){
     runs <- c()
     n <- length(lcg)
for (i in 1:length(lcg)-1){
runs[i] <- ifelse( lcg[i] >= lcg[i+1], -1, 1)
```

```
}
     S <- max(rle(runs)$length)
     p <- 2*(n-S)*(S+1)/factorial(S+2)
if(p>a){
return(list(S,"ACCEPT", p))
}else{
return(list(S,"REJECT", p))
}
}
```

This last function calculates the longest run, and gives a decision to reject or accept the null hypothesis of randomness based on a, a user selected significance. Again, we define *runs* to be an emty vector, and repeat the same steps for *runs* as in the first function, so it becomes a vector of 1 and -1 entries based on whether the lcg input is increasing or decreasing. Then S is defined to be the longest string of same values, obtained using the max() function on the rle()$length function. Then p is the probability of S being the largest run, and has a form defined previously. The function then returns "ACCEPT", along with S and p if $p > a$, and "REJECT" along with S and p otherwise.

2.2.6 Examples of Runs Test on LCGs

For both versions of the Runs Test, it is a requirement that all values tested are unique; something that is not the case for two of our LCGs; LCG2 and LCG3. Hence we cannot apply either version of the Runs Test to these LCGs. Then in this section we apply both versions of the Runs test to LCG1, LCGNR and RANDU.

Example 2.6. In this example the total number of runs is used to calculate a Runs Test. The outcome of the function for LCG1, LCGNR and RANDU are shown in the table below.

LCG	n	r	Z	Decision
LCG1	400	249	-2.0602	Reject
LCGNR	2000	683	1.2511	Accept
RANDU	2000	1327	-0.3183	Accept

As LCG2 and LCG3 do not meet the criteria to be tested by the Runs Test, we decide they fail this test by default.

We observe that LCG1 narrowly fails this test; a value of -2.0602 is just outside the -1.96 lower critical region bound for the standard normal distribution. In this case we reject the null hypothesis for the alternative hypothesis that the LCG output lacks indepence.

For LCGNR, a comfortable value of 1.2511 is obtained from the test and we deduce there is no evidence to reject the null hypothesis that the output of LCGNR is from a random source; no dependencies between values.

Finally, for RANDU, we witness the least suspect value of all three LCGs; a test statistic of -0.3183. This indicates we have no reason to suspect the null hypothesis is not true and deduce the output of RANDU to be sufficiently independent.

Example 2.7. Next we calculate the Runs Test using the longest run, and do not specify whether the longest run is of runs up or runs down. The calculated S will be considerably lower than $\frac{n}{2}$ and as such we remember that the probabilities calculated here are approximations only. The outcome for each LCG is displayed in the table below.

LCG	n	S	p	Decision
LCG1	400	6	0.1368	Accept
LCGNR	2000	6	0.3451	Accept
RANDU	2000	8	0.0099	Reject

Interestingly the results from this version of the Runs Test are not completely concordant with the previous version of the test. For LCG1, a longest run of 6 is deemed to occur with probability $p = 0.1368$ and we deem that acceptable, hence we cannot reject the null hypothesis in this case. Previously, in the last example we rejected the null hypothesis for LCG1.

Now for LCGNR, we also witness a longest run of 6 (but LCGNR is of course ran for much longer), and we calculate that this occurs with probability $p = 0.3451$, and thus strongly accept the null hypothesis as this is a very likely outcome.

Finally, for RANDU, which passed the previous version of the Runs Test, we see a longest run of 8, which is calculated to occur with probability $p = 0.0099$ and thus we conclude this is too unlikely outcome for a truly random source and thus reject the null hypothesis.

Chapter 3

Invariant Coordinate Selection

This chapter focuses on a method called Invariant Coordinate Selection, which is used to explore multivariate data. Although this may not seem initially applicable to the output of PRNGs, its use on one particular PRNG is rather infamous. The PRNG in question was a popular PRNG called 'RANDU', which is an example of a specific Linear Congruential Generator, with parameters $a = 65539$, $m = 2^{31}$, $c = 0$ and with initial value $x_0 =$ odd. Although its use was widespread in the 1960's and 1970's, since it was discovered how poor the output of this LCG actually was, it is now considered a bad random number generator. This section follows the paper written by David E. Tyler, Frank Critchley, Lutz Dümbgen and Hannu Oja entitled 'Invariant Coordinate Selection' [aFCDO09]. After providing an outline to the method, I will use the method to show the issue that arises within the RANDU LCG, and apply the technique to other popular PRNGs. As the method is a multivariate technique, to apply this method to PRNGs we take consecutive triplets of values from a LCG to be the coordinates of a point in 3-dimensions. Thus, say we have an output sequence of 300 values for a PRNG, then using this technique we will view these 300 values as being 100 points in 3-dimensions.

Although much of the mathematics in this chapter is not needed for analysing PRNGs using Invariant Coordinate Selection, it is a summary of the work covered in the aforementioned paper [aFCDO09]. It has been included as the mathematics here is incredibly helpful in understanding what this method is doing to any data it is applied to, and how it can identify any hidden structures that are not immediately apparent within data using less advanced methods.

3.1 Affine Equivariance

Before we begin defining affine equivariance we must define several properties of matrices from linear algebra.

Taken from 'Introduction to Linear Algebra' [Str09] by Strang, we define a symmetric matrix, found on page 109, a singular matrix from page 248, and the notion of positive definiteness from page 343.

Matrix A is a symmetric matrix if $A^T = A$, This can also be expressed in terms of elements as $a_{ij} = a_{ji}$.

A matrix A is singular if $\det(A) = 0$.

Matrix A is positive definite if $x^T Ax > 0$ for every nonzero vector x.

We define $Y \in R^p$ to be a multivariate random variable, with distribution function F_Y. We denote the set of all symmetric positive definite matrices with order p by $\mathcal{P}_p$.

We denote affine equivariant multivariate location by the following notation; $\mu(F_Y) \in R^p$, and affine equivariant multivariate scatter functionals using the notation; $V(F_Y) \in \mathcal{P}_p$. These are functions of the distribution F_Y. For the transformation $Y^* = AY + b$, where A is a nonsingular matrix and $b \in R^p$, both functions satisfy

$$\mu(F_{Y^*}) = A\mu(F_Y) + b$$

and

$$V(F_{Y^*}) = AV(F_Y)A^T.$$

The mean vector $\mu_Y = E[Y]$ and the variance-covariance matrix $\Sigma_Y = E[(Y - \mu_Y)(Y - \mu_Y)^T]$ are the standard examples of affine equivariant location and scatter functionals, respectively, given that they do in fact exist.

Having defined affine equivariance for the distribution F_Y, we next focus our attention to a sample of this distribution. Let us consider a p dimensional sample, and of size n using the notation $\mathbf{Y} = \{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$. Affine equivariant multivariate location and scatter statistics for the sample are denoted by $\hat{\mu}$ and $\hat{V}$, respectively. These statistics satisfy the following properties for A nonsingular, and any $b \in R^p$ that

$y_i \to y_i^* = Ay_i + b$ for $i = 1, \ldots n \Rightarrow (\hat{\mu}, \hat{V}) \to (\hat{\mu}^*, \hat{V}^*) = (A\hat{\mu} + b, A\hat{V}A^T)$.

Common examples of affine equivariant location and scatter statistics are the sample mean vector $\bar{y}$ and the sample variance-covariance matrix S_n, respectively. The properties affine equivariant location and scatter statistics applied to a sample; $\mathbf{Y} = \{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$, are simply a natural extension of the properties for the distribution F_Y, but with the conditions that they satisfy changed to apply to the sample $\mathbf{Y} = \{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$.

The sample mean vector and sample variance-covariance matrix will be used later in this section and as such, definitions are provided here. Both definitions are taken from pg 10 of 'Multivariate Analysis' by Mardia, Kent and Bibby [MKB88].

The sample mean of the ith variable is

$$\bar{y_i} = \frac{1}{n}\sum_{r=1}^{n} y_{ri}, \tag{3.1}$$

and the sample covariance between the ith and jth variables is

$$s_{ij} = \frac{1}{n}\sum_{r=1}^{n}(y_{ri} - \bar{y_i})(y_{rj} - \bar{y_j}). \tag{3.2}$$

Then the sample mean vector is then simply the vector of means y_i for $i = 1, \ldots, p$, and the sample covariance matrix is a p x p matrix of covariances, $S = (s_{ij})$.

3.2 Classes of Scatter Statistics

There are several concepts we must explain before proceeding any further. We will be making use of the Mahalanobis distance and weighted sample statistics and thus must explain these accordingly.

We may first consider the standard sample mean which has form

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

In the above case, the simplicity and commonness of the statistic may mean we take for granted what each term represents. The $\frac{1}{n}$ term means that each observation is weighted identically and is where this statistic differs from a weighted mean. A weighted mean can be written as

$$\hat{\mu}_w = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}.$$

Under the circumstances that $\sum_{i=1}^{n} w_i = 1$, $\hat{\mu}_w$ is described as being a normalised weighted mean. What is the relevance of weighted functions for invariant coordinate selection? Like many applications of a weight function; it is chosen to downplay outliers by assigning a small weighting to values that are considered as outliers. Deciding how each y_i is assigned a corresponding weight is another thing to consider. To do this, a distance known as the Mahalanobis distance is utilised, which will decide upon which of the y_i are outliers.

