CESAR AUGUSTO ROSA

NÚMEROS ALEATÓRIOS
GERAÇÃO, QUALIDADE E APLICAÇÕES

Santo André, 2016

UNIVERSIDADE FEDERAL DO ABC

CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO

CESAR AUGUSTO ROSA

NÚMEROS ALEATÓRIOS
GERAÇÃO, QUALIDADE E APLICAÇÕES

Orientador: Prof. Dr. Vinicius Cifú Lopes

Dissertação de mestrado apresentada ao Centro de Matemática, Computação e Cognição para obtenção do título de Mestre

ESTE EXEMPLAR CORRESPONDE A VERSÃO FINAL DA DISSERTAÇÃO
DEFENDIDA PELO ALUNO CESAR AUGUSTO ROSA,
E ORIENTADA PELO PROF. DR. VINICIUS CIFÚ LOPES.

SANTO ANDRÉ, 2016

Dedico este trabalho ao pequeno Hugo.

AGRADECIMENTOS

Agradeço ao meu pai, Fabio Luiz (*in memorian*) e à minha mãe, Flora, pelo empenho e sacrifício por garantir aos filhos a melhor educação possível. Agradeço à minha esposa Adriana pela paciência, compreensão e apoio. Aos meus irmãos Leilah e Fabinho, pelo incentivo e ajuda. Agradeço aos professores da UFABC pela disposição, aos colegas de mestrado pelo companheirismo, aos professores do IMPA, que me influenciaram, à distância. Agradeço à instituição UFABC por me proporcionar um curso com excelência. Agradeço aos professores que compõem a banca, Cristian Colleti e Gleiciane Aragão. Agradeço especialmente meu orientador, Prof. Vinicius Cifú Lopes, pela paciência, pelas ideias e por me corrigir as falhas.

Enfim agradeço ao meu pequeno Hugo, que com seus 4 anos, até assistia comigo aulas em vídeo e desenhava lindamente em meus cadernos de estudo. Sem isso nada teria graça.

"Ao contrário dos ilusionistas, que nunca revelam como operam seus truques, os matemáticos não sentem essa necessidade de guardar segredo."

(Martin Gardner, *Ah, Apanhei-te!*)

RESUMO

Este trabalho apresenta brevemente o conceito de aleatoriedade e de sequência de números aleatórios. Em especial são apresentados alguns exemplos de problemas que podem ser resolvidos utilizando "bons" números aleatórios associados a algoritmos computacionais. Também é explicada a diferença entre números verdadeiramente aleatórios e pseudoaleatórios. São introduzidos alguns métodos básicos de geração de números pseudoaleatórios, envolvendo alguns conceitos matemáticos, principalmente teoria elementar dos números. Em seguida são mencionados alguns testes de qualidade e métodos de transformação de variáveis. Ao final é apresentada uma sugestão de atividade para o ensino médio.

Palavras-chave: Simulação, probabilidade, números aleatórios, números pseudoaleatórios, congruências, método congruencial linear, teste de sequência.

ABSTRACT

This thesis briefly presents the concepts of randomness and random numbers sequence. More specifically, we discuss a few examples of problems which can be solved using the concept of "good" random numbers, associated to computational algorithms. We mention as well the difference between true random numbers and pseudo random numbers. Then we introduce a number of basic methods to generate pseudo random numbers, in which we discuss the related mathematical concepts, mainly from elementary numbers theory. A presentation follows, of some quality tests applicable to the theory, and methods of variable transformation. We close with a suggestion of how to use those concepts in an activity targeting high school students.

Keywords: Simulation, probability, random numbers, pseudo random numbers, congruencies, linear congruential methods, run test.

CONTEÚDO

INTRODUÇÃO

Pode causar certa estranheza para um público em geral – e aqui incluímos estudantes de ensino médio e fundamental – um trabalho sobre números aleatórios. Afinal, números são entidades que representam naturalmente quantidades e medidas. Quando acrescentamos o adjetivo *aleatório* parece vir imediatamente a pergunta: como assim *números aleatórios*? O foco da questão passa a ser *aleatório*. Mas o que vem a ser isso? Tecnicamente podemos simplesmente dizer, de acordo com Filho (1995, p. 61), que um número aleatório é uma variável aleatória com distribuição uniforme no intervalo (0,1). Mas esta definição técnica não deve obstar o senso comum do que seja aleatório; ao jogarmos um dado, o resultado é absolutamente imprevisível, embora saibamos algo sobre ele a priori. Cada face, considerando o dado honesto, tem uma chance em seis de ser observada em um lançamento, ou seja, tem probabilidade 1/6. Assim, se sabemos apenas a probabilidade de um evento ocorrer, então esse evento é dito aleatório.

Mas chegamos a um evento aleatório, e não a um número aleatório, ainda. Como afirmar que um número é aleatório? Se associarmos o número da face ao experimento do lançamento do dado, teremos os números de 1 a 6 sendo obtidos, aleatoriamente, conforme o experimento for sendo repetido. Vamos a um resultado: 6, 6, 3, 2, 3, 4, 5, 1, 5, 2. Essa sequência foi obtida no lançamento de um dado honesto por 10 vezes. Essa é uma sequência aleatória, pois cada face foi sorteada por mero acaso, considerando uma importante premissa: o dado é honesto. Um dado honesto é um dado que não favorece um resultado em detrimento de outro. (Se o dado fosse, por exemplo, mais pesado em um lado, ele favoreceria mais a observação de uma face do que as demais, o que naturalmente geraria um resultado viciado. A nova sequência ainda seria aleatória visto que os resultados ainda seriam imprevisíveis, embora não possua mais uma distribuição uniforme.) Então, repetindo um experimento uma grande quantidade de vezes, teremos uma sequência observada de valores. Agora, como saber se tal sequência é mesmo aleatória? Parece haver (e de fato há) um certo critério que pode determinar que uma dada sequência pareça ser mesmo aleatória. Isso dependerá de que tal sequência seja submetida a alguns testes estatísticos e que a hipótese de

aleatoriedade não seja rejeitada.

A geração de números aleatórios não é feita por mera curiosidade. Há um vasto campo de aplicações onde os números aleatórios constituem uma matéria-prima imprescindível na obtenção de resultados. A aplicação mais comum de números aleatórios é a simulação de sistemas, mais especialmente sistemas estocásticos (e aqui neste trabalho vamos adotar estocástico como sinônimo de aleatório, muito embora a rigor não seja). Esse tipo de simulação é também conhecida como Método Monte Carlo. Este método consiste basicamente na obtenção de resultados numéricos através de amostras aleatórias. Por este método, um experimento pode ser realizado repetidamente um grande número de vezes até se obter um resultado numérico — que pode ser uma probabilidade, ou um parâmetro populacional. Esse método pode ser aplicado se o sistema estudado possui aleatoriedade na ocorrência de eventos que a ele pertencem. Dessa forma os números aleatórios serão uma fonte de dados, aptos a gerarem as amostras aleatórias. Por exemplo, se queremos simular a formação de fila de automóveis em um semáforo instalado em um cruzamento, precisamos ter um gerador que gerará números aleatórios, que, especialmente manipulados, simularão uma chegada virtual de veículos ao cruzamento, obedecendo a um parâmetro de intervalo de tempo entre chegadas. Uma fila poderá se formar quando o semáforo estiver fechado e diminuirá assim que aparecer o sinal verde.

Há ainda aplicações em diversas áreas, como segurança da informação (geração de senhas aleatórias seguras), criptografia (com caracterização especial de imprevisibilidade do próximo bit aleatório), estética computacional, entretenimento (bingos, cassinos, etc), entre outras.

Neste trabalho apresentaremos alguns dos aspectos mais relevantes sobre números aleatórios, tais como aplicações, geração e testes de qualidade, bem como um exemplo de aplicação a ser abordado em sala de aula, no ensino médio. No Capítulo 1 apresentaremos alguns problemas que podem ser resolvidos com a ajuda de números aleatórios, além de exemplos de estimação de integrais definidas. No Capítulo 2 trataremos de números verdadeiramente aleatórios. No Capítulo 3 mostraremos os números pseudoaleatórios, bem como uma breve apresentação do ferramental matemático que rege uma substantiva quantidade de geradores. Ainda no mesmo capítulo iremos mostrar alguns geradores. Abordaremos testes de qualidade de números aleató-

rios no Capítulo 4, onde também mencionamos algumas transformações para se obter outros números aleatórios. Por fim, no último capítulo, apresentaremos uma sugestão de aplicação para o ensino médio. Desta forma cobrimos os principais pontos do tema de números aleatórios.

1

PROBLEMAS

Vamos apresentar neste capítulo alguns exemplos de problemas com duas abordagens diferentes: a clássica (ou probabilística, ou matemática) e a simulação usando números aleatórios.

1.1 PROBLEMA DO ÁLBUM DE FIGURINHAS

Um exemplo de aplicação simples e de fácil assimilação é a da coleção de figurinhas. Digamos ter adquirido um álbum de figurinhas e queremos completar o álbum. Assim, podemos estar interessados em saber qual é a quantidade esperada de figurinhas que devem ser adquiridas para que o álbum fique completo. Na Teoria das Probabilidades esse valor é conhecido como *esperança*. Não vamos considerar que se compram figurinhas em *pacotinhos* com 3, 4 ou 5 figurinhas, podemos retirar esse fato do problema sem perder sua essência e generalidade. É sabido, primeiramente, que quando se compra figurinhas num envelope, não sabemos quais são as figurinhas que estamos comprando, portanto as figurinhas são adquiridas de forma aleatória. Cada figurinha é numerada de 1 a N, onde N é o número total de figurinhas do álbum. É muito intuitivo admitir que quanto maior o número de figurinhas de um álbum, mais delas tenhamos que comprar. Isso indica claramente que a quantidade esperada de figurinhas a ser comprada é função de N. Estamos admitindo aqui que não iremos trocar figurinhas. E também vamos admitir que a obtenção aleatória de cada figurinha é independente uma da outra, ou seja, se obtivermos a figurinha número 12, a probabilidade de se obter novamente a figurinha 12, ou qualquer outra, não se altera.

Esse assunto já foi estudado fartamente, como podemos verificar no artigo de Carvalho (2010, p. 37). O pesquisador concluiu que o número esperado m de figurinhas a serem compradas é

$$m = N\left(1 + \frac{1}{2} + \ldots + \frac{1}{N}\right)$$

onde N é a quantidade de figurinhas do álbum.

Para entender como o pesquisador chegou a esta conclusão, teremos que estudar alguns aspectos da Teoria de Probabilidades. Como visto nos dois parágrafos anteriores, o *valor esperado*, ou simplesmente esperança, é uma quantidade que se espera ser observada, dado que seja feito um certo experimento aleatório. Esse valor pode se traduzir também como a média de valores observada em um experimento. Para exemplificar, vamos analisar um caso mais simples: Jogamos uma moeda não viciada 10 vezes. Qual é a quantidade mais provável de caras observadas? A resposta para essa pergunta é 5. Embora não saibamos, antes de realizar o experimento, qual será o número obtido, o resultado mais provável é 5. Se repetirmos o experimento muitas e muitas vezes, digamos 100 vezes, cada um consistindo de 10 lançamentos, obteremos 100 resultados, e quando calcularmos a média destes resultados, estaremos próximos de 5. Quanto maior for a quantidade de repetições deste experimento, mais próximo de 5 estaremos. Não vamos neste momento analisar o que há por trás deste aparente fenômeno. Porém podemos desde já definir que o valor médio esperado (ou esperança) é a somatória de todos os valores possíveis multiplicados cada um deles pela probabilidade de cada valor ser observado. Se num experimento aleatório podemos observar os valores $x_1, x_2, x_3, \ldots$, com probabilidades $p_1, p_2, p_3, \ldots$, respectivamente, então o valor médio esperado é $m = p_1x_1 + p_2x_2 + p_3x_3 + \ldots$

Para aplicar essa definição no caso da moeda lançada 10 vezes, vamos atribuir valor 1 para cara e 0 para coroa. Como a moeda é honesta, a probabilidade de obter cara é $\frac{1}{2}$. Assim, o valor esperado será de

$$m = \underbrace{\frac{1}{2}.1 + \frac{1}{2}.1 + \ldots + \frac{1}{2}.1}_{10\,\text{parcelas}} = \frac{1}{2}.10 = 5$$

Agora sabemos o que significa esperança, ou valor esperado, ou valor médio. Mas antes de chegar no problema das figurinhas, vamos ainda analisar uma outra situa-

ção: vamos supor que queiramos saber, em média, quantos lançamentos de um dado devemos ter até que a face 6 seja obtida. Sabemos que a face 6 deverá sair com probabilidade $\frac{1}{6}$, portanto contamos com a sorte para que a face 6 saia em apenas um lançamento. É muito mais provável não sair face 6 porque essa probabilidade é de $\frac{5}{6}$. Mas sabemos que podemos obter 6 com uma tentativa, com duas, três, etc., enfim, o número de tentativas pode ser qualquer valor inteiro positivo. Queremos agora saber, em média, quantas tentativas são necessárias até obter face 6. Desta forma, estamos diante de uma situação semelhante à do lançamento do dado: se há probabilidade p_1 de obter face 6 com uma tentativa, p_2 com 2 tentativas, p_3 com 3 tentativas, e assim por diante, resta apenas saber qual é a probabilidade p_n para n tentativas. Obter face 6 (sucesso) na n-ésima tentativa implica em obter face diferente de 6 (fracasso) nas $n-1$ tentativas anteriores. Assim, para uma tentativa considerando a probabilidade p como sucesso (obter face 6), teremos $1-p$ como fracasso (obter face diferente de 6) e consequentemente, $p_n = (1-p)^{n-1}p$.

A rigor, o número de tentativas independentes necessárias até se obter sucesso em um experimento é uma variável aleatória discreta que obedece a uma Distribuição Geométrica com parâmetro p, onde p é a probabilidade de obter sucesso. Uma referência deste assunto pode ser encontrado em Magalhães, Lima (2000, p. 76). Neste momento não faremos mais comentários a respeito do assunto de variáveis aleatórias, pois o conteúdo abordado até aqui é suficiente para o estudo do caso.

