Slice Sampling como alternativa ao Método de Rejeição

Carlos Henrique Mora Neto

1 julho, 2024

Resumo

Resumo: Esse artigo pretende analisar o algoritmo de slice sampling para avaliar se pode ser utilizado como uma alternativa viável e melhorada para o método de rejeição, incluindo simulações com dados reais e avaliações sobre custo computacional.

Palavras-chave: método de rejeição, slice sampling, simulação, geração de amostras

1 Introdução

1.1 Descrição do Problema

O método de rejeição, também conhecido como algoritmo de aceitação-rejeição, é um método conhecido e bem comumente utilizado para gerar amostras a partir de distribuições complexas. Este método é particularmente útil quando a distribuição de interesse é difícil de amostrar diretamente, mas pode ser descrita por uma função de densidade de probabilidade que é conhecida apenas até uma constante de normalização. A eficiência do método de rejeição reside na sua simplicidade e na flexibilidade para amostrar a partir de uma vasta gama de distribuições, desde que uma função de proposta adequada seja escolhida.

Entretanto, apesar de suas vantagens, o método de rejeição tem uma particularidade que pode ser indesejada, que é a alta quantidade de iterações necessárias no método para que a amostra gerada alcançe um resultado próximo à distribuição alvo. Dessa forma, buscaremos avaliar o método de Slice Sampling como uma alternativa plausível para esse método, tentando alcançar um algoritmo similarmente simples com maior eficiência.

1.2 Objetivos da Análise

Nessa análise, deseja se comparar diversas métricas relacionadas a custo computacional e eficiência de cada um dos métodos de interesse, a fim de se chegar a uma conclusão com relação aos seus potenciais de aplicação a dados reais.

1.3 Dados Utilizados

Para a realização dessa análise, foram selecionadas duas bases de dados que foram consideradas relevantes e que foram utilizadas para gerar as funções de densidade que serão geradas pelos métodos nesse estudo. Ambas foram selecionadas no Kaggle, que foram as seguintes:

- Cirtautas, 2023: Base completa de jogadores da NBA, com informações pessoais completas sobre cada um. Conta com 12844 dados e 22 colunas de variáveis, entretanto para esse estudo foi selecionada apenas uma como principal: a altura de cada jogador, em centímetros.
- Dabbas, 2018: Base de dados referentes à bilheteria de todos os filmes na história dos Estados Unidos. No total, abrange 15743 dados em 5 colunas, das quais apenas a quantidade de dinheiro absoluto obtido pelo filme em dólares foi escolhida. Além disso, foi necessário nessa base a exclusão de valores outliers do intervalo selecionado para a geração da amostra, a fim de reduzir o custo computacional que a aplicação dos métodos gastaria.

O objetivo da seleção de duas bases de dados se dá pelo fato de poder abranger maiores tipos de distribuições para testar os métodos, em quesito de formato e tamanho das densidades.

2 Metodologia

A metodologia utilizada envolveu os dois algoritmos escolhidos para comparação, que são:

2.1 Método de Rejeição Clássico

Dada que a distribuição-alvo a ser gerada é $f(x)$:

1. **Escolha da Distribuição Proposta:**
   - Escolha uma distribuição proposta $g(x)$ tal que seja fácil de amostrar e que majorize a distribuição-alvo $f(x)$.
   - Encontre uma constante c tal que $f(x) \leq c \cdot g(x)$ para todo x.
2. **Amostragem da Distribuição Proposta:**
   - Gere um valor x a partir da distribuição $g(x)$.

$$x \sim g(x)$$

3. **Geração de um Número Uniforme:**
   - Gere um valor u a partir de uma distribuição uniforme no intervalo $[0, 1]$.

$$u \sim \text{Uniforme}(0, 1)$$

4. **Critério de Aceitação:**

- Aceite x como uma amostra da distribuição $f(x)$ se:

$$u \leq \frac{f(x)}{c \cdot g(x)}$$

- Caso contrário, rejeite x e volte ao passo 2.

5. **Repetição:**
   - Repita os passos 2 a 4 até obter o número desejado de amostras n.