The Mahalanobis distance is an alternative to the Euclidean distance, and is one of the most important distances used in Cluster analysis. Taken from pg 17 of 'Multivariate Analysis' [MKB88] by Mardia, Kent and Bibby, the squared Mahalanobis distance between two values y_r and y_s is defined as

$$D_M^2 = (y_r - y_s)^T S^{-1} (y_r - y_s),$$

where S is the covariance matrix. In our case, we seek to compare our values y_i to a prescribed 'average' such as $\bar{y}$ or others of similar form, that will be described later in this section rather than the y_r and y_s expressed above. Looking at the form of the Mahalanobis distance is useful in explaining why it is used. If we consider the expression $(y-\mu)^T(y-\mu)$, this distance will assume data lies in a circle around the mean μ and thus does not consider any distribution besides a uniformly circular one around the mean. The Mahalanobis distance $(y-\mu)^T S^{-1}(y-\mu)$, does take into consideration how the values are distributed, through the appearance of the S^{-1} term, and thus the Mahalanobis distance may consider the points as being elliptically distributed around the mean.

We now consider three distinct, but similar classes of weighted sample mean and covariance matrices. The first an alternative to sample mean $\bar{y}$ and sample covariance matrix S_n are a weighted sample mean and weighted sample covariance matrix in which we base the weights upon the the Mahalanobis distance, as defined previously. Let's call the weighted sample mean and weighted sample covariance matrix as $\hat{\mu}$ and $\hat{V}$, respectively. They have the form

$$\hat{\mu} = \frac{\sum_{i=1}^{n} u_1(s_{o,i}) y_i}{\sum_{i=1}^{n} u_1(s_{o,i})}$$

and

$$\hat{V} = \frac{\sum_{i=1}^{n} u_2(s_{o,i})(y_i - \bar{y})(y_i - \bar{y})^T}{\sum_{i=1}^{n} u_2(s_{o,i})},$$

where $u_1(s_{o,i})$ and $u_2(s_{o,i})$ appropriate weight functions and $s_{o,i} = (y_i - \bar{y})^T S_n^{-1}(y_i - \bar{y})$.

The second class we consider is more robust than our first class. We derive this class of multivariate location and scatter statistics from the multivariate M-estimates, which are adaptively weighted sample means and sample covariances. They are the solutions of the M-estimate equations and are defined to be

$$\hat{\mu} = \frac{\sum_{i=1}^{n} u_1(s_i) y_i}{\sum_{i=1}^{n} u_1(s_i)}$$

and

$$\hat{V} = \frac{\sum_{i=1}^{n} u_2(s_i)(y_i - \hat{\mu})(y_i - \hat{\mu})^T}{\sum_{i=1}^{n} u_3(s_i)},$$

in which $u_1(s)$, $u_2(s)$ and also $u_3(s)$ are appropriate weight functions and $s_i = (y_i - \hat{\mu})^T \hat{V}(y_i - \hat{\mu})$.

These solutions are less self explanatory than the last class. For one, we now have three weight functions present. Second, we note the implicit nature of these solutions. They are implicit for $(\hat{\mu}, \hat{V})$ because the weight functions are dependent on the Mahalanobis distance relative to $(\hat{\mu}, \hat{V})$. This is not a problem for us as rather basic algorithms that compute these estimates are available.

A disadvantage of the M-estimates is that in higher dimensions they have a rather low breakdown point; they are bounded above by $\frac{1}{(p+1)}$. As a result of this, there are numerous estimates with a higher breakdown point. These however, are computationally intensive for large datasets an so not stated here.

We may make use of the initial affine equivariant location and scatter statistic to generalize $\hat{\mu}$ and $\hat{V}$. This is our third class of scatter statistics. If initial affine equivariant location and scatter statistic are $\hat{\mu}_0$ and $\hat{V}_0$ respectively, then $\hat{\mu}$ and $\hat{V}$ become

$$\hat{\mu} = \frac{\sum_{i=1}^{n} u_1(s_{o,i}) y_i}{\sum_{i=1}^{n} u_1(s_{o,i})}$$

and

$$\hat{V} = \frac{\sum_{i=1}^{n} u_2(s_{o,i})(y_i - \hat{\mu_0})(y_i - \hat{\mu_0})^T}{\sum_{i=1}^{n} u_2(s_{o,i})},$$

where we take $s_{o,i} = (y_i - \hat{\mu}_0)^T \hat{V}_0^{-1}(y_i - \hat{\mu}_0)$. Under a univariate setting, these weighted sample mean and variances are called one-step W-estimates, and so we extend this to the multivariate versions above and refer to them as the multivariate one-step W-estimates. The advantage of this method is the relative computational simplicity as opposed to one previously mentioned.

3.3 Comparing Scatter Matrices

We now need to understand some theory regarding linear algebra and the comparison of matrices to further our progress in understanding invariant coordinate selection. We note that comparing positive definite symmetric matrices is not something seldom seen. It is widespread in plenty of areas of statistics including multivariate analysis of variance (MANOVA), for example.

Like in many scenarios we are going to summarise the difference between two positive definite symmetric matrices through comparing the eigenvalues and eigenvectors of each matrix with respect to the other. Symmetric matrices possess several properties that make analysing eigenvalues and eigenvectors favourable. For symmetric matrices eigenvalues are real, and the eigenvectors of different eigenvalues are orthogonal. We begin at standard definition for eigenvectors and eigenvectors.

That is for a given square matrix A, a vector x is an eigenvector and λ is an eigenvalue of A if

$$Ax = \lambda x,$$

and $x \neq 0$. If A is symmetric then $\lambda \in \mathbb{R}$. If A is positive definite then every eigenvalue of A is positive.

Let us now consider two covariance matrices, $V_1, V_2 \in \mathcal{P}_p$. An eigenvalue, ρ_j and corresponding eigenvector, h_j, of V_2 relative to V_1 correspond to the solution of the matrix equation

$$V_2 h_j = \rho_j V_1 h_j,$$

where $j = 1, \ldots, p$. It can also be said that ρ_j and h_j are eigenvalues and eigenvectors, respectively of the matrix $V_1^{-1}V_2$. This relation is attained by multiplying both sides of the equality of the above equation by V_1^{-1}on the left side, giving

$$V_1^{-1}V_2 h_j = \rho_j h_j.$$

The above eigenvector/eigenvalue relation is now in the form of the standard definition for eigenvalues and eigenvectors, where $A = V_1^{-1}V_2$. The term $V_1^{-1}V_2$ is not symmetric. It is the case though, that ρ_j is also the eigenvalue of the symmetric matrix M, where $M = V_1^{-\frac{1}{2}}V_2V_1^{-\frac{1}{2}} \in \mathcal{P}$, where $V_1^{\frac{1}{2}} \in \mathcal{P}_p$ is the square root of V_1 and is unique, positive definite and symmetric.

Lemma 3.1.

$$V_1^{-1}V_2 h = \rho h \text{ if and only if } V_1^{-\frac{1}{2}}V_2V_1^{-\frac{1}{2}} q = \rho q \text{ for } q = V_1^{-\frac{1}{2}} h. \tag{3.3}$$

Proof. We take q from the above lemma to be $q = V_1^{-\frac{1}{2}} h$. Therefore, by definition of q, we have $h = V_1^{-\frac{1}{2}}$.

$$V_1^{-1}V_2 h = \rho h$$

We first multiply both sides of the equation by $V_1^{\frac{1}{2}}$, from the left hand side.

$$\Leftrightarrow V_1^{\frac{1}{2}}V_1^{-1}V_2 h = \rho V_1^{\frac{1}{2}} h$$

The term ρ represents an eigenvalue, and thus is a scalar value, hence we can move this term to the front of the expression on the right side of the equation. The two matrices $V_1^{\frac{1}{2}}$ and V_1^{-1} are then combined on the left side.

$$\Leftrightarrow V_1^{-\frac{1}{2}}V_2 h = \rho(V_1^{\frac{1}{2}} h)$$

We now use the fact that $q = V_1^{-\frac{1}{2}} h$ from the definition and apply this to the right hand side.

$$\Leftrightarrow V_1^{-\frac{1}{2}}V_2 h = \rho q$$

Finally we replace h on the left side by $h = V_1^{-\frac{1}{2}}$ and we are done.

$$\Leftrightarrow V_1^{-\frac{1}{2}}V_2V_1^{-\frac{1}{2}} q = \rho q$$

□

We can thus order the p eigenvalues in such a way that $\rho_1, \rho_2, \ldots, \rho_p > 0$, where all eigenvalues greater than 0 because the positive definite condition is satisfied for $V_1^{-\frac{1}{2}}V_2V_1^{-\frac{1}{2}}$. We choose corresponding orthonormal eigenvectors q_j, for $j = 1, \ldots, p$ such that $Mq_j = \rho_j q_j$. There is a relationship between the eigenvectors h_j, of V_2 relative to V_1 and the eigenvectors q_j of M. We can see this proportionality in more detail as

$$(Mq_j =)V_1^{-\frac{1}{2}}V_2V_1^{-\frac{1}{2}} q_j = \rho_j q_j$$

and

$$V_1^{-1}V_2 h_j = \rho_j h_j,$$

therefore $q_j \propto V_1^{-\frac{1}{2}} h_j$, as replacing q_j in (4.4) with $V_1^{-\frac{1}{2}} h_j$ yields (4.5). It is only proportional and not equal as replacing q_j with the same as above, but multiplied by any scalar, say $bV_1^{-\frac{1}{2}} h_j$ would also yield

equation (4.5). From this, it is the case that $h_i^T V_1 h_j = 0$ for $i \neq j$. First we note that as eigenvectors of different eigenvalues are orthogonal for a symmetric matrix therefore,

$$\begin{aligned} q_i \perp q_j \Rightarrow q_i^T q_j &= 0 \\ \Rightarrow (V_1^{-\frac{1}{2}} h_i)^T (V_1^{-\frac{1}{2}} h_j) &= 0 \\ = h_i^T (V_1^{\frac{1}{2}})^T V_1^{\frac{1}{2}} h_j &= 0, \end{aligned}$$

as $(AB)^T = B^T A^T$.