Resta-nos saber agora qual é o valor médio esperado (ou esperança) de tentativas até se obter face 6. Mas ao invés de fazer esse cálculo particular, vamos generalizar. Vamos calcular, para um experimento aleatório, o valor médio de tentativas necessárias até se obter o primeiro sucesso, dado que as tentativas são independentes uma da outra. Como vimos anteriormente, o valor médio é dado por:

$$m = p_1.1 + p_2.2 + p_3.3 + \ldots =$$
$$= p.1 + (1-p).p.2 + (1-p)^2.p.3 + \ldots$$

Essa soma infinita assemelha-se à soma de elementos de uma progressão geométrica, embora em absoluto não seja. Entretanto podemos reescrevê-la convenientemente assim:

$$\begin{aligned} m = p + p(1-p) + p(1-p)^2 + p(1-p)^3 + \ldots \\ + p(1-p) + p(1-p)^2 + p(1-p)^3 + \ldots \\ + p(1-p)^2 + p(1-p)^3 + \ldots \\ \vdots \end{aligned}$$

Vamos denotar cada linha i da soma acima, por L_i. Então teremos:

$$m = L_1 + L_2 + L_3 + \ldots$$

Podemos observar que L_1 é a soma dos termos de uma progressão geométrica cujo primeiro termo a_1 vale p e cuja razão q vale $(1-p)$. Sabemos do ensino médio, que a soma dos n primeiros termos de uma PG é dada pela expressão:

$$S_n = a_1 \frac{1-q^n}{1-q}$$

Cabe observar que a soma de L_1 possui infinitos termos. Assim, para n muito grande, tendendo ao infinito, devemos calcular o limite da soma no infinito:

$$\lim_{n \to +\infty} S_n = \lim_{n \to +\infty} a_1 \frac{1-q^n}{1-q}$$

Mas como temos $q = (1-p)$ e $a_1 = p$, então:

$$S = \lim_{n \to +\infty} S_n = \lim_{n \to +\infty} a_1 \frac{1-(1-p)^n}{1-(1-p)} = a_1 \frac{1 - \lim_{n \to +\infty}(1-p)^n}{p} = \frac{a_1}{p}$$

Logo, para L_1, temos $a_1 = p$, então

$$L_1 = \frac{p}{p} = 1$$

Podemos agora observar também que todas as linhas L_i são somas de elementos de uma PG. Em particular L_2 possui $a_1 = p(1-p)$ e é calculado assim:

$$L_2 = \frac{p(1-p)}{p} = 1 - p$$

Por sua vez L_3 é a soma de termos de uma PG cujo primeiro termo é $p(1-p)^2$, então calculamos:

$$L_3 = \frac{p(1-p)^2}{p} = (1-p)^2$$

Aplicando essa ideia indutivamente, podemos calcular a soma dos L_i:

$$m = L_1 + L_2 + L_3 + \ldots = 1 + (1-p) + (1-p)^2 + \ldots$$

Mas isso é a soma dos termos de uma nova PG infinita, cujo primeiro termo $a_1 = 1$ e razão $q = (1-p)$. Logo essa soma será:

$$m = \frac{1}{p}$$

Com isso concluímos que a quantidade esperada de repetições (independentes) que devem ser realizadas até se obter o primeiro sucesso é $m = \frac{1}{p}$, onde p é a probabilidade

de sucesso. Isso se aplica ao caso dos lançamentos de um dado até se obter a primeira face 6. Neste caso teremos

$$m = \frac{1}{\frac{1}{6}} = 6$$

Isso significa que, em média, são necessários 6 lançamentos de um dado para se obter uma face 6.

Agora vamos voltar ao álbum de figurinhas. Vamos denotar por N a quantidade total de figurinhas do álbum e por n a quantidade de figurinhas que já estão coladas no álbum, em um certo momento, sendo que n está entre zero e N ($0 \leq n \leq N$). A esta altura, compramos (sorteamos) uma figurinha. Ela terá probabilidade $q = \frac{n}{N}$ de ser repetida. Consequentemente, a probabilidade de sucesso p de não ser repetida é $p = 1 - q = 1 - \frac{n}{N} = \frac{N-n}{N}$.

De posse do valor de p ($p = \frac{N-n}{N}$), podemos desde já conhecer a quantidade média de figurinhas que devem ser adquiridas (sorteadas) até se obter a próxima figurinha não repetida. De acordo com o que fizemos há pouco, essa quantidade média é dada por $\frac{1}{p}$, que neste caso será $\frac{1}{p} = \frac{N}{N-n}$.

Agora vamos adotar a seguinte estratégia: ao iniciar a coleção, temos $n = 0$ figurinhas; logo a chance de sucesso ao adquirir a primeira figurinha é 1. Esta quantidade será designada por F_0. Em seguida precisaremos adquirir, em média, a quantidade F_1 de figurinhas até ser colada a segunda figurinha no álbum. Essa quantidade calculamos assim: $F_1 = \frac{N}{N-1}$. Da mesma forma, $F_3 = \frac{N}{N-2}$, $F_4 = \frac{N}{N-3}$ e assim por diante, até F_{N-1}. Então, o total esperado m de figurinhas que devem ser adquiridas até completar o álbum é o total de figurinhas esperado até colar a 1ª figurinha, mais o total esperado de figurinhas até colar a 2ª figurinha, mais o total esperado de figurinhas até colar a 3ª figurinha, e assim por diante, até colar a última figurinha, o que traduz da seguinte forma:

$$m = F_0 + F_1 + F_2 + \ldots + F_{N-1}$$

Substituindo os F_i pelas expressões em função de N, teremos:

$$m = 1 + \frac{N}{N-1} + \frac{N}{N-2} + \ldots + \frac{N}{2} + N =$$

$$= \frac{N}{N} + \frac{N}{N-1} + \frac{N}{N-2} + \ldots + \frac{N}{2} + N =$$
$$= N\left(1 + \frac{1}{2} + \ldots + \frac{1}{N-1} + \frac{1}{N}\right).$$

Assim, se conhecemos a quantidade N de figurinhas do álbum, podemos calcular a quantidade média de figurinhas que iremos adquirir, bastando para isso substituir o valor de N na fórmula acima. Por exemplo, o álbum de figurinhas oficial da Copa do Mundo 2014 possuía 649 figurinhas (ver PANINI). Sem considerar trocas, a quantidade de figurinhas a ser comprada, em média, para completar o álbum, pode ser calculada pela fórmula acima. Porém o cálculo é extenso e será necessária ajuda de uma planilha eletrônica ou um programa de cálculo, ou mesmo uma aplicação *online*, como o Wolfram Alpha (ver WOLFRAMALPHA). Para o álbum de 649 figurinhas da Copa do Mundo de 2014, o valor calculado com a aplicação *online* foi de 4577,67; ou seja espera-se adquirir em média 4578 figurinhas para se preencher o álbum — desconsiderando trocas.

No entanto, esse problema poderia ser tratado sob o ponto de vista da simulação (Monte Carlo) — assim como outro qualquer que envolva aleatoriedade. Vamos ver agora como poderemos resolver esse problema através da simulação, e consequentemente, dos números aleatórios. Vamos inicialmente analisar a mecânica do problema com o fluxograma da Figura 1.

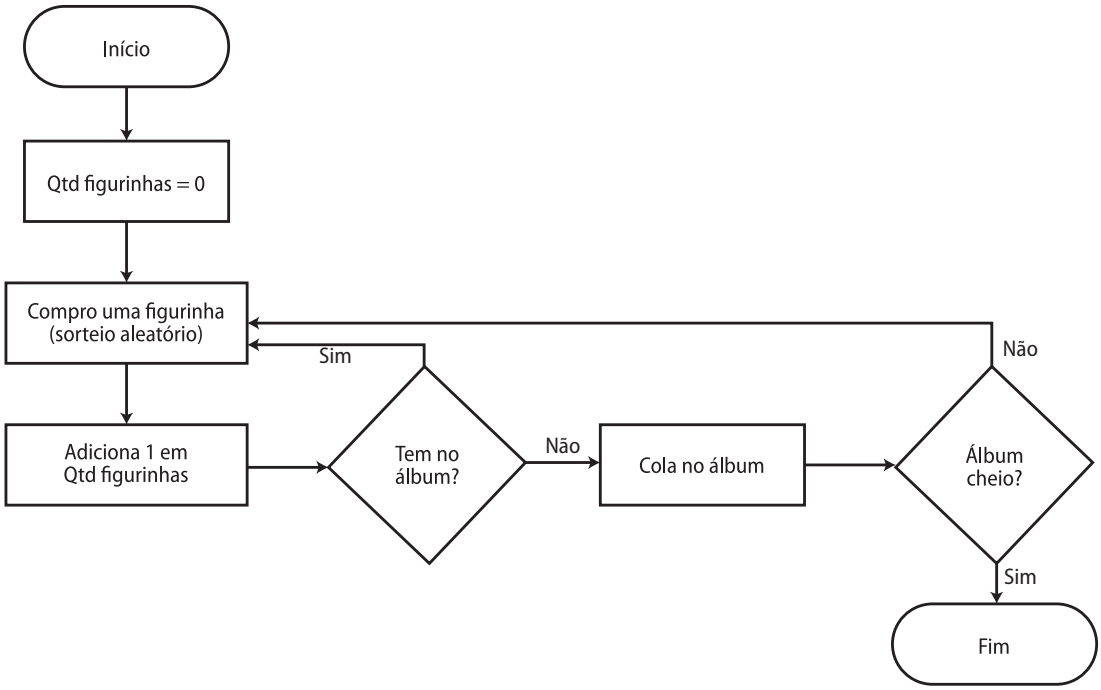


Figura 1: Fluxograma de uma coleção de figurinhas.

O esquema da Figura 1 pode ser facilmente implementado através de um programa de computador. Neste trabalho e para este exemplo apresentamos uma implementação em linguagem Perl. A seguir, o código fonte:

Listagem 1.1: Código-fonte Perl para simular o problema do álbum de figurinhas

```
use strict ;
use warnings;
srand(time ^$$);
my $qtd_fig = 30; # Quantidade de figurinhas do album
my %album; # Hash para 'colar' as figurinhas
for (my $i=1; $i <= $qtd_fig ; $i++)
        { $album{$i} = 0; }
my $cheio = 0; # var. booleana com status do album
my $qtd_coladas = 0;
my $qtd_compradas = 0;
while (1)
{
  $qtd_compradas ++;
  my $r = 1 + int(rand(30)); # Gera um inteiro aleatorio entre 1 e 30
  if ( $album{$r} == 0)
  {
        $album{$r} = 1 ;
        $qtd_coladas ++;
        if ( $qtd_coladas == $qtd_fig ) #Album cheio
                {
                        last;
                }
  }
}
print "Total de figurinhas compradas........: " . $qtd_compradas . "\n";
```

O código anterior é simples, ele apenas segue a ideia da Figura 1. Não iremos comentar sintaxe de linguagens e estruturas de controle, do tipo *For* ou *If*, pois o foco do trabalho não é esse. No entanto chama-nos a atenção, neste código, duas declarações: *srand* e *rand*. São basicamente dois procedimentos de geração de aleatoriedade, disponibilizados pelos desenvolvedores da linguagem. O procedimento *srand*, de acordo com o *site* de documentação do Perl (ver PERL), inicializa o gerador de números aleatórios da função *rand*, dando a ele um novo valor semente. Sem nenhum parâmetro, *srand* escolhe uma semente (semi-)aleatória. Neste exemplo, *srand* gera uma semente com base no tempo do sistema, operado com o valor do *Process ID* do programa. Já a função *rand* retorna um número aleatório entre 0 e 1, usando a semente gerada por *srand*.

Embora tenhamos falado sobre declarações específicas de uma linguagem de programação, surgiu um outro conceito no parágrafo anterior: semente. Isso terá importante papel na determinação de números aleatórios, e será um dos temas deste trabalho. Esse assunto será abordado posteriormente.

De volta ao álbum de figurinhas, vamos fazer neste ponto um comparativo dos dois métodos de resolução do seguinte problema: *"Compramos um álbum de figurinhas com 30 cromos ao todo. Dado que não serão feitas trocas, quantas figurinhas em média teremos que comprar para completar o álbum?"*

O método baseado em Teoria das Probabilidades afirma que basta usarmos $N = 30$ na fórmula

$$m = N\left(1 + \frac{1}{2} + \ldots + \frac{1}{N-1} + \frac{1}{N}\right)$$

que obteremos m, o número médio esperado de figurinhas a serem adquiridas. Efetuando o cálculo usando o Wolfram Alpha, conseguimos o valor $m = 119{,}8496$.

Por outro lado, se executarmos o programa do código acima uma grande quantidade de vezes, chegaremos aproximadamente ao mesmo número. Realizamos uma simulação 10 vezes, sendo que cada simulação executa o código acima para 1000 (hum mil) álbuns e calcula a média das quantidades obtidas. Obtivemos 10 resultados, a saber:

119,506; 119,673; 117,992; 120,567; 120,703;
119,069; 119,714; 118,463; 121,025; 118,986.

Chegamos sempre a resultados muito próximos entre si e do valor teórico. Este exemplo é uma amostra da capacidade de resolução de problemas utilizando o método Monte Carlo e números aleatórios.

1.2 PROBLEMA DA FORMAÇÃO DO TRIÂNGULO

Um outro interessante problema a ser estudado através de uma simulação Monte Carlo é o problema da formação de um triângulo:

Se dois pontos estão localizados de forma aleatória em um intervalo unitário, então eles dividem o intervalo em três segmentos. Qual é a probabilidade de que estes três segmentos formem um triângulo?

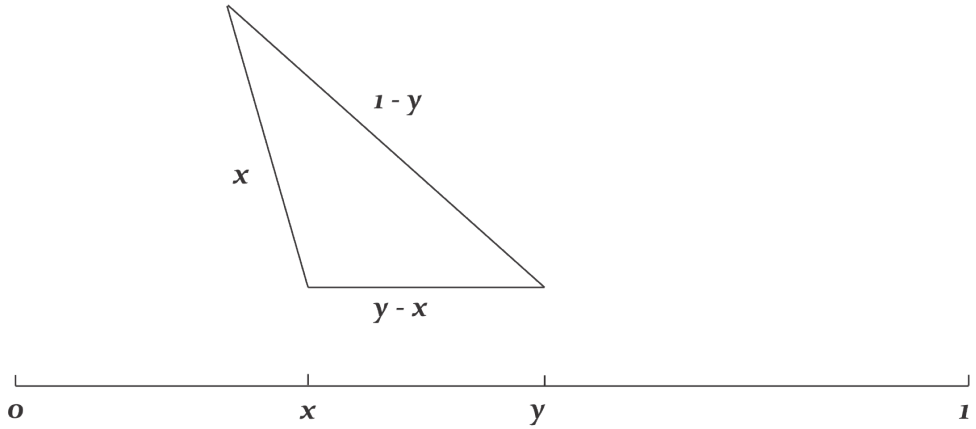


Figura 2: Dados aleatoriamente x e y, com $0 < x < y < 1$, podemos obter um triângulo caso seja satisfeita a desigualdade triangular.