A partir daí, os n valores de x obtidos compõem a amostra gerada desejada.

2.2 Método de Slice Sampling

Dada que a distribuição-alvo a ser gerada é $f(x)$, seguem-se os seguintes passos:

1. **Escolha de um Valor Inicial:**
   - Escolha um valor inicial aleatório x_0.
2. **Amostragem Uniforme no Intervalo:**
   - Gere um valor a a partir de uma distribuição uniforme no intervalo $[0, f(x_0)]$.

$$a \sim \text{Uniforme}(0, f(x_0))$$

3. **Determinação dos Segmentos de Linha:**
   - Imagine uma linha horizontal em $y = a$. Determine todos os segmentos de linha abaixo da curva $f(x)$.
4. **Amostragem de x Uniformemente nos Segmentos:**
   - A partir de todos os segmentos de linha, desenhe um valor de x uniformemente.

$$x \sim \text{Uniforme}(\text{segmentos de linha abaixo de } y = a)$$

5. **Repetição:**
   - Repita a partir do passo 2 até obter o número desejado de amostras n.

Dessa forma, os n valores de x gerados se tornam a amostra obtida.

3 Resultados/Aplicações

3.1 Aplicação das Metodologias

Para aplicação das metodologias, inicialmente foram estimadas as distribuições de cada uma das bases de dados, gerando os seguintes gráficos:

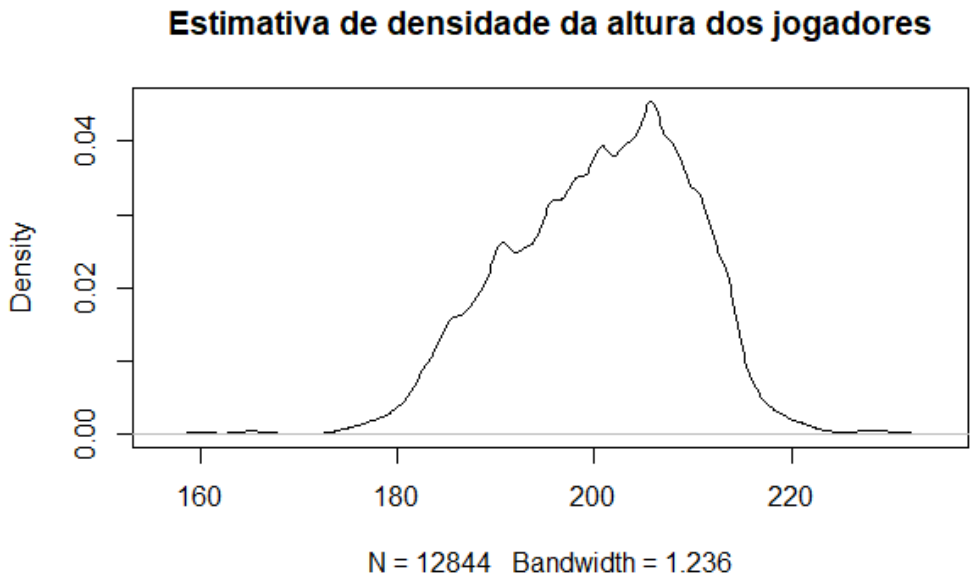

Figura 1: Distribuição real da primeira base de dados (Altura dos jogadores na NBA)