$$\begin{aligned} &= h_i^T V_1^{\frac{1}{2}} V_1^{\frac{1}{2}} h_j = 0 \\ &= h_i^T V_1 h_j = 0, \end{aligned}$$

as $V_1^{\frac{1}{2}}$ is symmetric hence $(V_1^{\frac{1}{2}})^T = V_1^{\frac{1}{2}}$.

From this we derive the simultaneous diagonalisation of V_1 and V_2 as

$$H^T V_1 H = D_1$$

and

$$H^T V_2 H = D_2,$$

where $H = [h_1 \dots h_p]$, and D_1 and D_2 are diagonal matrices with positive entries $D_1^{-1} D_2 = \Delta = \text{diagonal}\{\rho_1, \dots, \rho_p\}$. This is because $H^T V_1 H = (h^T V_1 h)$ and $h_i^T V_1 h_j = 0$, so only the diagonal entries $h_i^T V_1 h_i \neq 0$, for $i = 1, \dots, p$, which are found on the diagonal of $H^T V_1 H$, and similarly for the case of V_2.

We can normalise h_j to take $D_1 = I$ without any loss of generality. By doing so $h_j^T V_1 h_j = 1$. The spectral decomposition of the matrix $V_1^{-\frac{1}{2}} V_2$ is

$$V_1^{-\frac{1}{2}} V_2 = H \Delta H^{-1}.$$

By then plotting the data using the coordinates $Z = H^T Y$, it may reveal structures in the values that previously would not have been spotted.

3.4 Invariant Coordinate Systems

It is critically important that the transformation $Z = H^T Y$ is invariant, so that any analysis of this transform is applicable to the original dataset Y. Fortunately, this is the case, and this invariant nature of $Z = H^T Y$ is now explored in this section. Now, any theory or properties stated from this section onwards are in regards the functional or population forms of scatter matrices, and not the sample version. This is due to the sample version following a special case based on empirical distributions. It does still hold true for the sample case.

We assume that for $Y \in R^p$, with distribution F_Y, that $V_1(F)$ and $V_2(F)$ are two scatter functionals and both uniquely defined at F_Y.

Let $H(F) = [h_1(F) \dots h_p(F)]$ be a matrix of eigenvectors as previously described, with $\rho_1(F) \geq \rho_p(F)$ be the corresponding eigenvalues, when we take $V_1 = V_1(F)$ and $V_2 = V_2(F)$.

The resulting variables from the transformation $Z = H(F_Y)^T Y$ is invariant under any affine transformation.

Suppose then, that the roots $\rho_1(F_Y), \ldots, \rho_p(F_Y)$ are all distinct. Then for the affine transformation $Y^* = AY + b$, (A nonsingular),

$$\rho_j(F_{Y^*}) = \gamma\rho_j(F_Y) \text{ for } j = 1, \ldots, p \tag{3.4}$$

for some $\gamma > 0$.

Further, the components of $Z = H(F_Y)^T Y$ and of $Z^* = H(F_{Y^*})^T Y^*$ differ at most coordinatewise by location and scale. For some constants $\alpha_1, \ldots, \alpha_p$, $\alpha_j \neq 0$ for $j = 1, \ldots, p$ and $\beta_1, \ldots, \beta_p$,

$$Z_j^* = \alpha_j Z_j + \beta_j \text{ for } j = 1, \ldots, p \tag{3.5}$$

The transformed variable $Z = H(F_Y)^T Y$ is called an invariant coordinate system, and the method this system is obtained by is referred to as invariant coordinate selection or ICS. This can be generalised in a way that allows for the possibility of multiple roots.

First allow Y, Y^*, Z and Z^* be defined as stated previously. For Y of distribution F_Y, we again let $V_1(F)$ and $V_2(F)$ be uniquely defined scatter functionals. Suppose that the roots $\rho_1(F_Y), \ldots, \rho_p(F_Y)$ are comprised of m distinct values denoted $\rho_{(1)} > \cdots > \rho_{(m)}$, and $\rho_{(k)}$ has multiplicity p_k for $k = 1, \ldots, m$, therefore we have that $p_1 + \ldots, p_m = p$. Then (4.4) holds true and furthermore, suppose that Z is partitioned in a way such that $Z^T = (Z_{(1)}^T, \ldots, Z_{(m)}^T)$, with $Z_{(k)} \in R^{p_k}$. Then for some nonsingular matrix, say C_k, of order p_k and some p_k dimensional vector β_k,

$$Z_{(k)}^* = C_k Z_{(k)} + \beta_k \text{ for } k = 1, \ldots, m. \tag{3.6}$$

Which means that the components of $Z_{(k)}^*$ and the components of $Z_{(k)}$ span the same space.

3.5 Invariant Coordinate Selection under a Mixture of elliptical distributions

Although so far only the only distributions considered so far when Y is elliptically symmetric. In reality, the data may be from several different elliptical distributions, which collectively may be considered as a mixture distribution. What this means is that there may be multiple elliptical distributions, say k, each corresponding to a proportion of the overall data, and each with a different mean and spread function but equal shape matrices.

Then again we assume that for $Y \in R^p$, with distribution F_Y, that $V_1(F)$ and $V_2(F)$ are two scatter functionals and both uniquely defined at F_Y. Suppose that Y has the density

$$f_Y(y) = \det(\Gamma)^{-\frac{1}{2}} \sum_{j=1}^{k} \alpha_j g_j\{(y - \mu_j)^T \Gamma^{-1}(y - \mu_j)\},$$

where $\alpha_j > 0$m for $j = 1, \ldots, k$ and $\alpha_1 + \cdots + \alpha_k = 1$ are the proportions of corresponding to the k elliptical distributions, $\Gamma \in \mathcal{P}_p$ and $g_1, \ldots, g_k$ are non negative functions. Suppose that the centres $\mu_1, \ldots, \mu_k$ span some q dimensional hyperplane ($0 < q < p$), then there exists at least one root, $\rho_{(j)}$ for $j = 1, \ldots, m$ with multiplicity $\geq p - q$. If no root has multiplicity stictly $> p - q$, then there is a root with multiplicity $= p - q$, say $\rho_{(t)}$ such that

$$\text{Span}\{\Gamma^{-1}(\mu_j - \mu_k) | j = 1, \ldots, k-1\} = \text{Span}\{H_q(F_Y)\},$$

where $H_q(F_Y) = [h_1(F_Y), \ldots, h_{p_1+\cdots+p_{t-1}}(F_Y), h_{p_1+\cdots+p_{t+1}}(F_Y), \ldots, h_p(F_Y)]$.

3.6 Invariant Coordinate Selection applied to data sets including PRNGs

3.6.1 Invariant Coordinate Selection applied to data sets

In the paper 'Invariant Coordinate Selection' [aFCDO09] and also here, little advice has been given as what pair of scatter matrices should be chosen when using the ICS method. Thus, one might assume that this choice is not of great importance, and for many data sets, this is actually the case. Diagnostic plots; plots generated by plotting the data transformed by ICS (of form; $Z = H^T Y$) are not strongly affected by the choice of scatter variables when the data is from a mixture of distributions explained in the previous section, or from an independent component model. The explanation of independent component models has been omitted here as is not relevant to the testing of random number generators however an explanation of ICS applied to these models can be found in 'Invariant Coordinate Selection' [aFCDO09]. It must be noted that for other data sets the diagnostic plots can be rather susceptible to change based on choice of scatter matrices.

Then what advice can be given on choosing scatter matrices for Invariant Coordinate Selection? This is a rather difficult question to address, as a definitive answer does not exist. Using different scatter matrices can reveal many different structures within the data as there is no single departure from a distribution, data can depart from a distribution in many different ways. From this then, we can deduce there does no exist a certain pair of scatter matrices that will always reveal these structures best. Thus, it is best work through this problem by testing various different pairs of scatter matrices and then considering the results of all these different pairs to determine which pair revealed the most about structures within the data set.

3.6.2 Invariant Coordinate Selection applied to PRNGs

Although traditional PRNGs, included the ones studied here do not sample values from elliptical distributions, Invariant Coordinate Selection may still be applied to them. The reason for this is given in the form of a remark on pg 4 of in [aFCDO09] and is reiterated as follows.

The class of distributions for which all affine equivariant location functions are equal and all equivariant scatter functionals are proportional to each other - a requirement for Invariant Coordinate Selection, is not limited to only elliptical distributions. It is also true for the distribution F_Y, where $Y = AZ + \mu$, and the distribution of Z is exchangeable and symmetric for all components. This means that $Z \sim DJZ$, where J is any permutation matrix and D corresponds to any diagonal matrix with diagonal entries ± 1. The is the largest possible class for which the properties hold. The classes included in this umbrella of distributions includes uniform distributions within a unit cube. A string of observations from a (valid) PRNG will be sampled from a uniform distribution on the interval $[0, 1]$, and thus grouping these values into trios will correspond to a single value sampled uniformly from a unit cube. This forms the basis for why the method of Invariant Coordinate Selection is a valid approach to PRNG testing.