Tomando dois pontos, x e y, aleatoriamente entre 0 e 1, vamos denotar por x o ponto de menor coordenada, e y o outro ponto. Desta forma teremos os segmentos x, $y - x$ e $1 - y$ (Figura 2). Para verificar se formam um triângulo, vamos usar a *desigualdade triangular*, que requer que em um triângulo, a soma de dois lados quaisquer seja sempre maior que o terceiro lado. Há três desigualdades a considerar:

(I) $x < (y - x) + (1 - y) \Leftrightarrow 2x < 1 \Leftrightarrow x < \frac{1}{2}$

(II) $(y - x) < x + (1 - y) \Leftrightarrow 2y < 2x + 1 \Leftrightarrow y < x + \frac{1}{2}$

(III) $(1 - y) < x + (y - x) \Leftrightarrow 1 < 2y \Leftrightarrow y > \frac{1}{2}$

Ainda devemos considerar a condição $x < y$.

Podemos calcular a probabilidade de que um triângulo seja formado se as condições (I), (II) e (III) são satisfeitas dado que $x < y$. Cada uma das condições pode ser representada por polígonos, conforme Figura 3.

A condição (I) é formada pelos triângulos A, B, C e E.
A condição (II) é formada pelos triângulos B, C, D, E, F e o quadrado G.

A condição (III) é formada pelos triângulos A, B, D e F.
A condição $x < y$ é formada pelos triângulos A, B, C e D.

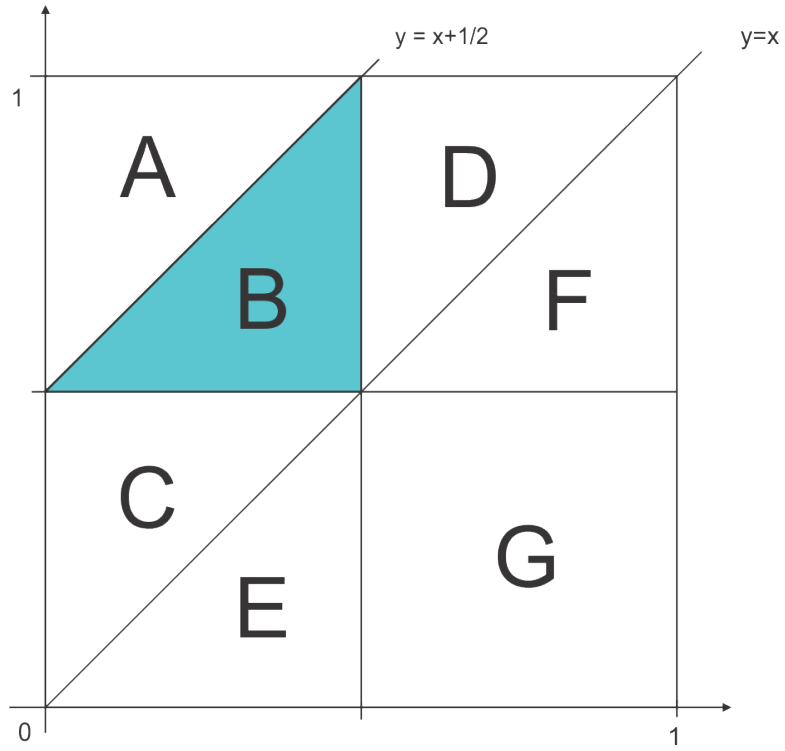


Figura 3: Gráfico auxiliar para cálculo de probabilidade de um triângulo ser formado.

Assim queremos calcular a seguinte probabilidade condicional:

$$\mathcal{P}(\textit{ Formar Triângulo } \mid x < y) = \frac{\mathcal{P}(I \cap II \cap III \cap (x < y))}{\mathcal{P}(x < y)}$$

O numerador a ser calculado equivale à intersecção de todas as condições, que resulta na área do triângulo B, em destaque na Figura 3. Esta área é igual a $\frac{1}{8}$. Como $\mathcal{P}(x < y) = \frac{1}{2}$, a probabilidade de formar um triângulo é:

$$\mathcal{P}(\textit{ Formar Triângulo } \mid x < y) = \frac{1/8}{1/2} = \frac{1}{4}$$

□

Esta forma de calcular a probabilidade é o método usado na Teoria de Probabilidades, com apoio de argumentos geométricos. Entretanto também podemos estimar esta probabilidade com o uso de números aleatórios e um algoritmo, numa simulação

computacional, tal qual o problema das figurinhas. Vamos considerar os seguintes passos:

(1) Sorteamos a_1 e a_2, onde $0 < a_1 < 1$ e $0 < a_2 < 1$;

(2) $x := min\{a_1, a_2\}$;

(3) $y := max\{a_1, a_2\}$;

(4) Se $x < \frac{1}{2}$ e $y < x + \frac{1}{2}$ e $y > \frac{1}{2}$, contamos um triângulo;

(5) Voltar ao passo (1) mais $N - 1$ vezes;

(6) A probabilidade (aproximada) de formar um triângulo é o números de triângulos dividido por N.

Uma implementação desse algoritmo é mostrada em seguida, desenvolvida em um programa em linguagem Perl. Está fixada a quantidade de 50 mil experimentos.

Listagem 1.2: Código-fonte Perl para simular o problema do fechamento do triângulo

```
use strict ;
use warnings;
use Math::Random::MT::Perl qw(srand rand);
my $a1; #numero sorteado
my $a2; #numero sorteado
my $x ; #menor dos numeros sorteados
my $y ; #maior dos numertos sorteados
my $qtd_triang = 0; #Quantidade de triangulos
my $N = 50000; #Quantidade de experimentos
my $tot=0 ;
for (my $i=0 ; $i < $N; $i++)
{
  $a1 = rand();
  $a2 = rand();
  if ($a1 < $a2)
    { $x = $a1 ;
      $y = $a2 ;  }
  else   {
    $x = $a2;
    $y = $a1;
  }
  if ( ($x<0.5) and ($y<($x+0.5)) and ($y>0.5) )
    {   $qtd_triang++;  }
}
my $prob = ($qtd_triang / $N);
print "Probabilidade estimada: $prob \n";
```

A quantidade de experimentos pode ser parametrizada no código, mas a ideia é que quanto maior essa quantidade, mais próximo da verdadeira probabilidade (25%) estaremos. Podemos observar esse resultado quando fazemos por exemplo, uma certa quantidade de ensaios, cada um com uma quantidade de experimentos diferente. Apresentaremos aqui uma tabela (Tabela 1) de resultados onde foram realizados 100 ensaios com 50 experimentos cada e obtivemos a média e a variância. Os mesmos 100 ensaios e os mesmos indicadores foram realizados para 500, 5.000 e 50.000 experimentos cada.

Quantidade de experimentos	**50**	**500**	**5000**	**50000**
Média	26.64	25.00	24.99	24.99
Variância	29.59	3.60	0.35	0.04

Tabela 1: Resultados obtidos do programa Perl que simula o problema do fechamento do triângulo.

O que podemos observar é que com 500 experimentos já obtivemos a resposta correta, embora alguma variância ainda tenha sido observada. Mas por exemplo, com 100 ensaios de 50 mil experimentos aleatórios, a média foi de praticamente 25% e a variância quase zero.

A conclusão é que o algoritmo chegou ao resultado aproximadamente correto com uma pequena margem de erro. Tal desvio pode ser cada vez menor, desde que seja aumentada a quantidade de experimentos. Mas não apenas isso: deve-se ter uma sequência de números aleatórios com "qualidade", ou seja, que sejam de fato aleatórios ou que se pareçam como tais. Essa característica será discutida mais à frente. Este é mais um exemplo da capacidade de resolução de problemas com o uso de números aleatórios.

1.3 CÁLCULO DE INTEGRAIS DEFINIDAS

Uma clássica aplicação de números aleatórios é o cálculo estimado de integrais definidas, através de uma simulação Monte Carlo. Segundo Gentle (2003, p. 229), em geral, para se estimar expressões matemáticas complexas, os métodos numéricos por

aproximação são preferidos, embora o Monte Carlo seja uma alternativa. Há casos em que este método é a única forma viável de tratar o problema computacionalmente.

Vamos abordar aqui uma breve ideia de como isso funciona. Vamos supor que queremos calcular (estimar) o valor da seguinte integral, a que chamaremos θ:

$$\theta = \int_a^b g(x)\,dx$$

onde $g(x)$ é uma função contínua em $[a, b]$, com $a < b$. O valor médio de g é, por definição,

$$E(g) = \frac{\int_a^b g(x)\,dx}{b-a} = \frac{\theta}{b-a}.$$

Então, $\theta = (b-a) \cdot E(g)$ e $E(g)$ será substituído por uma média discreta $\frac{1}{n}\sum_{i=1}^{n} g(x_i)$.

Agora vamos adotar alguns passos para calcular θ :

1. Sorteamos aleatoriamente n valores de x_i, representados por $x_1, x_2, \ldots, x_n$
2. Calculamos $g(x_i)$ para $1 \leq i \leq n$
3. Calculamos a média da seguinte forma: $\dfrac{1}{n}\displaystyle\sum_{i=1}^{n} g(x_i)$
4. Multiplicamos a média do item anterior por $(b-a)$

O resultado é uma estimativa de θ.

1.4 SOMA DE RIEMANN

Um outro exemplo para estimar integrais é calcular a *soma de Riemann*, através do uso de sorteios aleatórios de números em um intervalo $[a, b]$. Uma referência para soma de Riemann é Guidorizzi (1991, p. 298).

Seja um intervalo real $[a, b]$. Vamos definir uma partição $\mathcal{P}$, finita, como o conjunto $\{x_0, x_1, x_2, \ldots, x_n\}$, com $a = x_0 < x_1 < x_2 < \ldots < x_n = b$. Desta forma, com a partição $\mathcal{P}$ obtemos n subintervalos do intervalo $[a, b]$ (A reunião de todos os subintervalos equivale ao intervalo todo).

Vamos definir também, para cada índice i ($i = 1, 2, \ldots, n$), a amplitude do intervalo $[x_{i-1}, x_i]$ com o símbolo Δx_i, sendo que $\Delta x_i = x_i - x_{i-1}$. Cabe esclarecer que as ampli-

tudes não são necessariamente iguais, mas arbitrárias (aleatórias).

Agora vamos admitir uma função f definida em $[a, b]$ e seja $\mathcal{P}$ uma partição, como foi definida anteriormente. Para cada i ($1 \leq i \leq n$), seja c_i um número no intervalo $[x_{i-1}, x_i]$, escolhido aleatoriamente. Uma ideia desta definição está na Figura 4.

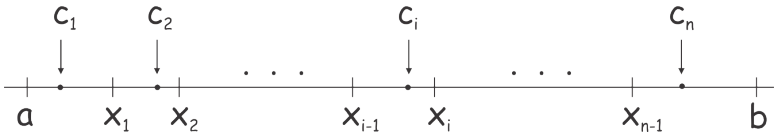


Figura 4: Partição de um intervalo $[a, b]$.

A soma de Riemann de f, no intervalo $[a, b]$, relativa à partição $\mathcal{P}$ e aos valores c_i é definida como

$$\sum_{i=1}^{n} f(c_i)\Delta x_i = f(c_1)\Delta x_1 + f(c_2)\Delta x_2 + \ldots + f(c_n)\Delta x_n$$

sendo que a integral de f no intervalo $[a, b]$ pode ser aproximada pela soma acima.

Desta forma, uma soma de Riemann pode ser obtida com um conjunto de números aleatórios entre a e b. Após formar cada subintervalo $[x_{i-1}, x_i]$ podemos sortear, para todo i um c_i aleatório entre x_{i-1} e x_i e calcular $f(c_i)$. Assim, ao final, obteremos a soma de Riemann, com a ajuda de números aleatórios, e por conseguinte uma estimativa da integral de f em $[a, b]$.

2

NÚMEROS VERDADEIRAMENTE ALEATÓRIOS

Aleatoriedade é um termo comum atualmente, e parece ser, de senso comum, conhecido como falta de um padrão, ou como falta de um comportamento previsível. Se soubermos como foi formada uma sequência numérica, ou seja, se conhecermos sua lei de formação, ou se for possível determinar sua lei de formação, então essa sequência não será aleatória, mas sim determinística. Ou, na melhor das hipóteses, chamaremo-la de pseudoaleatória (veremos isso mais à frente). Porém se não houver certeza sobre qual será o próximo elemento da sequência, será dita aleatória. Exemplos já foram abordados no capítulo anterior. Números aleatórios, como também já mencionamos, são essenciais em algumas aplicações tais como criptografia, amostras estatísticas, simulações Monte Carlo, entre outras. Esta última é atualmente, exclusivamente computacional, ou seja, necessitamos de números aleatórios em grande quantidade e com muita rapidez. Além disso tais números precisam ser de fato aleatórios, ou concebidos de tal forma que se pareçam com números verdadeiramente aleatórios.

Isso nos remete a um fato sutil: se tais números precisam se parecer com números aleatórios, ou seja, se devem possuir alguma característica que os categoriza como aleatórios, então deve haver algo verdadeiramente aleatório para servir como uma base de comparação. De fato há o que chamamos de *Números Verdadeiramente Aleatórios*, ou a sigla em inglês TRN (*True Random Numbers*). Para obter tais números bastaria observar um experimento aleatório e anotar os resultados. O lançamento de uma moeda é o exemplo mais básico de aleatoriedade: atribuindo 1 a cara e 0 a coroa, quando lançamos e repetimos o experimento uma grande quantidade de vezes, obteremos uma sequência aleatória de zeros e uns. Este processo é de fato um *gerador de números verdadeiramente aleatórios*, que doravante designaremos TRNG (*True Random Number Generator*). De posse de uma sequência de zeros e uns que foram obtidos pelo lan-

çamento sucessivo de uma moeda, podemos obter uma sequência de outros números aleatórios da forma que desejarmos. Com a sequência em mãos, podemos dividi-la em subsequências menores, todas com o mesmo tamanho. Por exemplo, se temos uma sequência muito grande de zeros e uns podemos segmentar esta sequência de cinco em cinco dígitos. Com um pacote de cinco dígitos podemos formar um número de 5 *bits*, ou seja, um número de 0 a $2^5 - 1 = 31$. Assim, naturalmente, esta sequência (suficientemente grande) de zeros e uns nos fornece números aleatórios entre 0 e $2^k - 1$, onde k é um número natural arbitrário. Podemos também obter números aleatórios (inteiros) entre a e b, com $a < b$, mesmo que a e b não sejam da forma $2^k - 1$, do seguinte modo: fazemos $c = b - a$, e geramos números aleatórios entre 0 e c. Como c não é da forma $2^k - 1$, escolhemos α tal que 2^α é o menor inteiro da forma 2^k maior do que c. Então bastará dividir a sequência de zeros e uns em pedaços de tamanho α e converter essa cadeia, que representa um número do sistema binário, em um número de base decimal. Pode ocorrer desse número estar no intervalo $]c, 2^\alpha]$, e portanto fora do intervalo $[0, c]$. Neste caso, tal valor deve ser descartado e uma nova sequência deve ser considerada, até que a sequência forme um número que esteja no intervalo $[0, c]$. Somando a obteremos um número aleatório entre a e b. Devemos observar atentamente que a obtenção dos números aleatórios conforme foi descrito é realizada mediante um *algoritmo*.