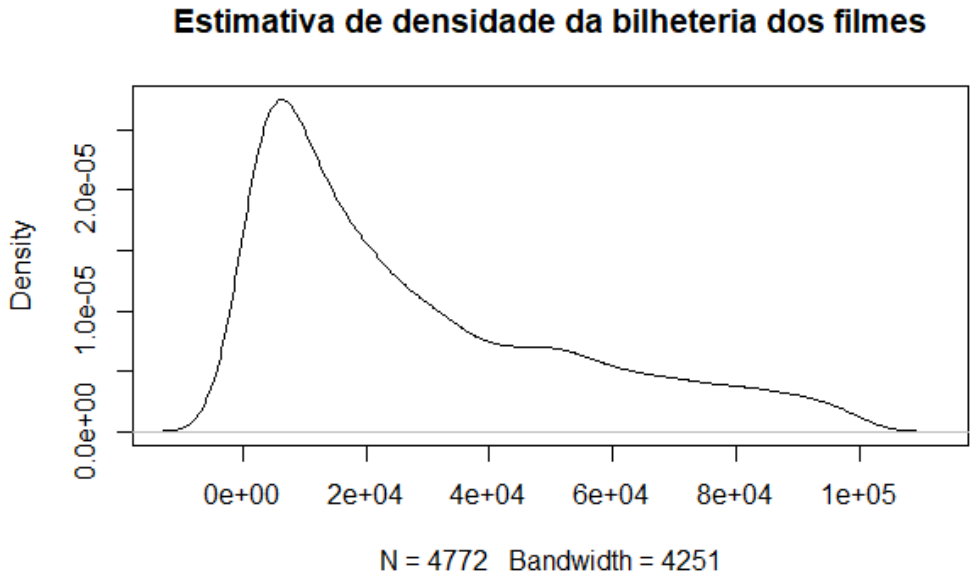

Figura 2: Distribuição real da segunda base de dados (bilheteria dos filmes)

A partir disso, foram feitos testes para se encontrar a verdadeira distribuição do primeiro grupo de dados, testando se havia a possibilidade de uma normalidade na base. Dessa forma, através da biblioteca "MASS"do R, foram geradas estimativas dos possíveis parâmetros para essa distribuição caso se aplicasse a uma Normal ($\mu = 200.555097$ / $\sigma = 9.110736$) e se aplicou um teste de Kolmogorov-Smirnov com tais parâmetros, que gerou os seguintes resultados:

Estatística de Teste (D)	**p-valor**
0.098647	$< 2.2 \times 10^{-16}$

Tabela 1: Resultados do teste de Kolmogorov-Smirnov

Como o p-valor indicou um valor muito próximo de 0, e bem abaixo do valor de rejeição 0.05, conclui-se que a distribuição não segue uma normalidade. Portanto, foram aplicadas as estimativas kernel em ambas as densidades para se definir a função distribuição alvo em ambos os casos.

Já para a distribuição $g(x)$ do método de rejeição, foram usadas distribuições Uniformes com parâmetros definidos pelos valores mínimo e máximo de cada uma das bases.