When testing the quality of PRNGs, especially when viewing the flaws in the generator RANDU, we use the one step W-estimate as one of the scatter matrices, and the other the sample covariance matrix, S_n. The form of the one step of the W-estimate is taken from pg 22 of [aFCDO09] and has form

$$\hat{V} = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{(y_i - y_j)(y_i - y_j)^T}{\{(y_i - y_j)^T S_n^{-1} (y_i - y_j)\}^2}. \tag{3.7}$$

This form of this W-estimate is slightly different to what was shown previously.

3.7 Invariant Coordinate Selection in R

I have written two functions in R that will make it easy to perform Invariant Coordinate Selection on a PRNG. The first function will calculate the one-step W-estimate based on pairwise differences, requiring a single input of the values of a PRNG. These must be inputted in matrix form where the of columns represents the how the output of a PRNG values are grouped. For example to see the interesting features of the PRNG RANDU this must be inputted as triples, so the matrix must have 3 columns. I was originally going to require an input, say *col*, where this number is the users choice for the number of columns. This turned out to be inconvenient, as was the only way to ensure the PRNG had the correct number values the PRNG must run for $col \times n$ to ensure the number is divisible by the column number, otherwise the calculation would fail. The function takes a considerably long amount of time to run by nature and this became rather troublesome, so it worked better by requiring the input of the function to be in the correct form, rather than requiring user specification. The function is displayed below.

```
W.estimate1 <- function(Y) {
         n <- nrow(Y)
         res <- 0
         for (i in 1:(n-1)) {
                    for (j in (i+1):n) {
                              yi <- Y[i,]
                              yj <- Y[j,]
                              z <- yi - yj
                              numer <- z %*% t(z)
                              denom <- as.numeric(t(z) %*% solve(cov(Y)) %*% z)^2
                              res <- res + numer/denom
                    }
          }
          return(2*res/(n*(n-1)))
}
```

The second function written plots a pairs plot of the transformed matrix Z, using the W.estimate function written above. This function is given below.

```
invariant.plot <- function(Y){
   V <- solve(cov(Y)) %*% W.estimate1(Y)
   eigs <- eigen(V)
   H <- eigs$vectors
   return(pairs(Y %*% H))
}
```

3.8 Examples of Invariant Coordinate Selection on LCGs

We return to the five LCGs that we have been testing within previous sections, and this time apply Invariant Coordinate Selection to the output of these LCGs. However, unlike for our other tests, Invariant Coordinate Selection is used to reveal hidden structures within data, and does not simply output a definitive answer as to 'accept' or 'reject' the output of the LCG as being from a specified distribution, or being sufficiently random. Our final output here is a plot of the output, transformed in such a way that some structures may be revealed. It is due to this, that a decision must be made by the user based upon the plot generated from the function I have written. Let us then consider our LCGs once again. As Invariant Coordinate Selection treats triplets of consecutive values from an LCG as coordinates in three dimensions, the amount of values outputted from each LCG is increased to a number divisible by three.

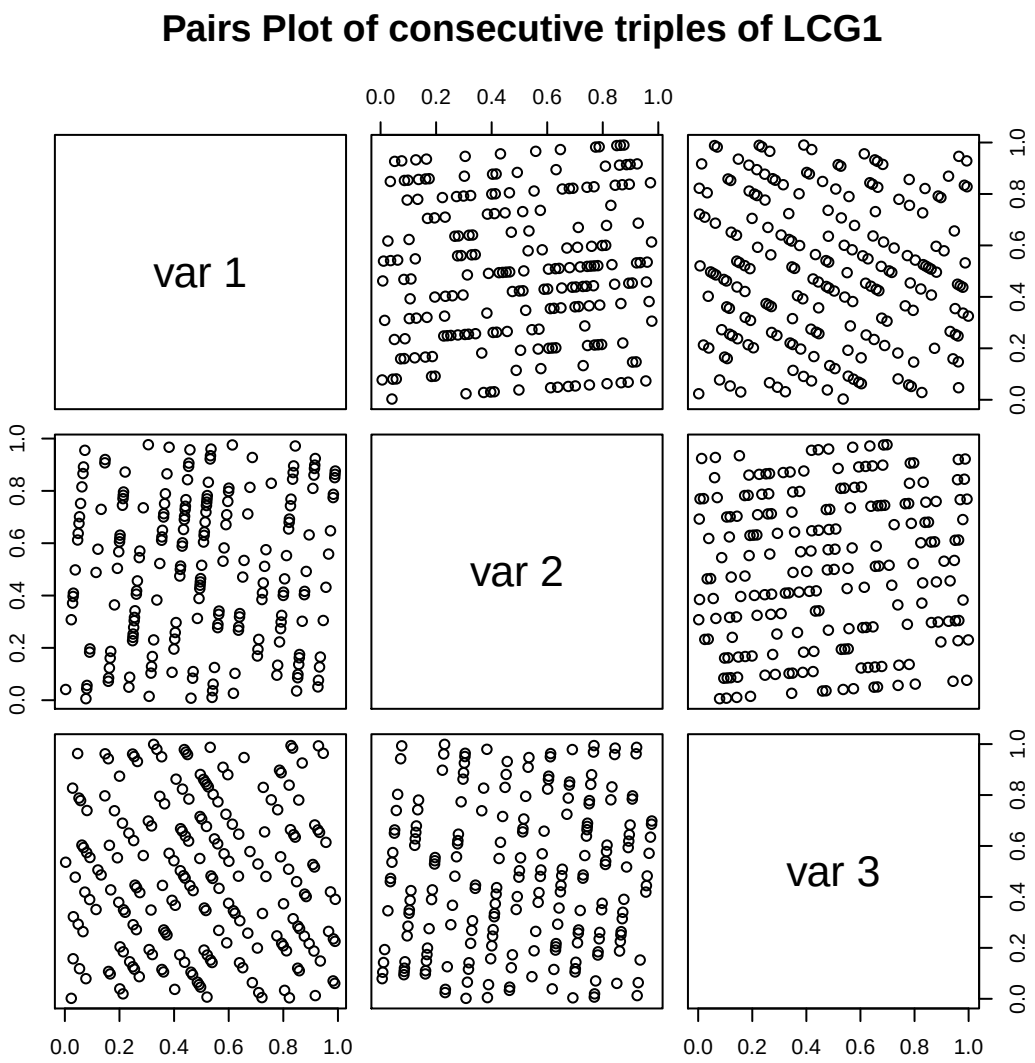

Figure 3.1: This figure shows a Pairs Plot of consecutive triples of LCG1. Planes of values are already clearly visible in the plots and we can immediately deduce that the output of LCG1 is incredibly flawed.

Example 3.1. LCG1 from Example 0.3 is adapted and now ran for $n = 600$, so is divisible by three and the method works. Next, the string of 600 values is converted into a matrix with 3 columns and 200 rows. We first consider a pairs plot of this matrix, before apply Invariant Coordinate Selection.

Even within this pairs plot, it is apparent that the output values of LCG1 are not sufficiently random. We see a number of planes running through the data, where a number of points are found on each of these planes, and no points in between them. LCG1 was created simply to illustrate how each test can be used to assess the quality of a PRNG, and even before these tests it is evident that LCG1, LCG2 and LCG3 are poor LCGs due to such a small period length, and multiplier. With that said, let us apply Invariant Coordinate Selection to see if it reveals this already apparent structure within the data more clearly. A pairs plot of the transformed data for LCG1 is generated using the R function I have written.

In this transformed pairs plot, we observe that the data has been transformed and we do see the planes on which the observables appear from a different angle. Though the improvement is not noticable due to LCG1 being so incredibly poor to begin with. The verdict here would be that LCG1 definitely fails our ICS test, as we can easily view the planes in which values are selected, and is seemingly not uniformly

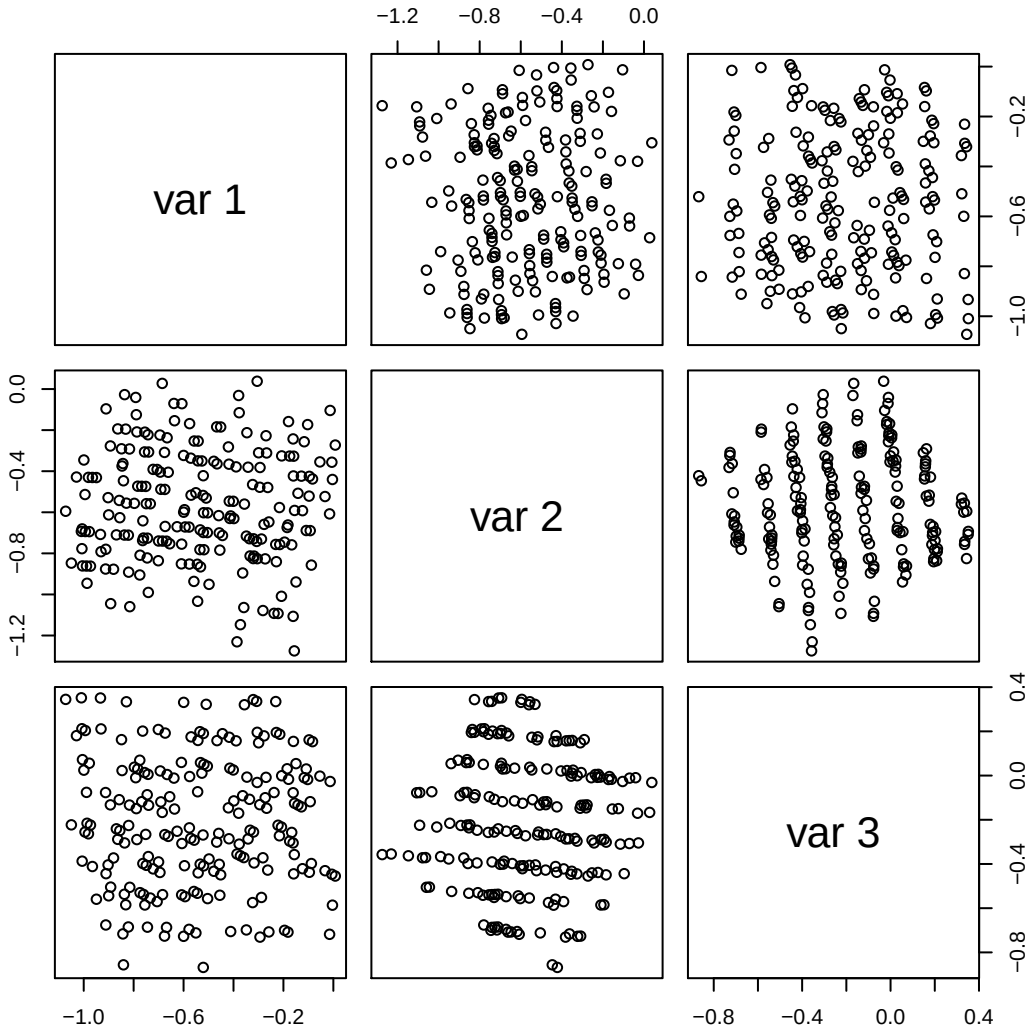

Figure 3.2: This figure shows a Pairs Plot of ICS transformed consecutive triples of LCG1, and does not reveal much more about the data that we could already see with the initial pairs plot. It is worth noting that we see a reduced number of planes within the data, hence indicating even less randomness than initially thought from the initial pairs plot.