Convenciona-se usar, entretanto, números aleatórios uniformemente distribuídos entre 0 e 1, conforme Gentle (2003, p. 5), ou Knuth (1998, p. 10), ou Ross (2010, p. 518). Representamos pela notação $U(0, 1)$, que tecnicamente é uma *variável aleatória contínua uniforme*, no intervalo entre 0 e 1. Embora não obrigatória, adotar essa representação permite que números sejam mais facilmente transformados em outras distribuições de probabilidade a serem usados em aplicações de forma geral. Não é difícil observar que no caso de números aleatórios uniformemente distribuídos, gerados no intervalo $[a, b]$, para obter uma sequência de números aleatórios no intervalo $[0, 1]$, basta subtrair a e dividir cada um deles por $b - a$.

Para alguma aplicação que requeira números aleatórios em grande quantidade, é evidente que lançar moedas ou jogar dados é inviável. Há, todavia, formas alternativas de se obter TRNs, em larga escala; tais fontes são das mais diversas naturezas: conversão de imagens astronômicas (espaciais), de ruídos elétricos de um semicondu-

tor, decaimento radioativo, etc.

Uma das fontes possíveis de TRNs pode ser encontrada em arquivos especiais em sistemas computacionais Unix, como o */dev/random* e */dev/urandom*. Tais arquivos armazenam resultados provenientes de atividades do sistema, como interações de usuários com o *mouse*, via teclado, coleta de uso de interfaces de rede, de disco, de memória. Em geral, especialistas informam que tal fonte de números possui baixa entropia, o que caracterizaria este gerador como de baixa qualidade, conforme Katzgraber (2010, p. 4).

Podemos reproduzir aqui uma amostra deste tipo de TRN gerados em um sistema operacional Unix (Solaris). O comando seguinte,

```
cat /dev/urandom | tr -dc '[0-9]'
```

produz resultados como:

```
085438354963116165543388880304397076196693696443917327747451258053752271767894616928179102445624904328246104132347244566882262501044083838618685754094086219166843289981860744847209073095096660569361364995795014475999998018661472064523762470292289641906076357930317516753991949029256022480355754072604357464911429863434478553553674886745736144548668884317972650497569804887344729329297519644836913570525066500573315554485541660952746626010621669999244314077364005045895598096897435427483884167921648659266816911437544659102217808742209710175627046666272224211542149162084469126664188622965994261114220279656675929698848574840228719112215208078224559646482128136329416262612974148841022052447377046731460161980621008915242232434483990205783361212106530700258618170715087228522613227796309079522343185284208022497764716092382145316525218103163007532802874465144480826279469415235219176914788352646667609724451611086532166804013432862872470623003968907046732851191222384170329546008120806399409508914075862730975185 81
```

A Web possui também alguns *sites* que oferecem TRNs, de graça, ou até como serviço pago. Podemos conferir um em RANDOM.ORG. Neste *site* há diversas opções grátis, como lançamento de uma moeda, de dados, dígitos aleatórios, embaralhador

de cartas, sorteios de loterias, etc. De acordo sua apresentação, os números são de fato verdadeiramente aleatórios e provenientes de observação de ruído atmosférico.

3

GERADORES DE NÚMEROS PSEUDOALEATÓRIOS

Neste capítulo vamos definir o que são números pseudoaleatórios, mostrar alguns métodos de geração de tais números, bem como fazer uma breve apresentação de fundamentos matemáticos que constituem estes geradores.

3.1 ALGUNS MÉTODOS

Já no início da era dos computadores imaginou-se se haveria alguma forma efetiva de obter números aleatórios sem ter que se recorrer a métodos mecânicos ou manuais. A resposta para essa questão é SIM e talvez a primeira proposta tenha sido o método criado pelo matemático John von Neumann, em 1946, conhecido como o método do *quadrado do meio*. Consiste em sacar os dígitos centrais de um número, elevá-lo ao quadrado e obter, assim, um novo número. Se, por exemplo, desejamos obter uma sequência de números aleatórios de quatro dígitos, tomamos um número inicial de quatro dígitos (que será conhecido como semente) e elevamo-lo ao quadrado. Desse valor calculado, tomamos os quatro dígitos centrais e obtemos assim um novo número aleatório. Um exemplo numérico: partindo da semente $X_0 = 4928$, temos $X_0^2 = 24285184$. Sacando os 4 dígitos centrais ficamos com $X_1 = 2851$. Este processo pode seguir indefinidamente, com a obtenção de mais números aleatórios.

Cabe aqui uma observação importante: sendo cada número determinado pelo valor anterior, como pode essa sequência ser aleatória? De acordo com Knuth (1998, p. 3), o próprio von Neumann concordava que seria um pecado admitir que os números obtidos assim fossem aleatórios. De fato a sequência não é aleatória, mas o que importa é

Teste de algumas sementes de 4 dígitos	
Semente	**Comportamento**
2008	Finalizou com o ciclo 6100, 2100, 4100, 8100
2009	Finalizou com o ciclo 9600, 1600, 5600, 3600
2010	Mesmo caso da semente 2008
2011	Degenerou para zero
2012	Mesmo caso da semente 2008
2013	Mesmo caso da semente 2008
2014	Mesmo caso da semente 2008
2015	Degenerou para zero

Tabela 2: Alguns resultados do método do quadrado do meio

que parece ser. Isso para certas aplicações é suficiente e a falsa aleatoriedade não será relevante, embora para outras não. Exatamente pelo fato de serem gerados através de algum método matemático ou algorítmico, tais números são ditos *pseudoaleatórios* e tais geradores serão aludidos aqui como PRNG (*Pseudo Random Numbers Generators*).

Houve nos anos 1950, segundo Knuth (1998, p. 4), diversos trabalhos sobre esse método, que acabaram por contestar sua aplicabilidade pura, embora outras pesquisas endossassem alguns casos especiais. Vamos abordar um exemplo: se quisermos gerar números de 4 dígitos, podemos iniciar com uma semente e observar os resultados; há casos onde a sequência degenera para zero e outros onde fecha um ciclo *(Ver Tabela 2)*.

Ainda segundo Knuth (1998, p. 4), N. Metropolis utilizando o método do quadrado do meio, conseguiu uma sequência de 750.000 números de 38 bits antes de ocorrer uma degeneração, além da sequência ter sido aprovada em testes estatísticos de aleatoriedade. Mas apesar disso o método ainda apresenta mais fraquezas do que virtudes. A título didático, Knuth (1998, p. 5), também apresentou um método próprio, que batizou de algoritmo K, mas sua própria conclusão foi a de que números aleatórios não podem ser gerados com métodos escolhidos aleatoriamente.

Convém introduzir uma definição: por *semente* devemos entender o valor inicial X_0, a partir do qual os demais números serão gerados. Além disso, consideremos

uma sequência $a_0, a_1, a_2, ...$ de números aleatórios. Sejam k e n naturais de forma que $0 \leq k < n$. O *período* da sequência é o menor valor $n - k$, quando $a_n = a_k$ e também quando $a_{n+j} = a_{k+j}$, para qualquer $j \geq 0$.

O método de von Neumann, bem como outros que serão abordados são, como convencionamos chamar, PRNGs. E mais, por não serem verdadeiramente aleatórios, sempre deverão ser submetidos a testes de aleatoriedade, ou testes de qualidade. Mais à frente esse assunto será abordado.

Uma importante observação que devemos mencionar: tanto o método do quadrado do meio quanto os demais métodos a serem apresentados aqui, são capazes de gerar sequências de inteiros pseudoaleatórios, mas para efeito prático, sempre se faz uma redução de tais números a uma sequência de variáveis aleatórias de distribuição uniforme no intervalo [0,1]. Desta forma e com uma transformação apropriada, a sequência poderá representar outras distribuições de probabilidade.

Os PRNGs possuem algumas vantagens em relação aos TRNGs, segundo Katzgraber (2010, p. 4), considerando aspectos computacionais:

- A geração é rápida;
- Os PRNGs não requerem equipamento especial;
- Se necessário, pode-se reproduzir a exata sequência de números aparentemente aleatórios. Por exemplo, para verificação de como um programa funciona ou para reproduzir um teste nas mesmas condições.

Por outro lado paga-se um preço:

- PRNGs possuem período finito e começam a se repetir a partir de um certo ponto. Por esta razão, é desejável que este período seja o mais longo possível;
- Pode haver alguma autocorrelação na sequência gerada, em particular quando se faz algum agrupamento. Isso é indesejável pois pode comprometer a aleatoriedade. Veremos esse fenômeno na Figura 7.

A característica principal de um PRNG é que uma sequência pode ser gerada de forma recorrente: cada x_n da sequência pode estar em função de seus antecessores:

$$x_n = f(x_{n-1}, x_{n-2}, ..., x_{n-k})$$

Uma recorrência precisa de valores iniciais, então no mínimo haverá um valor inicial (semente) caso ela dependa de apenas um antecessor, ou vários (bloco de sementes, ou *seed block*). Observamos que, no caso de um único antecessor, a primeira repetição de valor já fecha um ciclo periódico, mas no caso de vários antecessores, podem-se repetir valores dentro do período, sem fechar o ciclo. O objetivo será obter uma função f de modo que a sequência gerada seja longa e a mais aleatória possível.

Após o método de von Neumann surgiu outro método que apresenta resultados mais interessantes: D. H. Lehmer apresentou em 1949 o *método congruencial linear*. Este método tornou-se bastante popular por apresentar vantagens em relação ao método anterior e também por possuir propriedades matemáticas interessantes. O método consiste basicamente em escolher adequadamente quatro parâmetros, a saber:

- m, o módulo, sendo m um inteiro positivo.
- a, o multiplicador, com $0 \leq a < m$.
- c, o incremento, com $0 \leq c < m$.
- X_0, o valor inicial ou semente, com $0 \leq X_0 < m$.

Para definir a sequência, usamos uma operação aritmética chamada "módulo" (mod), que explicaremos na próxima seção. A sequência será obtida por uma recorrência linear assim

$$X_{n+1} := (aX_n + c) \bmod m, \text{ para } n \geq 0$$

Com esta recorrência teremos uma sequência de período finito de no máximo m. A sequência pode prosseguir sendo gerada infinitamente, mas o período será finito e a partir do fim do primeiro período, os elementos começarão a se repetir.

Vamos, antes de prosseguir na análise deste tipo de gerador, esclarecer alguns elementos de um ferramental matemático importante: congruências.

3.2 UM POUCO DE ARITMÉTICA

Uma vez que o método linear congruencial é baseado em aritmética modular (ou congruências), cabe agora escrevermos um pouco sobre alguns fundamentos de uma parte dessa disciplina matemática. Esse assunto também é conhecido como Teoria Elementar dos Números.

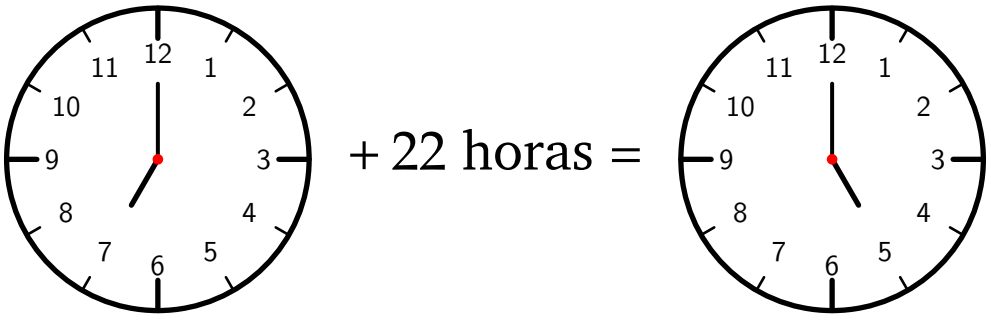
Dizemos que dois números inteiros a e b são congruentes módulo m, onde m é um número inteiro positivo, se os restos das divisões de a e b por m são iguais, ou seja, se m divide $a - b$. Em símbolos:

$$a \equiv b \bmod m \iff m|(a-b)$$

Em seguida veremos uma aplicação que ilustra uma forma lúdica do uso de congruências.

3.2.1 *Aritmética do Relógio*

A aritmética modular, ou congruência, pode ser facilmente entendida quando a aplicamos às contas relacionadas ao ponteiro de horas do relógio. Há 12 horas disponíveis para as quais um ponteiro pode apontar. Então, se quisermos somar (ou subtrair) uma certa quantidade de horas a uma dada posição inicial dos ponteiros, podemos fazê-lo utilizando o módulo 12. Desta forma obteremos a posição final do ponteiro das horas. Por exemplo, se a posição inicial do relógio marca 07:00, 22 horas além o relógio marcará $7 + 22 = 29 \equiv 5 \bmod 12$, ou seja, 05:00. Em palavras, 29 é congruente a 5 módulo 12. Segue uma ilustração (criada com código LaTeX):



Uma vez que congruências são operações com números inteiros, também podemos subtrair qualquer quantidade inteira de horas a uma certa posição inicial de horas do relógio. Por exemplo, se tivermos inicialmente o relógio indicando 07:00, 33 horas antes o relógio indicava 10 horas. Observemos: $7 - 33 = -26 \equiv 10 \bmod 12$. Aqui, para sair de -26 e chegar em 10, somamos a 26 múltiplos de 12 até obter um número positivo: $-26 + 3 \times 12 = 10$.

3.2.2 *Algumas Propriedades de Congruências*

Da definição decorre que congruência é uma *relação de equivalência*. Segundo Hefez (2011, p. 110), formalmente podemos descrever isto conforme a proposição abaixo:

Proposição 3.2.1. Seja $m > 0$ um inteiro positivo e sejam quaisquer a, b e $c \in \mathbb{N}$. Então:

(I) $a \equiv a \bmod m$ (propriedade reflexiva),

(II) Se $a \equiv b \bmod m$, então $b \equiv a \bmod m$ (propriedade simétrica),

(III) Se $a \equiv b \bmod m$ e $b \equiv c \bmod m$, então $a \equiv c \bmod m$ (propriedade transitiva).

Prova: A prova destas propriedades é bastante simples. Vamos a elas:

(I) Como $m|0$, então $m|(a - a)$, logo $a \equiv a \bmod m$.

(II) Se $a \equiv b \bmod m$ então $m|(a - b)$, o que implica em $a - b = k.m$, com $k \in \mathbb{Z}$. Desse modo, $b - a = (-k)m$, então $m|(b - a)$ e $b \equiv a \bmod m$.