3.2 Resultados e comparações

3.2.1 Base de dados 1

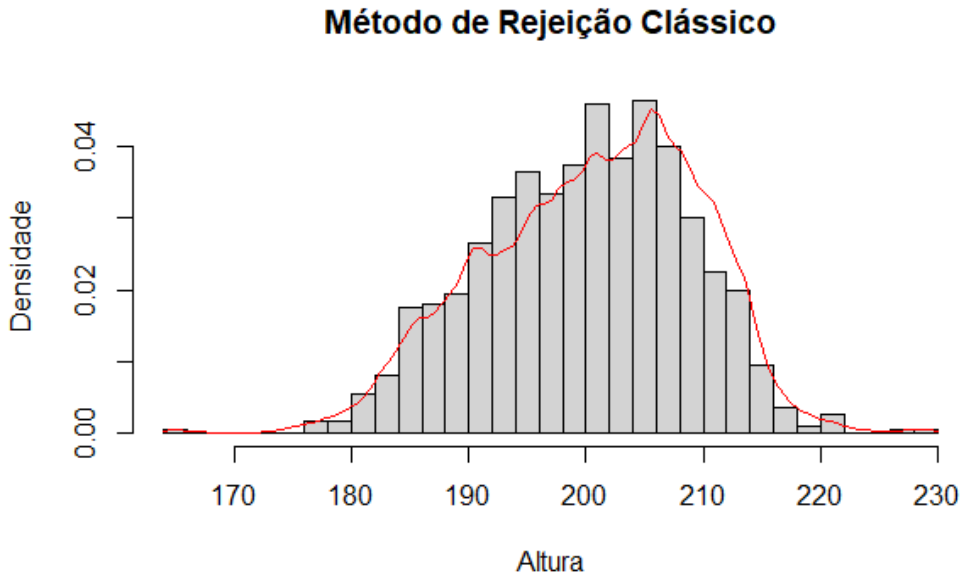

Figura 3: Histograma dos resultados do Método de Rejeição com n = 1000

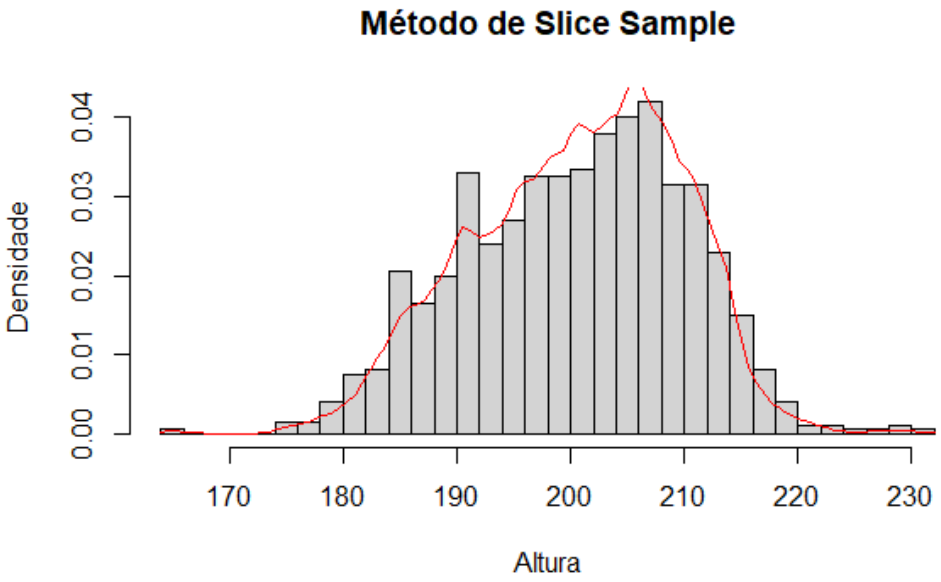

Figura 4: Histograma dos resultados do Método de Slice Sampling com n = 1000

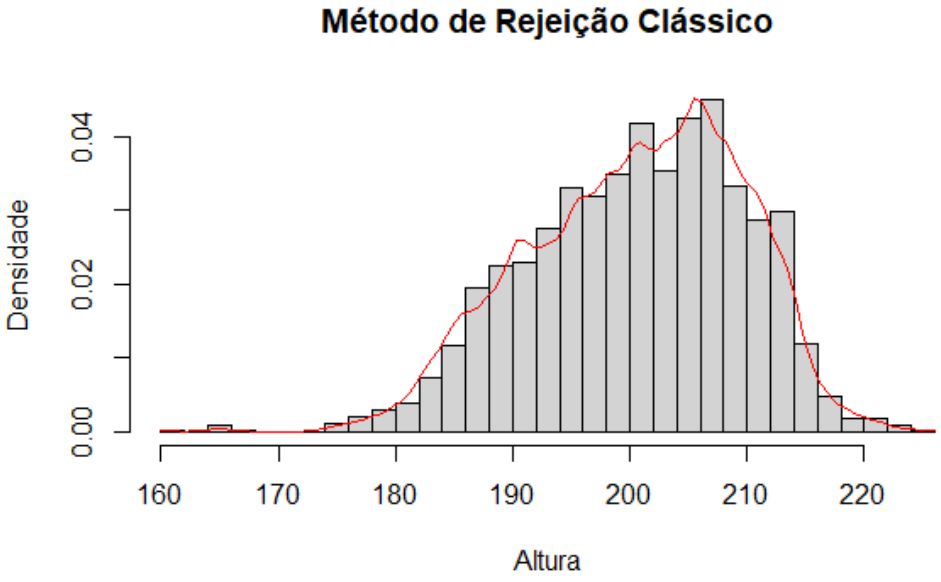

Figura 5: Histograma dos resultados do Método de Rejeição com n = 2000

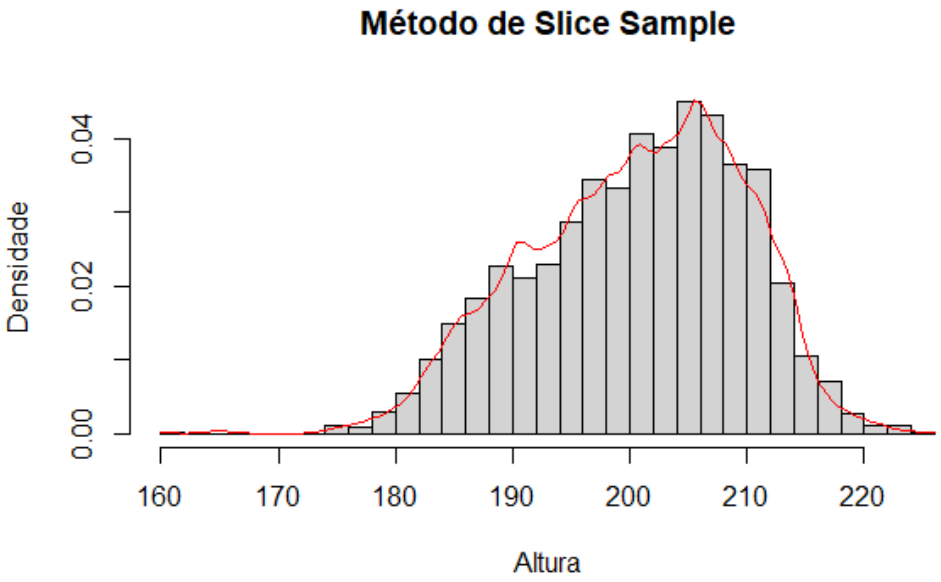

Figura 6: Histograma dos resultados do Método de Slice Sampling com n = 2000