chosen on the interval $[0, 1]$.

Example 3.2. We next return to LCG2 from Example 0.4. Our pairs plot for LCG2 3.3 is rather lacking, as the period length is so small, many values are repetitions hence we observe very few points on the plot. In these plots, $n = 90$ was the length of the LCGs output, as the function I wrote would not work for values any larger. This is due to the nature of the function, and as we observe so many repetitions for LCG2, the denominator of the function becomes 0 and thus the function fails. The initial pairs plot of LCG2 shows decent randomness; there doesn't seem to be an apparent pattern, such as there was is for LCG1 (and LCG3 see next example). Let us then consider a plot of the data after the transformation.

We can begin to see planes of values within the data after the transformation. Several planes are now viewable, however this is not as clear as what happens for RANDU. This is because n is rather small in this example, however the function breaks down for n any larger in the case of LCG2. We can see planes within the data that span from the top right corner of the plots to the bottom left. The planes are viewable enough for us to suspect LCG2 of being poor, however this is something we are already rather clear about.

Example 3.3. LCG 0.5 is adapted and ran for $n = 3000$ values. The string of 3000 values are converted to a matrix with 3 columns and 1000 rows. We first consider a pairs plot of this matrix.

Although the plot 3.5 is rather saturated in terms of values, we are still able to detect planes where the values occur and we observe that these values are not randomly distributed. This calls in to question the quality of LCG3 immediately. Let us now apply Invariant Coordinate Selection to LCG3.

This plot **??** is slightly improved over the initial pairs plot LCG3 3.5. It contains the same points from the pairs plot of LCG3, however this transformation reveals a view of the points at an angle where we observe a reduced number of planes, and they are seen more clearly. It is now even more clear that LCG3 is a poor PRNG; after transformation we see values are found only only nine planes. The verdict here is that LCG3 strongly fails Invariant Coordinate Selection as already partially viewable structures within the data are now even more obvious.

Example 3.4. We now consider LCGNR 0.6, which is ran for $n = 3000$ values and converted to a matrix with 3 columns and 1000 rows. An initial pairs plot of this matrix is first analysed.

Although again this plot 3.7 is rather saturated in terms of values, we cannot identify a pattern, or any planes on which values appear. This plot does appear sufficiently random. Let us then apply Invariant Coordinate Selection, and then consider a pairs plot of the transformed data.

Unlike in our previous examples, this transformed pairs plot has not revealed any hidden structures within the data. Points appear random within the unit square, which we would expect of a good PRNG. Our verdict is obvious; we deduce that LCGNR is outputting values that appear random when consecutive triples are treated as coordinates in three dimensions. We conclude that LCGNR has passed Invariant Coordinate Selection, as has failed to reveal any non randomness within the data.

Example 3.5. Finally we consider the famous RANDU 0.7, which is ran for $n = 3000$ values for our analysis here. We convert the output into a matrix with 3 columns and 1000 rows. We first consider a pairs plot of the output.

We obvserve another rather saturated pairs plot 3.9, but with no obvious pattern to the points, and certainly no planes are visible in this data set. This plot does not hint at the output of RANDU being poor or being sampled from planes. We then apply Invariant Coordinate Selection and observe a huge difference in plots.

Unlike LCGNR where we saw no change from applying ICS, we witness a huge change for RANDU. The transformed data is now sitting on 15 planes, and no longer appears random within the a three dimensional unit square. RANDU therefore fails Invariant Coordinate Selection, despite passing some of our earlier tests comfortably.

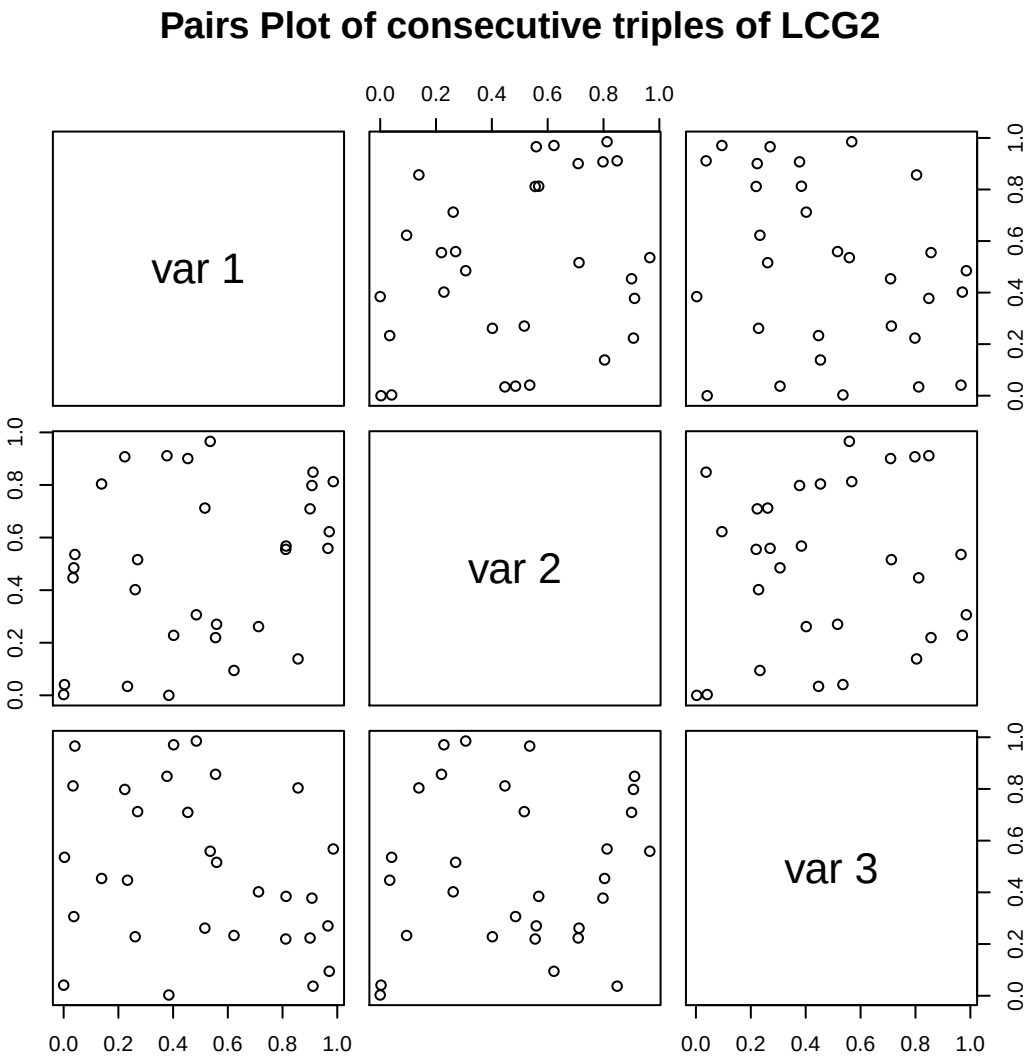

Figure 3.3: This figure shows a Pairs Plot of consecutive triples of LCG2. This plot appears rather random; perhaps mainly due to the fact so few unique values are present. No real planes containing data points have become visible within this plot.