(III) De fato, se $a \equiv b \bmod m$ e $b \equiv c \bmod m$, então existem inteiros p e q tais que

$$a - b = p.m \qquad \text{e} \qquad b - c = q.m$$

Então
$a - c = (a - b) + (b - c) = p.m + q.m = (p + q)m$, logo $a \equiv c \bmod m$.

□

Por ser a congruência uma relação de equivalência, há uma certa semelhança com a igualdade dos inteiros. Vejamos as proposições a seguir:

Proposição 3.2.2. Seja $m > 0$ um inteiro positivo e sejam quaisquer a, b, c e d inteiros. Se $a \equiv b \bmod m$ e $c \equiv d \bmod m$, então:

(I) $(a+c) \equiv (b+d) \bmod m$.

(II) $ac \equiv bd \bmod m$.

Prova: seguem as demonstrações (elementares):

(I) $m|(a-b)$ e $m|(c-d) \implies m|((a-b)+(c-d)) \implies m|((a+c)-(b+d)) \implies (a+c) \equiv (b+d) \bmod m$.

(II) $m|(a-b) \implies \exists\, q \in \mathbb{Z}$ tal que $a-b = mq \implies ad-bd = mqd \implies ad = mqd+bd$. Temos também $m|(c-d) \implies \exists\, p \in \mathbb{Z}$ tal que $c-d = mp \implies ac-ad = mpa$. Somando as duas igualdades obtidas, $ac-bd = m(pa+qd) \implies m|(ac-bd) \implies ac \equiv bc \bmod m$.

□

Proposição 3.2.3. Seja $m > 0$ um inteiro positivo e sejam quaisquer a e b inteiros. Então:

(I) Se $a \equiv b \bmod m$ e $n|m$, com $n > 0$, então $a \equiv b \bmod n$.

(II) Se $a \equiv b \bmod m$ e $c > 0$, então $ac \equiv bc \bmod mc$.

(III) Se $a \equiv b \bmod m$ e se um inteiro $d > 0$ divide a, b e m, então $\frac{a}{d} \equiv \frac{b}{d} \bmod \frac{m}{d}$.

Estas propriedades também tem provas bastante elementares e estão dadas a seguir:

(I) $n|m \implies m = n.q$, com algum $q \in \mathbb{Z}$. Além disso, temos $m|(a-b) \implies nq|(a-b) \implies a-b = n(pq) \implies a \equiv b \bmod n$.

(II) Se $a \equiv b \bmod m$ e $c > 0$ então $m|(a-b) \implies a-b = mq$, com algum $q \in \mathbb{Z}$. Multiplicando a última igualdade por c, ficamos com $c(a-b) = mqc \implies ac-bc = mc(q) \implies ac \equiv bc \bmod (mc)$.

(III) $a \equiv b \bmod m \implies m|(b-a) \implies b-a = m.q$, com $q \in \mathbb{Z}, q > 0$. Assim $\frac{b}{d} - \frac{a}{d} = q.\frac{m}{d} \implies \frac{a}{d} \equiv \frac{b}{d} \bmod \frac{m}{d}$.

□

Podemos notar que todos os possíveis restos da divisão de algum inteiro por m são os números $0, 1, 2, \ldots, m-1$, em qualquer ordem. Esse conjunto de números chamaremos de *sistema completo de resíduos (restos)* módulo m. Como se observa, tal conjunto possui m elementos.

Outra característica interessante que podemos notar é que um certo número inteiro pode ser associado a outros inteiros quando estes deixam o mesmo resto na divisão por m. Este conjunto de inteiros que deixam o mesmo resto na divisão por algum inteiro m, com $m > 0$ é chamado de *classe de congruência módulo m*.

Tomemos um exemplo: para $m = 5$, teremos 5 diferentes classes de congruências, a saber:

$$\{\ldots, -10, -5, 0, 5, 10, \ldots\}$$

$$\{\ldots, -9, -4, 1, 6, 11, \ldots\}$$

$$\{\ldots, -8, -3, 2, 7, 12, \ldots\}$$

$$\{\ldots, -7, -2, 3, 8, 13, \ldots\}$$

$$\{\ldots, -6, -1, 4, 9, 14, \ldots\}$$

Observando a primeira das classes listadas acima, podemos afirmar que tais números são congruentes entre si, ou seja, -10 é congruente a -5, que é congruente a 0 e assim para todos os demais. Embora exista uma semelhança entre congruência e igualdade nos inteiros, ressaltamos que a congruência está relacionada à operação "resto". Dois números inteiros são congruentes entre si se deixam o mesmo resto na divisão por um m inteiro dado. Apesar disso eles não são necessariamente iguais.

Gentle (2003, p. 8), observa que para um $m > 0$ dado, há exatamente m classes de congruências módulo m, sendo que a união de todas as classes forma o conjunto de todos os inteiros.

Se denotarmos por $\overline{a}$ a classe de congruência que deixa resto a na divisão por m, teremos um conjunto finito, que podemos denotar por $\mathbb{Z}/m\mathbb{Z}$, que será a reunião de todas as classes de congruência módulo m. Então escrevemos

$$\mathbb{Z}/m\mathbb{Z} = \{\overline{a},\ a \in \mathbb{Z}\}$$

Cada classe de congruência $\overline{a}$ corresponde a um resíduo a e consiste das translações de a por múltiplos de m, isto é, $\overline{a} = \{a + km,\ k \in \mathbb{Z}\}$.

Assim, por ser a congruência uma relação satisfazendo as Proposições 3.2.1 e 3.2.2, o conjunto $\mathbb{Z}/m\mathbb{Z}$, acrescido das operações de soma e multiplicação é chamado de *anel dos inteiros módulo m*. (Obs.: Anel é um conjunto dotado de operações de adição e multiplicação e possui certas propriedades. Uma referência para este assunto pode ser visto em Martinez, Moreira, Saldanha e Tengan (2010, p. 39)).

3.3 MÉTODO LINEAR CONGRUENCIAL

O método, como vimos, consiste em obter valores recursivamente através da relação

$$X_{n+1} := (aX_n + c) \bmod m, \ \ n \geq 0 \tag{3.1}$$

Essa sequência X_n, como vimos anteriormente, pode ser devidamente transformada em uma sequência com distribuição uniforme no intervalo $[0,1]$, representada como $U(0,1)$. Para isso fazemos a simples transformação:

$$U_n = \frac{X_n}{m}$$

Os valores de X_0, a, c e m devem ser escolhidos a priori e em seguida toda a sequência poderá ser obtida. Vejamos um exemplo: seja $X_0 = a = 7$, $c = 2$ e $m = 18$. A sequência obtida é:

$$7, 15, 17, 13, 3, 5, 1, 9, 11, 7, 15, 17 \ldots$$

Podemos observar que o período é igual a 9, isto é, obtivemos nove números entre 0 e 17. Quando um dos números que já foi gerado anteriormente, aparecer de novo, implica que o período chegou ao fim e um ciclo idêntico será gerado novamente. Vamos agora então observar os números do primeiro ciclo: pareceram embaralhados. É algo como selecionar ao acaso 9 números entre 0 e 17, em sequência, sem reposição. Porém, indesejavelmente apenas números ímpares foram gerados e isso não parece ser então tão aleatório.

Alterando o incremento de 2 para 1, obtivemos outra sequência:

$$7, 14, 9, 10, 17, 12, 13, 2, 15, 16, 5, 0, 1, 8, 3, 4, 11, 6, 7, 14, 9, \ldots$$

Agora, a situação foi diferente: o período foi o máximo, 18, logo pares e ímpares foram obtidos. Além disso, tal como no exemplo anterior, os números "pareceram"

aleatórios, tal como se fizéssemos um sorteio sem reposição dos números 0 a 17. Mas ainda houve um incoveniente: pares e ímpares se alternam e isso mais uma vez levanta suspeitas. A efetividade do método, ao que parece, dependerá de boas escolhas dos parâmetros. Faremos isso.

As sequências de números pseudoaleatórios gerados por este método, serão, como esperamos, regidas por uma escolha adequada dos parâmetros. O incremento c pode ser igual a zero, conforme a ideia original de Lehmer. Neste caso, $c = 0$, de acordo com Knuth (1998, p. 11), usa-se o nome *método linear congruencial multiplicativo*, enquanto que com $c \neq 0$ denota-se *método linear congruencial misto*. A ideia é a de que $c \neq 0$ permitiria obter períodos mais longos.

De início podemos rejeitar os casos $a = 1$ e $a = 0$. Os resultados serão degenerados. Por exemplo, vamos analisar o caso $X_0 = 7, a = 1$, $c = 2$ e $m = 18$. A sequência obtida consiste de uma progressão aritmética com módulo:

$$7, 9, 11, 13, 15, 17, 1, 3, 5 \ldots$$

Não parece aleatório, portanto só vale a pena analisar os casos em que $a \geq 2$.

Vamos agora retrabalhar e generalizar a equação (3.1). Antes de mais nada, convém ressaltar que uma equação de recorrência como esta pode ser resolvida e que sua solução pode ser entendida como calcular qualquer termo da recorrência, não em função do seu termo anterior (ou mais do que um termo), mas sim em função apenas dos parâmetros iniciais, encontrando uma fórmula. E mais, por estarmos trabalhando com um conjunto finito (inteiros positivos menores que m), vamos sempre considerar que os resultados a cada operação serão reduzidos ao resto módulo m.

O caso $c = 0$ é um caso muito simples. Sabemos que $X_n \equiv aX_{n-1} \bmod m$, então $X_1 \equiv aX_0 \bmod m$. Deduzimos X_2 em função de X_0 da seguinte forma: se $X_2 \equiv aX_1 \bmod m$, então $X_2 \equiv a(aX_0) \bmod m$, ou seja, $X_2 \equiv a^2X_0 \bmod m$. Analogamente. e de forma recursiva, concluímos que

$$X_n \equiv a^n X_0 \bmod m$$

É uma progressão geométrica com módulo.

Quando $c \neq 0$, temos um trabalho adicional: partindo da recorrência $X_n \equiv aX_{n-1} + c \bmod m$, teremos para:

- $n = 1$: $X_1 \equiv aX_0 + c \bmod m$
- $n = 2$: $X_2 \equiv aX_1 + c \equiv a(aX_0 + c) + c = a^2X_0 + ac + c \bmod m$
- $n = 3$: $X_3 \equiv aX_2 + c \equiv a^3X_0 + a^2c + ac + c \bmod m$

$$\vdots$$

- $n = n$: $X_n \equiv a^nX_0 + (a^{n-1} + a^{n-2} + \ldots + a + 1)c \bmod m$

Da Álgebra, sabemos que $(a - 1)$ divide $(a^n - 1)$, e isto equivale a:

$$\frac{a^n - 1}{a - 1} = a^{n-1} + a^{n-2} + \ldots + a + 1$$

Desta forma podemos reescrever a equação da seguinte forma:

$$X_n \equiv a^nX_0 + \frac{(a^n - 1)c}{a - 1} \bmod m.$$

Há mais propriedades matemáticas interessantes na geração de uma sequência linear congruencial. Antes de estudar uma dessas propriedades vamos antes analisar duas proposições:

Proposição 3.3.1. Dados a, b e c inteiros positivos, a equação $ax + by = c$ possui soluções inteiras $x, y \in \mathbb{Z}$ se e somente se, $\text{mdc}(a, b)|c$.

Prova: Vamos supor que (x_0, y_0) é solução. Assim, $ax_0 + by_0 = c$. Temos que $\text{mdc}(a, b)|a$ e $\text{mdc}(a, b)|b$, então

$$\text{mdc}(a, b)|(ax_0 + by_0) = c.$$

Por outro lado, se $\text{mdc}(a, b)|c$, então existe $d \in \mathbb{Z}$ tal que $c = d.\text{mdc}(a, b)$. Sabemos que, das propriedades de divisão euclidiana e do máximo divisor comum, existem $m, n \in \mathbb{Z}$ tais que $am + bn = \text{mdc}(a, b)$. Essa identidade também é conhecida como Relação de Bézout. Assim, $c = d.(am + bn) \Rightarrow c = a(dm) + b(dn)$, portanto

$$c = ax_0 + by_0.$$

□

Proposição 3.3.2. Dados a, b e $m \in \mathbb{Z}$, com $m > 0$, a equação $ax \equiv b \bmod m$ possui solução única módulo m, $x \in \mathbb{Z}$, se $\text{mdc}(a, m) = 1$.

Prova: Se $\text{mdc}(a, m) = 1$, então existem $x_0, y_0 \in \mathbb{Z}$ tais que $ax_0 + my_0 = 1$ (Relação de Bézout). Multiplicando a equação por b, teremos $ax_0b + my_0b = b$. Fazendo $x = x_0b$ e $y = y_0b$, ficaremos com $ax + my = b$, portanto $ax \equiv b \bmod m$.

Para provar a unicidade, vamos supor que existem x e y de forma que $ax \equiv b \bmod m$ e $ay \equiv b \bmod m$. Subtraindo uma equação da outra obteremos $a(x - y) \equiv b - b = 0 \bmod m$. Assim, $m|a(x - y)$, e como $mdc(a, m) = 1$, temos que $m|(x - y)$, portanto $x \equiv y \bmod m$. Com isto concluímos a unicidade.

□

Agora temos condição de avaliar a seguinte proposição (Knuth (1998, p. 11)):

Proposição 3.3.3. Em uma sequência linear congruencial $X_{n+1} := (aX_n + c) \bmod m$, se a e m são primos entre si, então X_0 sempre se repetirá.

Prova: Seja a sequência linear congruencial

$$X_0, X_1, X_2, \ldots, X_{k-1}, X_k, \ldots, X_{r-1}, X_r, \ldots$$

gerada por $X_{n+1} := (aX_n + c) \bmod m$, com $\text{mdc}(a, m) = 1$. Consideremos X_r como o primeiro valor repetido desta sequência. Se X_r for igual a um X_k para algum k tal que $0 < k < r$, necessariamente $X_{r-1} \neq X_{k-1}$. Com isso teríamos X_r determinado por X_{k-1} e X_{r-1}, o que é absurdo pois X_n é unicamente determinado por X_{n-1} quando a e m são primos entre si, conforme Proposição 3.3.2. Logo, $k = 0$.