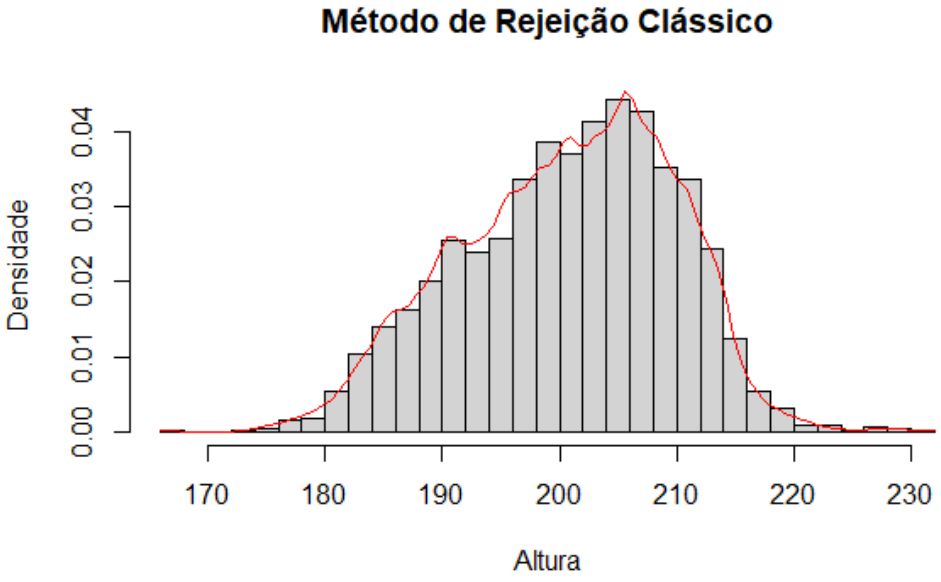

Figura 7: Histograma dos resultados do Método de Rejeição com n = 5000

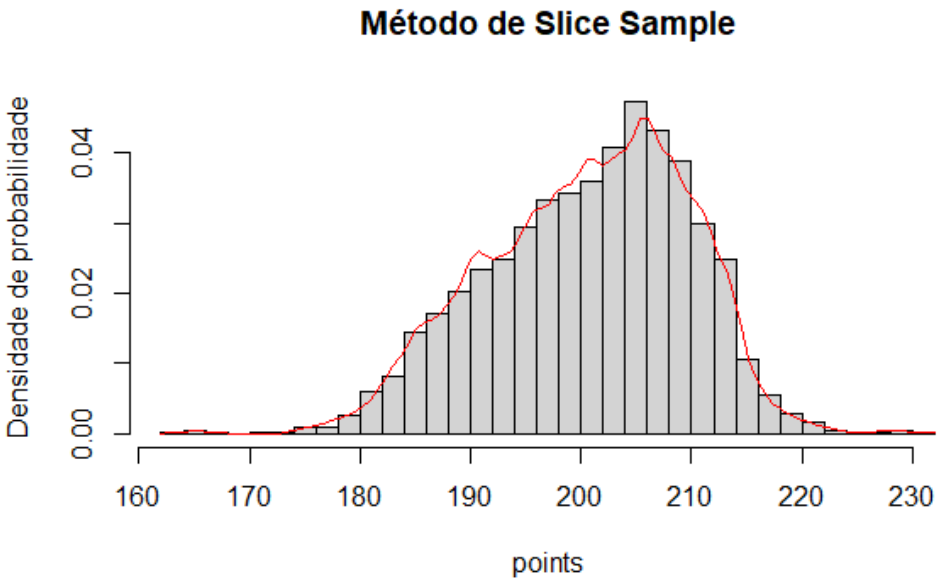

Figura 8: Histograma dos resultados do Método de Slice Sampling com n = 5000

Tamanho de n	**Tempo de execução (Método de Rejeição)**	**Tempo de execução (Slice Sampling)**
1000	0.1318319 secs	1.433258 mins
2000	0.1039279 secs	2.727945 mins
5000	0.441499 secs	6.160423 mins

Tabela 2: Comparação de tempo de execução entre métodos de amostragem na base de dados 1

Observando o teste da primeira base de dados testada, nota-se que o tempo de execução do método Slice Sampling tende a ser consideravelmente maior em relação ao Método de Rejeição, mas em compensação seus resultados aparentam ser mais precisos quando comparados em números iguais de amostras. Para esse caso, foi utilizada a **root.accuracy = 0.01**, o que reduz o intervalo de seleção do modelo Slice Sampling e, apesar de aumentar seu tempo, não deixou o algoritmo inutilizável e permitiu resultados melhores, fazendo dele um método superior ao de rejeição clássico.

