ALGORITMOS PROBABILISTICOS

A GERAÇÃO DE NÚMEROS ALEATÓRIOS

ALGORITMOS PROBABILÍSTICOS
O PAPEL DA ALEATORIEDADE NO MUNDO MODERNO

- **O que são Números Aleatórios?**
  - Sequências de números sem um padrão previsível.
  - São a base para simular a imprevisibilidade de eventos e a complexidade do mundo real.
- **Por que são importantes?**
  - Permitem modelar sistemas complexos.
  - Garantem segurança em transações digitais.
  - Criam experiências dinâmicas em software e jogos.
- **Tipos de Geradores:**
- **Geradores de Números Aleatórios Verdadeiros (TRNGs):**
  - Baseados em fenômenos físicos (ex: ruído eletrônico).
- **Geradores de Números Pseudo-Aleatórios (PRNGs):**
  - Baseados em algoritmos matemáticos que produzem sequências longas e não-repetitivas.
  - São a base da computação moderna.

ALGORITMOS PROBABILÍSTICOS APLICAÇÕES EM SIMULAÇÃO E MODELAGEM

•**Análise de Risco Financeiro (Simulação de Monte Carlo)**

•**Exemplo:** Simular milhares de cenários possíveis para o preço de uma ação.
•A aleatoriedade é usada para modelar o "comportamento" imprevisível do mercado, permitindo que analistas avaliem o risco de um portfólio de investimentos.

•**Engenharia Civil e Sismologia**
•**Exemplo:** Simular como um prédio se comportaria durante um terremoto.
•A intensidade e a frequência de um tremor de terra são eventos aleatórios.
Geradores de números aleatórios permitem testar a resiliência de estruturas em diversas condições de estresse.

ALGORITMOS PROBABILÍSTICOS ENTRETENIMENTO E INDÚSTRIA CRIATIVA

•**Videogames (Geração Procedural)**

•**Exemplo:** A criação de mundos, masmorras e itens em jogos como *Minecraft* ou *Diablo*.
•Números aleatórios são utilizados para gerar conteúdo infinito e único, garantindo que cada partida seja uma nova experiência.

•**Música e Artes Visuais**

•**Exemplo:** Composição musical algorítmica ou arte generativa.
•Artistas e compositores usam a aleatoriedade como uma ferramenta para explorar novas formas, padrões e melodias, criando obras que não poderiam ser concebidas manualmente.

ALGORITMOS PROBABILÍSTICOS SEGURANÇA E CRIPTOGRAFIA

•**Geração de Chaves Criptográficas**

•**Exemplo:** Criar a chave privada de uma carteira de Bitcoin ou as chaves usadas para criptografar uma comunicação via WhatsApp.
•A segurança desses sistemas depende inteiramente de chaves que são verdadeiramente imprevisíveis. Um gerador fraco pode levar à quebra da criptografia.

•**"Salting" de Senhas (Salting)**

•**Exemplo:** Armazenar senhas de usuários de forma segura em um banco de dados.
•Um valor aleatório (salt) é adicionado à senha antes de ser "hashada". Isso impede ataques de dicionário e rainbow tables, mesmo que o banco de dados seja comprometido.

ALGORITMOS PROBABILÍSTICOS
CIÊNCIA DE DADOS E MACHINE LEARNING

•**Divisão de Dados para Treinamento (Train-Test Split)**

•**Exemplo:** Em um modelo de IA que prevê a probabilidade de uma doença, é crucial que os dados de treinamento e teste sejam divididos aleatoriamente para garantir a imparcialidade e a robustez do modelo.
•A aleatoriedade assegura que o modelo não seja treinado em uma amostra enviesada.

•**Algoritmos de Floresta Aleatória (Random Forest)**

•**Exemplo:** Um algoritmo de aprendizado de máquina usado para classificação e regressão.
•Ele constrói múltiplas "árvores de decisão" de forma aleatória e combina seus resultados para uma previsão mais precisa. O elemento aleatório previne o *overfitting* (ajuste excessivo do modelo aos dados).