□

O caso de um gerador congruencial linear multiplicativo puro, ou seja, quando $c = 0$, é digno de menção. Neste caso em particular, o período teórico máximo m não pode ser alcançado e há uma razão para isso. Vamos observar a forma deste gerador:

$$X_{n+1} := aX_n \bmod m$$

Caso $X_n = 0$ a sequência degenerará imediatamente para zero. E mais, se houver algum d divisor de m e se X_n é múltiplo de d, de acordo com a Proposição 3.2.3, todos

os sucessores $X_{n+1}, X_{n+2}, \ldots$ dessa sequência serão múltiplos de d. Desta forma queremos que todo X_n seja relativamente primo com m. A quantidade de inteiros positivos que são relativamente primos com m é dado por uma função especial chamada *fi de Euler* e denotada por $\varphi(m)$. Veremos isso um pouco adiante.

Observando o termo geral da sequência multiplicativa:

$$X_n \equiv a^n X_0 \bmod m, \tag{3.2}$$

temos que o período vai depender de a e de m. Vamos agora considerar um k inteiro positivo, o menor possível tal que

$$a^k \equiv 1 \bmod m$$

Quando isto ocorrer, é fácil observar que a sequência começará se repetir, pois teremos

$$X_k \equiv a^k X_0 \equiv 1.X_0 \bmod m \Rightarrow X_k \equiv X_0 \bmod m.$$

Então o período não poderá ser maior do que k. Esse valor de k que procuramos é resultado de um teorema fundamental da teoria elementar dos números: o Teorema de Euler–Fermat.

Na seção 3.2 conhecemos as classes de congruência módulo m. Uma classe $\overline{a}$ é um conjunto de inteiros positivos que deixam resto a quando divididos por m. A notação é:

$$\overline{a} = \{x \in \mathbb{Z},\ x \equiv a \bmod m\}$$

O conjunto que reúne todas as classes de congruência módulo m pode ser denotada por $\mathbb{Z}/m\mathbb{Z}$. Assim:

$$\mathbb{Z}/m\mathbb{Z} = \{\overline{a}\ ,\ a \in \mathbb{Z}\} = \{\overline{0}, \overline{1}, \ldots, \overline{m-1}\}$$

Isto significa que $\mathbb{Z}/m\mathbb{Z}$ é a reunião de todas as classes de congruência módulo m. Como só existem m classes de congruências, o conjunto $\mathbb{Z}/m\mathbb{Z}$ é finito.

Há um subconjunto notável de $\mathbb{Z}/m\mathbb{Z}$, denotado por $(\mathbb{Z}/m\mathbb{Z})^*$ e definido como sendo todas as classes cujos elementos são primos com m. Assim:

$$(\mathbb{Z}/m\mathbb{Z})^* = \{\overline{a}\ , \text{mdc}(a, m) = 1\}$$

Agora podemos definir a função *fi de Euler* como sendo a quantidade de elementos de $(\mathbb{Z}/m\mathbb{Z})^*$:

$$\varphi(m) := |(\mathbb{Z}/m\mathbb{Z})^*|$$

(Obs.: Estamos adotando o símbolo "$|\ \ |$" para designar a quantidade de elementos do conjunto).

Em outras palavras, $\varphi(m)$ conta quantos inteiros positivos entre 1 e m são primos com m. Por exemplo:

$\varphi(4) = |\{1,3\}| = 2$
$\varphi(8) = |\{1,3,5,7\}| = 4$
$\varphi(12) = 4$, etc.

Após a definição da função *fi de Euler*, podemos enunciar o Teorema de Euler-Fermat.

Teorema 3.3.1. (Euler–Fermat) Sejam $m, a \in \mathbb{N}$ com $m > 1$ e a e m primos entre si. Então,

$$a^{\varphi(m)} \equiv 1 \bmod m.$$

A demonstração deste teorema vai além do propósito deste trabalho, mas pode ser encontrado nas seguintes referências: Hefez (2011, p. 132), ou Martinez, Moreira, Saldanha e Tengan (2010, p. 47).

Observando a equação (3.4), quando tivermos o menor k tal que $a^k \equiv 1 \bmod m$, é porque k é o período. Se o menor valor de k que satisfaz $a^k \equiv 1 \bmod m$ for igual a $\varphi(m)$, diremos que a é *raiz primitiva* de m, ou raiz primitiva módulo m.

Não é difícil observar que quando m é primo, teremos $\varphi(m) = m - 1$. Então podemos obter maiores períodos quando m é primo e quando a é raiz primitiva de m. Vamos fazer um exemplo.

Sejam $m = 31$ e $a = 9$. Então vamos calcular os números aleatórios gerados por

$$X_n := 9X_{n-1} \bmod 31$$

escolhendo, por exemplo, a semente $X_0 = 17$. O resultado segue:

$$17, 29, 13, 24, 30, 22, 12, 15, 11, 6, 23, 21, 3, 27, 26, 17 \ldots$$

O período foi 15. Então 15 é o menor inteiro positivo tal que $9^{15} \equiv 1 \bmod 31$. Mas $\varphi(31) = 30$, logo 9 não é raiz primitiva de 31.

Então vamos mudar o multiplicador, vamos considerar $a = 24$ e calcular novamente a sequência. Assim, obteremos:

$$17, 5, 27, 28, 21, 8, 6, 20, 15, 19, 22, 1, 24, 18, 29,$$

$$14, 26, 4, 3, 10, 23, 25, 11, 16, 12, 9, 30, 7, 13, 2, 17, \ldots$$

Agora conseguimos o período 30. E há uma razão para isso: 24 é raiz primitiva de 31 (ou raiz primitiva módulo 31). Significa que o menor k inteiro positivo tal que $24^k \equiv 1 \bmod 31$ é $k = 30$.

Obter um período longo não é a única característica desejada em um gerador. Ele deve apresentar números que pareçam ser aleatórios. Isso implica que deve passar por testes, sejam eles visuais, gráficos ou estatísticos. Vamos agora entender porque esse exemplo acima não seria exatamente uma boa sequência. Um critério visual, apenas observando os números, não desperta desconfiança. Entretanto, agrupando-os em pares com sobreposição (*overlapping)*, (5,27), (27,28), (28,21),...(2,17), (17,5), e em seguida traçando um gráfico de pontos pareados, obteremos a imagem da Figura 5, onde há uma aparente correlação.

Os geradores congruenciais lineares, segundo Katzgraber (2010, p. 6), são reconhecidamente problemáticos por possuir algum tipo de correlação e não são recomendados para simulações numéricas, apesar da rapidez de sua geração em computador. Podem ser usados, no entanto, como geradores de sementes para outros tipos de geradores mais complexos.

Notadamente um gerador congruencial linear que foi muito usado e que possui uma característica indesejável, será analisado em seguida.

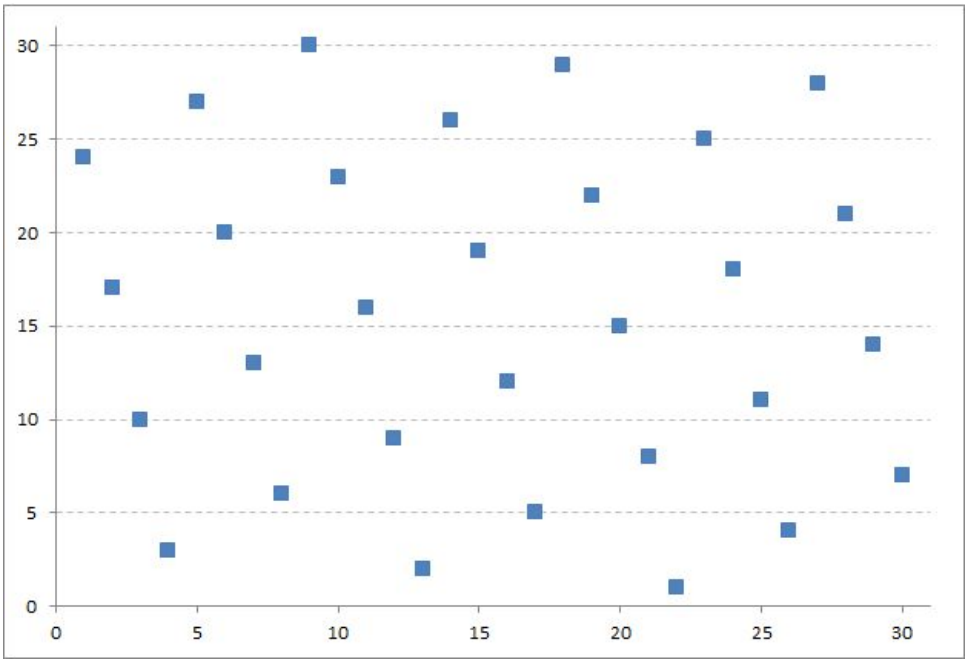


Figura 5: *Pares com sobreposição. Gráfico construído em Microsoft Excel.*

3.3.1 *Gerador RANDU*

RANDU foi um gerador congruencial linear multiplicativo programado em *mainframes* IBM nas décadas de 1960 e 1970, com a finalidade de gerar PRNs com rapidez. Isso foi possível por usar o multiplicador $a = 65539$, incremento $c = 0$ e módulo $m = 2^{31}$, proporcionando mais velocidade na obtenção dos PRNs em máquinas de 32-bits. Entretanto existe uma correlação entre os números obtidos, conforme podemos verificar com a seguinte manipulação: Se $X_n \equiv 65539X_{n-1} \bmod 2^{31}$, então $X_n \equiv 65539^2 X_{n-2} \bmod 2^{31}$. Então,

$$X_n \equiv (65536+3)^2 X_{n-2} \bmod 2^{31}$$

$$X_n \equiv (2^{16}+3)^2 X_{n-2} \bmod 2^{31}$$

$$X_n \equiv 2^{32}X_{n-2} + 6.2^{16}X_{n-2} + 9X_{n-2} \bmod 2^{31}$$

Como $2^{32} \equiv 0 \bmod 2^{31}$, temos $X_n \equiv 6.2^{16}X_{n-2} + 9X_{n-2} \bmod 2^{31}$. Assim,

$$X_n \equiv 6(65539-3)X_{n-2} + 9X_{n-2} \bmod 2^{31}$$

Como $X_{n-1} \equiv 65539X_{n-2} \bmod 2^{31}$, teremos

$$X_n \equiv 6X_{n-1} - 9X_{n-2} \bmod 2^{31}$$

Se a sequência for gerada com $3n$ valores, podemos agrupá-la em triplas (v_i, v_{i+1}, v_{i+2}) e desta forma obteremos uma matriz $n \times 3$. Estas triplas, devidamente apresentadas em um gráfico, mostrarão que os pontos situam-se em 15 planos de $\mathbb{R}^3$. Em geral os pontos parecerão aleatoriamente situados no espaço, mas se rotacionado adequadamente, o gráfico revelará os pontos sobre os planos (Figuras 6 e 7).

Observação: as Figuras 6 e 7 foram geradas pelo *software* R, com o seguinte código:
library(rgl)
with(randu, plot3d(x, y, z, axes = TRUE, xlab = , ylab = , zlab =))
rgl.viewpoint(theta = 0, phi = 0, fov = 0, zoom = 0.7)

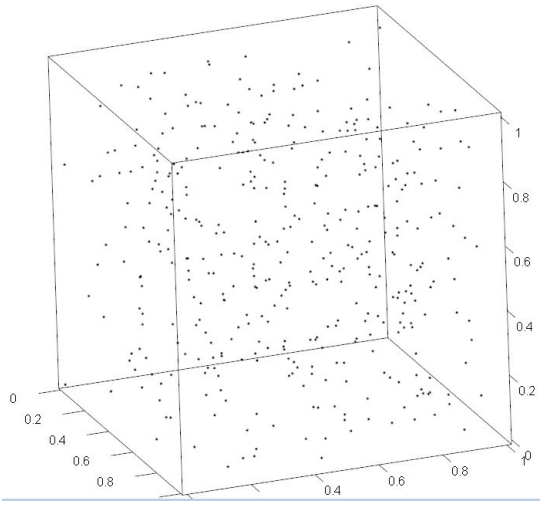


Figura 6: *Pontos gerados pelo RANDU, agrupados em triplas. Parecem aleatoriamente distribuídos no espaço.*

Desde sua introdução por Lehmer, as propriedades dos geradores lineares congruenciais multiplicativos tem sido largamente estudados. Há o fato apontado por Marsaglia (1968, p. 25), que demonstra que, como dissemos anteriormente, há um defeito na sequência de números aleatórios gerados. Esse defeito não pode ser removido por escolhas arbitrárias de módulos, multiplicadores ou semente. Se *n-uplas* $(v_1, v_2, ..., v_n)$ de valores da sequência forem vistos como pontos em um cubo *n*–dimensional, então todos os pontos estarão localizados em um número relativamente pequeno de hiperplanos. É o caso do RANDU.

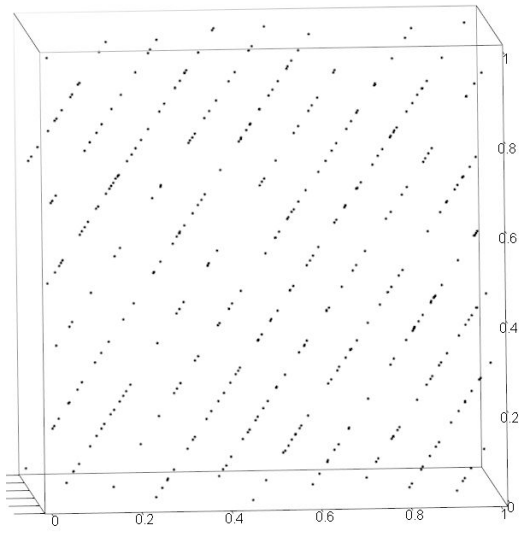


Figura 7: *Pontos gerados pelo RANDU. Quando visualizados sob certo ângulo, revelam que os pontos situam-se sobre 15 planos.*

3.3.2 *Outros Geradores*

Há inúmeros exemplos de PRNGs, com base nos lineares congruenciais, que podem apresentar resultados semelhantes, melhores ou até piores. Knuth (1998, p. 26), aponta que, do contrário do que se imagina, um incremento na expressão da sequência poderia gerar menos aleatoriedade. Por exemplo, uma expressão do tipo

$$X_{n+1} := ((aX_n) \bmod (m+1) + c) \bmod m$$

será menos aleatória em comparação a $X_n := (aX_n + c) \bmod m$.

Também é possível pensar em sequências em que X_n é determinado por mais do que um antecessor. O exemplo mais simples deste caso é a sequência de Fibonacci, já adaptada ao módulo m,

$$X_{n+1} := X_n + X_{n-1} \bmod m$$

Knuth (1998, p. 26), classificou-a como de aleatoriedade insatisfatória. Porém podemos usá-la de forma mais genérica:

$$X_n := (X_{n-j} \odot X_{n-k}) \bmod m, \qquad 0 < j < k$$

Este gerador é conhecido como *Lagged Fibonacci*, ou seja, uma sequência semelhante à sequência de Fibonacci, mas onde cada termo X_n da sequência depende de valores deslocados de $n - j$ e $n - k$. Decorre da equação desta sequência que são necessários k valores para semente. Isto é chamado de bloco de sementes ou *seed block*, de tamanho k. Segundo Katzgraber (2010, p. 7), comumente se usa $m = 2^M$, com $M = 32$ ou 64. O símbolo $\odot$ representa um dos operadores: adição, multiplicação ou OU exclusivo (XOR).