3.2.2 Base de dados 2

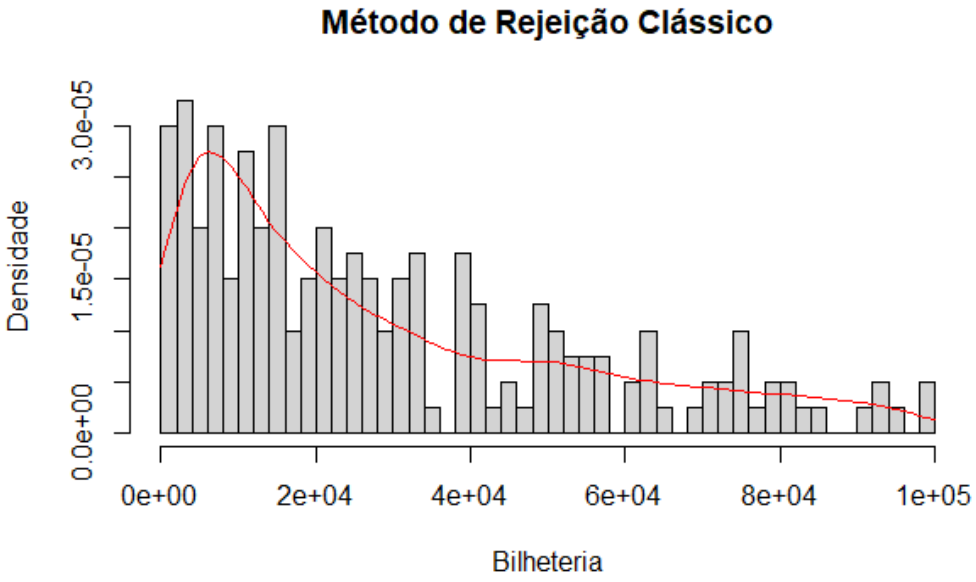

Figura 9: Histograma dos resultados do Método de Rejeição com n = 200

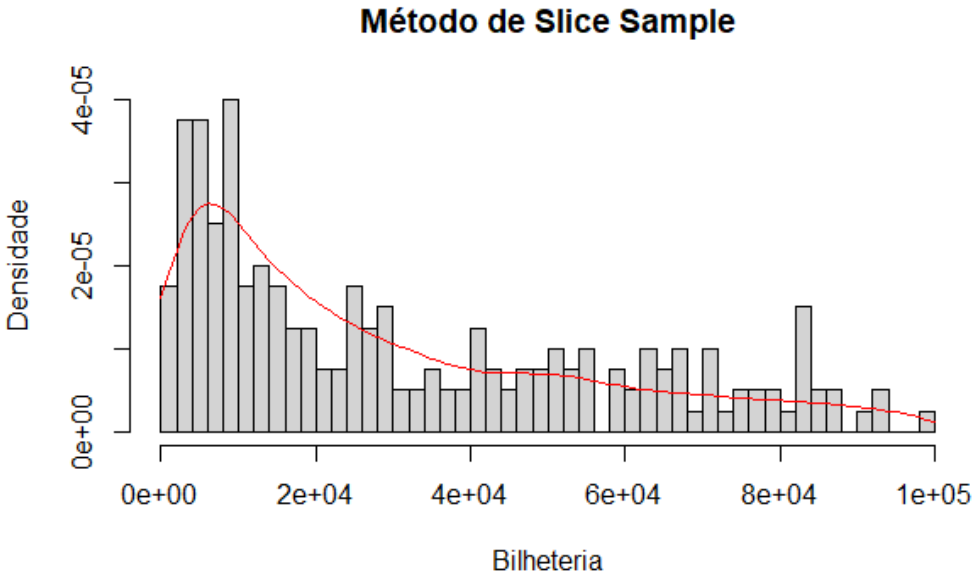

Figura 10: Histograma dos resultados do Método de Slice Sampling com n = 200

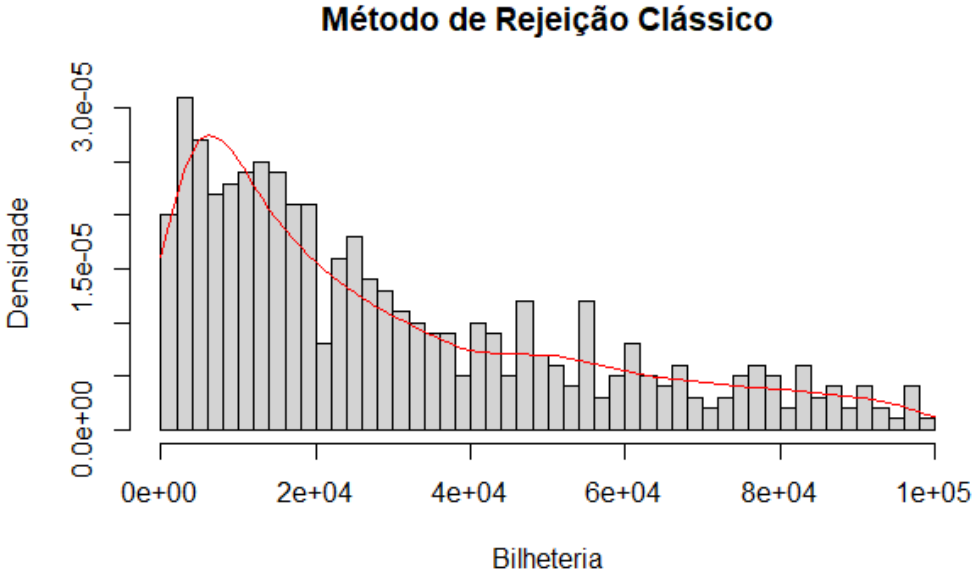

Figura 11: Histograma dos resultados do Método de Rejeição com n = 500

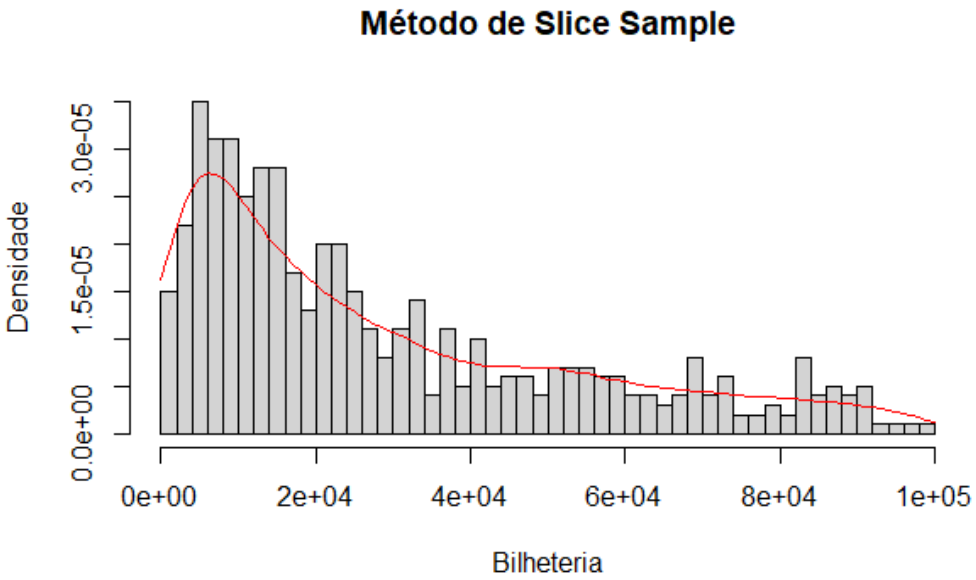

Figura 12: Histograma dos resultados do Método de Slice Sampling com n = 500

Tamanho de n	**Tempo de execução (Método de Rejeição)**	**Tempo de execução (Slice Sampling)**
200	0.076092 secs	10.45552 mins
500	0.06464791 secs	32.27823 mins

Tabela 3: Comparação de tempo de execução entre métodos de amostragem na base de dados 2

Já levando em consideração a segunda base, os resultados podem ser analisados de forma diferente: o tempo de execução do método de Slice Sampling continua levando bem mais tempo, mesmo depois de aumentar consideravelmente o valor da acurácia (**root.accuracy = 0.5**), e dessa maneira a precisão da amostra desse método se mostrou pior em comparação com o outro método, contradizendo a conclusão da outra base de dados. Caso a acurácia tivesse sido mantida constante, a amostra teria sido ainda um pouco melhor em comparação ao método de rejeição, mas o custo computacional de memória e tempo que isso gastaria chegaria em um nível que não compensaria a pequena diferença de melhora na seleção da amostra.