ALGORITMOS PROBABILÍSTICOS
TESTE DE FREQUÊNCIA (TESTE QUI-QUADRADO)

- Como aplicar:
- Conte quantas vezes cada número, de 1 a 25, apareceu em todos os sorteios da sua planilha.
- Calcule a frequência esperada: (Total de dezenas sorteadas) / (25). Por exemplo, se você tem 100 sorteios, o total de dezenas é 100×15=1500. A frequência esperada para cada número seria 1500/25=60.
- Compare a frequência observada (a contagem que você fez) com a frequência esperada usando o teste Qui-Quadrado. Se a diferença for grande, é um indício de que a distribuição não é uniforme e pode não ser aleatória.

ALGORITMOS PROBABILÍSTICOS
TESTE DE SEQUÊNCIAS (RUNS TEST)

Este teste verifica a existência de padrões ou tendências na ordem em que os números aparecem. Ele é útil para ver se há "correntes" de números que se repetem ou se alternam de forma previsível.

- **Como aplicar:**

1. Escolha uma propriedade para os números, como "ímpar" ou "par", "alto" ou "baixo" (por exemplo, abaixo ou acima da mediana, que é 13).

2. Para cada sorteio, crie uma sequência com base nessa propriedade. Por exemplo, para o sorteio (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15), a sequência de par/ímpar seria: Í-P-Í-P-Í-P-Í-P-Í-P-Í-P-Í-P-Í.

3. Conte o número de "sequências" (runs). No exemplo acima, a sequência Í-P-Í-P... tem 15 "runs" (uma sequência de ímpar, uma de par, etc.).
4. Compare o número de runs observado com o número esperado de runs para uma sequência aleatória. Se o número de runs for consistentemente muito baixo (indicando padrões longos) ou muito alto (indicando alternância extrema), a aleatoriedade pode ser questionada.

ALGORITMOS PROBABILÍSTICOS
TESTE DE POKER

- O teste de poker é usado para verificar se certas combinações de dezenas ocorrem com a frequência esperada. A ideia é tratar cada sorteio como uma "mão" de poker e ver a distribuição dessas mãos.
- **Como aplicar:**
  - Defina algumas "mãos" de interesse. Por exemplo:
    - "Sequência": um sorteio com 5 ou mais números consecutivos (ex: 7, 8, 9, 10, 11).
    - "Trinca de pares": um sorteio com 3 pares de números consecutivos (ex: 2 e 3, 8 e 9, 12 e 13).
    - "Muitos ímpares": um sorteio com 12 ou mais números ímpares.
  - Calcule a probabilidade teórica de cada uma dessas "mãos" ocorrerem em um sorteio aleatório.
  - Conte a frequência com que essas "mãos" realmente aparecem na sua planilha.
  - Compare a frequência observada com a teórica. Se uma "mão" aparece com muito mais ou muito menos frequência do que o esperado, isso pode ser um indício de não-aleatoriedade.

ALGORITMOS PROBABILÍSTICOS
TESTE DE GAP (TESTE DE LACUNAS)

- Este teste analisa a distância entre as ocorrências de um número específico. A distribuição dessas lacunas deve seguir um padrão específico (distribuição geométrica) se os sorteios forem aleatórios.
- **Como aplicar:**
  - Escolha um número, por exemplo, o "10".
  - Percorra a planilha e anote a distância (em número de sorteios) entre cada vez que o número "10" foi sorteado.
  - Analise a distribuição dessas distâncias. Em uma sequência aleatória, é esperado que a maioria das lacunas seja pequena e que poucas sejam muito grandes. Se você encontrar muitas lacunas muito longas ou muito curtas, pode ser um sinal de que o número não está sendo sorteado aleatoriamente.

ALGORITMOS PROBABILÍSTICOS

Neste contexto, o estudo de diferentes algoritmos geradores é essencial para entender como essa aleatoriedade é construída e quais são suas características.
O algoritmo **SPUTINIK**, em particular, representa uma abordagem interessante para a geração de variáveis aleatórias com base em uma distribuição uniforme. Sua implementação, originalmente em C, utiliza um método que transforma um número aleatório de distribuição uniforme, gerado pela macro myrand, em uma nova variável que se aproxima de uma distribuição normal.
O objetivo deste trabalho é, portanto, não apenas traduzir o algoritmo SPUTINIK da linguagem C para Python, mas também realizar uma análise aprofundada de sua aplicação. Por meio de uma série de execuções e da visualização de dados, buscamos responder à seguinte questão:
que tipo de variável aleatória este algoritmo gera?
Para isso, utilizaremos gráficos e distribuições de frequência para demonstrar visualmente as características do algoritmo e validar a nossa análise.