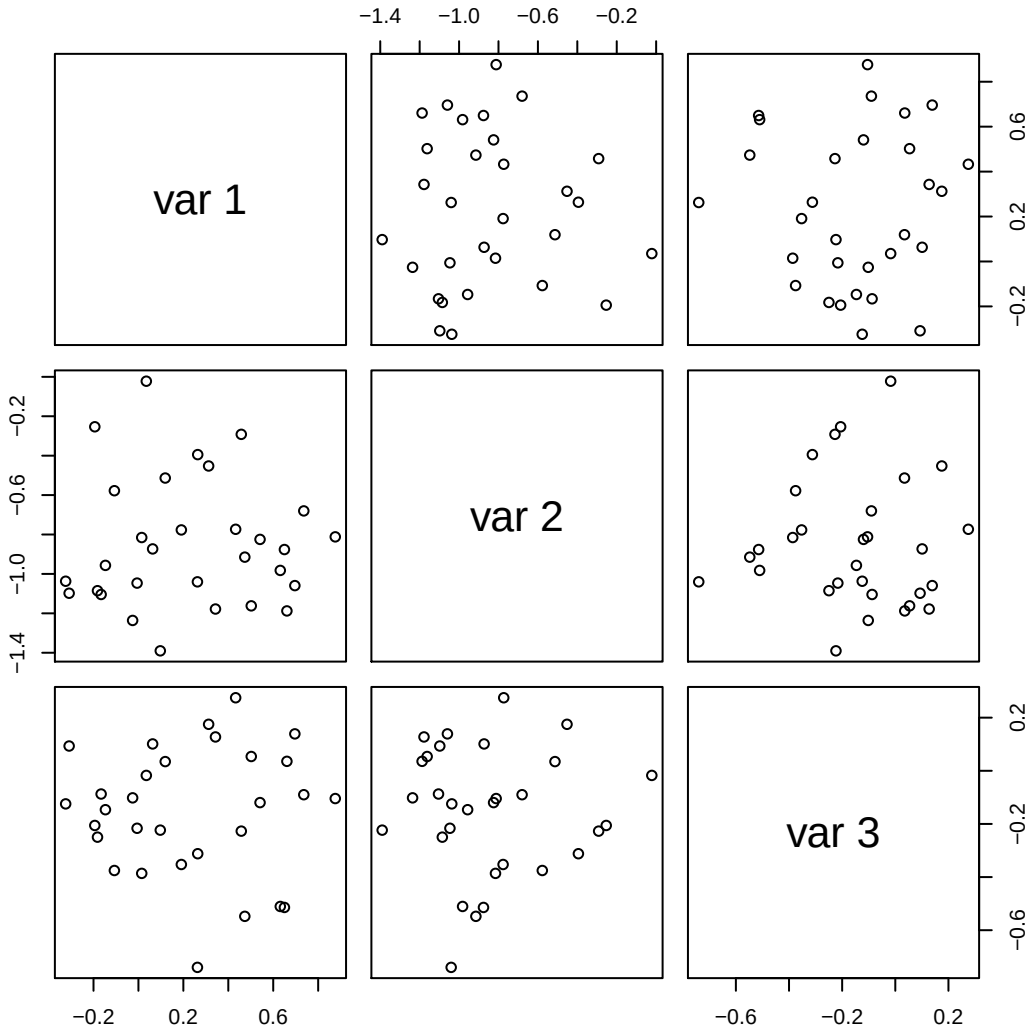

Figure 3.4: This figure shows a Pairs Plot of ICS transformed consecutive triples of LCG2. We see a small improvement over the initial pairs plot of the data in this case; diagonal planes have become visible, where the data points lie.

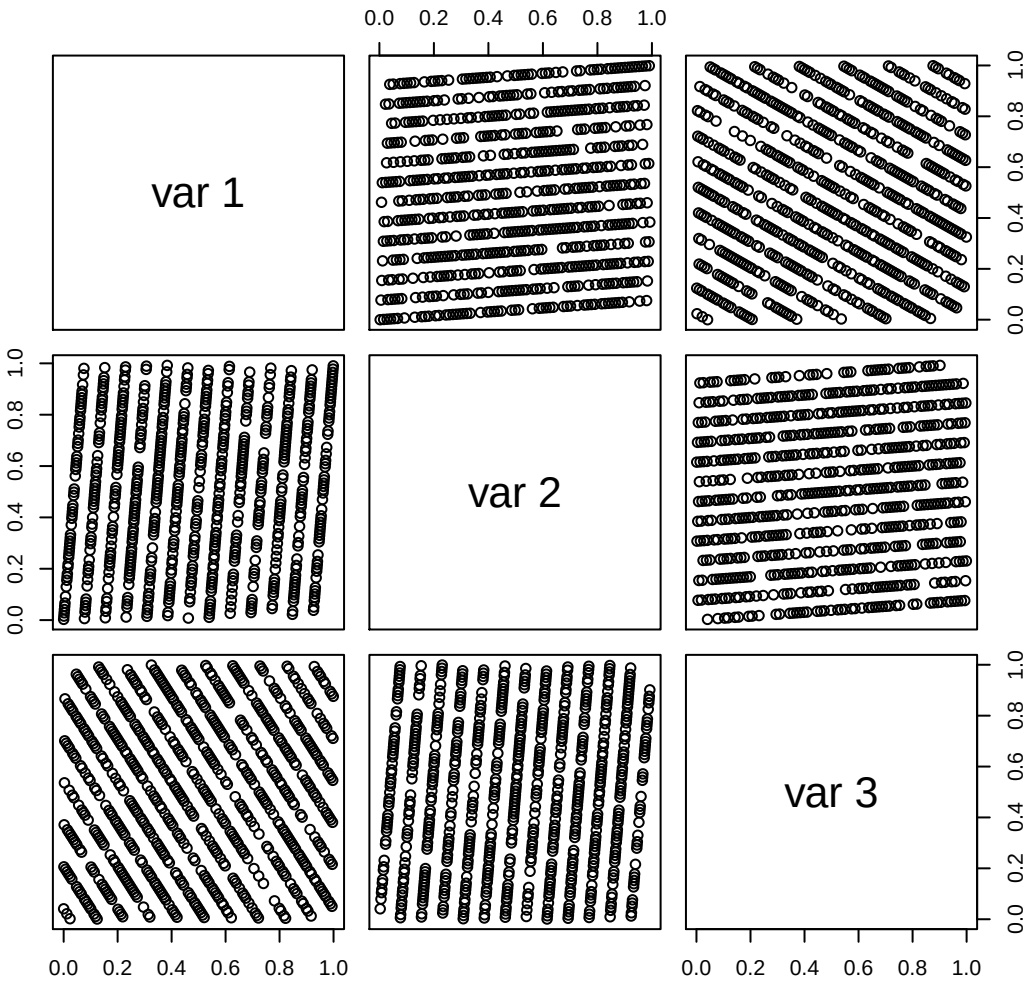

Figure 3.5: This figure shows a Pairs Plot of consecutive triples of LCG3. Due to the large sample size for LCG3, we observe planes within the data even in this initial plot. It is already clear we would reject randomness based soley on this plot.

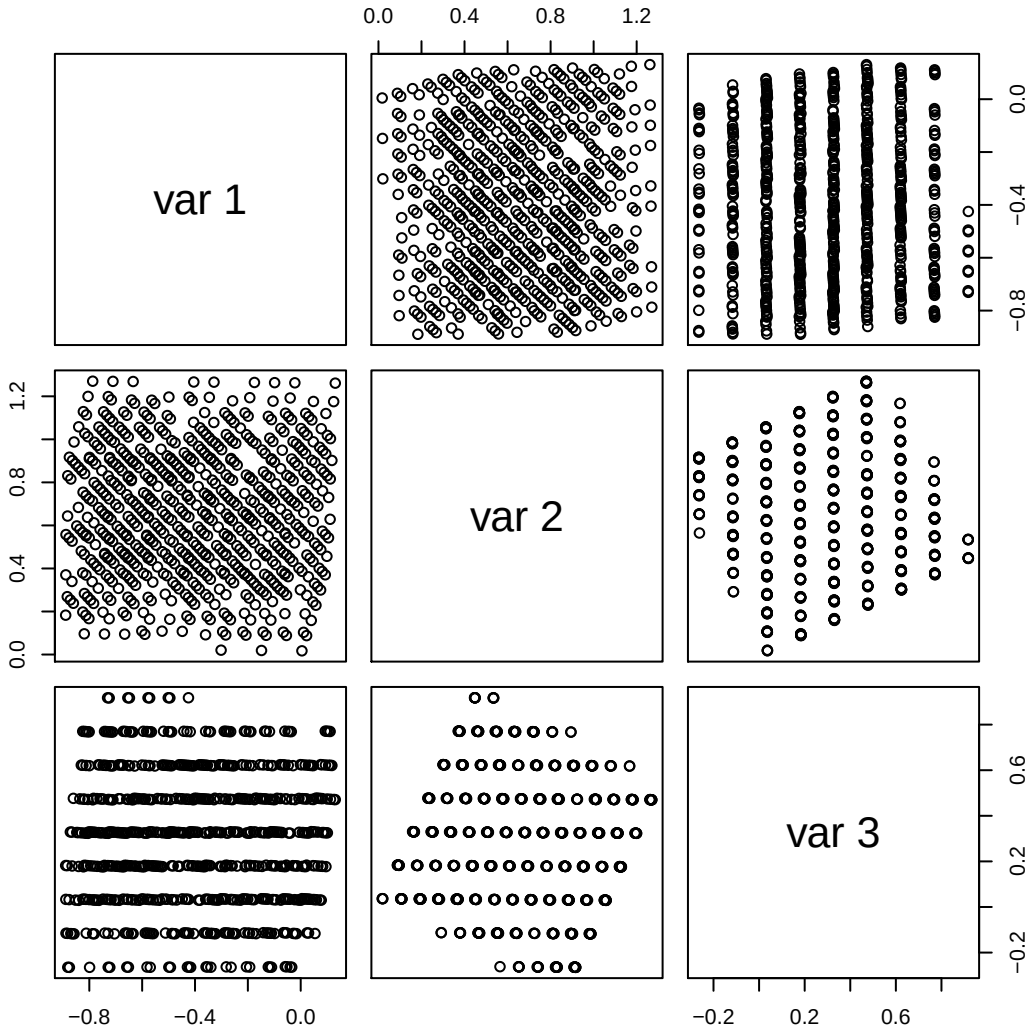

Figure 3.6: This figure shows a Pairs Plot of ICS transformed consecutive triples of LCG3. We see a reduced amount of planes present in this plot compared to the initial pairs plot for LCG3, indicating even less randomness to the output of LCG3 than initially thought.