4 Conclusão/Discussão

Em conclusão, com base nas análises realizadas, pode-se inferir que a resposta para a dúvida de qual o método mais eficiente para geração de amostras não é simples, e vai variar a cada caso. Sendo dois algoritmos de relativamente simples implementações, a maior diferença e comparação a ser feita em relação aos métodos se torna exclusivo em relação aos seus custos e desempenhos.

Analisando as duas bases selecionadas e a comparação feita, é possível se denotar como ponto principal para ser feita essa comparação é o tamanho do intervalo de x: para a primeira base, na qual a variável varia entre 160 e 230 aproximadamente, os tempos de execução para o método de Slice Sampling não se mostraram absurdos, e pode-se utilizar um valor otimizado da root.accuracy para obter um resultado superior ao método de Rejeição. Já na segunda base, nos quais os valores variam bem mais (de 0 até mais de 100000), os tempos de execução se tornaram absurdos, fazendo inviável a utilização desse método, a não ser com uma acurácia muito abaixo do esperado, causando resultados não adequados, o que não justifica a utilização desse algoritmo.

Referências

Cirtautas, J. (2023). NBA Players. https://www.kaggle.com/datasets/justinas/nba-players-data/data

Dabbas, E. (2018). Boxofficemojo Alltime Domestic Data. https://www.kaggle.com/datasets/eliasdabbas/boxofficemojo-alltime-domestic-data