De acordo com Gentle (2003, p. 33), o *Lagged Fibonacci* é um bom gerador desde que seja feita uma boa escolha dos parâmetros j, k e m. Se m for primo, o período pode ser maior do que $m^k - 1$, mas em geral, como dito antes, m é usualmente potência de 2 e o período máximo possível para $m = 2^M$ é $2^{k-1}2^{M-1}$.

Cabe aqui uma explicação sobre o operador binário XOR, para o qual adotaremos o símbolo $\oplus$. Este é um operador lógico entre dois valores (operandos), que resulta 1 (verdadeiro) quando os operandos possuem valores lógicos distintos, ou 0 (falso) quando os operandos possuem o mesmo valor lógico. Em lógica elementar, pode ser também designado por disjunção exclusiva, quando aplicado a duas proposições, p e q. A operação só será verdadeira quando p é verdadeira ou q é verdadeira, mas não ambas. No cotidiano é comum nos depararmos com situações do tipo: "para sobremesa, você pode escolher fruta ou doce", quando se quer na realidade dizer que podemos escolher apenas uma delas, não ambas.

A operação $\oplus$ pode ser apresentada como uma tabela–verdade, conforme podemos verificar na Tabela 3. Para operar dois números inteiros positivos, devemos reescrevê-los na notação "base 2" e em seguida operar *bit* a *bit*, conforme a Tabela 3 (Aqui, um *bit* é um símbolo 0 ou 1). Por exemplo, para calcular $5 \oplus 9$ devemos escrever os operandos na base 2 e aplicar $\oplus$ a cada par de *bits* correspondentes (primeiro com primeiro, segundo com segundo, etc.): $(0101)_2 \oplus (1001)_2 = (1100)_2$, sendo que $(1100)_2$ equivale a 12 em numeração decimal. Assim, $5 \oplus 9 = 12$. Para conhecer mais sobre bases e sistemas de numeração, podemos consultar Hefez (2010, capítulo 4).

ou exclusivo (XOR)		
p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 3: Tabela–verdade $p \oplus q$.

Especificamente o operador XOR é tratado num contexto de outro tipo de gerador conhecido como *generalised feedback shift register* (registradores de deslocamento linear). Nesta categoria de geradores encontram-se atualmente os PRNGs mais eficientes, como o WELL e o *Mersenne Twister*.

Este último foi desenvolvido por Makoto Matsumoto e Takuji Nishimura, em 1997, mas ainda permanece sendo relançado, como o *TinyMT*, lançado em 2015. A origem do nome Mersenne Twister é devido à utilização de um primo especial, conhecido como um dos primos de Mersenne, $2^{19937} - 1$ e "twister" associado a um tornado, algo embaralhado, retorcido. Os criadores também acharam curioso o nome por conter as iniciais de ambos (MT). O Mersenne Twister hoje está implementado em diversos softwares matemáticos, estatísticos ou de uso geral, tais como Matlab, Python, R, etc, através do código mt19937(), exatamente por possuir as melhores características que um PRNG poderia conter: um período gigantesco de $2^{19937} - 1$, rápido, usa pouca memória, distribuído em código fonte em linguagem C padrão. Os autores advertem que o algoritmo é para uso em simulações Monte Carlo e não deve ser usado para criptografia. Os criadores possuem uma página na internet onde se podem obter muitas informações e o código-fonte. Ver o endereço em MATSUMOTO; NISHIMURA na bibliografia.

Os PRNGs, para que sejam usados com alguma finalidade, devem ter algumas boas características que foram comentadas até aqui. Vamos então analisar alguns testes e validações para os PRNGs.

4

TESTES E TRANSFORMAÇÕES

Ao longo deste texto falamos em boas propriedades de números aleatórios, ou seja, características que os tornariam próximos de números verdadeiramente aleatórios ou que ao menos se pareçam como tais. Naturalmente, podemos aplicar inicialmente testes estatísticos com o objetivo de avaliar se uma coleção de números pode ser comparada a uma variável aleatória de distribuição uniforme no intervalo $[0, 1]$. Genericamente esse tipo de teste é denominado como Teste de Hipótese. Mais especificamente, é um Teste de Aderência.

4.1 TESTE DE HIPÓTESE. TESTE DE ADERÊNCIA.

A estrutura do teste é formular duas hipóteses, que chamaremos H_0 (hipótese nula) e H_1 (hipótese alternativa) da seguinte forma:

> H_0: Os números gerados representam uma variável aleatória com distribuição uniforme no intervalo $[0, 1]$.
>
> H_1: Os números gerados não representam uma variável aleatória com distribuição uniforme no intervalo $[0, 1]$.

A hipótese H_0, caso não seja rejeitada, implica em aceitar que devem-se ao acaso as discrepâncias entre o conjunto de números aleatórios e a variável aleatória de distribuição uniforme. Após aplicar o teste, devemos exclusivamente rejeitar ou não rejeitar a hipótese H_0.

Importante também é observarmos que esse teste não considera a ordem dos números da sequência. Portanto, está longe ainda de avaliar com segurança se a sequência

possui propriedades que a caracterizariam como uma boa sequência de números aleatórios. Esse tipo de teste (de aderência) apenas assegura, com uma pequena margem de erro, que o conjunto de valores possui distribuição uniforme naquele intervalo.

O primeiro teste de aderência que vai rejeitar ou não a hipótese H_0 é o teste de *chi-quadrado* (pronunciamos qui quadrado). Esse teste consiste em classificar valores observados $x_1, x_2, \ldots, x_n$ em k classes que sejam mutuamente exclusivas. Sejam $f_1, f_2, \ldots, f_k$ as quantidades de elementos de cada classe e sejam $p_1, p_2, \ldots, p_k$ as quantidades esperadas em cada classe de acordo com a distribuição uniforme $U(0,1)$. (Aqui vamos adotar $U(0,1)$ a representação de uma distribuição uniforme nesse intervalo). É fácil observar aqui que

$$\sum_{i=1}^{k} f_i = \sum_{i=1}^{k} p_i = n$$

É possível mostrar que se os vetores $f = (f_1, f_2, \ldots, f_k)$ e $p = (p_1, p_2, \ldots, p_k)$ são provenientes de uma mesma população (neste caso de uma variável aleatória $U(0,1)$, então a estatística seguinte:

$$\chi^2 = \sum_{i=1}^{k} \frac{(f_i - p_i)^2}{p_i} \tag{4.1}$$

é proveniente de uma variável aleatória X com distribuição chi-quadrado com $k-1$ graus de liberdade. Vale lembrar que o significado de graus de liberdade (ver Bussab e Morettin (2006, p. 392)) é que se temos k compartimentos para armazenar n valores, podemos fazer a distribuição livremente em $k-1$ compartimentos, porque o último compartimento obrigatoriamente deverá conter a diferença entre n e o que já foi distribuído.

Dado um nível de significância α, o teste consistirá em rejeitar H_0 sempre que o χ^2 for maior do que um valor teórico previamente tabelado, conhecido como valor crítico. Este valor denotamos por χ^2_{crit}. Se o valor crítico for menor que o calculado, a hipótese H_0 deve ser rejeitada, ou seja, implicará que os números gerados aleatoriamente não estão distribuídos segundo uma variável aleatória $U(0,1)$. O valor crítico de χ^2 pode ser encontrado em uma tabela ou pode ser calculado dinamicamente por algum pacote

estatístico computacional.

Mas há, na verdade, a possibilidade de cometermos um erro na decisão após a aplicação do teste. Se rejeitamos H_0 quando ela for verdadeira, cometeremos um erro chamado de erro tipo I. Na prática, se adotarmos $\alpha = 5\%$, implica que temos 5% de probabilidade de cometermos o erro tipo I. Nos testes de hipótese como este, sempre devemos ter algum nível de significância. Mais detalhes podem ser vistos em extensa bibliografia relacionada à Estatística, como em Magalhães, Lima (2000, p. 244).

No capítulo sobre TRN's apresentamos uma sequência de números aleatórios provenientes de uma coleta bruta do dispositivo */dev/urandom* (ver página 25), que podemos supor, após uma reorganização, como sendo pertencentes a uma variável aleatória de distribuição $U(0,1)$, portanto passível de ser testada pelo teste de chi-quadrado. A reorganização consistirá em particionar a sequência da seguinte forma: tomamos os 4 primeiros algarismos, que representam um número que pode variar de 0 a 9999 e dividimo-lo por 10^4, assim obtendo um número entre 0 e 1. Repetimos o procedimento para os 4 próximos e assim por diante até o fim, de forma a se obter uma nova sequência de números aleatórios. Após a reorganização podemos montar uma tabela com 6 classes e suas respectivas frequências observadas e esperadas (*Tabela 4*).

Intervalos	$0 \vdash \frac{1}{6}$	$\frac{1}{6} \vdash \frac{2}{6}$	$\frac{2}{6} \vdash \frac{3}{6}$	$\frac{3}{6} \vdash \frac{4}{6}$	$\frac{4}{6} \vdash \frac{5}{6}$	$\frac{5}{6} \vdash 1$
frequência observada	46	48	44	49	31	35
frequência esperada	42.17	42.17	42.17	42.17	42.17	42.17

Tabela 4: Dados do */dev/urandom*. Total de 253 números.

Usando (4.1), chegamos a $\chi^2 = 6,5173$. Temos aqui 5 graus de liberdade e se escolhermos $\alpha = 0,05$, o valor crítico é ${\chi^2}_{crit} = 11,07$. Esse valor pode ser obtido através de uma tabela de valores da estatística chi-quadrado ou pode mesmo ser calculado dinamicamente através de um *software* estatístico. Nessas condições, como ${\chi^2}_{crit} > \chi^2$, não rejeitamos H_0, logo a distribuição dos dados do */dev/urandom* representa uma variável $U(0,1)$.

Embora ainda exista outros testes de aderência, como o de Kolmogorov–Smirnoff (ver Bussab e Morettin (2006, p. 404)), eles não são suficientes para assegurar a desejada qualidade de uma sequência de números aleatórios. Portanto outros tipos de testes devem ser aplicados.

4.2 OUTROS TESTES

O teste chi-quadrado não considerou a ordem em que os números são obtidos. Ao contrário, apenas a frequência de classes (partições) foi considerada, a fim de avaliar se os números estavam uniformemente distribuídos. Porém a ordem é absolutamente essencial na análise da qualidade da sequência. Há uma boa quantidade de testes que podem ser aplicados para essa verificação. Um teste bastante conhecido é o "run test".

4.2.1 *"Run Test"*

Este teste é usado para decidir se os números de uma sequência são originados por algum processo aleatório. *Run test* é também conhecido como Teste de Sequências, segundo Morettin e Toloi (2004, p. 61).

Uma sequência, neste contexto, é definida como uma série de crescimento ou de decrescimento de valores. O número de crescimentos e de decrescimentos é o tamanho da sequência. Em uma sequência de números aleatórios X_n, a probabilidade de X_{i+1} ser maior ou menor do que X_i segue uma distribuição binomial, que forma a base do teste.

Em primeiro lugar, em um teste de sequências, devemos contar o número de elementos da sequência. Não há uma maneira única de contar, mas de maneira geral temos que transformar a sequência em uma cadeia com dois símbolos (binária). Por exemplo, o lançamento de um dado pode produzir um número par (P) ou ímpar (I). Desta forma, 20 lançamentos de um dado pode produzir uma sequência aleatória como esta:

$$PIIPPPPPIPIPIPPPPIIP$$

Nesta série o número de sequências é 11, pois há 11 agrupamentos contíguos de P's ou I's.

Numa sequência de números aleatórios podemos denotar valores acima da mediana como positivos e abaixo da mediana como negativos. (Mediana é um número m tal que metade dos valores são menores do que m, e metade maiores do que m). Uma sequência será então uma série de valores consecutivamente positivos (negativos). O teste pode ser definido como um teste de hipótese com:

H_0: A sequência foi gerada de forma aleatória.

H_1: A sequência não foi gerada de forma aleatória.

A estatística calculada nesse teste é:

$$Z = \frac{R - \overline{R}}{S_R} \tag{4.2}$$

onde R é o número observado de sequências, $\overline{R}$ é o número esperado de sequências e S_R é o desvio padrão do número de sequências. Calculamos os valores de $\overline{R}$ e S_R da seguinte forma:

$$\overline{R} = \frac{2n_1n_2}{n_1 + n_2} + 1 \tag{4.3}$$

$$S_R = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1 + n_2)^2(n_1 + n_2 - 1)}} \tag{4.4}$$

sendo que n_1 e n_2 denotam o número de valores positivos e negativos da série, respectivamente. Logo $n_1 + n_2$ representa o total de elementos e α é o nível de significância do teste. A referência da utilização destas fórmulas pode ser obtida em NIST/SEMATECH.

Para uma amostra grande, com $n_1 > 20$ e $n_2 > 20$, podemos aproximar a estatística a uma distribuição normal padrão. Rejeitamos a hipótese nula H_0 se

$$|Z| > Z_{1-\frac{\alpha}{2}}$$

onde $Z_{1-\frac{\alpha}{2}}$ é o valor da variável Z (normal padrão), chamado de valor crítico, situado no limite onde a probabilidade é de $1 - \frac{\alpha}{2}$. Para mais informações sobre distribuição

normal, podemos consultar Magalhães e Lima (2000, p. 181). Então, com $\alpha = 5\%$, $Z_{1-\frac{\alpha}{2}} = 1,96$, logo, a estatística com valor absoluto maior do que $1,96$ indicará não-aleatoriedade. Também mais informações sobre *"run test"* podemos ver em Knuth (1998, p. 66-69) e em Morettin e Toloi (2006, p. 61).

Podemos aplicar o teste na amostra do */dev/urandom*, mas antes precisamos agrupar a lista de números. Adotando números de 4 dígitos, obteremos: 0854, 3835, 4963, etc., até o final da lista. Assim, conseguimos ao todo 253 números. Para obter o número de sequências, contamos a quantidade de crescimentos e decrescimentos. O teste gerou os seguintes resultados:

Mediana: 4716
Números de sequências: $R = 135$
$n_1 = 127$
$n_2 = 126$
$\overline{R} = 127,5$
$S_R = 7,94$
$Z = 0,94$

Como $0,94 < 1,96$, não rejeitamos H_0, ou seja, a sequência foi gerada de forma aleatória.