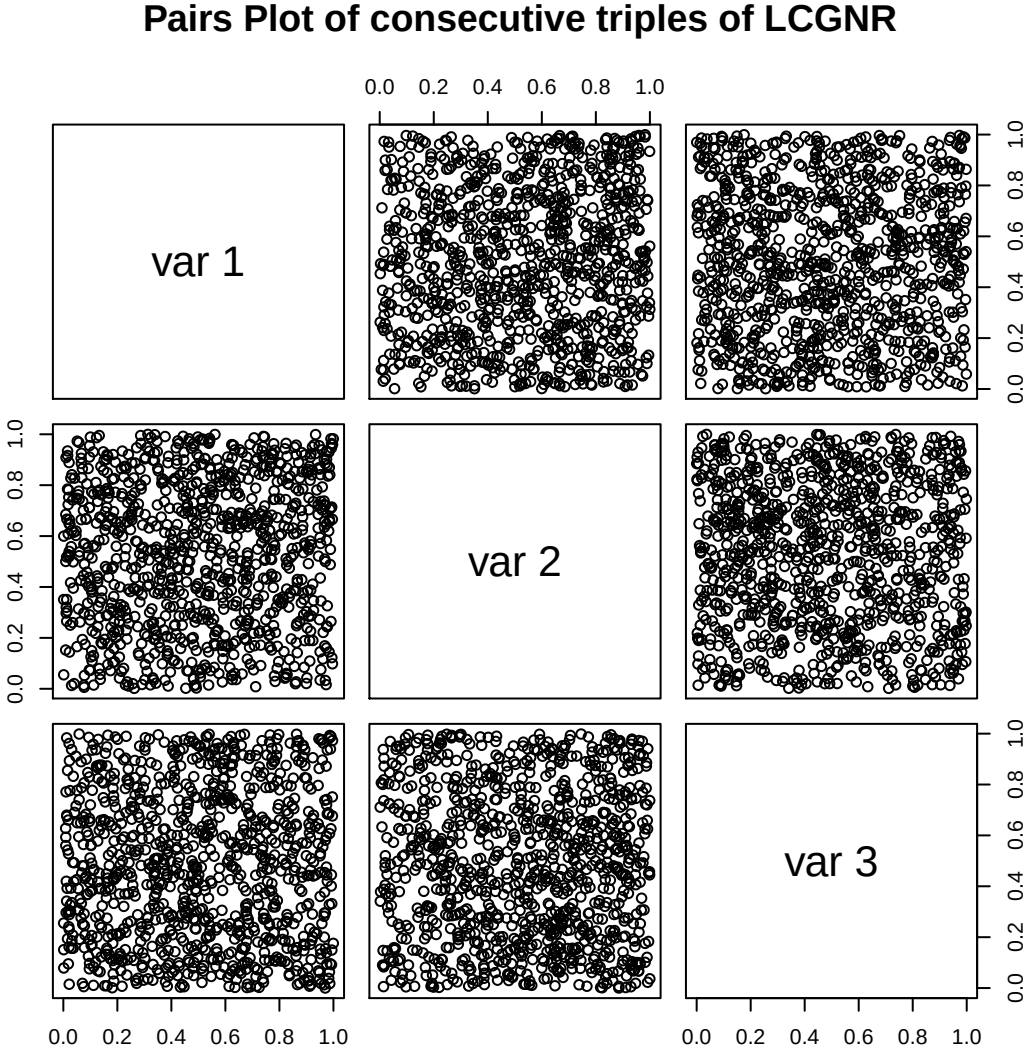

Figure 3.7: This figure shows a Pairs Plot of consecutive triples of LCGNR. We certainly cannot see any planes within the data here, the data looks to be randomly distributed and is a vast improvement over initial pairs plots of previous LCGs.

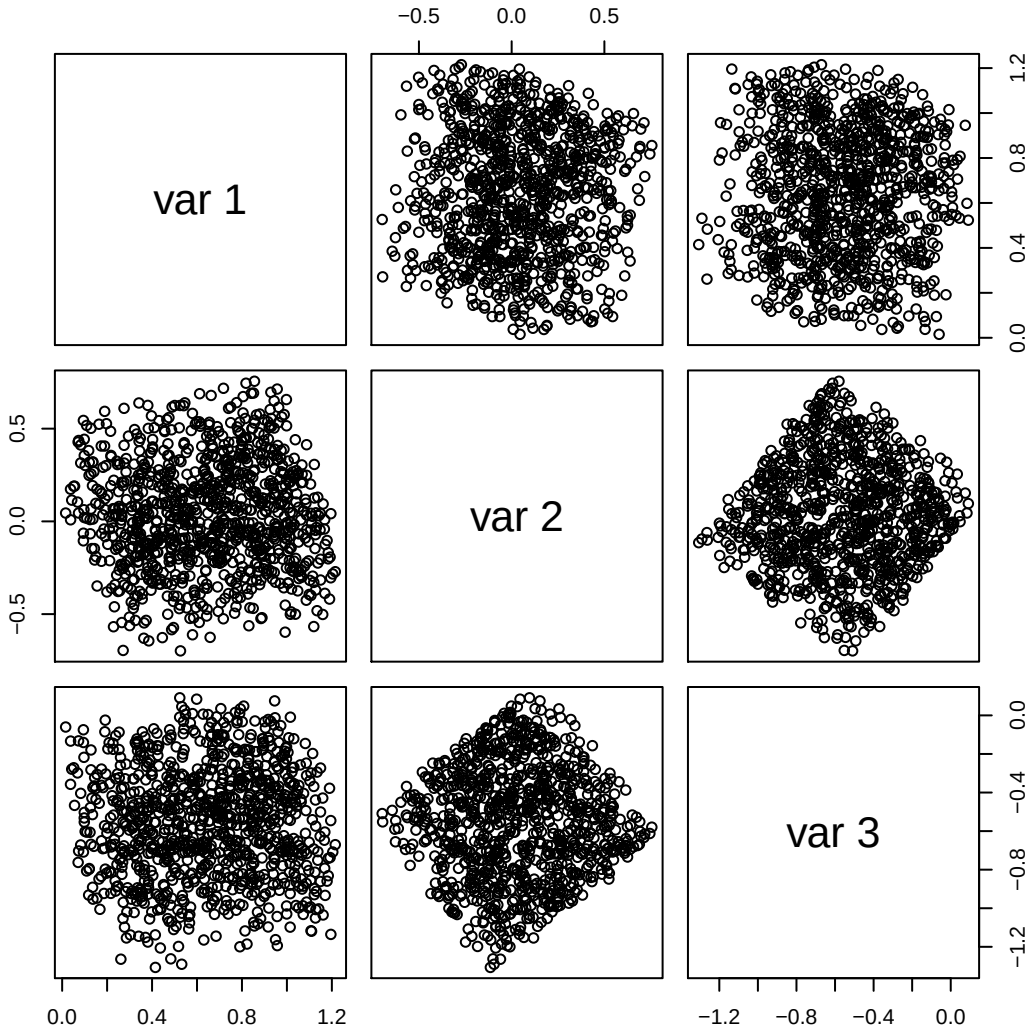

Figure 3.8: This figure shows a Pairs Plot of ICS transformed consecutive triples of LCGNR. Although ICS has been applied to the data, we see no real change to the distribution of the data. It still appears randomly distributed and no planes whatsoever are visible in the plot. We simply cannot reject our null hypothesis of randomness within the data here, and conclude that LCGNR has passed ICS.

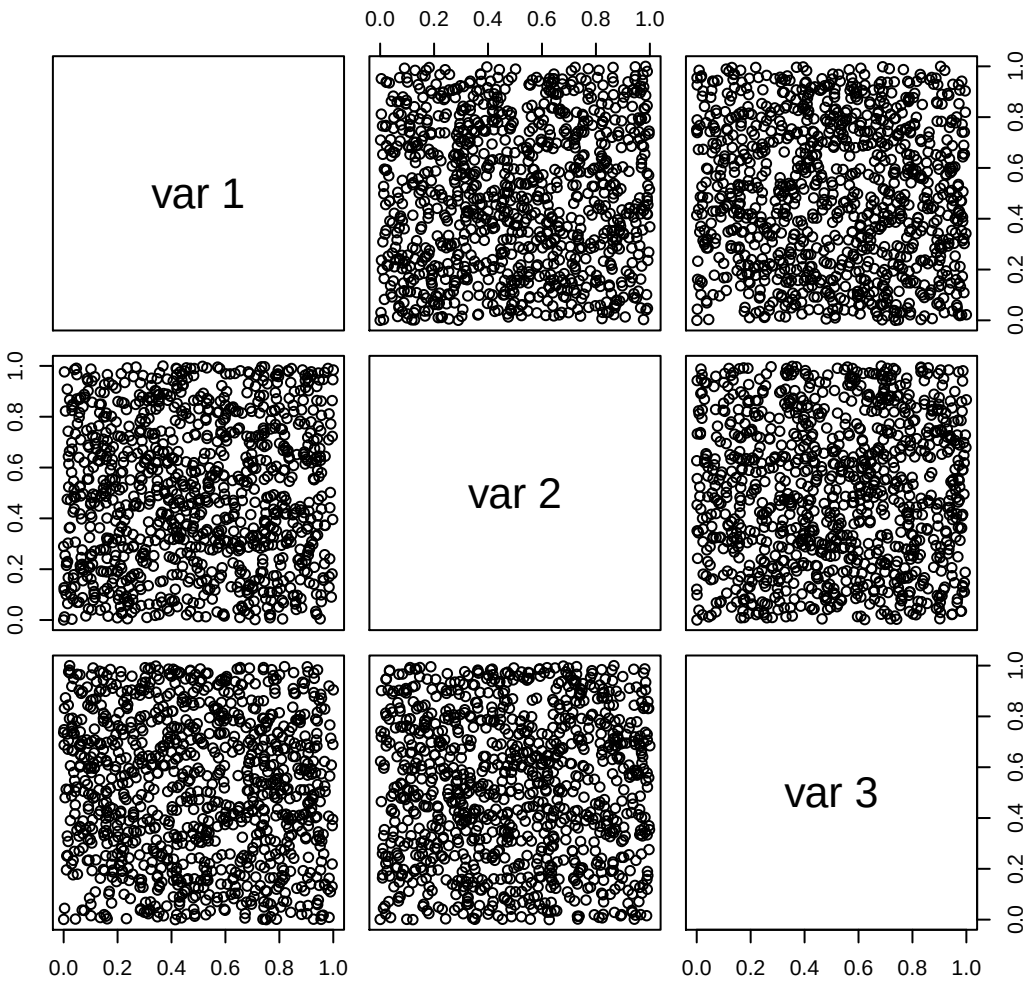

Figure 3.9: This figure shows a Pairs Plot of consecutive triples of RANDU. This initial pairs plot is very similar to the pairs plot for LCGNR. We cannot spot any planes within the data and the distribution of values appear sufficiently random. We would not suspect a lack of randomness from seeing this plot for RANDU.

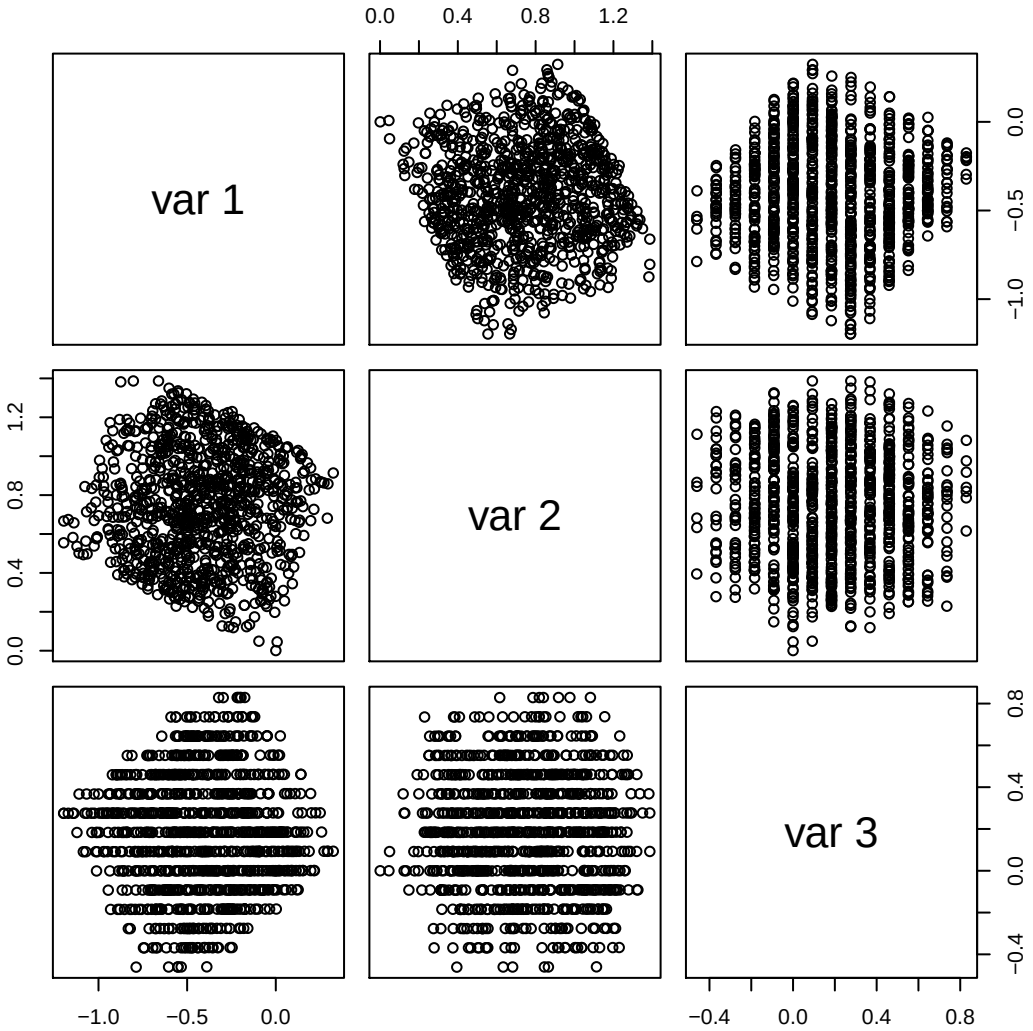

Figure 3.10: This figure shows a Pairs Plot of ICS transformed consecutive triples of RANDU. We witness an incredible difference here between the initial pairs plot of RANDU.15 planes are now clearly visible in this plot, where all values are found. This is also starkly different from the transformed pairs plot for LCGNR, which showed no such planes within the data. We would strongly reject a null hypothesis of randomness in this case and deduce that ICS has helped uncover a hidden structure within the generator RANDU.

Chapter 4

Conclusion

In this section I intend to bring together the results of how the 5 LCGs performed in the tests in each section and then reflect upon this. The results are summarised within the table below.

Test	LCG1	LCG2	LCG3	LCGNR	RANDU
Chi-square	Pass	Fail	Fail	Pass	Pass
Kolmogorov-Smirnov 1	Fail	Fail	Fail	Pass	Pass
Kolmogorov-Smirnov 2	Fail	Fail	Fail	Pass	Fail
Spearman's Rank Correlation Coefficient	Fail	Pass	Fail	Pass	Fail
Runs Test 1	Fail	Fail (Does not meet criteria)	Fail (Does not meet criteria)	Pass	Pass
Runs Test 2	Pass	Fail (Does not meet criteria)	Fail (Does not meet criteria)	Pass	Fail
Invariant Coordinate Selection	Fail	Fail	Fail	Pass	Fail
Total Passed	2	1	0	7	3

We see that LCGNR passes all tests in this report, and is a typical trait of widely used PRNGs. Good PRNGs pass most statistical tests, or are not considered good and will not see widespread use. The obvious exception to this is RANDU, although it is no longer considered a good PRNG, nor is it widely used anymore. We see it managed to pass three of the seven tests here and although this is not a considerable amount, it is possible to see why it could have been regarded more highly than it is today. Its output was still far superior to the user created LCGs; LCG1, LCG2 and LCG3; as these were chosen more to showcase the the various tests than as serious generators. LCG2 breaks a fundamental rule with regards to LCGs, LCG3 is ran for too long compared to its period length, and although LCG1 does not break any rules, the parameters are simply too small to be taken seriously as a generator and this is reflected in the amount of tests it passed, only two out of a total of seven.

Overall, it is clear that for a LCG to earn the title of being a good PRNG, it is necessary for it to pass a large number of tests. The best way to do this is to adhere to the rules relating to period length, and to also use huge numbers for the parameter values, such is the case for LCGNR, which was determined to be the best generator tested in this report.

Bibliography

[aFCDO09] David E. Tyler amd Frank Critchley, Lutz Dümbgen, and Hannu Oja. Invariant coordinate selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):549–592, 2009.

[And83] D. André. Sur le nobre de permutations de n éléments qui présentent s séquences. *Comptes Rendus*, 97:1356–1358, 1883.

[Bra68] James V. Bradley. *Distribution-Free Statistical Tests*. Prentice-Hall, 1968.

[FP61] E. C. Fieller and E. S. Pearson. Tests for rank correlation coefficients: II. *Biometrika*, 48(1/2):29–40, June 1961.

[FPH57] E. C. Fieller, E. S. Pearson, and H. O. Hartley. Tests for rank correlation coefficients: I. *Biometrika*, 44(3/4):470–481, December 1957.

[GN96] Priscilla E. Greenwood and Mikhail S. Nikulin. *A Guide to Chi-Squared Testing*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc, 1996.

[KDS51] M. G. Kendall, S. T. David, and A. Stuart. Some questions of distribution in the theory of rank correlation. *Biometrika*, 38(1/2):131–140, June 1951.

[Ken49] M. G. Kendall. Rank correlation and product-moment correlation. *Biometrika*, 36(1/2):177–193, June 1949.

[Knu69] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 1969.

[Mey00] Carl D. Meyer. *Matrix Analysi and Applied Linear Algebra*. SIAM, 2000.

[MKB88] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academia Press, Inc, 1988.

[MN98] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

[Mor48] P. A. P. Moran. Rank correlation and product-moment correlation. *Biometrika*, 35(1/2):203–206, May 1948.

[Str09] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley - Cambrige Press, fourth edition, 2009.

[UC08] Graham Upton and Ian Cook. *A Dictionary of Statistics*. Oxford University Press, second edition, 2008.

[Wol73] Hollander Wolfe. *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc, 1973.