4.2.2 *Mais Testes*

Como comentamos anteriormente, há uma boa quantidade de testes para avaliar a qualidade de sequências de números aleatórios. Fizemos uma avaliação visual, podemos fazer uma avaliação gráfica simples, podemos observar um gráfico da sequência pareada em *overlapping*, um gráfico tridimensional com os dados agrupados em triplas (caso do RANDU) e testes estatísticos. Podemos somar a estes testes, outros como o de simples correlação e de momentos (estes podem ser vistos em Katzgraber (2010, p. 9-10)).

Há ainda, em última instância, testes agrupados em *suítes*, como o DIEHARD, de G. Marsaglia. Mais informações sobre o DIEHARD podem ser obtidas em MARSAGLIA.

4.3 TRANSFORMAÇÕES

Até aqui usamos sequências de números aleatórios, essencialmente como uma variável aleatória com distribuição uniforme entre 0 e 1. Apesar de, por exemplo no caso das figurinhas, sortearmos números entre 1 e N, com N um inteiro positivo, convencionamos adotar que uma sequência de números aleatórios pertence a uma variável aleatória uniforme entre 0 e 1. No entanto se houver necessidade de se obter uma sequência aleatória de outra distribuição de probabilidade, será possível obtê-la através de uma transformação. Por exemplo, podemos obter uma amostragem de uma variável aleatória de distribuição normal com média μ e variância σ^2 mediante uma transformação de um conjunto de variáveis aleatórias de distribuição uniforme entre 0 e 1.

Há diferentes técnicas para transformar uma uniforme em outra distribuição, sendo uma delas o *método da transformação inversa*.

É possivel fazer essa transformação usando a função de distribuição acumulada da variável da qual desejamos obter a sequência. Define-se uma função de distribuição acumulada (de probabilidade) como:

$$F(x) = \mathcal{P}(X \leq x)$$

onde X é a variável e x é um possível valor que X pode assumir. Como $F(x)$ representa uma probabilidade, o contradomínio é o intervalo $[0, 1]$. Mais informações sobre esta função podem ser vistas em Bussab e Morettin (2006, p. 138).

Conforme Ross (2010, p. 520), seja U uma variável aleatória uniforme no intervalo [0,1]. Para qualquer variável de distribuição contínua F (aqui devemos entender F como uma função de distribuição acumulada), se definirmos uma variável aleatória Y como

$$Y = F^{-1}(U)$$

então Y tem distribuição F. Aqui F^{-1} é a função inversa de F. Para demonstrar, lembre que F é crescente e faça:

$$P(Y \leq y) = P(F^{-1}(U) \leq y) = P(U \leq F(y)) = P(u \in [0, F(y)]) = F(y)$$

Uma visualização gráfica deste método de transformação, para o caso contínuo, é fornecida por Bussab e Morettin (2006, p. 236). Para o caso discreto, é apresentada

uma pequena alteração no gráfico da função de distribuição acumulada.

O método pode ser aplicado com eficiência, por exemplo, para se obter uma variável de distribuição *exponencial* de acordo com Ross (2010, p. 521): dada uma variável aleatória exponencial com função de distribuição acumulada $F(x) = 1 - e^{-x}$, então temos que achar o valor x tal que $F(x) = u$, dado que u pertence a uma $U(0,1)$. Assim, temos

$$1 - e^{-x} = u$$

então $e^{-x} = 1 - u$. Aplicamos logaritmo nos dois lados da igualdade: $\ln(e^{-x}) = \ln(1 - u)$ e concluímos que $x = -\ln(1 - u)$. Aqui encontramos $F^{-1}(U) = -\ln(1 - U)$, sendo U uma variável uniforme no intervalo (0,1). Assim, F^{-1} possui distribuição exponencial com média 1. Da mesma forma, $-\ln(U)$ também tem distribuição exponencial com média 1 pois $(1 - U)$ é também uniforme entre 0 e 1.

Há outros métodos de obtenção de números aleatórios com distribuição de probabilidade que não a uniforme entre 0 e 1. Ross (2010, p. 521), apresenta o *método da rejeição* e o *método polar* para obtenção de variáveis aleatórias normais e mais exemplos de obtenção de distribuição de Poisson, binomial, entre outras, além de uma abordagem do método de *redução de variância*.

5

UMA APLICAÇÃO PARA ENSINO MÉDIO

O assunto referente à aleatoriedade pode despertar curiosidade em alunos do ensino médio. Um tema conhecido é o das loterias, que pode ser abordado após os alunos terem estudado probabilidades. Podemos lançar questões do tipo: "Será que os sorteios são aleatórios?", "Como é realizado o sorteio?" ou ainda "Se os sorteios são aleatórios, porque historicamente há uma dezena que aparece com maior frequência?"

Uma abordagem como essa pode introduzir intuitivamente o conceito (a ideia) de aleatoriedade. Ao que nos parece, um sorteio de bolas em um globo giratório parece ser algo absolutamente aleatório, imprevisível. O que mais poderia se parecer com um sorteio? Lançar moeda, jogar um dado, girar uma roleta, etc. Podemos comentar que executar um experimento desses várias vezes e anotar seus resultados seria uma forma de criar uma "tábua" de aleatoriedade, uma fonte de números aleatórios.

5.1 PROBLEMA DE MONTY HALL

Para ilustrar a utilidade de números aleatórios, o professor poderia simular em classe o clássico problema a seguir: o professor convida um aluno para participar de um jogo e o enuncia em seguida: "Sobre a mesa há três envelopes: A, B e C. Em um deles fingiremos que há um cheque de mil reais, enquanto nos outros dois fingiremos que há uma moeda de um real. Você pode escolher um dos envelopes". Após a escolha de um envelope pelo aluno, o professor abre um dos envelopes não escolhidos e revela uma moeda. Restam portanto dois envelopes fechados, sendo que um deles é o que foi escolhido pelo aluno. Em seguida o professor propõe ao aluno: "Você gostaria de mudar sua escolha para o outro envelope? Ou prefere ficar com o mesmo?". A questão

principal do problema é: qual é a melhor estratégia para ganhar o prêmio de mil reais? Mudar de envelope ou manter a escolha inicial? Ou seria indiferente?

Esse é o famoso problema de Monty Hall, em homenagem ao apresentador de um programa televisivo norte-americano. Uma análise mais profunda e pitoresca desse problema e sua enorme repercussão pode ser conferida em Mlodinow (2008, p. 51).

Desde que não conheçam o problema e sua solução, o que se supõe é que os alunos achem em sua maioria, que parece ser indiferente trocar ou não. O principal argumento para não ver nenhuma vantagem em trocar a escolha inicial é a de que restaram dois envelopes fechados e que um deles há um prêmio e no outro não. Haveria assim, 50% de chance para cada um dos envelopes conter o prêmio. Embora questionável, esse raciocínio parece bastante razoável e difícil de rebater rapidamente. Seria importante que os alunos defendessem racionalmente seus pontos de vista. Se alguns optarem por trocar, por acharem mais vantajoso, deverão fazê-lo com alguma justificativa razoável.

Uma ideia inicial para se questionar a tese dos 50% seria a seguinte: há três envelopes na mesa, com a mesma probabilidade de conter o prêmio. Logo, uma escolha de um dos envelopes tem $1/3$ de chance de ser o premiado. Como pode o fato de abrir um envelope mudar a probabilidade anterior? De fato, não muda. Ademais, já sabemos *a priori* que um envelope seria aberto, então a probabilidade deveria ser 50% no início, o que é um absurdo. Se a probabilidade do envelope inicial conter o cheque é ainda de $1/3$, então a probabilidade do outro envelope ainda fechado conter o cheque, deve ser de $2/3$, pela definição de probabilidade — que supõe-se que os alunos já conheçam.

A questão central é a de que o professor, no papel de Monty Hall, precisa escolher um de dois envelopes para abrir e, com certa probabilidade, só pode abrir um determinado (no caso do aluno ter escolhido o outro com a moeda). Vale lembrar que tanto o professor, quanto o apresentador de TV *sabem* previamente o conteúdo de cada envelope.

O professor poderia então propor uma solução experimental. Por que não *simular* o problema? Os passos para se fazer esta atividade seriam razoavelmente simples e

poderiam ser realizados com a colaboração de todos. Números aleatórios e uma tabela são os ingredientes. Um modelo seria a tabela mostrada na Figura 8 adiante. E para obter números aleatórios, uma boa opção é o lançamento de um dado.

Um experimento consistiria em separar a turma em três equipes, cada uma com uma função, seguindo as seguintes intruções:

- **Sorteio do envelope do prêmio:** A primeira equipe fica encarregada de lançar um dado e obter o resultado de onde estará o prêmio. Caso a face sorteada for 1 ou 2, o prêmio estará no envelope A. Se a face sorteada no dado for 3 ou 4, envelope B. E finalmente se sair 5 ou 6, envelope C. O resultado é anotado.
- **Sorteio do envelope escolhido:** A segunda equipe segue exatamente os passos da primeira equipe. Porém o resultado é o envelope que o participante do jogo escolhe. O resultado é anotado.
- **Preenchimento da tabela:** A terceira fica responsável por preencher a tabela e calcular o placar.

O cálculo do placar é feito da seguinte forma: vamos supor que o sorteado para conter o prêmio seja o envelope A, e que a escolha do concorrente do jogo seja pelo envelope B. Num jogo real o apresentador revelaria o conteúdo do envelope C (que não contém o prêmio) e perguntaria ao concorrente se ele deseja trocar a escolha para o envelope A. Caso a opção seja por trocar (coluna “Troca”), o concorrente ganha o prêmio. Caso permaneça com a escolha original, não ganha o prêmio. Assim, marcamos 1 na coluna “Troca” e zero na coluna “Não Troca”. Esse critério é adotado então, qualquer que seja a combinação de prêmio e escolha.

É conveniente, entretanto, que o professor explique que apenas um experimento não é suficiente para simular. Assim, solicita que cada uma das equipes repita o procedimento 20 vezes, digamos. Na prática, a primeira equipe sorteia 20 resultados e os anota, a segunda equipe faz o mesmo e a terceira equipe lança todos os resultados e faz o placar final. Surpreendentemente o resultado dará clara vantagem na troca da escolha inicial em relação à não troca. Esse exemplo pode despertar nos estudantes uma noção mais clara sobre probabilidades e aleatoriedade.

	A	B	C	Troca	Não Troca
Prêmio					
Escolha					

Prêmio					
Escolha					

Prêmio					
Escolha					

Figura 8: Tabela para auxiliar problema de *Monty Hall*

Opcionalmente ao lançamento do dado, o professor poderá, caso possua recursos como acesso à Internet, computador e projetor, abrir algum endereço de Internet a fim de obter uma listagem de números aleatórios de forma rápida e eficiente. Talvez a melhor sugestão seja RANDOM.ORG. Há a opção de gerar grande quantidade de números aleatórios em sequência — que neste exemplo estão entre 1 e 3. O professor pode explicar que em situações diferentes, de caráter científico e/ou profissional, grande quantidade de números aleatórios em sequência precisa ser obtida em um curto tempo, o que inviabiliza a obtenção manual, como o lançamento do dado.

5.2 CONCLUSÃO

Números aleatórios têm incontestável utilidade, sobretudo após a tecnologia tornar-se inerente ao desenvolvimento científico. As simulações, o método Monte Carlo, entre outras aplicações, encontram nos números aleatórios um insumo essencial para obtenção de resultados. Ademais, a geração de números aleatórios, seus métodos e aplicações ainda possuem campo para pesquisa, por ser uma área relativamente nova, iniciada em meados do século 20.

Para estudantes de ensino médio, uma vez que conheçam os fundamentos de probabilidades, os números aleatórios e simulações podem ser um interessante complemento, além de ilustrar resolução de problemas e aplicações práticas.

BIBLIOGRAFIA

[1] CARVALHO, P. C. P. Quantas figurinhas comprar para completar o álbum da copa? *Revista do Professor de Matemática*, n. 73, 2010, pp. 37-41.

[2] FILHO, C. P. *Introdução à Simulação de Sistemas*. Editora da UNICAMP, 1995.

[3] GENTLE, J. E. *Random Numbers Generation and Monte Carlo Methods*. Springer, 2003.

[4] GUIDORIZZI, H. L. *Um Curso de Cálculo.* Livros Técnicos e Científicos, 1991.

[5] HEFEZ, A. *Elementos de Aritmética*. SBM, 2011.

[6] KATZGRABER, H. G. *Random Numbers in Scientific Computing: An Introduction.* Lecture given at the International Summer School Modern Computational Science, Oldenburg, Germany, 2010. Disponível em *http://arxiv.org/pdf/1005.4117.pdf*. Acesso em 20 nov. 2014.

[7] KNUTH, D. E. *The Art of Computer Programming: Seminumerical Algorithms.* 3 ed., v. 2. Addison Wesley, 1998.

[8] MAGALHÃES, M. N.; LIMA, A. C. P. *Noções de Probabilidade e Estatística.* USP, 2000.

[9] MARSAGLIA, G. Random Numbers Fall Mainly in The Planes. *Proceedings of the National Academy of Sciences of the United States of America,* v. 61, 1968, pp. 25-28.

[10] MARSAGLIA, G. *The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness*. Disponível em *http://stat.fsu.edu/pub/diehard/*. Acesso em 4 jun. 2015.

[11] MARTINEZ, F. B.; MOREIRA, C. G.; SALDANHA, N.; TENGAN, E. *Teoria dos Números, um passeio com primos e outros números familiares pelo mundo inteiro.* IMPA, 2010.

[12] MATSUMOTO, M.; NISHIMURA, T. *Mersenne Twister Home Page.* Disponível em *http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html*. Acesso em

25 jul. 2015.

[13] MLODINOW, L. *O Andar do Bêbado*. Zahar, 2008.

[14] MORETTIN, P. A.; BUSSAB, W. O. *Estatística Básica*. Saraiva, 2006.

[15] MORETTIN, P. A.; TOLOI, C. M. C. *Análise de Séries Temporais*. Blücher, 2006.

[16] NIST/SEMATECH. *e-Handbook of Statistical Methods.* Disponível em *http://www.itl.nist.gov/div898/handbook/eda/section3/eda35d.htm* Acesso em 7 set. 2015

[17] PANINI. Disponível em *http://www.torcidapanini.com.br* Acesso em 28 out. 2014.

[18] PERL. Disponível em *http://perldoc.perl.org/* Acesso em 20 jul. 2015.

[19] RANDOM.ORG. Disponível em *https://www.random.org* Acesso em 21 jul. 2015.

[20] ROSS, S. *Probabilidade: Um Curso Moderno com Aplicações*. Bookman, 2010.

[21] WOLFRAMALPHA. Disponível em *http://www.wolframalpha.com/* Acesso em 5 ago. 2015